Production-Driven Patch Generation

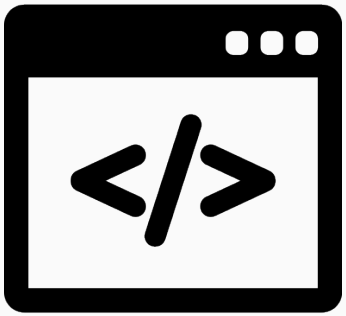
Thomas Durieux[1], Youssef Hamadi[2], Martin Monperrus[1]

May 25, 2017

[1]INRIA & University of Lille, France [2]Ecole Polytechnique, France

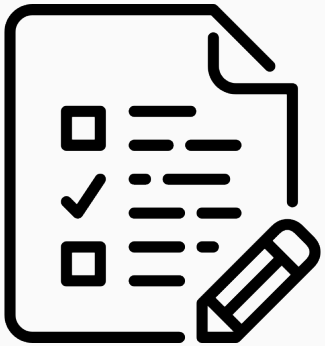
Windows sends more than 100 million error reports per day!

Automatic Program Repair



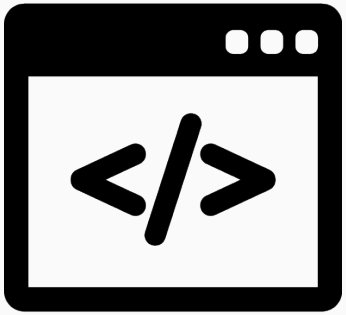
Buggy Application

Repair Strategy



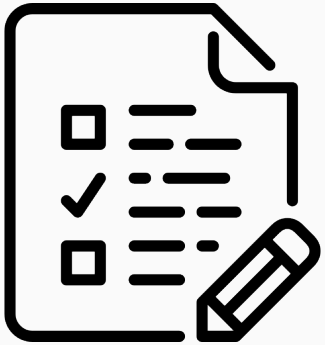
Oracle (e.g: Crash)

Offline - Automatic Program Repair



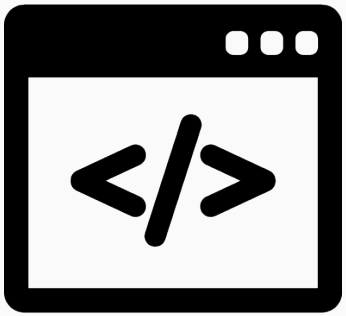
Buggy Program

GenProg,
Nopol, ...



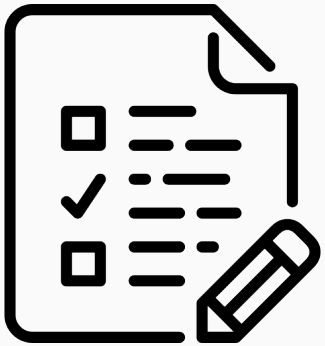
Failure Oracle:
Failing Tests
Regression Oracle:
Passing Tests

Online - Automatic Program Repair



Running Program

Repair Strategy



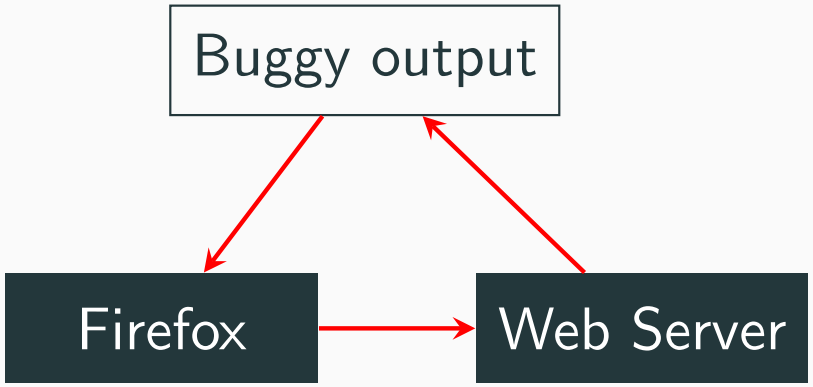
Failure Oracle: ?

Regression Oracle: ?

