VISiBLE

...

Visual Steerable Symbolic Execution

The problem?

Exhaustive Unit-Testing Is Difficult.

Structure

- How we solve it - Symbolic Execution.
- Demo
- How we built it
- Conclusion and Extensions

VISiBLE

Visual. Steerable. Symbolic Execution.

VISiBLE

Visual. Steerable. Symbolic Execution.

VISiBLE

Visual. Steerable. Symbolic Execution.

VISiBLE

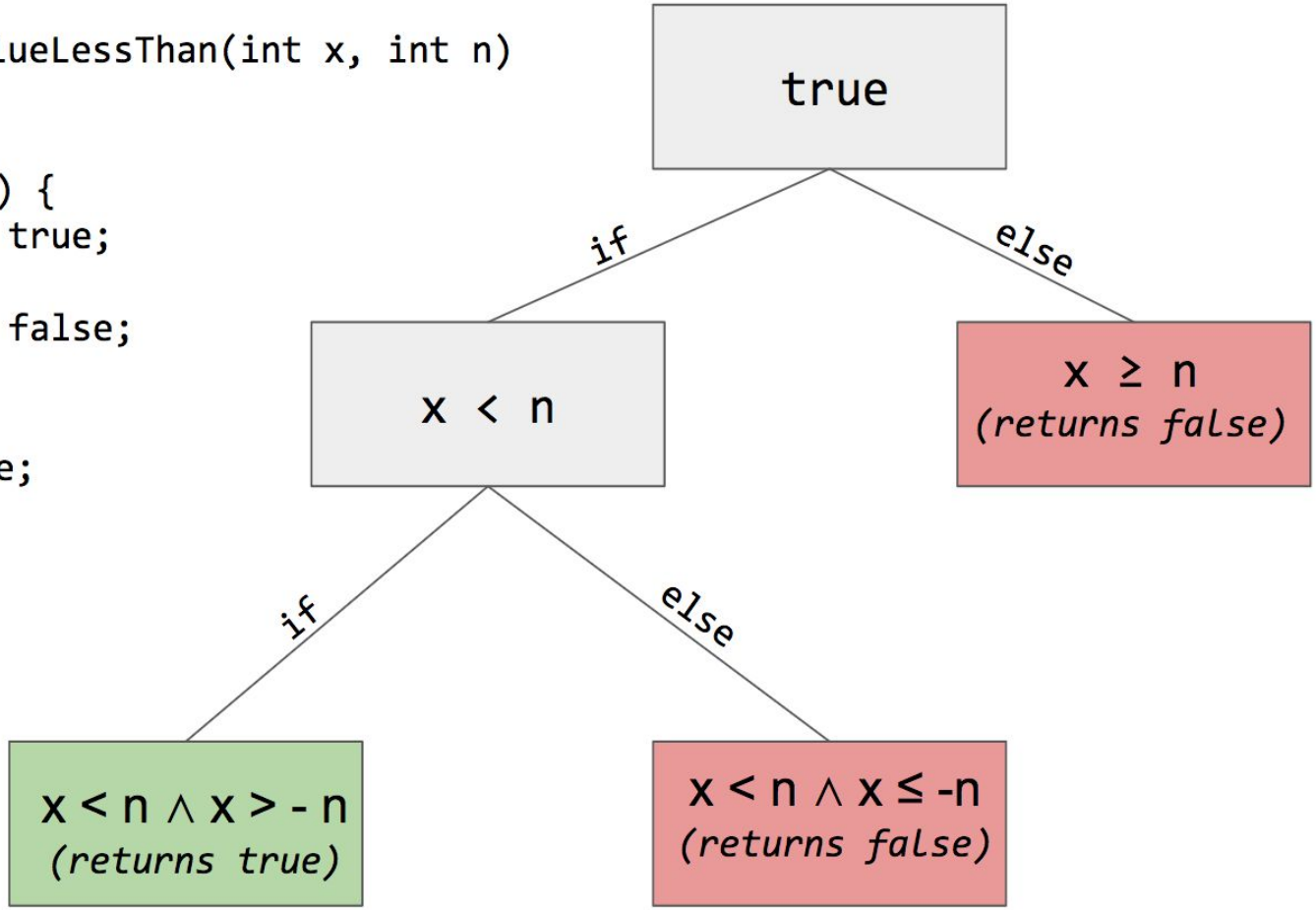
Visual. Steerable. Symbolic Execution.

