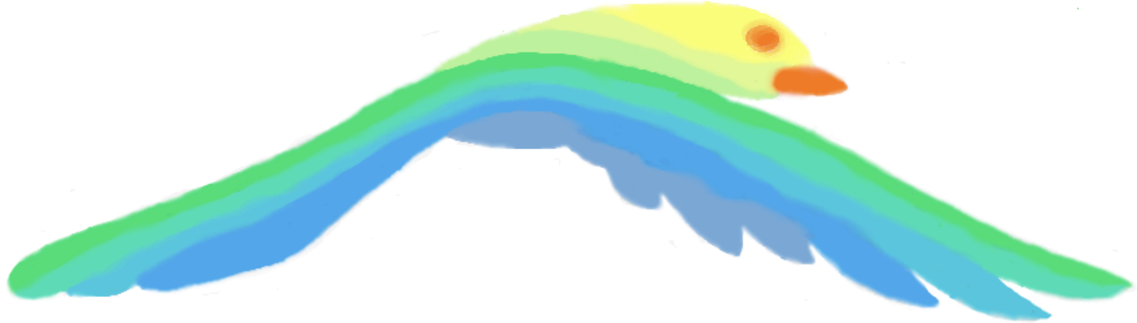
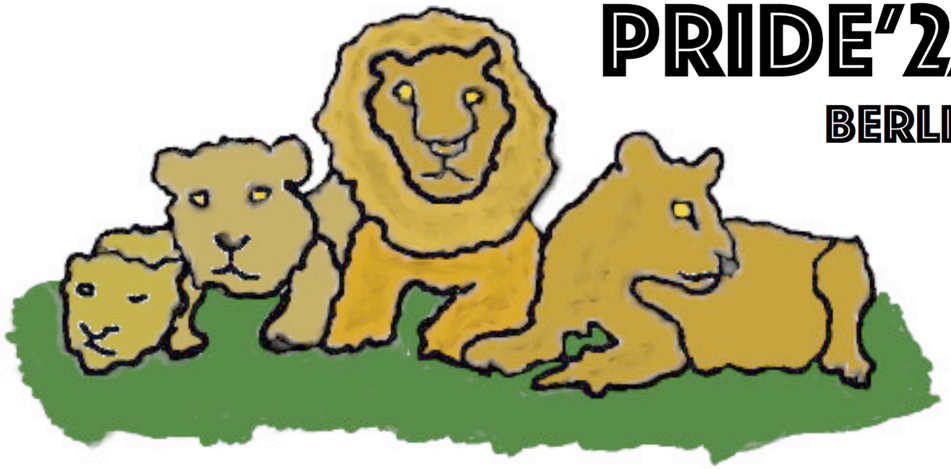


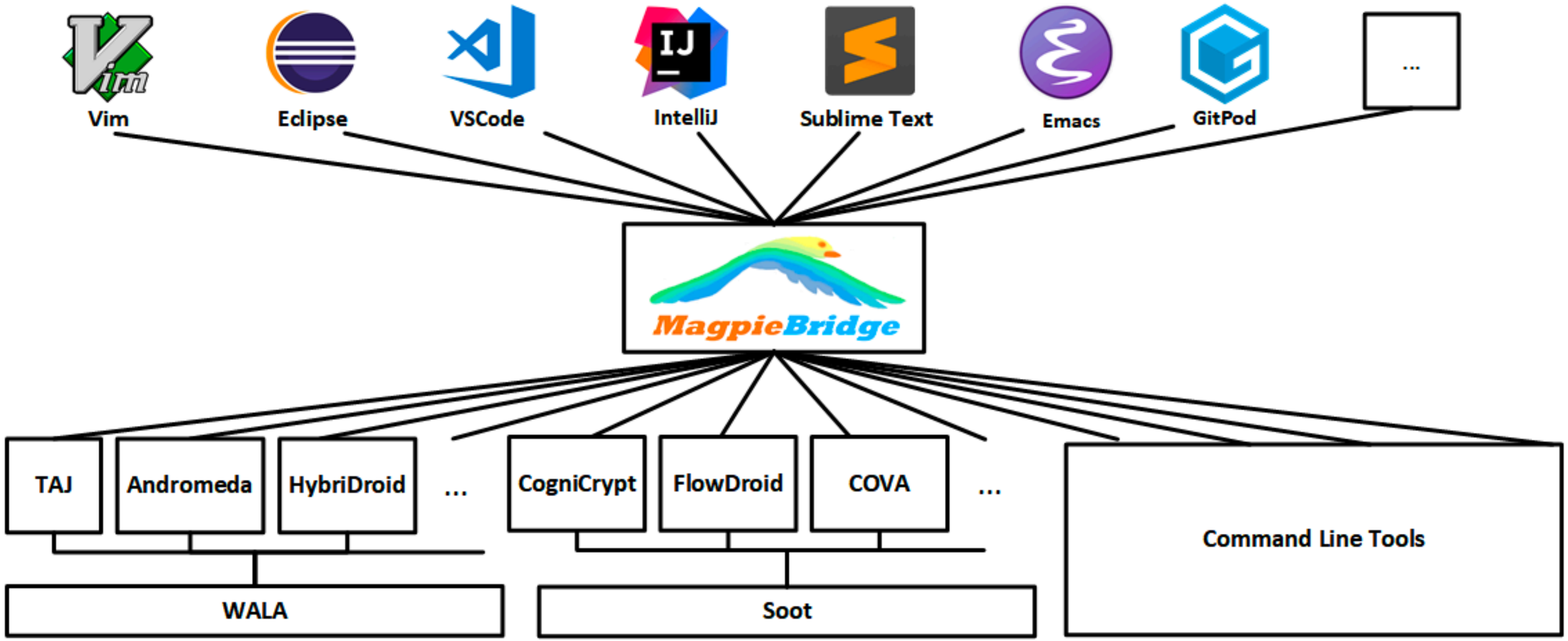
MagpieBridge Tutorial



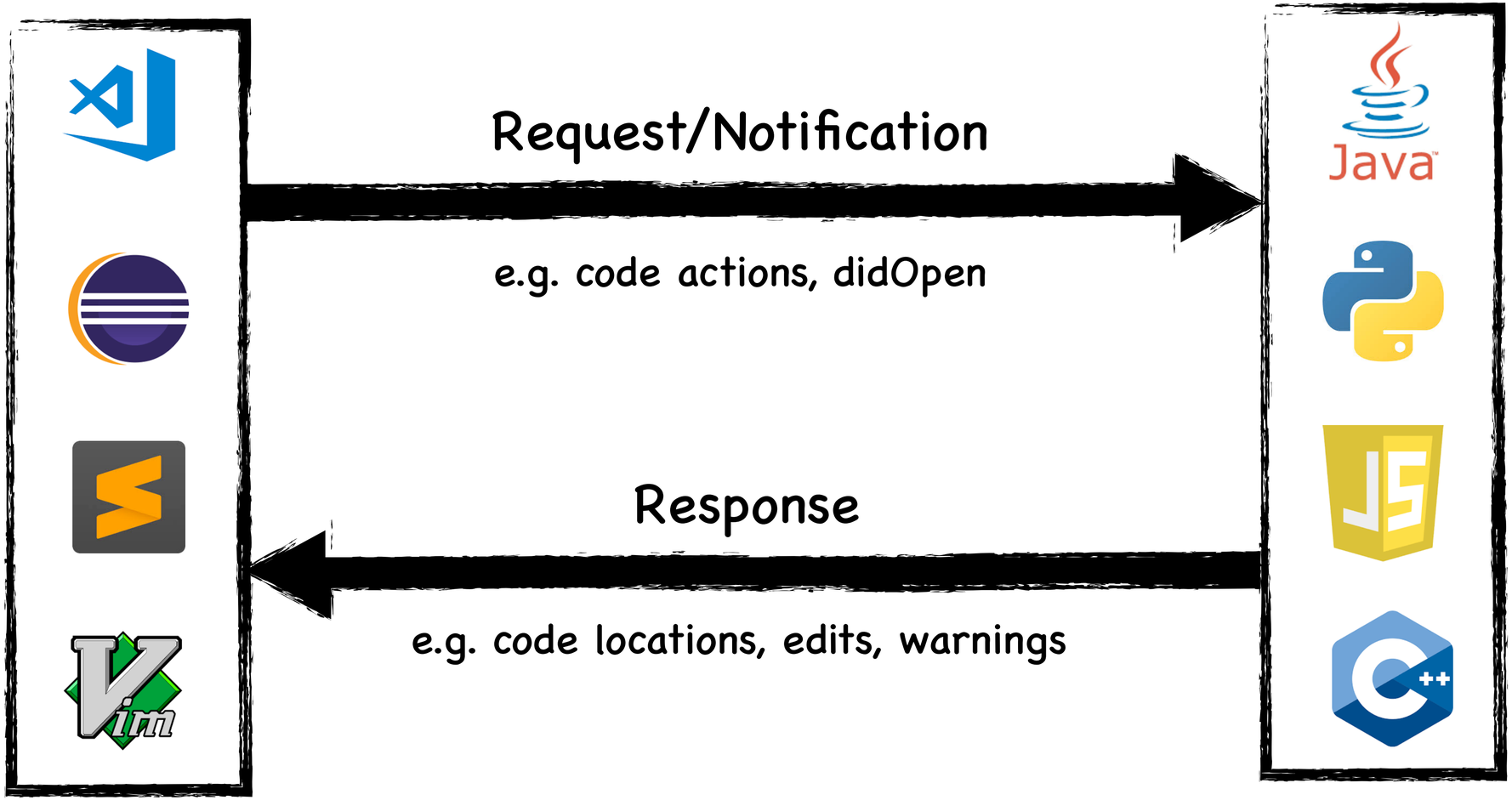
PRIDE'22

BERLIN

MagpieBridge



Language Server Protocol (LSP)