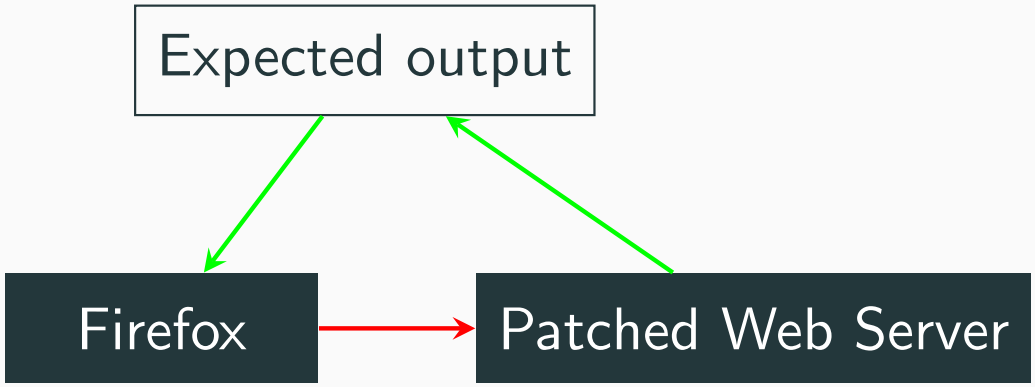
Outline

Itzal Concept

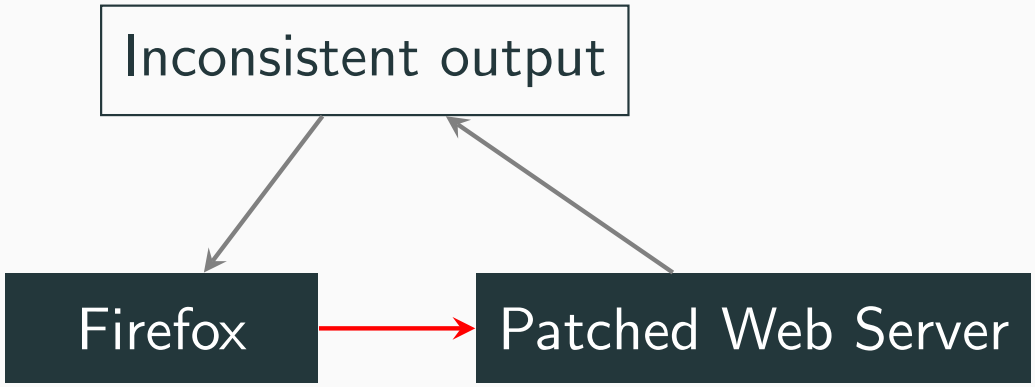
Itzal Concept



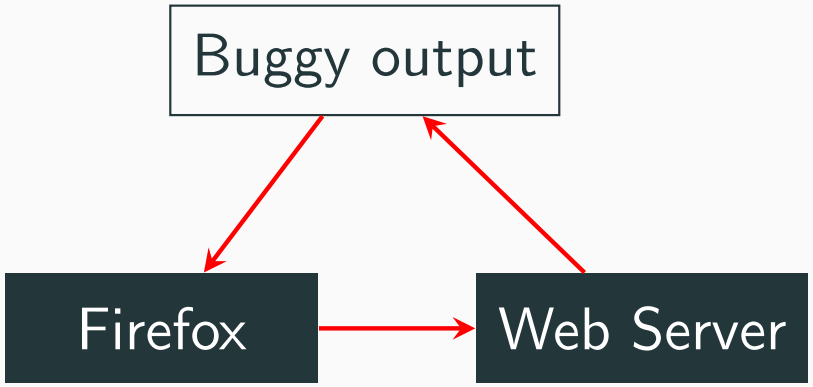
Itzal Concept



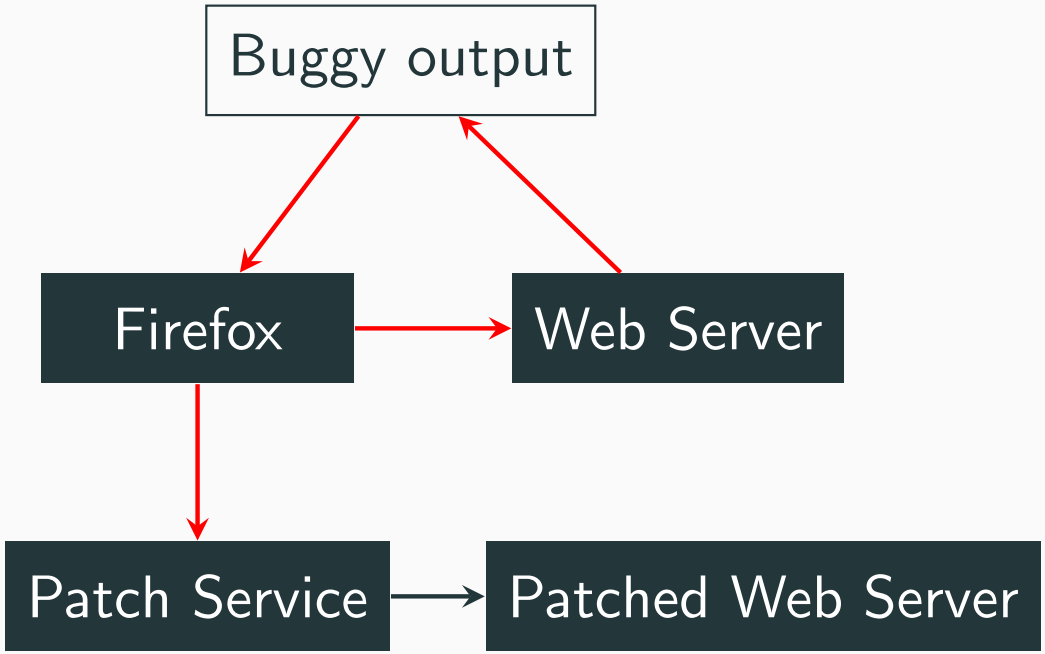