What is symbolic execution?

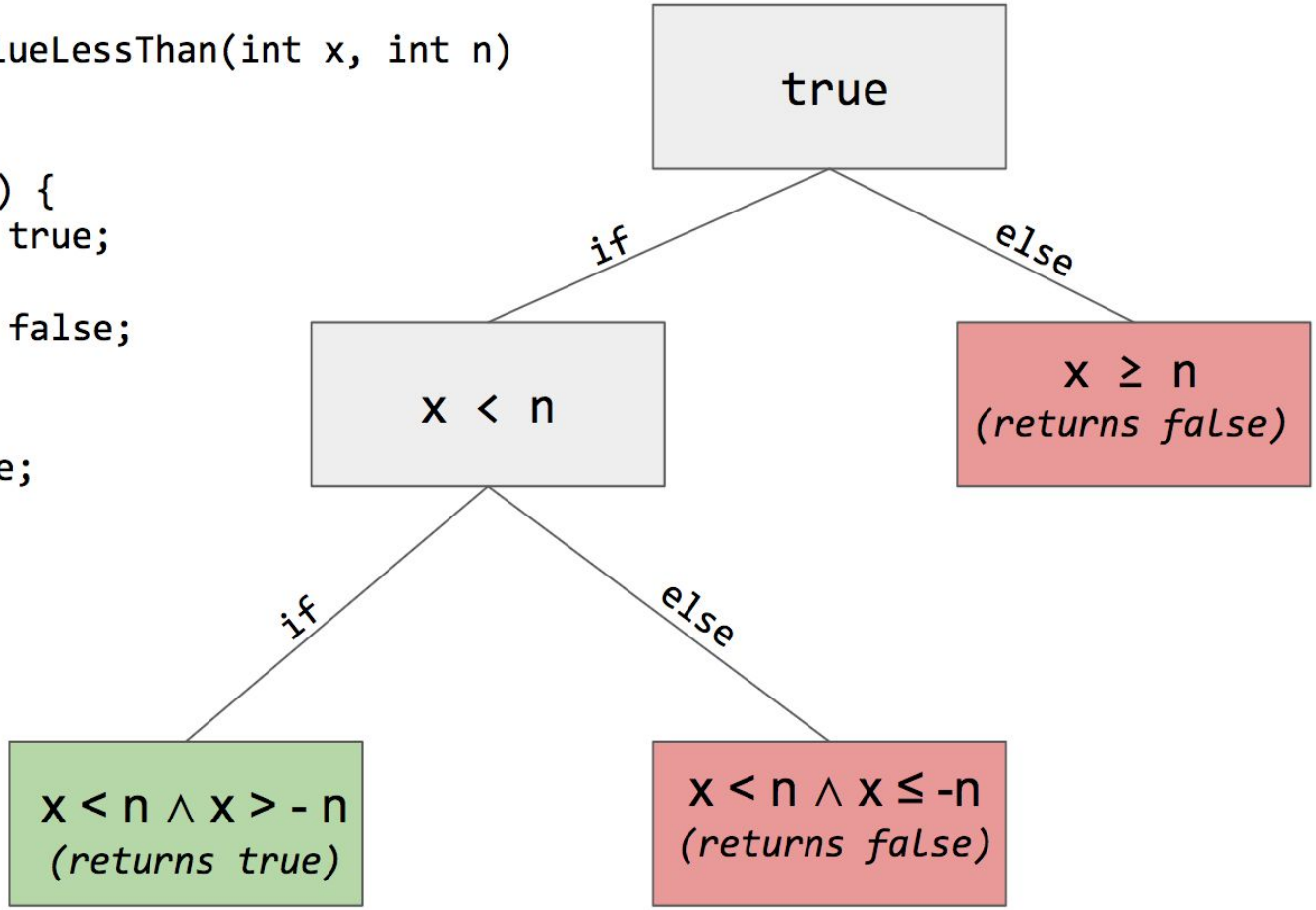
- Execute a program using arbitrary inputs instead of concrete ones.