Developer Tool

Language Server

```
[Trace - 01:15:49 PM] Received notification 'textDocument/didOpen'
Params: {
  "textDocument": {
    "uri": "file:///Users/llinghui/Projects/testTargets/DemoProject/src/main/java/tutorial2/Example.java",
    "languageId": "java",
    "version": 1,
    "text": "package tutorial2;\nimport java.security.NoSuchAlgorithmException;\n\nimport javax.crypto.Cipher;\nimport javax.crypto.NoSuchPaddi
  }
}
```

A file is opened in IDE

```
[Trace - 01:15:49 PM] Sending notification 'window/showMessage'
Params: {
  "type": 3,
  "message": "First Analysis started analyzing the code."
}
```

To show a message in IDE

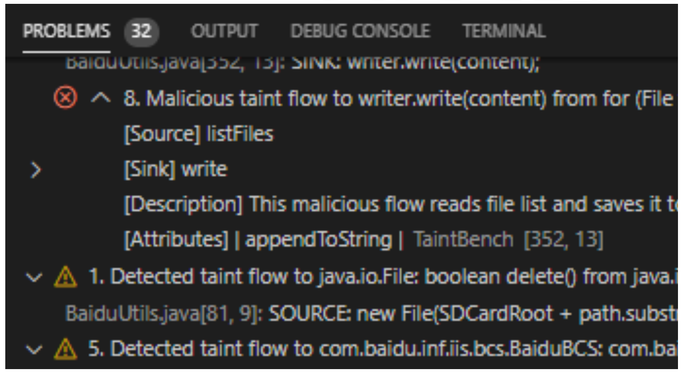
```
[Trace - 01:16:14 PM] Sending notification 'textDocument/publishDiagnostics'
Params: {
  "uri": "file:///Users/llinghui/Projects/testTargets/DemoProject/src/main/java/tutorial2/Example.java",
  "diagnostics": [
    {
      "range": {
        "start": {
          "line": 10,
          "character": 43
        },
        "end": {
          "line": 10,
          "character": 63
        }
      },
      "severity": 1,
      "code": "AES/ECB/PKCS5Padding\");",
      "source": "First Analysis",
      "message": "Insecure parameter! The first parameter with the value AES/ECB/PKCS5Padding containts ECB should be replaced by any of CBC, C
      "relatedInformation": []
    }
  ]
}
```

To display diagnostics in code

IDE Features for Diagnostics

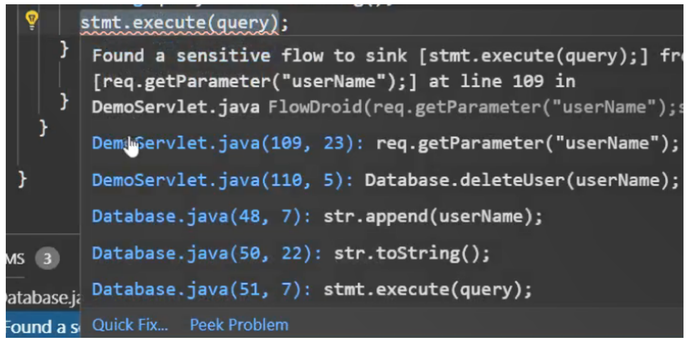
Markers with different severities

(ERROR, WARNING, INFO)



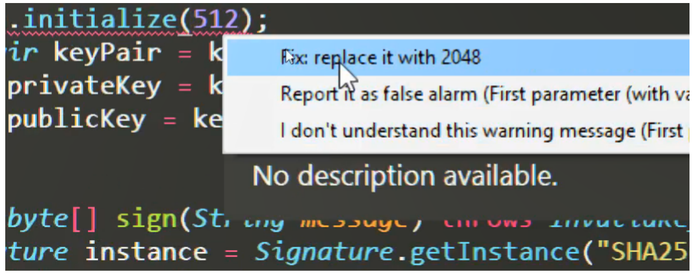
Related code in hover message

(Witness path)



Advanced features

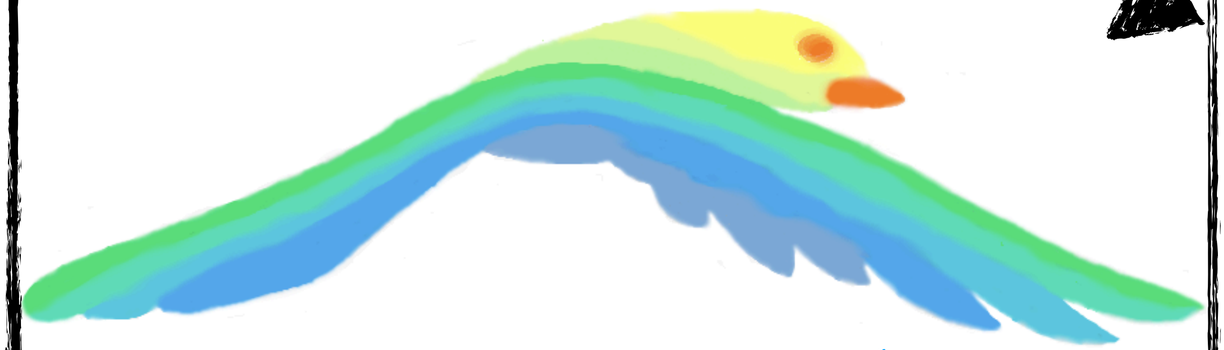
(Quick fixes, Open URI, False alarm reporting ...)



Developer Tool

A Reusable IDE Integration Framework

addAnalysis(Analysis, language)



MagpieBridge

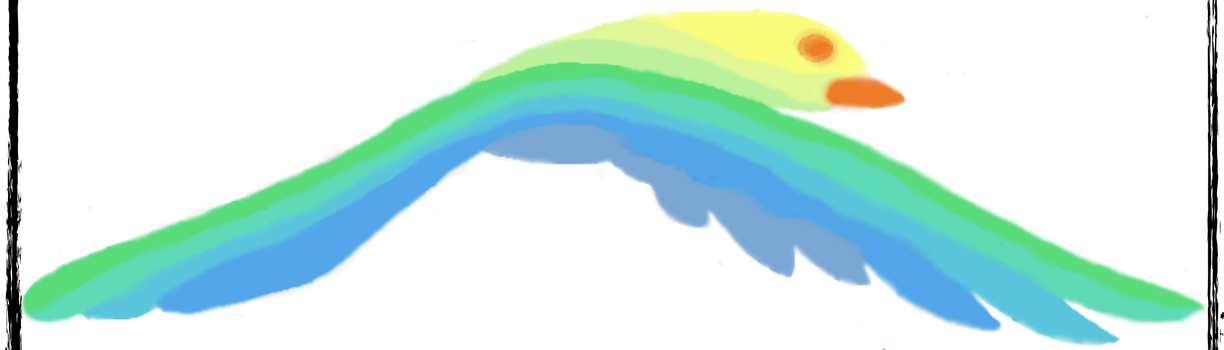
Analysis

Produces AnalysisResult (e.g message, source code position, related code) which can be converted into LSP messages

Developer Tool

MagpieServer

A Reusable IDE Integration Framework



MagpieBridge

ServerAnalysis

Analyses written in Java

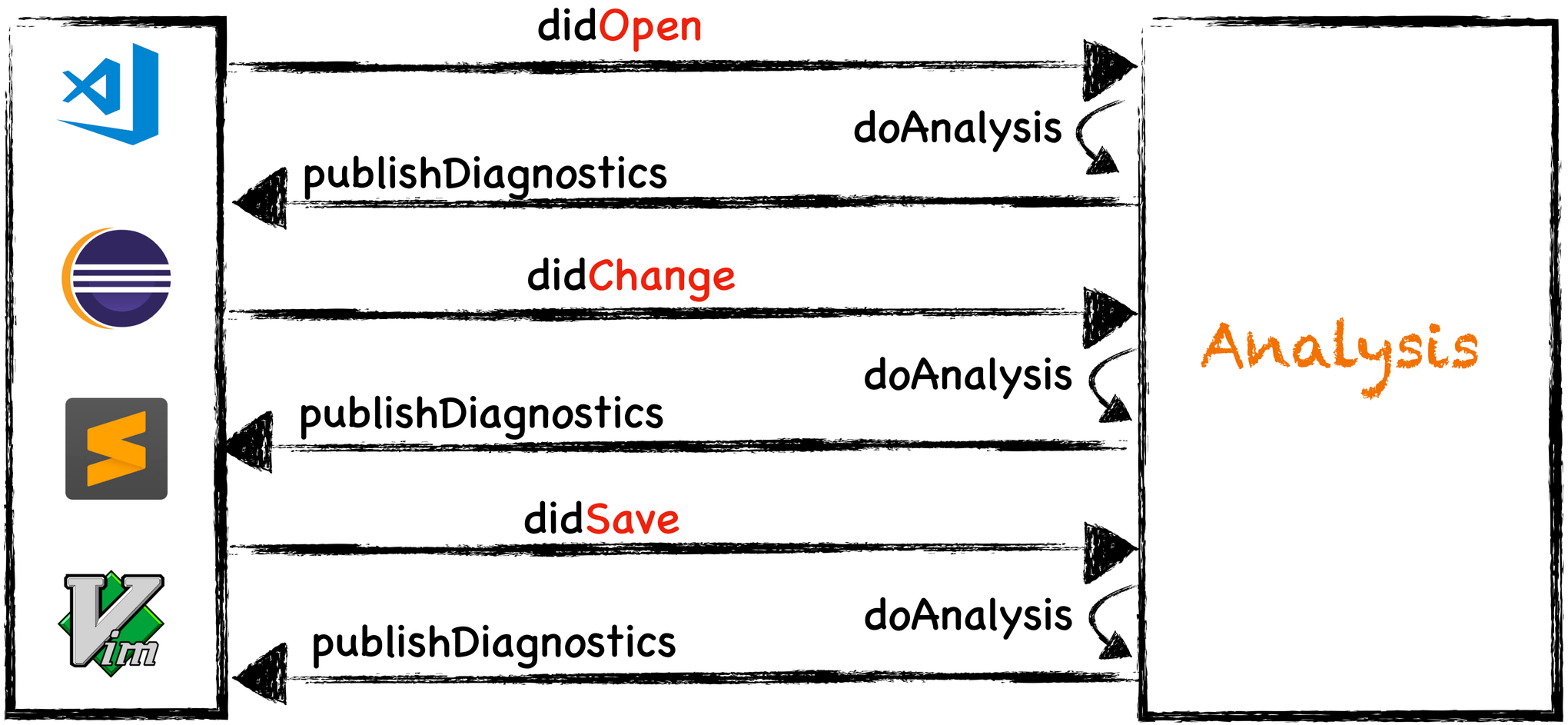
ToolAnalysis

Command Line Tools

Developer Tool

MagpieServer

Instant Detection



Developer Tool

MagpieServer

Task 1

- Display analysis results from a JSON file

{} preparedResults.json > ...

```
1   {
2       "results":[
3           {
4               "fileName": "Example.java",
5               "msg": "Insecure parameter! The first parameter with the value AES/EC
6               "startLine": 11,
7               "startColumn": 43,
8               "endLine":11,
9               "endColumn":63,
10              "repair":"AES/CBC/PKCS5Padding"
11          }
12      ]
13  }
```