Itzal Concept



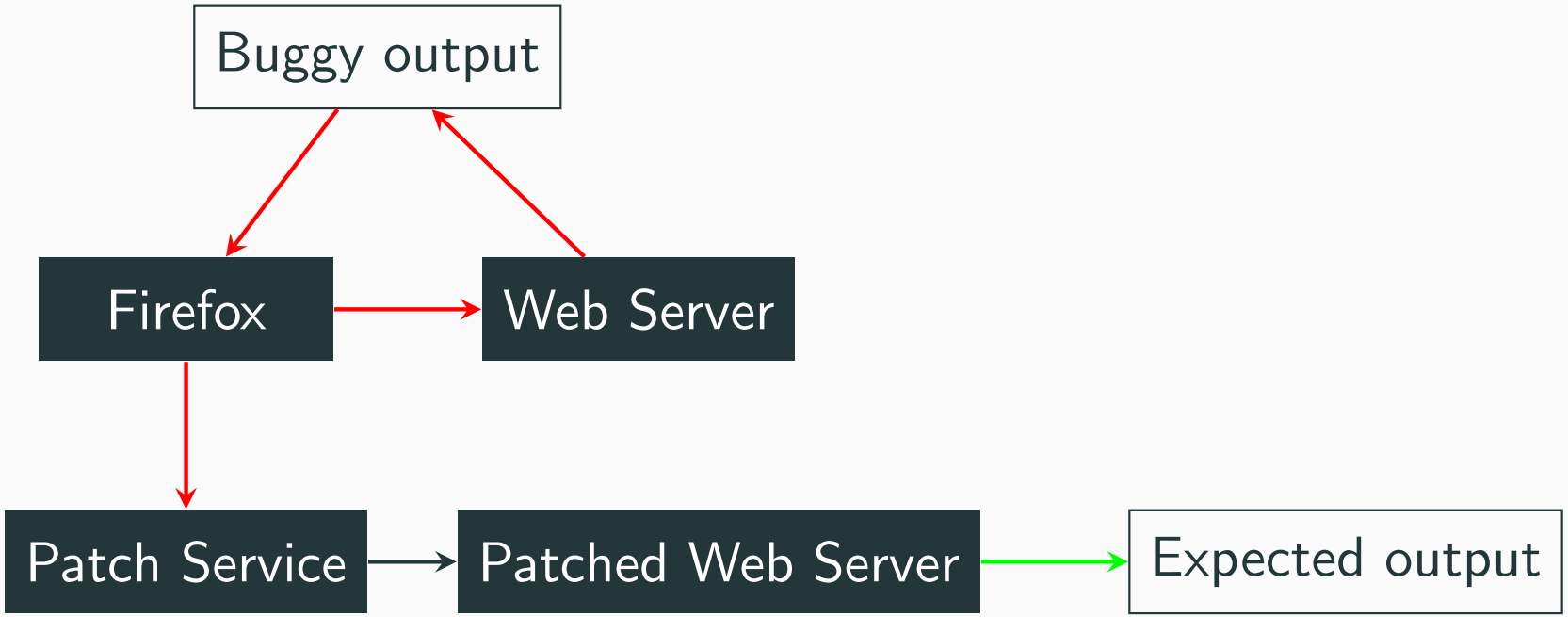
Itzal Concept



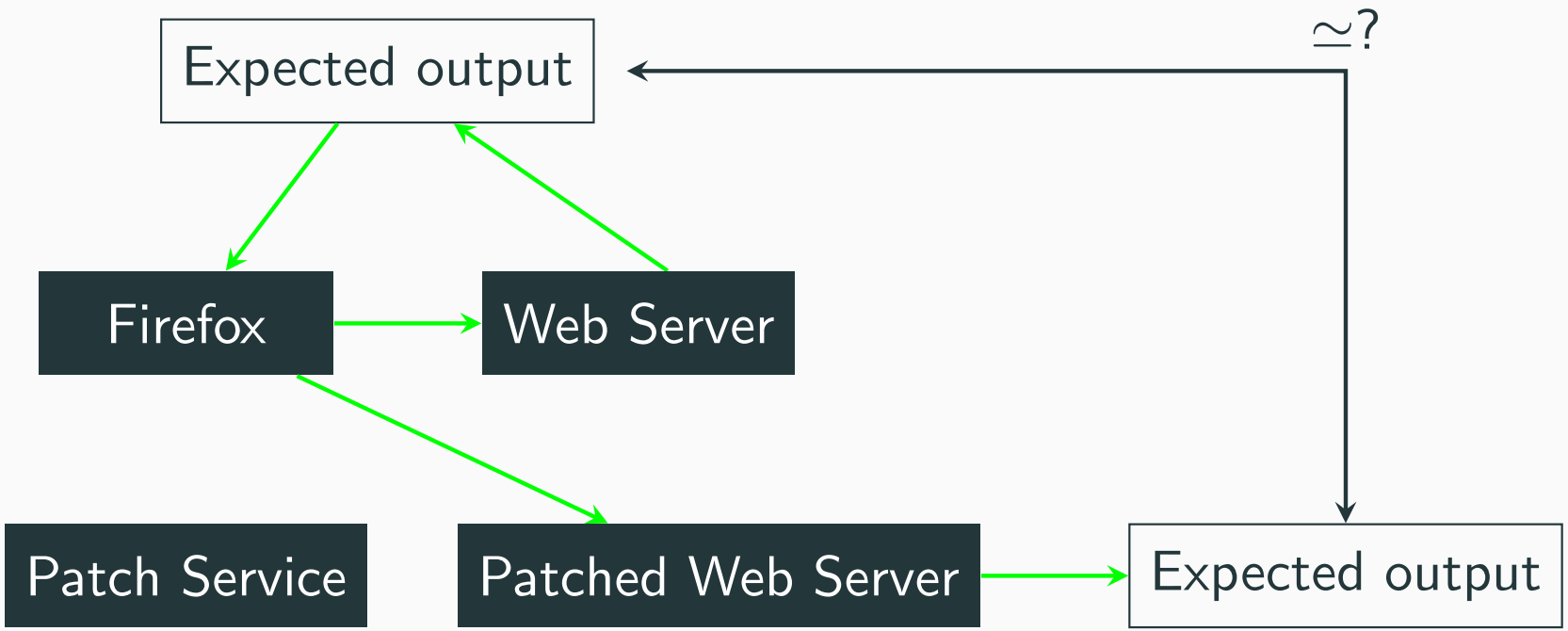
Itzal Concept



Itzal Concept



Itzal Concept



Itzal Concept

Highlights

- Patch generation in production.
- Patch regression with production inputs.
- Sandboxed repair environment repair the program without affecting the production.

Requirement

A request based application to reproduce the requests in a duplicated environment.