- Explore all possible execution paths of a program.

```
public boolean absValueLessThan(int x, int n)
{
     if(x < n) {
          if (x > - n) {
               return true;
          } else {
               return false;
          }
     } else {
          return false;
     }
}
```



```
public boolean absValueLessThan(int x, int n)
{
     if(x < n) {
          if (x > - n) {
               return true;
          } else {
               return false;
          }
     } else {
          return false;
     }
}
```



Test-case generation

Demo

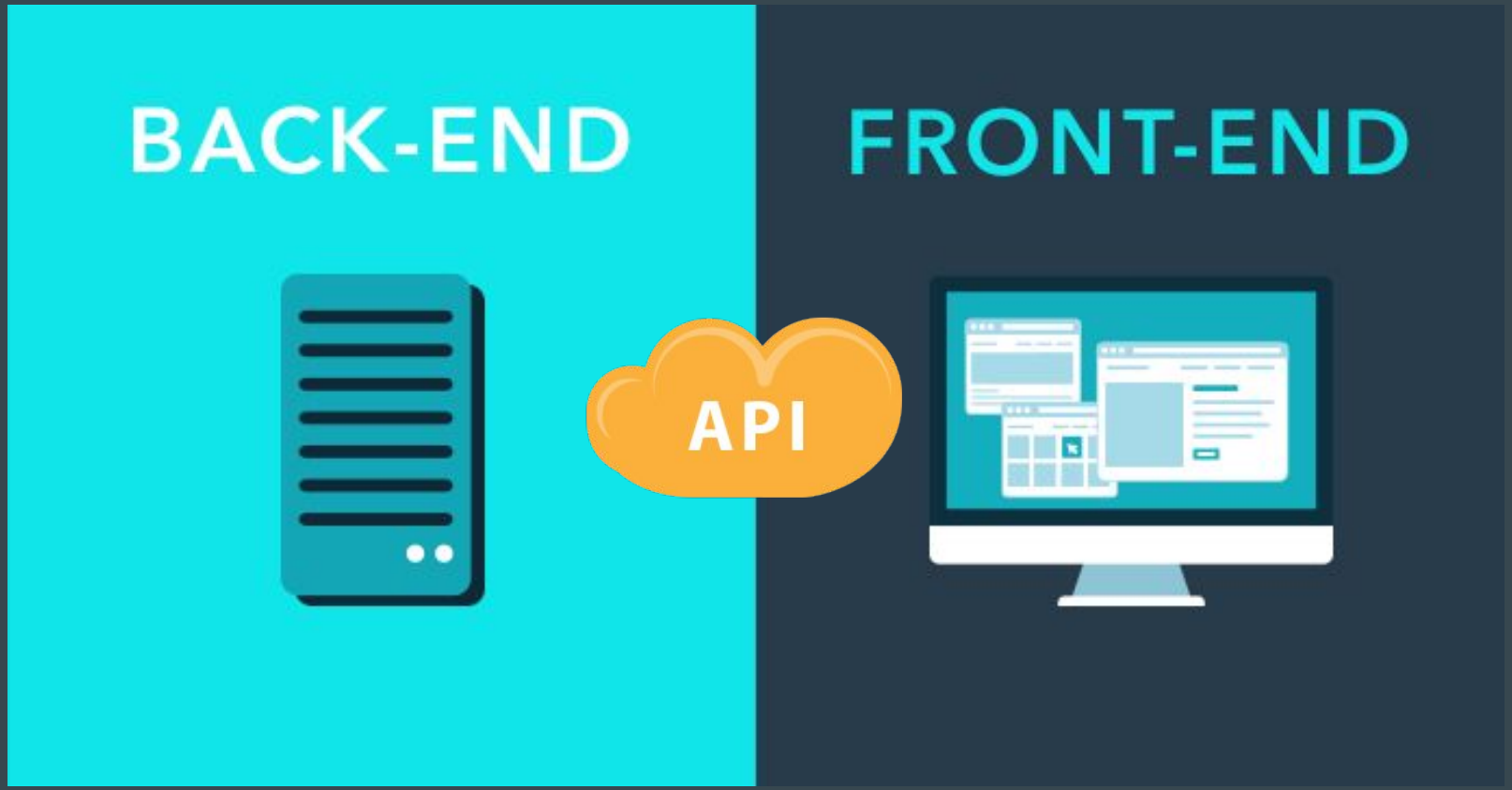
```
public static void maxOfThree(int x, int y, int z) {
    int max  = Integer.MIN_VALUE;
    if (x >= y) {
      if (x >= z) {
        max = x;
      }
      else {
        max = z;
      }
    }
    else {
      if (y >= z) {
        max = y;
      }
      else {
        max = z;
      }
    }
  }
```

```
 public static boolean absLessThan(int x, int n)
{
    if (x < n) {
      if (x > -n) {
        return true;
      } else {
        return false;
      }
    } else {
        return false;
    }
  }
```