Expected Outcome

```java
1   package tutorial2;
2   import java.security.NoSuchAlgorithmException;
3
4   import javax.crypto.Cipher;
5   import javax.crypto.NoSuchPaddingException;
6
7   public class Example{
8       public static void main(String... args)
9           String data = "some data";
10          String key = "secret key";
11          Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
12          Encrypt.encrypt(cipher, data, key);
13      }
14  }
```

Insecure parameter! The first parameter with the value AES/ECB/PKCS5Padding containts ECB should be replaced by any of CBC, GCM, PCBC, CTR, CTS, CFB or OFB. First Analysis(AES/ECB/PKCS5Padding");)

Peek Problem (⌥F8) No quick fixes available

```java
1   package tutorial2;
2   import java.security.NoSuchAlgorithmException;
3
4   import javax.crypto.Cipher;
5   import javax.crypto.NoSuchPaddingException;
6
7   public class Example{
8       public static void main(String... args) throws NoSuchAlgorithmEx
9           String data = "some data";
10          String key = "secret key";
11          Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
12          ey);
13      }
14  }
```

Fix: replace it with AES/CBC/PKCS5Padding

Create a MagpieServer

```
ServerConfiguration config = new ServerConfiguration();
// log the LSP messages between MagpieServer and IDE.
File traceFile = new File ("magpie_server_trace.lsp");
config.setLSPMessageTracer(new PrintWriter(traceFile));

MagpieServer server = new MagpieServer(config);

// either launch on Socket, good for debugging
MagpieServer.launchOnSocketPort(5007, () -> server);
// or launch on Standard I/O
server.launchOnStdio();
```

Add an Analysis

Add as many analyzers as you want

```
MagpieServer server = new MagpieServer(config);
String language = "java";
server.addAnalysis(Either.forLeft(new ServerAnalysis() {
    @Override
    public String source() {
        return “My Dummy Analysis";
    }

    @Override
    public void analyze(Collection<? extends Module> files,
                        AnalysisConsumer server,
                        boolean rerun) {

    }
}), language);
```

Files that have been touched in IDE

A server that consumes the analysis results

Indicates when to rerun the analysis

Add an Analysis

```
MagpieServer server = new MagpieServer(config);
String language = "java";
server.addAnalysis(Either.forLeft(new ServerAnalysis() {
    @Override
    public String source() {
        return “My Dummy Analysis";
    }

    @Override
    public void analyze(Collection<? extends Module> files,
                        AnalysisConsumer server,
                        boolean rerun) {
    MagpieServer magpieServer = (MagpieServer) server;
    if (rerun) {
        Set<AnalysisResult> results = doYourAnalysis(files);
        magpieServer.consume(results, source());
    }
}
}), language);
```

Fill up AnalysisResult

```
public class FirstResult implements AnalysisResult {
  public Kind kind(){ return Kind.Diagnostic; }


  public String toString(boolean useMarkdown){

  }

  public Position position(){
      URI + Code Range
  }


  public Iterable<Pair<Position, String>> related(){

  }


  public DiagnosticSeverity severity(){
         Return DiagnosticSeverity.ERROR;
  }

  public Pair<Position, String> repair(){

  }

  public String code(){

  }

}
```

```
"uri": "file:///Users/linghuiluo/Projects/testProject/src/Example.java"
```

preparedResults.json > ...

```
1   {
2       "results":[
3           {
4               "fileName": "Example.java",
5               "msg": "Insecure parameter! The first parameter with the valu
6               "startLine": 11,
7               "startColumn": 43,
8               "endLine":11,
9               "endColumn":63,
10              "repair":"AES/CBC/PKCS5Padding"
11          }
12      ]
13  }
```

Useful Classes:

magpiebridge.util.SourceCodeInfoFinder
magpiebridge.util.SourceCodeReader
magpiebridge.util.SourceCodePositionUtils
magpiebridge.util.FileUtils
magpiebridge.core.analysis.dataflow.sarif.SARIFToAnalysisResultConverter
magpiebridge.core.analysis.dataflow.sarif.AnalysisResultToSARIFConverter

Configure MagpieServer to Analyze

```
ServerConfiguration config = new ServerConfiguration();

config.setDoAnalysisByOpen(true);

MagpieServer server = new MagpieServer(config);
```

```
    config.setDo
m setDoAnalysisByFirstOpen(boolean doAnalysisByFirstOpen)
m setDoAnalysisBySave(boolean doAnalysisBySave)
m setDoAnalysisByOpen(boolean doAnalysisByOpen)
m setDoAnalysisByIdle(boolean doAnalysisByIdle, long timeOut)
```

Task 2

- Do taint analysis on whole project and provides feedbacks
- Display data-flow graph for each analysis finnding

Expected Outcome

```
3   public class SecondExample {
4     public static void main(String... args) {
5       Demo demo = new Demo();
6       String s1 = demo.source();
7       String s2 = "foo" + s1;
8       String s3= "bar" + s2;
9       demo.sink(s3);
10      View data-flow graph: http:/127.0.0.1:51925/flow
11  }
12
```

Project Information

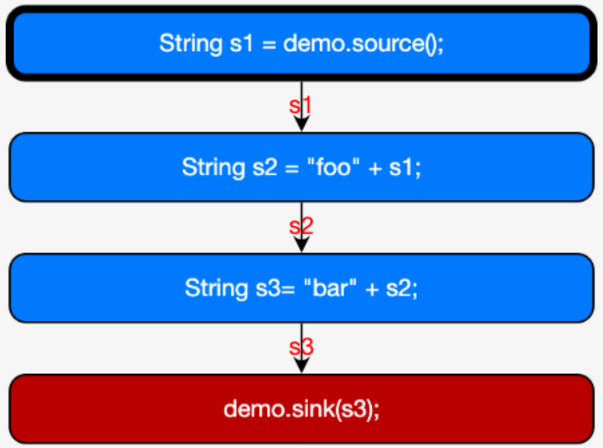
Project Name

SecondExample.java
(4 locations)

- Line 6
- Line 7
- Line 8
- Line 9

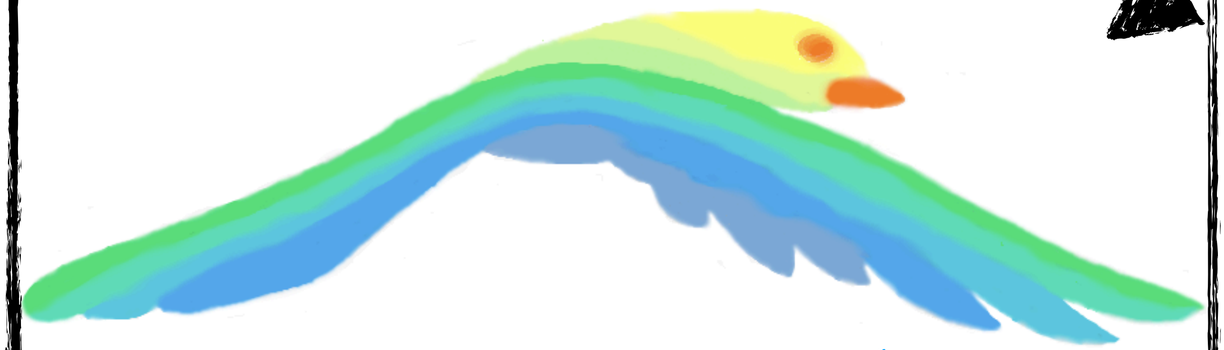
Welcome to MagpieBridge Data-Flow Page!