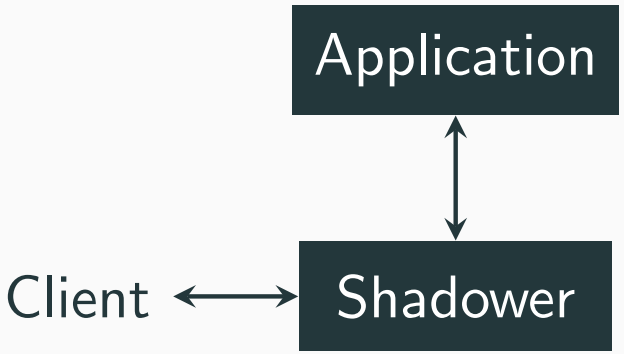
Outline

Itzal Architecture

Client: e.g. a browser

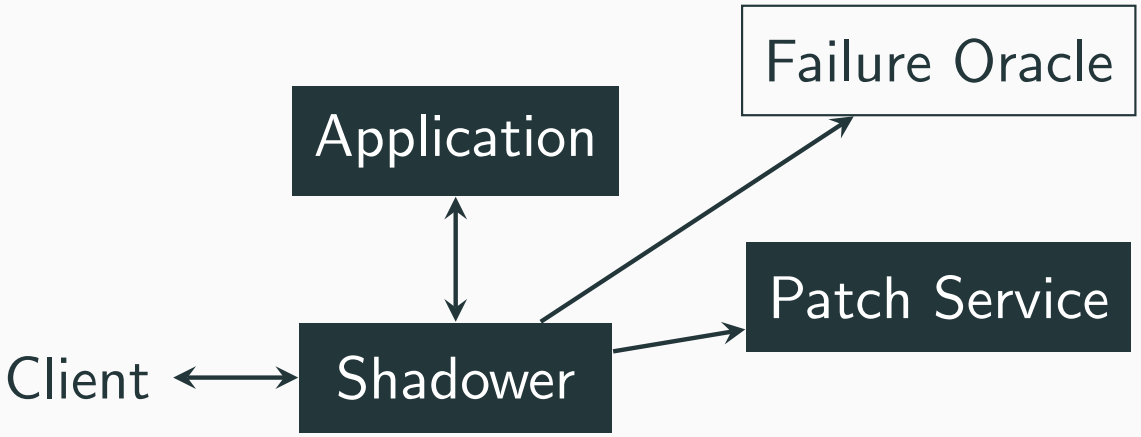
Application: e.g. a web server

Itzal Architecture



Shadower: intercepts and duplicates the requests

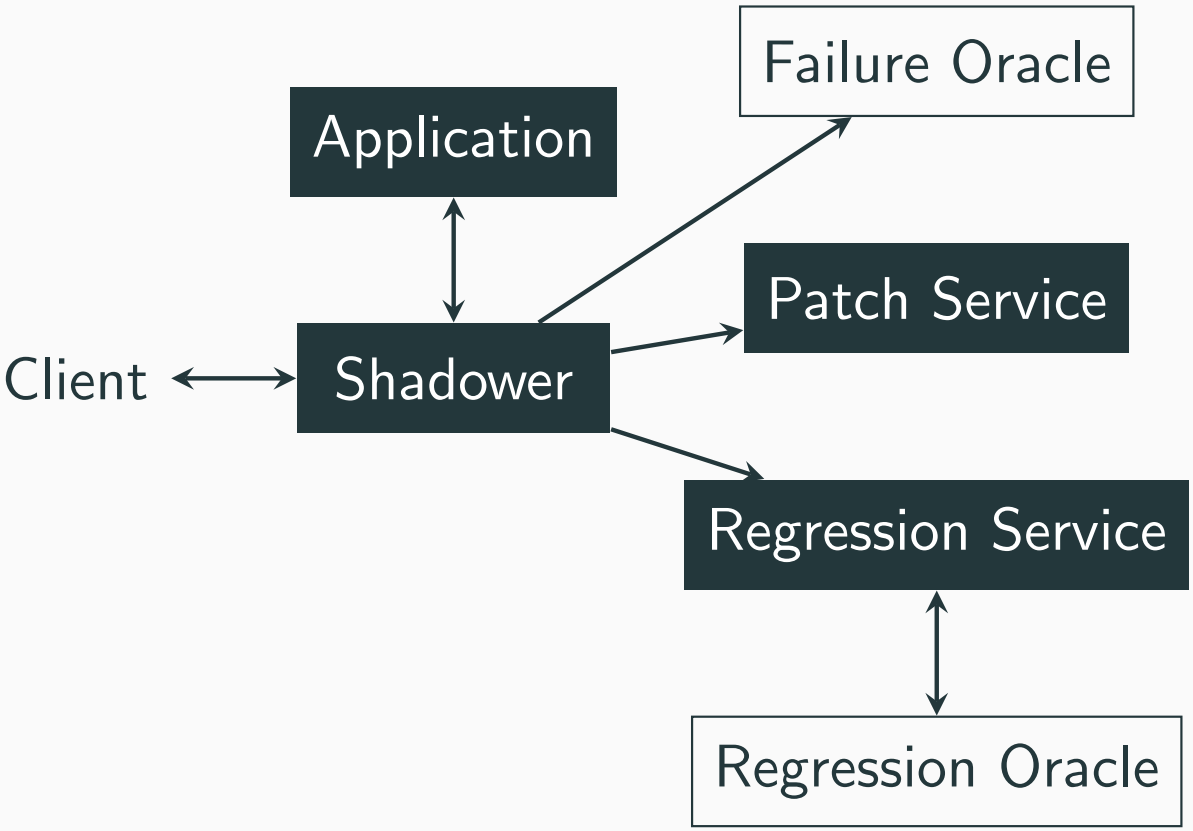
Itzal Architecture



Patch Service: generates patches that fix the requests