How we built it

The Recipe


mockito
docker
NASA
node
JS


BACK-END
FRONT-END
API

The Front-end

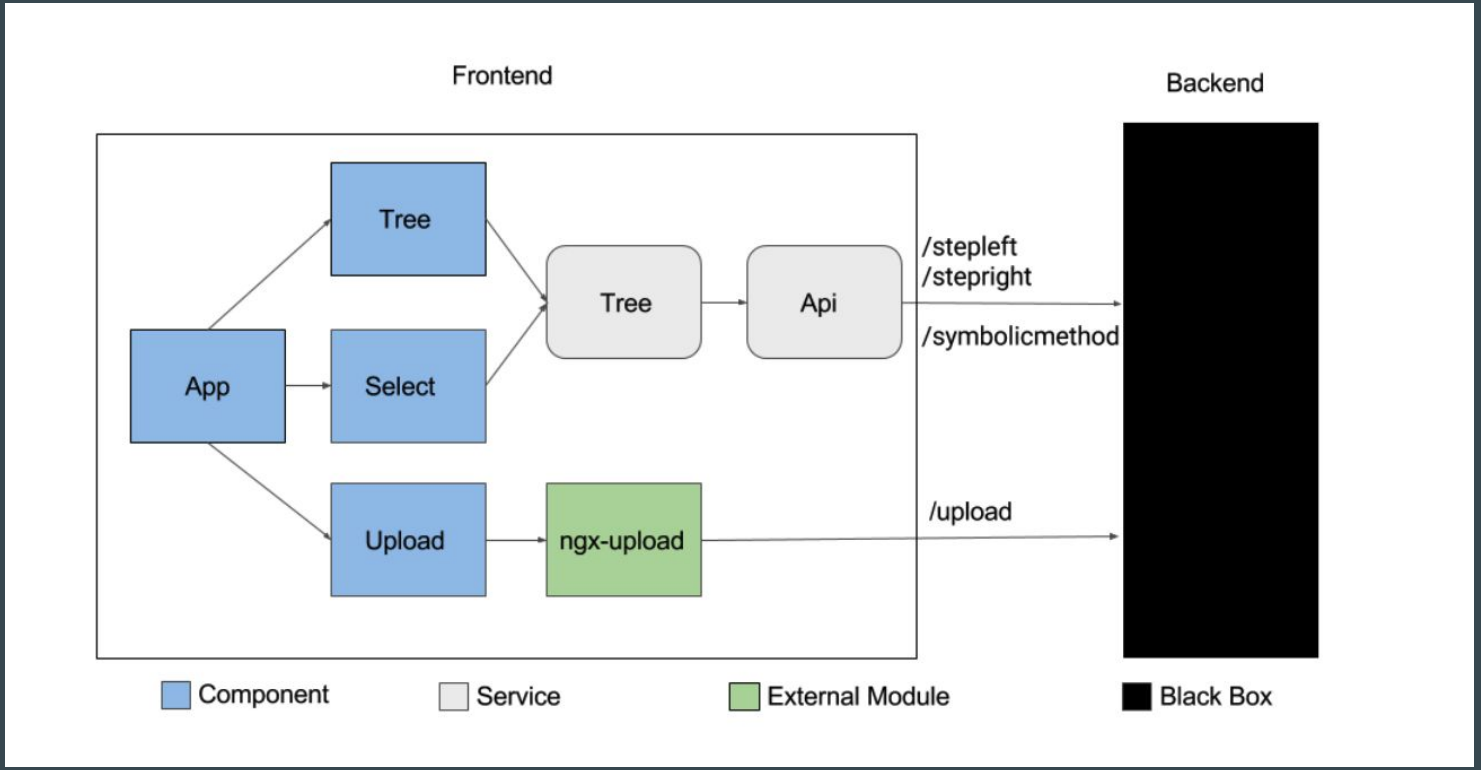
Where the visualisation happens.