Error: Find a taint flow from String s1 = demo.source(); to demo.sink(s3);



Reusable Project Services

addProjectService(language, IProjectService)



MagpieBridge

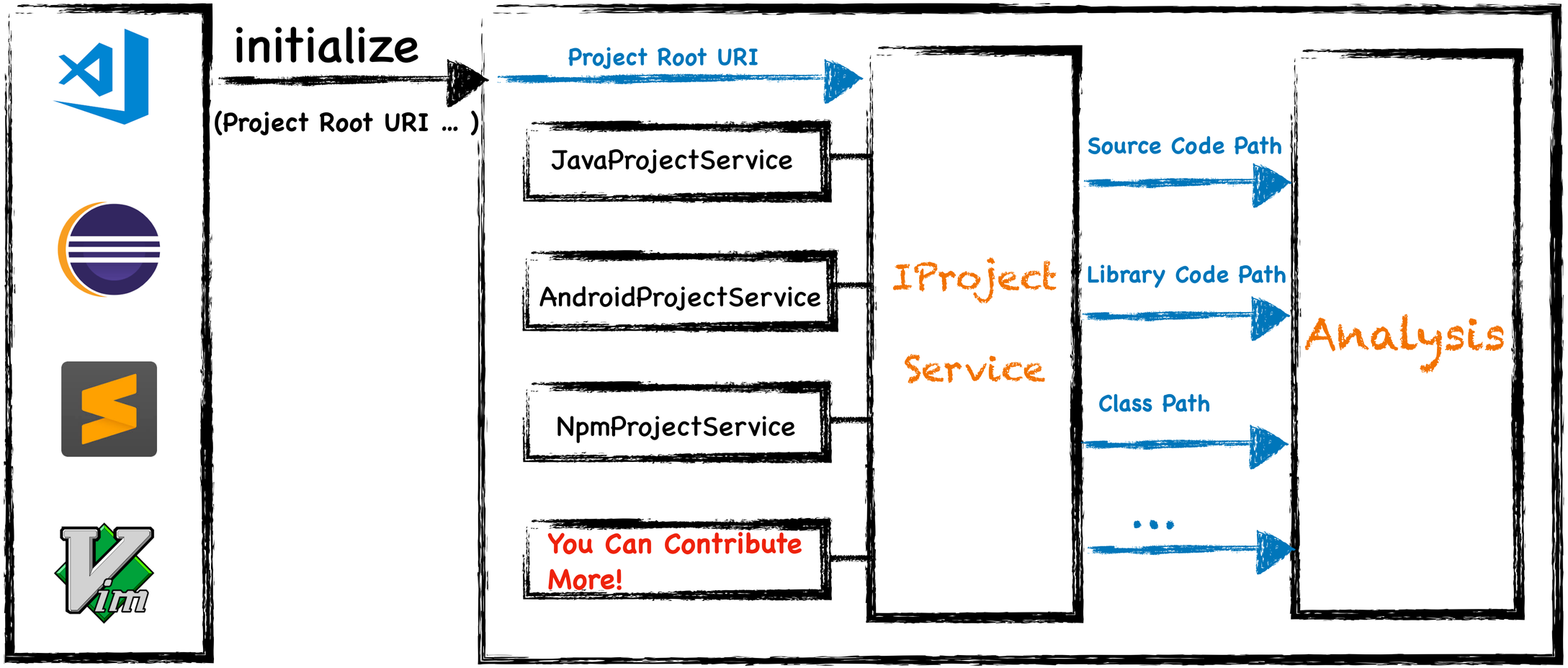
IProject Service

Resolves project scope by leveraging popular build tools (e.g. Maven, Gradle, NPM)

Developer Tool

MagpieServer

Reusable Project Services



Developer Tool

MagpieServer

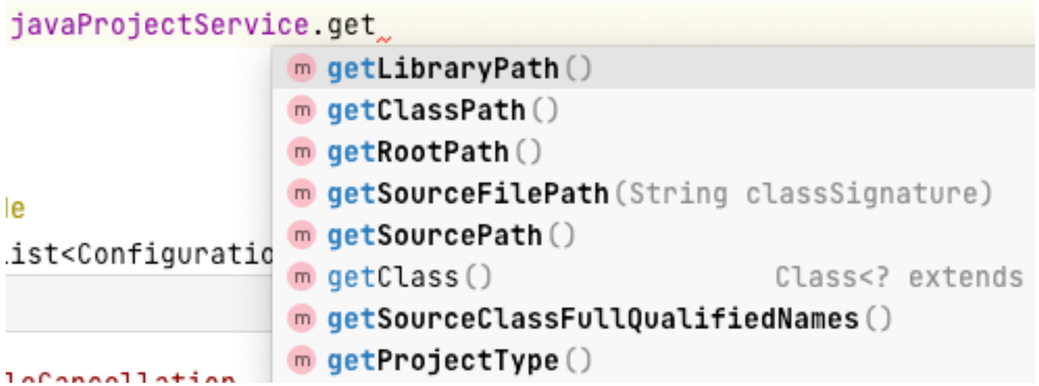
Add a ProjectService

```
MagpieServer server = new MagpieServer(config);
IProjectService javaProjectService = new JavaProjectService();
server.addProjectService(language, javaProjectService);
```

Get Path Information

```
new ServerAnalysis() {
    @Override
    public void analyze(Collection<? extends Module> files,
                        AnalysisConsumer server,
                        boolean rerun) {
    MagpieServer magpieServer = (MagpieServer) server;

    JavaProjectService ps = (JavaProjectService) magpieServer
                                                 .getProjectService("java").get();
    Set<Path> classPath = ps.getClassPath();
    Set<Path> libPath = ps.getLibraryPath();
    Set<AnalysisResult> results = doYourAnalysis(classPath, libPath);
    magpieServer.consume(results, source());
    }
}
```



Use FlowAnalysisResult

New feature since version 0.1.5

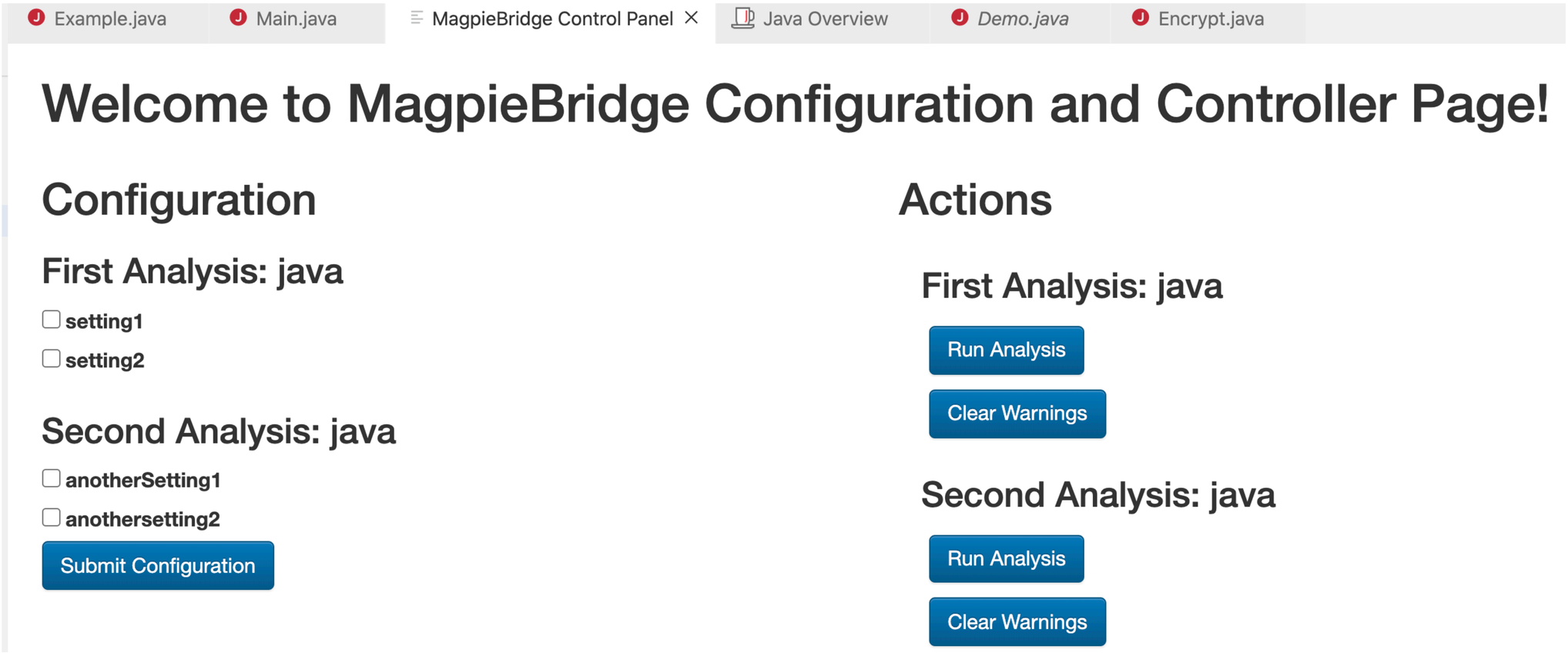
```
Set<AnalysisResult> results = new HashSet<>();
...
results.add(new FlowAnalysisResult(kind, sink.toPosition(),
   msg, related, DiagnosticSeverity.Error, repair, sink.code));
```