Failure Oracle: detects if a request is passing or failing

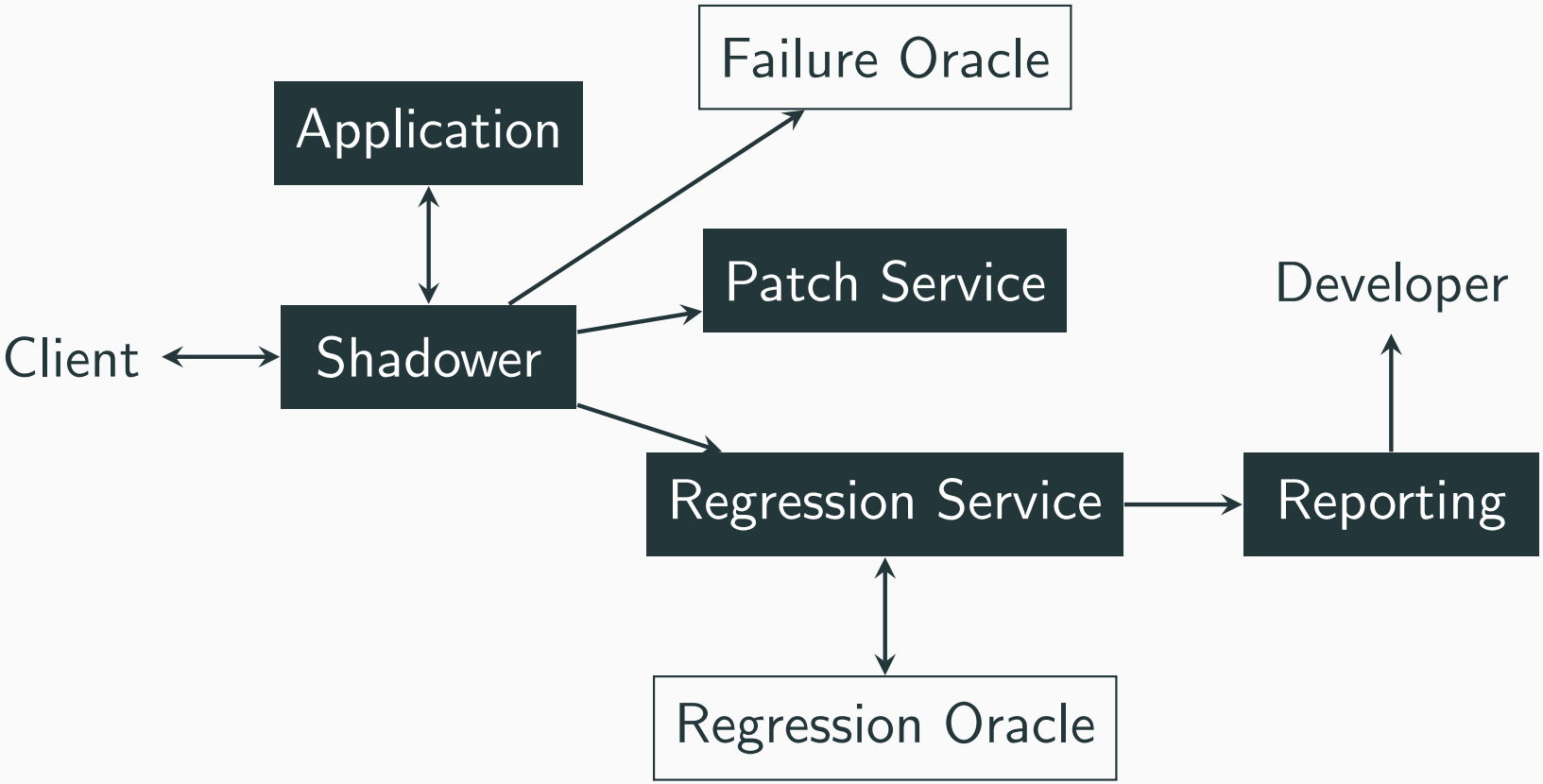
Itzal Architecture



Regression: executes passing request on patched applications

Regression Oracle: compares the output of the application and the patched application

Itzal Architecture



Reporting: communicates the patches to the developers (Dashboard, Pull Request, ...)