Front-end

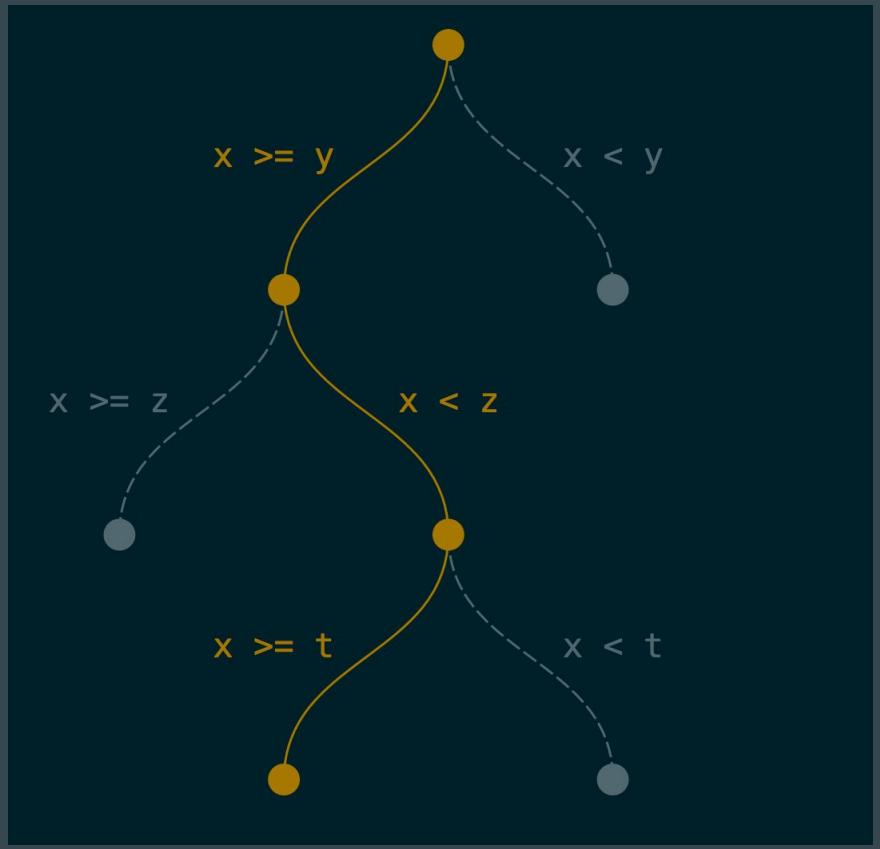
Structure

Drawing

Structure

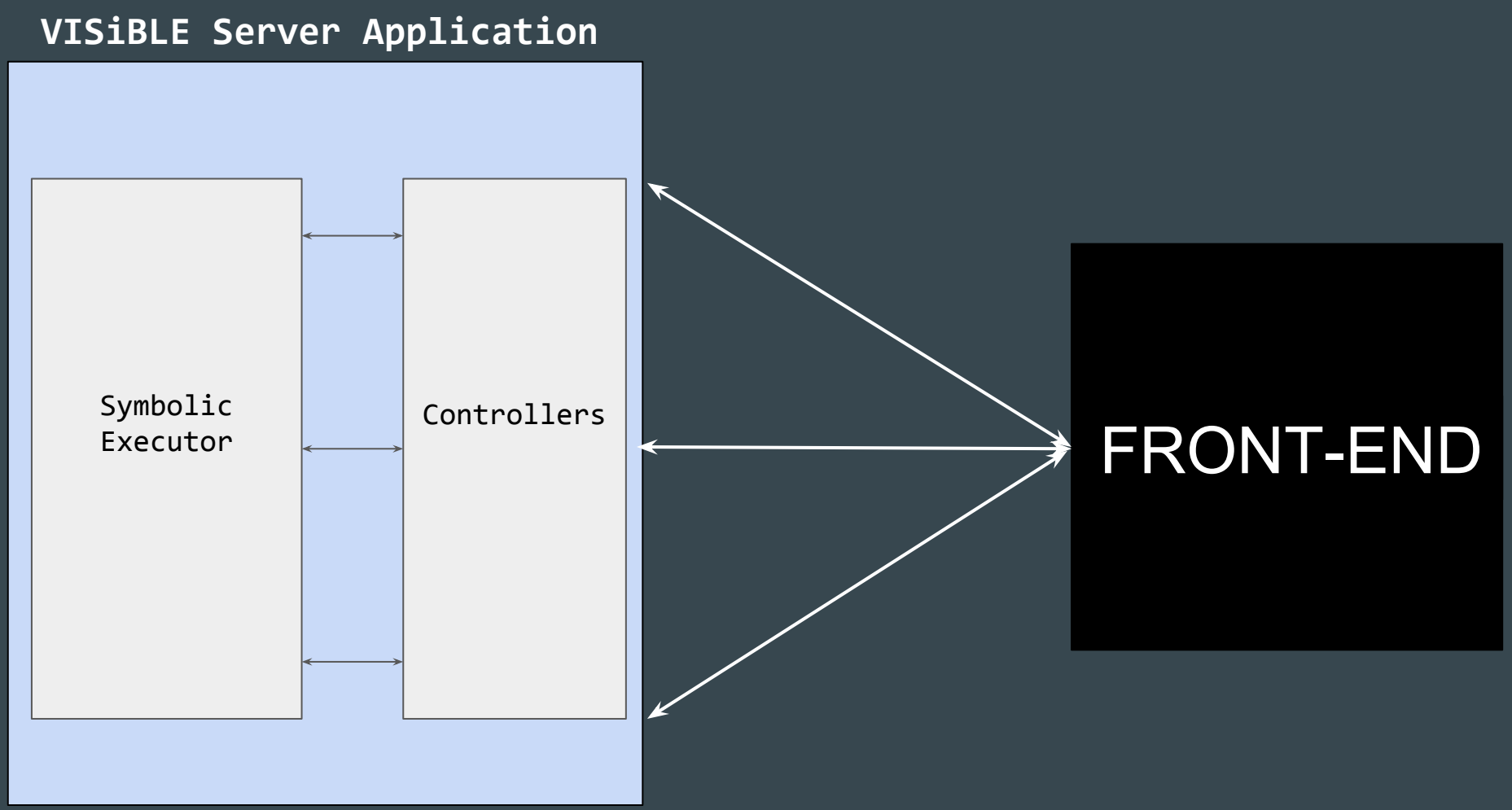

Frontend
Backend
Tree
Tree
Api
/stepleft
/stepright
/symbolicmethod
App
Select
Upload
ngx-upload
/upload
Component
Service
External Module
Black Box