statements that are on the data-flow path

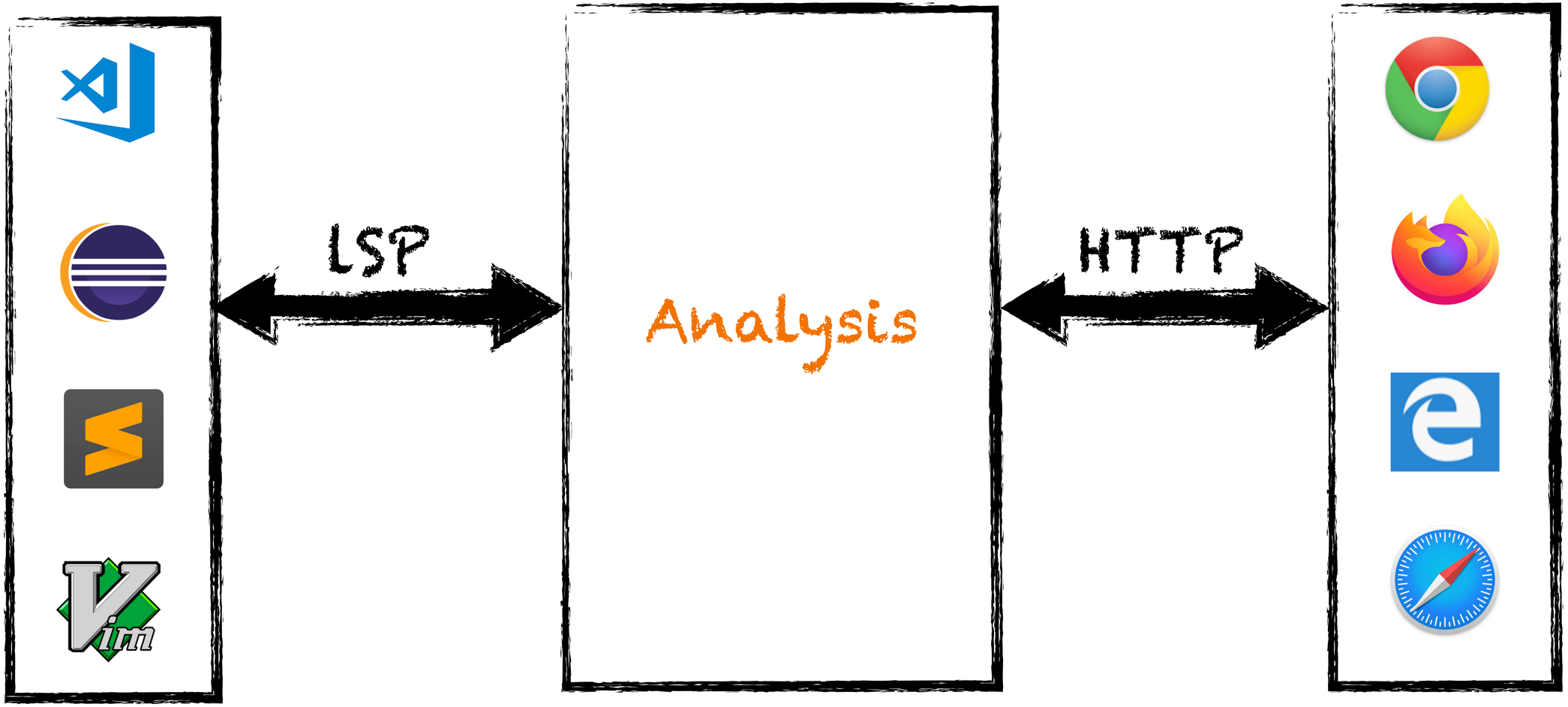
```
ServerConfiguration config = new ServerConfiguration();
config.setShowDataFlowGraph(true);
MagpieServer server = new MagpieServer(config);
```

Task 3

- Provide configuration and allow user control over the analyses.