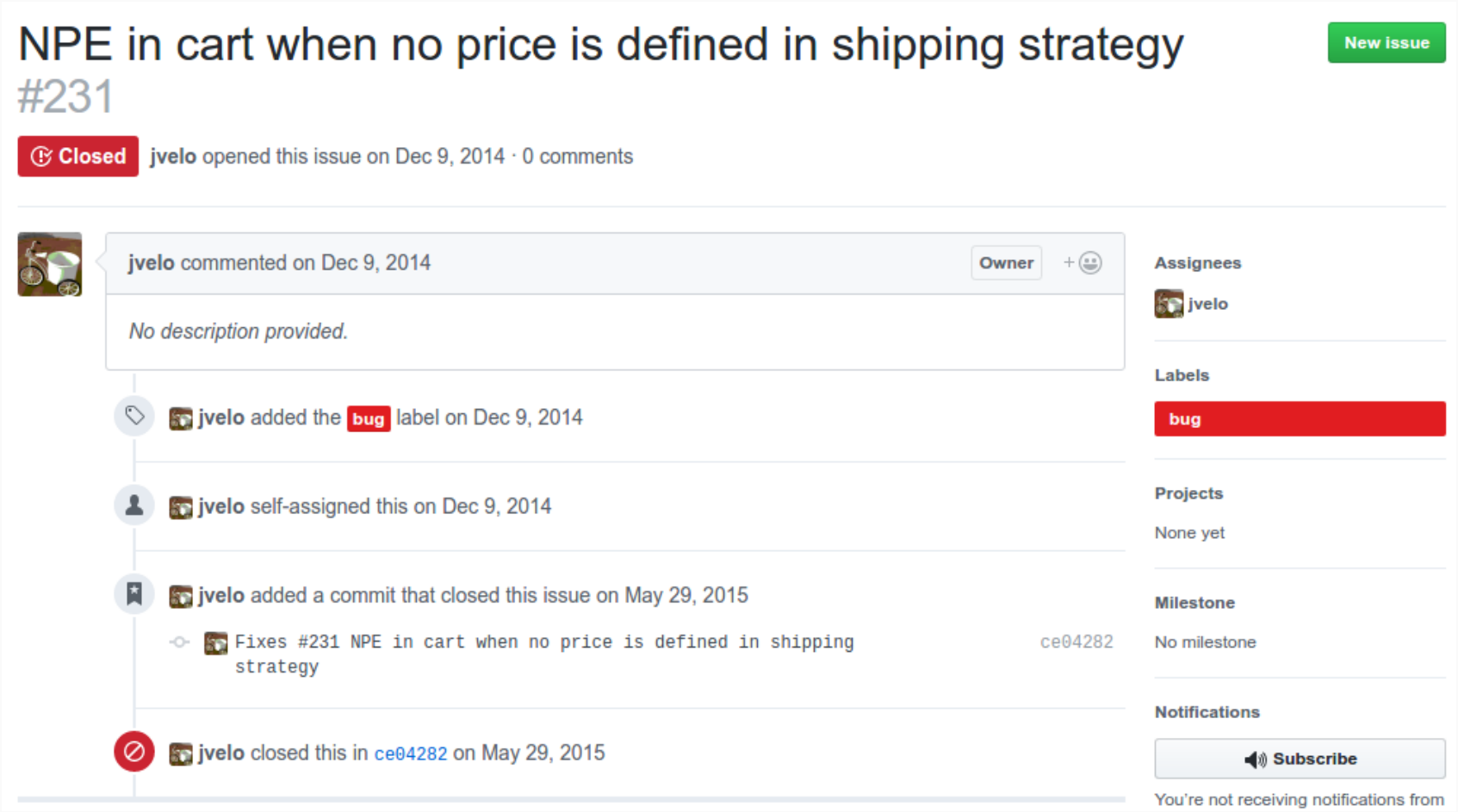
Itzal Oracle

- Failure Oracle: decides if a request is valid or not (e.g. $HTTP_{status} \neq 5xx$)
- Regression Oracle: decides if the patch does not modify the behavior for all non-failing requests.

Outline

Itzal Prototype

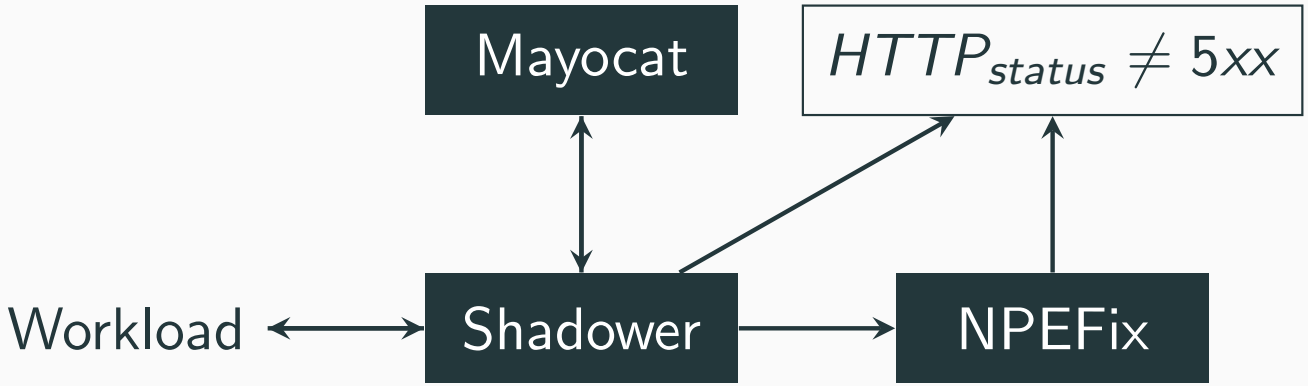
Itzal Prototype



Generated Patch for Mayocat

```
@@ FlatStrategyPriceCalculator.java
@@ -37,2 +37,5 @@
+    if (carrier.getPerItem() == null) {
+        return null;
+    }
     price = price.add(carrier.getPerItem().
         multiply(BigDecimal.valueOf(
         numberOfItems)));
```

Itzal Prototype



Outline

Conclusion

Take Away

- Patch generation in production.
- Patch regression with production inputs.
- Sandboxed repair environment repair the program without affecting the production.

Future Work

- Evaluation on other bugs (extremely difficult)
- Repair other type of bugs

Summary

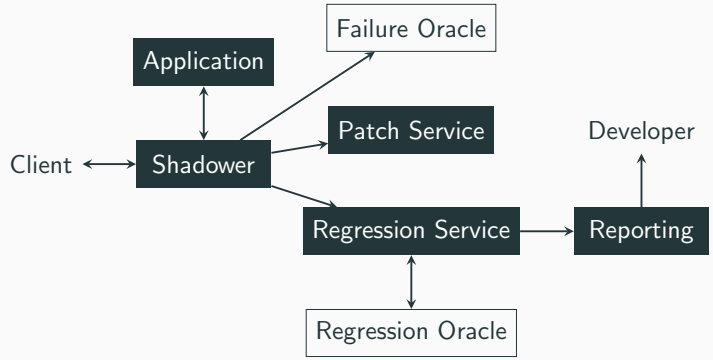
Online - Automatic Program Repair

Running Program

Repair Strategy

Failure Oracle: ?
Regression Oracle: ?