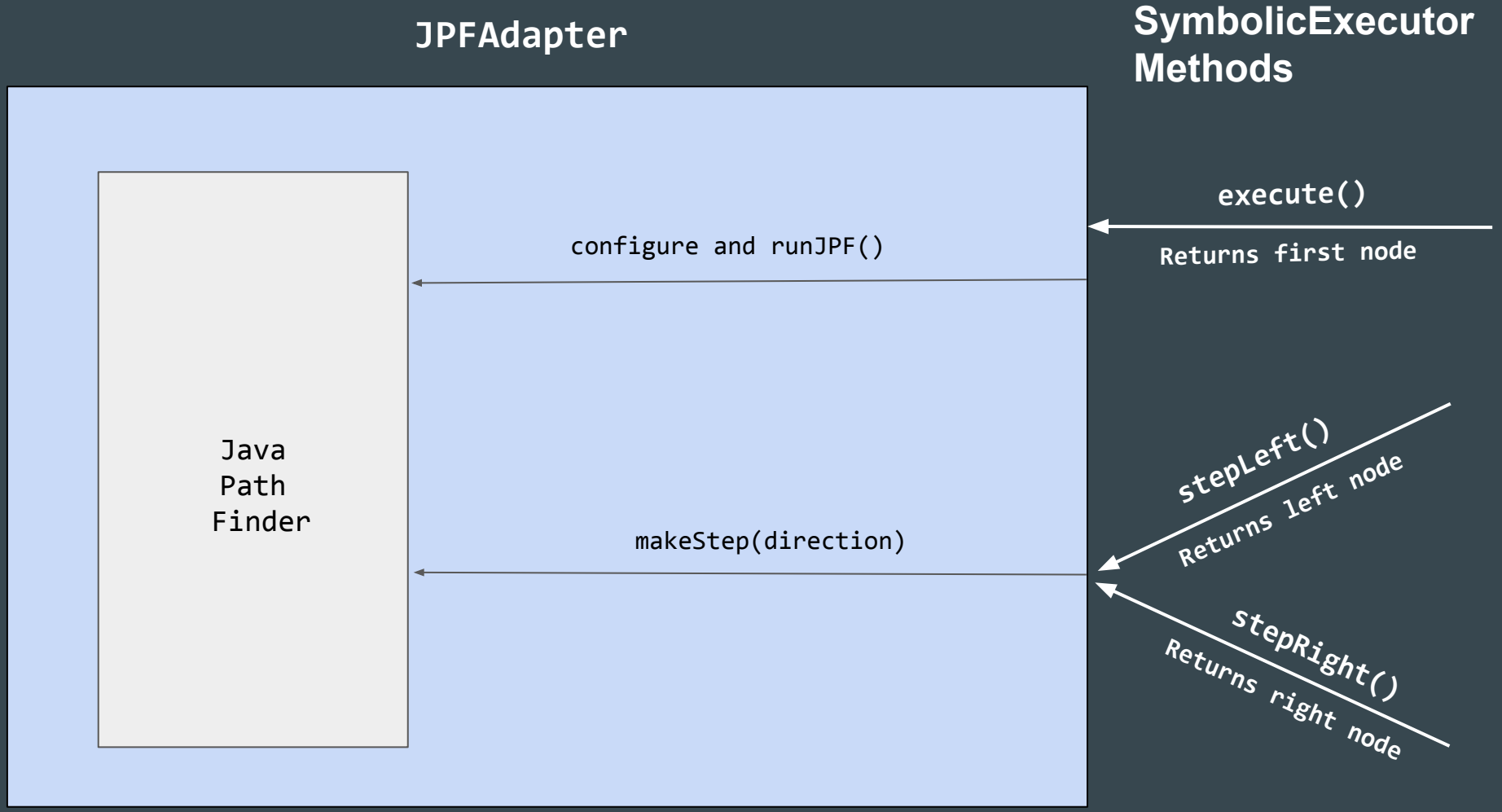
Drawing



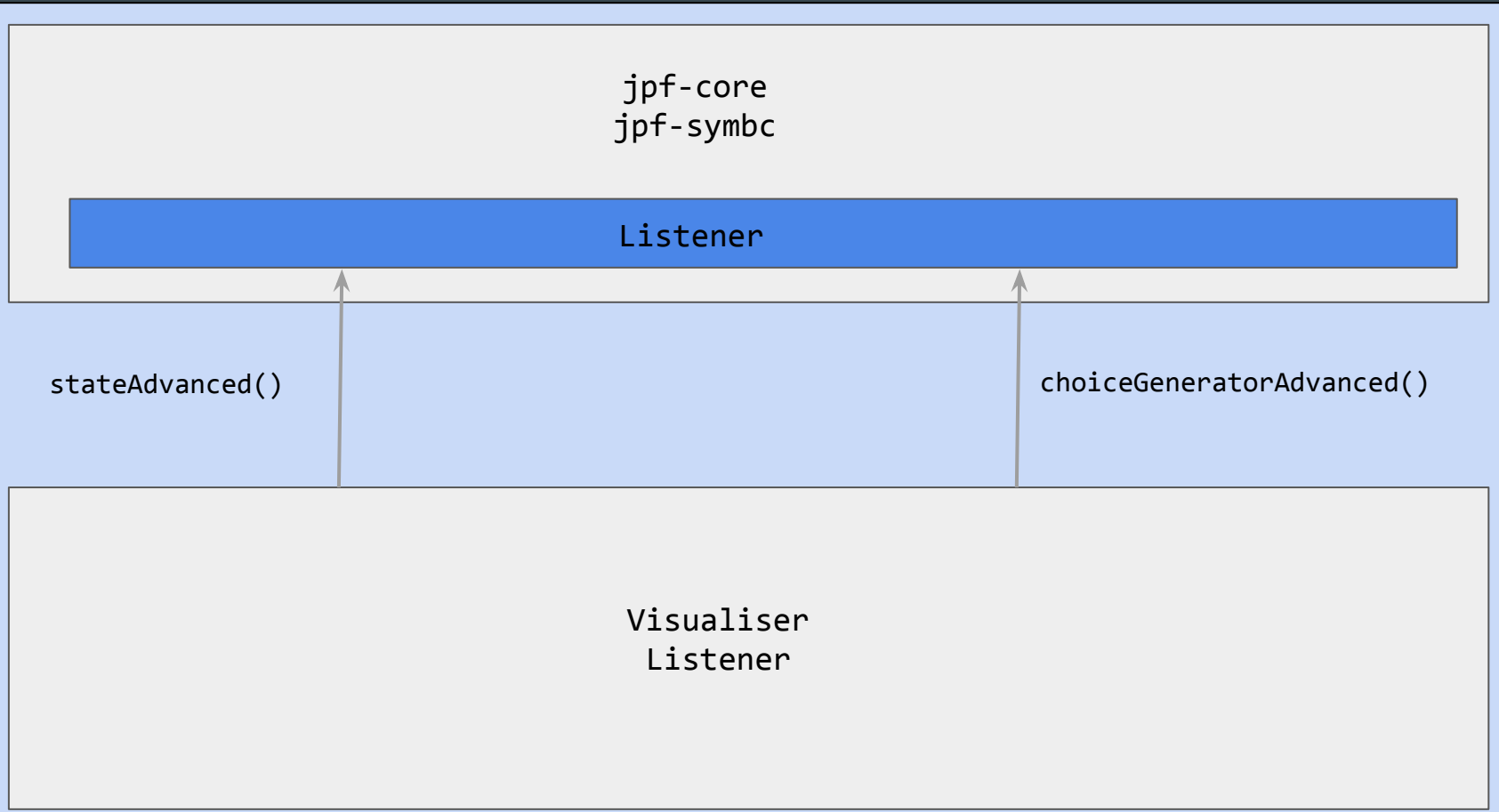
The Back-end

The magic of Symbolic Execution


VISiBLE Server Application
Symbolic Executor
Controllers
FRONT-END


JPFAdapter
SymbolicExecutor Methods
Java Path Finder
execute()
Returns first node
configure and runJPF()
stepLeft()
Returns left node
makeStep(direction)
stepRight()
Returns right node

Java Path Finder



Extensions

Extending the range of supported types

Restore previously explored states

Smart visualisation

Stack/heap visualisation

IDE integration

Non-deterministic choices

Conclusion

Thank you!

Questions?