MagpieBridge Provides More User Control

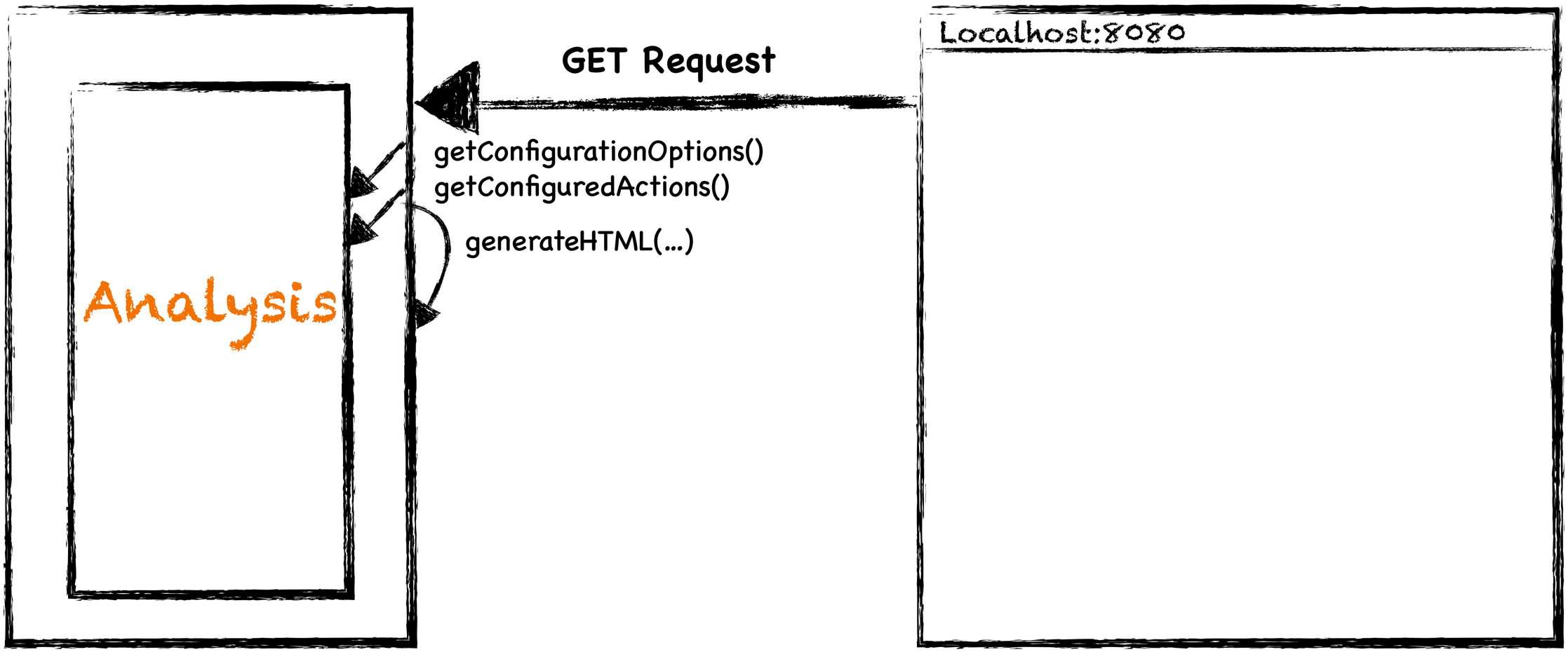


Developer Tool

MagpieServer

Web Browser

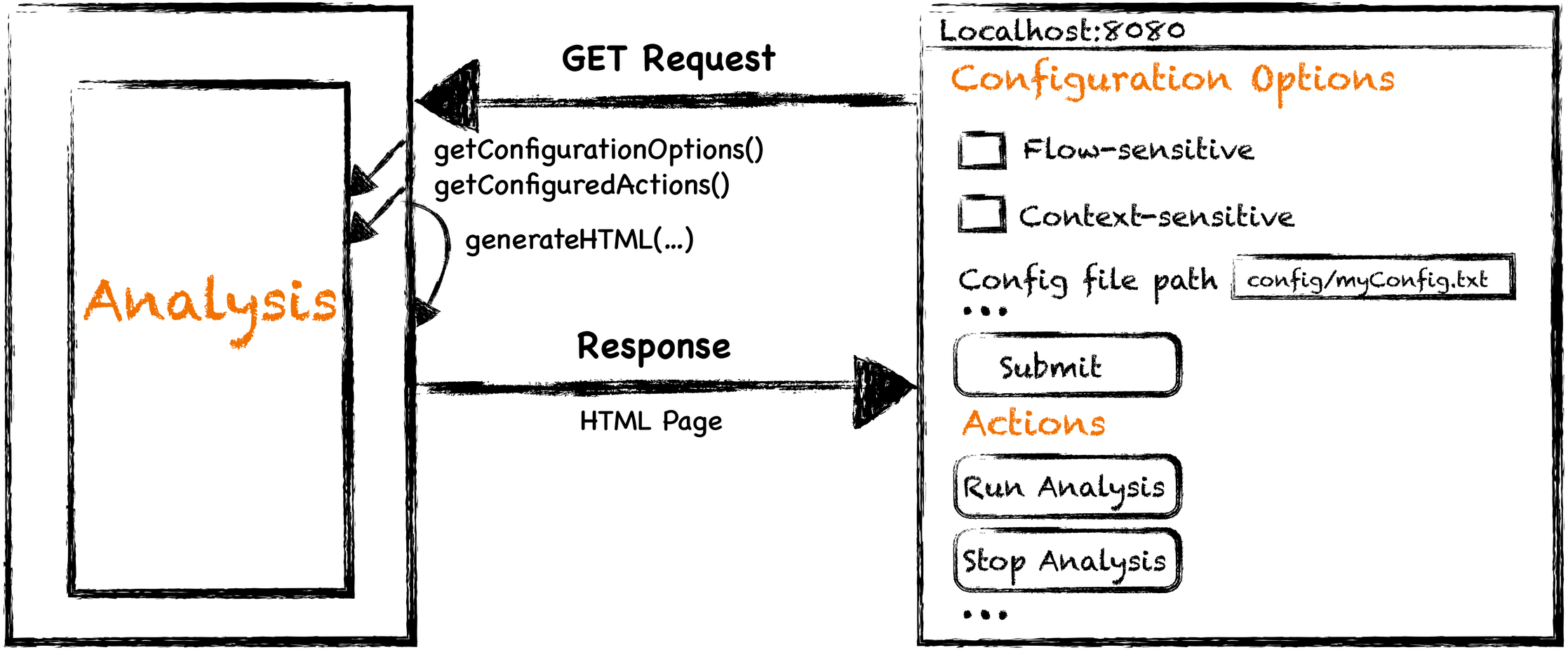
MagpieBridge Provides More User Control



MagpieServer

Web Browser

MagpieBridge Provides More User Control



MagpieServer

Web Browser

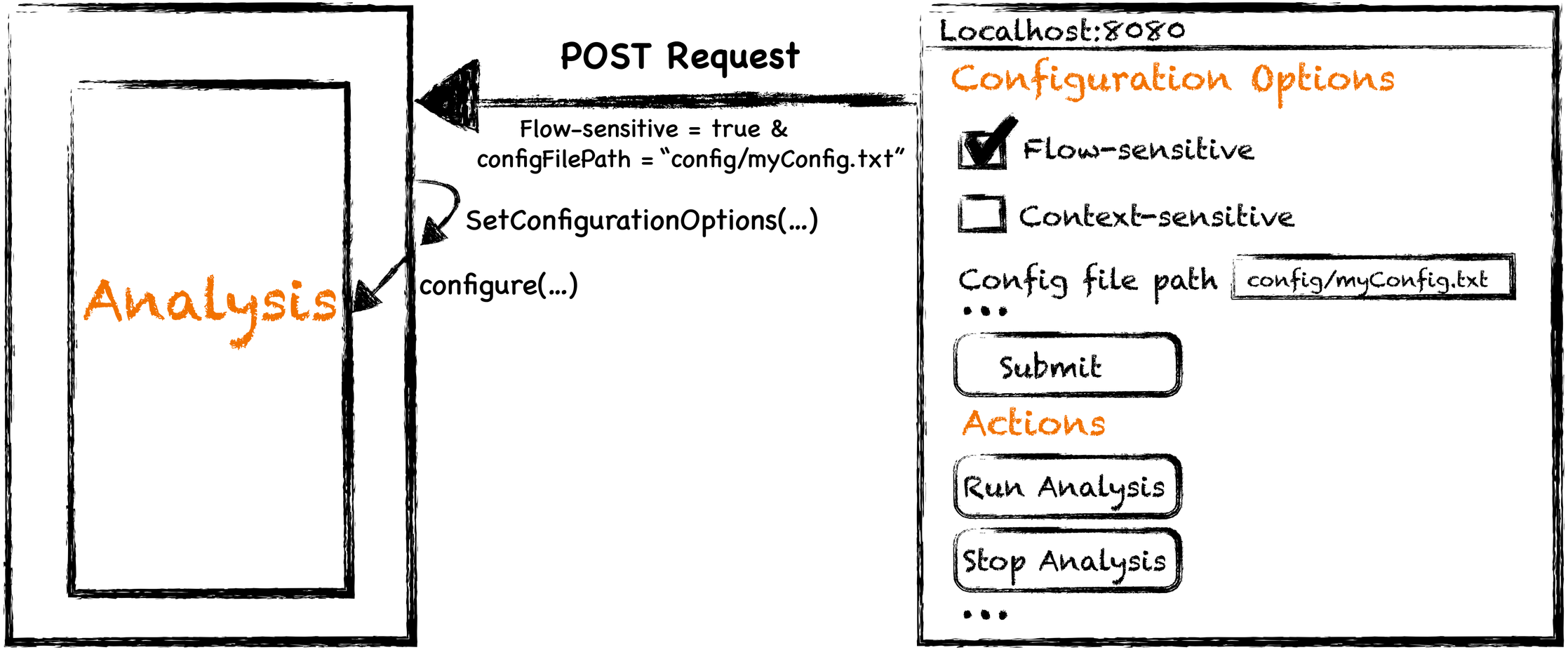
Setup Analysis Configuration Options

The HTML page is rendered from the provided options.

```
new ServerAnalysis() {
@Override
public List<ConfigurationOption> getConfigurationOptions() {
  ConfigurationOption op1 = new ConfigurationOption("setting1", OptionType.checkbox);
  ConfigurationOption op2 = new ConfigurationOption("setting2", OptionType.checkbox);
  List<ConfigurationOption> options = new ArrayList<ConfigurationOption>();
  options.add(op1);
  options.add(op2);
  return options;
}
```

text input, container etc.

MagpieBridge Provides More User Control



MagpieServer

Web Browser

Setup Analysis Configuration Options

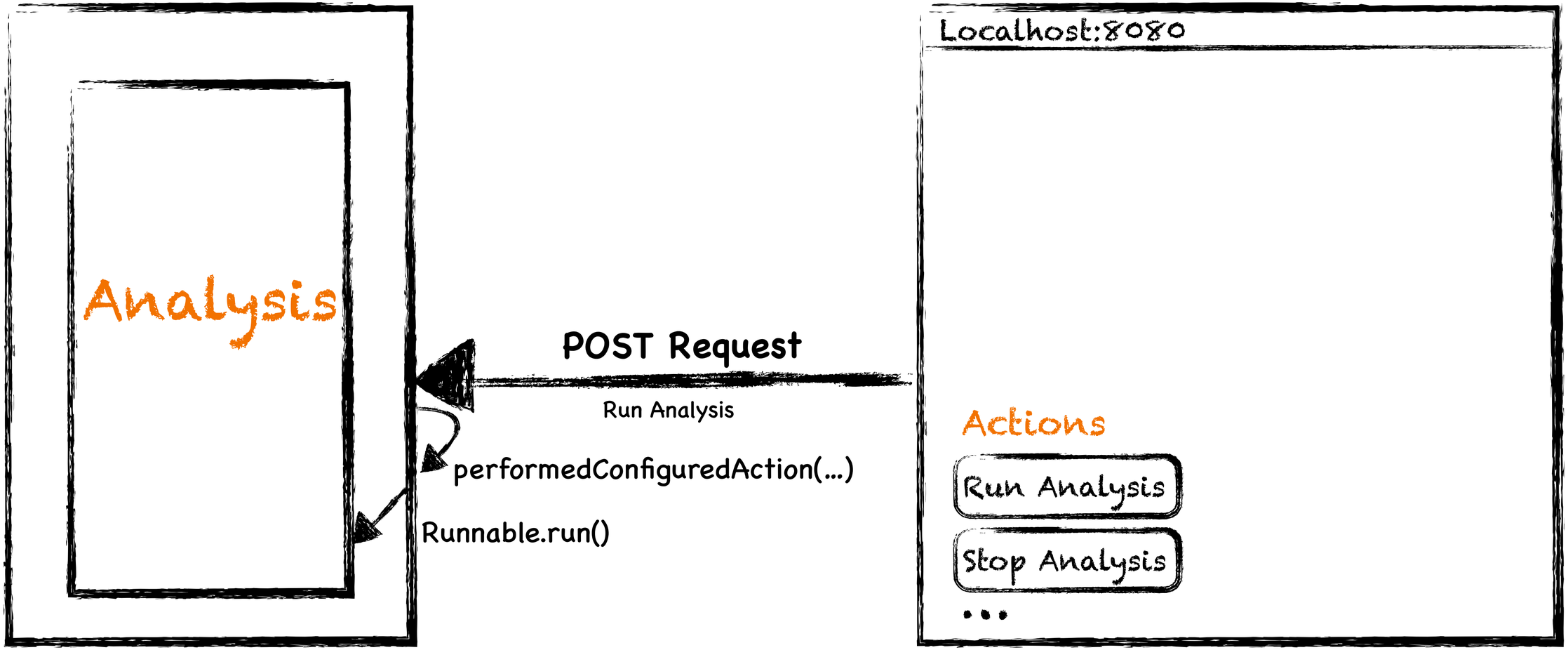
```
new ServerAnalysis() {


@Override
public void configure(List<ConfigurationOption> configuration) {
  // TODO Configure your analysis.
  }

}
```

Configured options from the HTML page

MagpieBridge Provides More User Control



Implement Actions For Buttons

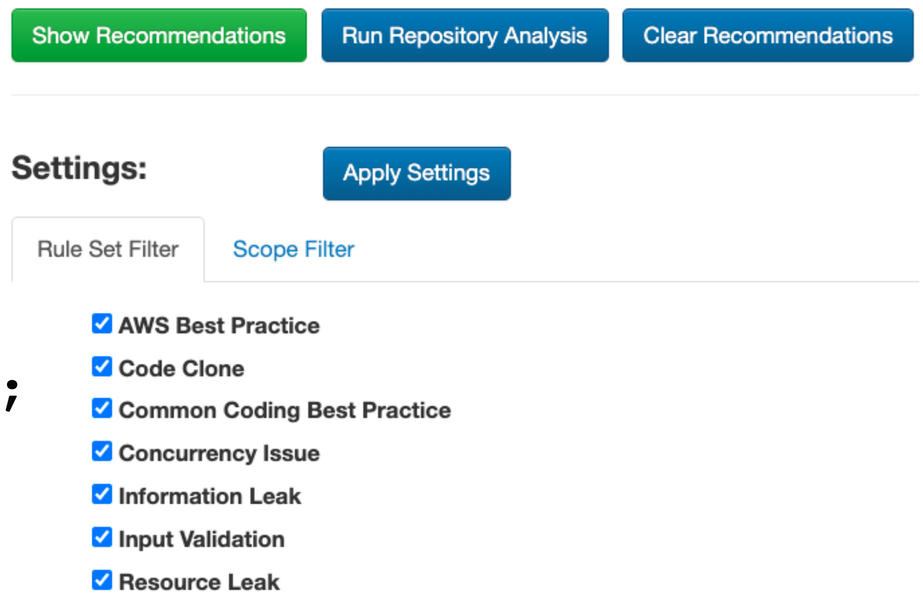
A runnerable object

```
new ServerAnalysis() {
@Override
public List<ConfigurationAction> getConfiguredActions() {
  List<ConfigurationAction> actions = new ArrayList<ConfigurationAction>();
  ConfigurationAction clearWarnings = new ConfigurationAction("Clear Warnings", () -> {
    final String msg = source() + " warnings are cleared. ";
    this.magpieServer.forwardMessageToClient(new MessageParams(MessageType.Info, msg));
    this.magpieServer.cleanUp();
  });
  actions.add(clearWarnings);
  return actions;
}
}
```

Add “Run Analysis” button by default

```
ServerConfiguration config = new ServerConfiguration();
config.setShowConfigurationPage(true, true);
MagpieServer server = new MagpieServer(config);
```

Want a Nice Page?




```
MagpieServer server = new MagpieServer(config);
String url = createYourHTTPSever();
server.addHttpServer(url);
```