Itzal Architecture



Reporting: communicates the patches to the developers (Dashboard, Pull Request, ...)

Generated Patch for Mayocat

```
@@ FlatStrategyPriceCalculator.java
@@ -37,2 +37,5 @@
+    if (carrier.getPerItem() == null) {
+        return null;
+    }
     price = price.add(carrier.getPerItem().
         multiply(BigDecimal.valueOf(
         numberOfItems)));
```


Conclusion

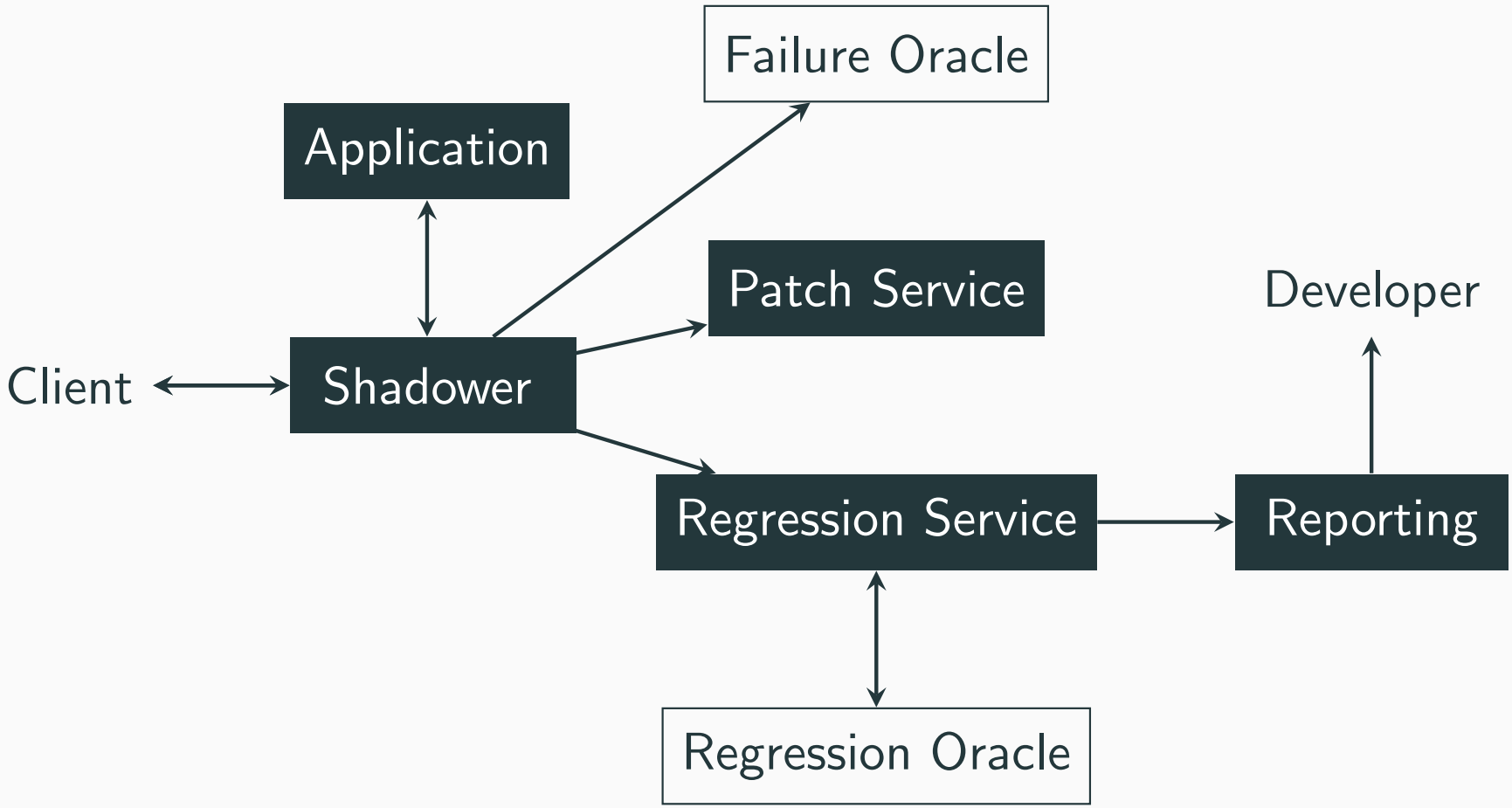
Take Away

- Patch generation in production.
- Patch regression with production inputs.
- Sandboxed repair environment repair the program without affecting the production.

Future Work

- Evaluation on other bugs (extremely difficult)
- Repair other type of bugs

Summary


Failure Oracle
Application
Patch Service
Developer
Client
Shadower
Regression Service
Reporting
Regression Oracle