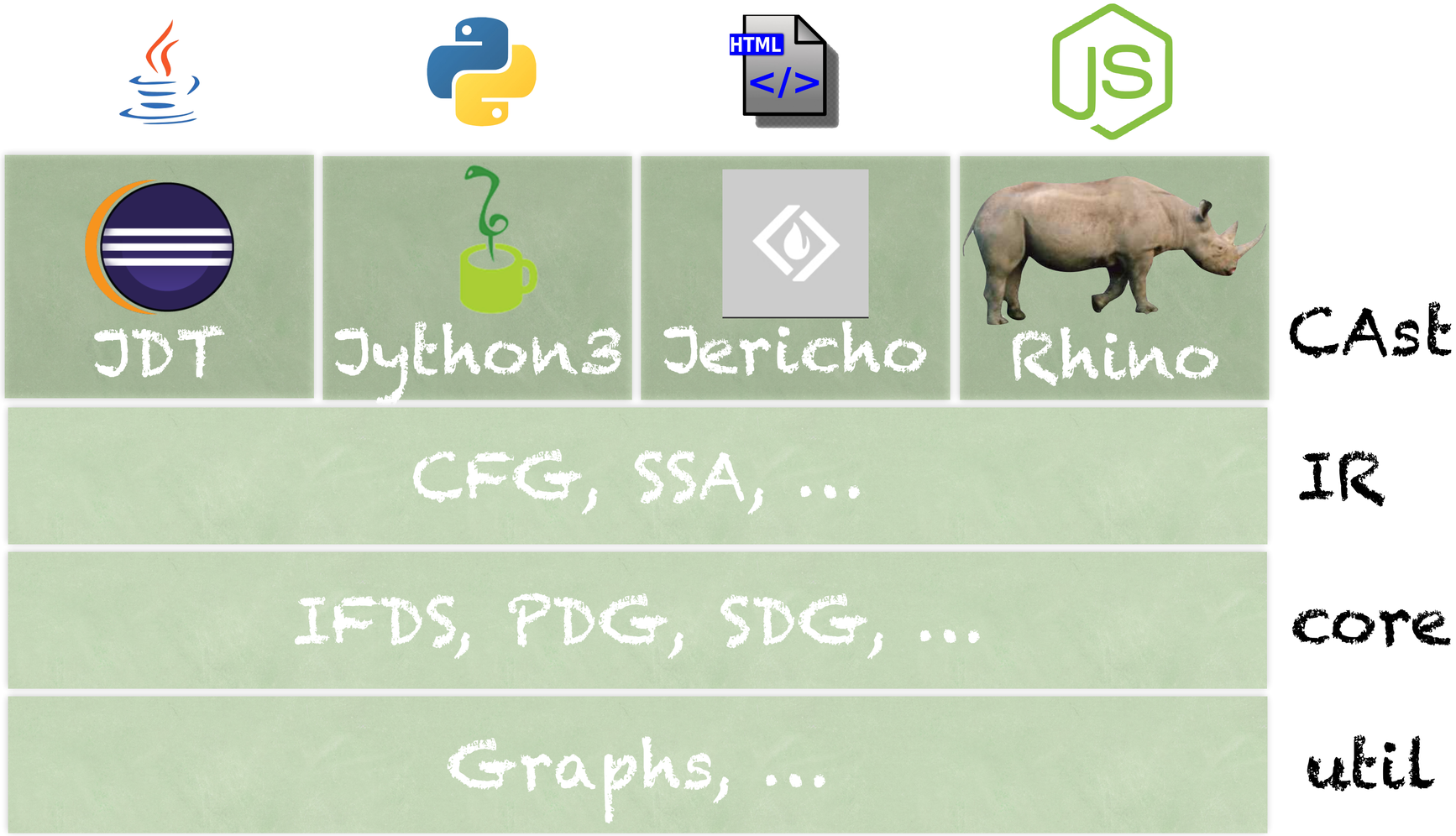
The default HTTP server:

magpiebridge.core.analysis.configuration.MagpieHttpServer

Task 4

- Write taint analysis in WALA and integrate with MagpieBridge

WALA



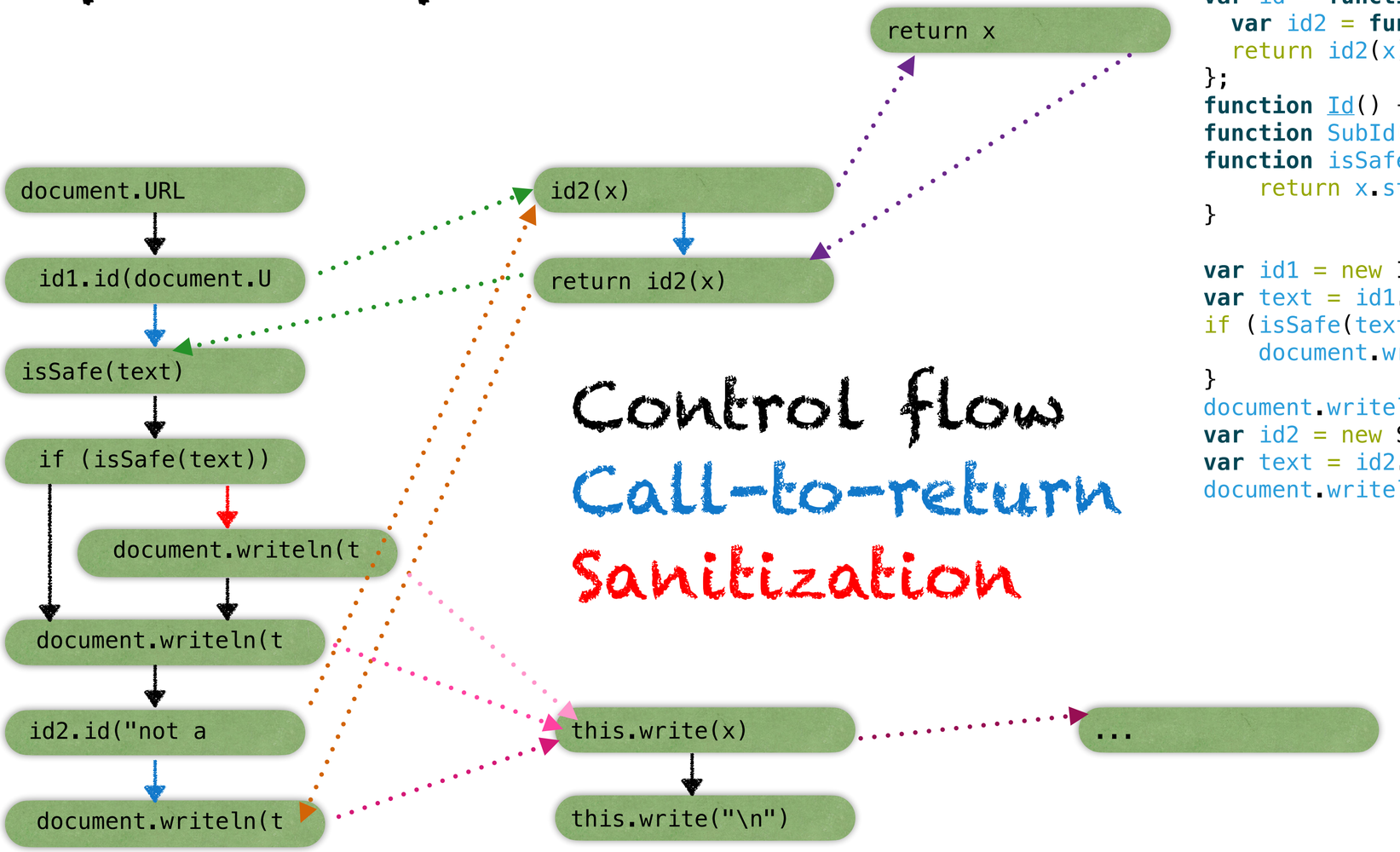
https://github.com/wala/WALA

Example

```
var document = {
  URL: "whatever",
    writeln: function Document_prototype_writeln(x) {
      this.write(x).
      this.write("\n")
    },
    write: function Document_prototype_write(x) { } };
var id = function _id(x) {
  var id2 = function _id2(x) { return x };
  return id2(x)
};
function Id() { this.id = id }
function SubId() { }; SubId.prototype = new Id()
function isSafe(x) {
    return x.startsWith("https:");
}

var id1 = new Id();
var text = id1.id(document.URL);
if (isSafe(text)) {
    document.writeln(text);
}
document.writeln(text);
var id2 = new SubId();
var text = id2.id("not a url");
document.writeln(text);
```

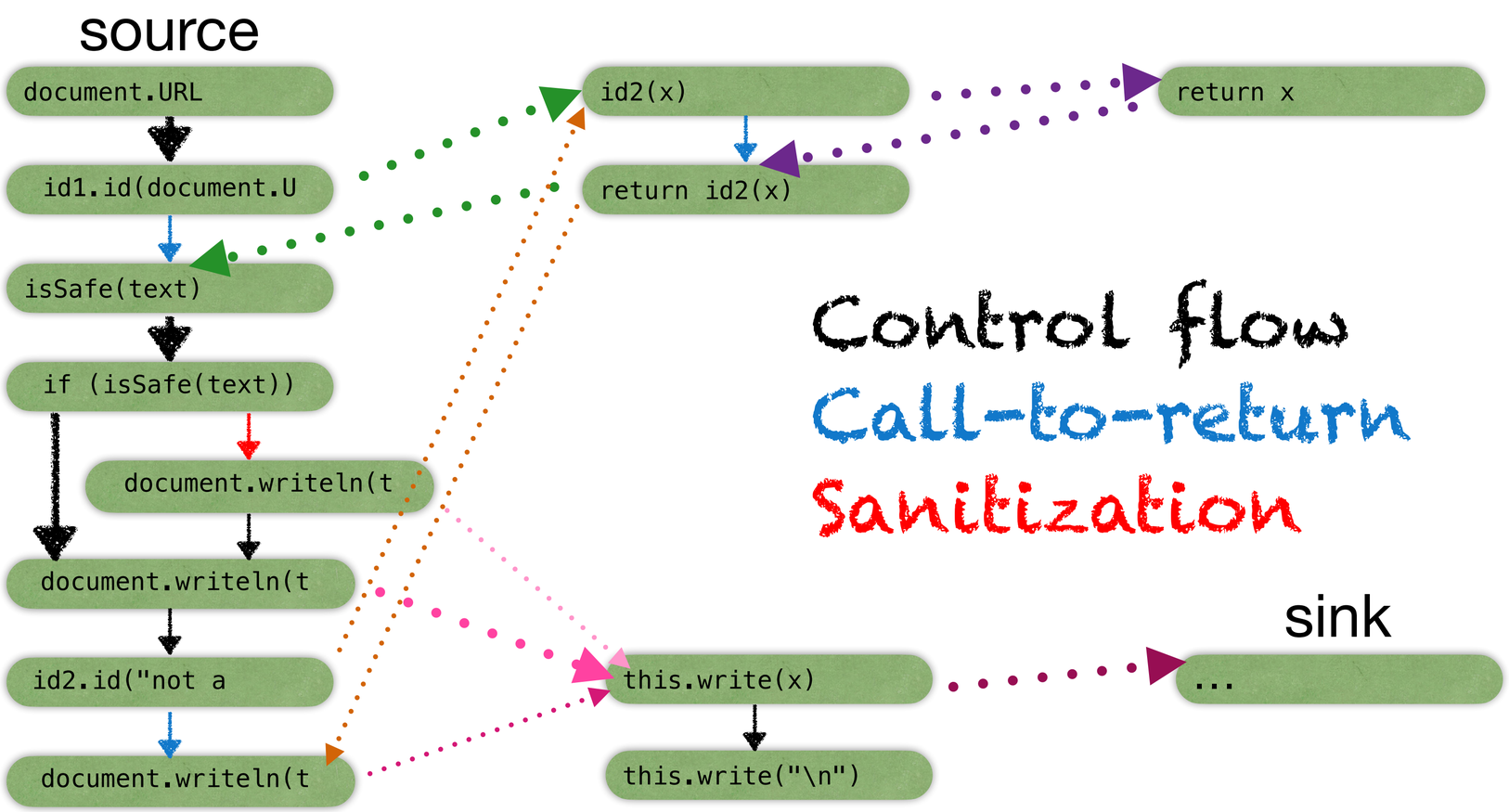
Supergraph




```
var document = {
  URL: "whatever",
    writeln: function Document_prototype_writeln(x) {
       this.write(x).
       this.write("\n")
    },
    write: function Document_prototype_write(x) { } };
var id = function _id(x) {
  var id2 = function _id2(x) { return x };
  return id2(x)
};
function Id() { this.id = id }
function SubId() { }; SubId.prototype = new Id()
function isSafe(x) {
    return x.startsWith("https:");
}

var id1 = new Id();
var text = id1.id(document.URL);
if (isSafe(text)) {
    document.writeln(text);
}
document.writeln(text);
var id2 = new SubId();
var text = id2.id("not a url");
document.writeln(text);
```

IFDS Supergraph



IFDS in WALA

```
public void analyze(Collection<? extends Module> files, AnalysisConsumer server, boolean rerun) {
    try {
      builder = makeBuilder(files, server);
      callGraph = builder.makeCallGraph(builder.getOptions());
      ISupergraph<BasicBlockInContext<IExplodedBasicBlock>, CGNode> supergraph =
        ICFGSupergraph.make(callGraph);
      ZhinuDomain domain = new ZhinuDomain();

      PartiallyBalancedTabulationSolver<
              BasicBlockInContext<IExplodedBasicBlock>,
              CGNode,
              Pair<Integer, List<CAstSourcePositionMap.Position>>> solver =
          PartiallyBalancedTabulationSolver.createPartiallyBalancedTabulationSolver(
              new ZhinuProblem(supergraph, domain, this, callGraph), null);
      TabulationResult<BasicBlockInContext<IExplodedBasicBlock>, CGNode,
              Pair<Integer, List<CAstSourcePositionMap.Position>>>
          R = solver.solve ();

      Set<List<CAstSourcePositionMap.Position>> result = HashSetFactory.make();
      Collection<BasicBlockInContext<IExplodedBasicBlock>> reached = R.getSupergraphNodesReached();
      reached.forEach((sbb) -> {
            if (sinkFinder().apply(sbb)) {
              SSAInstruction inst = sbb.getDelegate().getInstruction();
              CAstSourcePositionMap.Position sink =
                  ((AstMethod) sbb.getMethod()).debugInfo().getInstructionPosition(inst.iIndex());
              R.getResult(sbb).foreach(i -> {
                  Pair<Integer, List<CAstSourcePositionMap.Position>> trace = domain.getMappedObject(i);
                  for (int j = 0; j < inst.getNumberOfUses(); j++) {
                    if (inst.getUse(j) == trace.fst) {
                      List<CAstSourcePositionMap.Position> x = new LinkedList<>(trace.snd);
                      x.add(sink);
                      result.add(x);
      }}}); }});
```

paths

IFDS Flow Functions

```
public IUnaryFlowFunction getReturnFlowFunction(
      BasicBlockInContext<IExplodedBasicBlock> call,
      BasicBlockInContext<IExplodedBasicBlock> src,
      BasicBlockInContext<IExplodedBasicBlock> dest) {
    int retVal = call.getLastInstruction().getDef();
    return (d1) -> {
      Pair<Integer, List<CAstSourcePositionMap.Position>> vn =
        domain.getMappedObject(d1);
      MutableIntSet result = IntSetUtil.make();
      supergraph.getPredNodes(src).forEachRemaining(pb -> {
        SSAInstruction inst = pb.getDelegate().getInstruction();
        if (inst instanceof SSAReturnInstruction && inst.getUse(0) == vn.fst) {
          List<CAstSourcePositionMap.Position> x = new LinkedList<>(vn.snd);
          if (src.getMethod() instanceof AstMethod) {
            CAstSourcePositionMap.Position next =
              ((AstMethod) src.getMethod()).debugInfo()
                .getInstructionPosition(inst.iIndex());
            if (next != null) {
              x.add(next); } }
          Pair<Integer, List<CAstSourcePositionMap.Position>> key=Pair.make(retVal, x);
          if (!domain.hasMappedIndex(key)) {
            domain.add(key); }
          result.add(domain.getMappedIndex(key)); } });
      return result; }; }
```

Results to MagpieBridge

```
Set<AnalysisResult> ars = HashSetFactory.make();
result.forEach(lp -> { lp.forEach(step -> {
  ars.add(
    new AnalysisResult() {

      @Override
      public Kind kind() {
        return Kind.Diagnostic; }

      @Override
      public String toString(boolean useMarkdown) {
        return "tainted flow step”; }

      @Override
      public CAstSourcePositionMap.Position position() {
        return Print.fixIncludedURL(step); }

      @Override
      public Iterable<Pair<CAstSourcePositionMap.Position, String>> related() {
        Set<Pair<CAstSourcePositionMap.Position, String>> trace = HashSetFactory.make();
        lp.forEach(e -> {
          trace.add(Pair.make(e, "flow step”)); });
        return trace; }

      @Override
      public DiagnosticSeverity severity() {
        return DiagnosticSeverity.Warning; }

      @Override
      public String code() {
        return new SourceBuffer(position()).toString();
      } } ) } ) } );
server.consume(ars, source());
```

Developing with MagpieBridge

```
ServerAnalysis analysis = new JSWalaIFDSTaintAnalysis(uriCache);
Supplier<MagpieServer> createServer = () -> { …
      MagpieServer server = new MagpieServer(config);
      Either<ServerAnalysis, ToolAnalysis> either = Either.forLeft(analysis);
      server.addAnalysis(either, “javascript”);
      return server;
    };
if (args.length > 0) {
  AnalysisConsumer ac = new AnalysisConsumer() {
        @Override
        public void consume(Collection<AnalysisResult> results, String s) {
          results.forEach(ar -> {
                Position pos = ar.position();
                try {
                  SourceBuffer sb = new SourceBuffer(pos);
                  System.err.println(ar.toString(false)+" '"+sb+"' “+pos); …
        }
      };

  Set<Module> modules = HashSetFactory.make();
  for (String arg : args) {
    File f = new File(arg);
    modules.add(new SourceFileModule(f, f.getName(), null));
  }

  analysis.analyze(modules, ac, false);
} …
} else
  createServer.get().launchOnStdio(); …
```

MagpieBridge IDE taint analysis

demog.js

UNREGISTERED

demog.js

```javascript
var document = {
    URL: "whatever",
    writeln: function Document_prototype_writeln(x) {
      this.write(x)
      this.write("\n")
    },
    write: function Document_prototype_write(x) { } };
var id = function _id(x) {
  var id2 = function _id2(x) { return x };
  return id2(x)
};
function Id() { this.id = id }
function SubId() { }; SubId.prototype = new Id()
function isSafe(x) {
    return x.startsWith("https:");
}

var id1 = new Id();
var text = id1.id(document.URL);
if (isSafe(text)) {
    document.writeln(text);
}

var text2 = id1.id("not a url");
document.writeln(text2);

document.writeln(text);
```

walajstaint, Line 27, Column 24 main 233 Spaces: 4 JavaScript

MagpieBridge command line taint

TutorialWithWALA — -zsh — 89×10

...ta — -zsh | ...~ — -zsh | ...A — -zsh | ...~ — -zsh | ...~ — -zsh

```
(base) dolby@Julians-MBP-2 TutorialWithWALA % java -cp target/TutorialWithWALA-1.0-SNAPSH
OT.jar magpiebridge.Main data/demog.js 2>/dev/null
tainted flow step 'document.URL' demog.js [460->472] (line 19)
tainted flow step 'id1.id(document.URL)' demog.js [453->473] (line 19)
tainted flow step 'id2(x)' demog.js [272->278] (line 10)
tainted flow step 'return x' demog.js [249->257] (line 9)
tainted flow step 'return id2(x)' demog.js [265->278] (line 10)
tainted flow step 'document.writeln(text)' demog.js [585->607] (line 27)
tainted flow step 'this.write(x)' demog.js [95->108] (line 4)
(base) dolby@Julians-MBP-2 TutorialWithWALA %
```

Tutorial Code:

https://github.com/MagpieBridge/Tutorials-PRIDE-2022

Want more? Visit

https://github.com/MagpieBridge

to find Tutorials and Examples