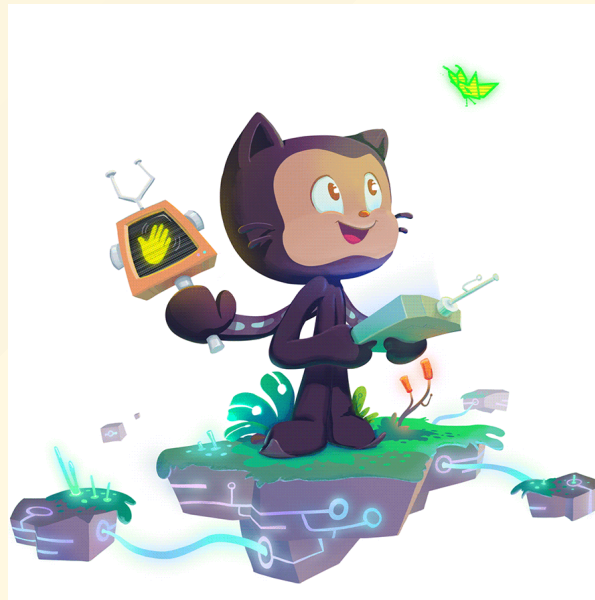
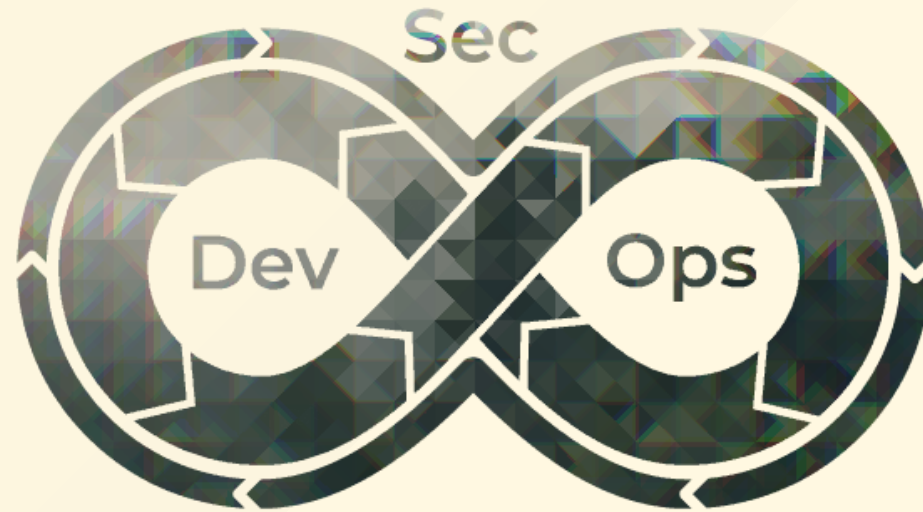


Introduction aux outils de tests de sécurité d'application statique (SAST)

by Adrien Pessu  GitHub



Introduction



Search in code

```
grep setDangerousHTML index.ts
```

Remediation

```
sed -i 's/setDangerousHTML/void/g' index.ts
```

Merci de votre attention



Introduction



Definition

OWASP

Source code analysis tools, also known as Static Application Security Testing (SAST) Tools, can help analyze source code or compiled versions of code to help find security flaws.

Workflow

 Code =>  Models =>  Patterns =>  Results!



Code

Source Code Analysis

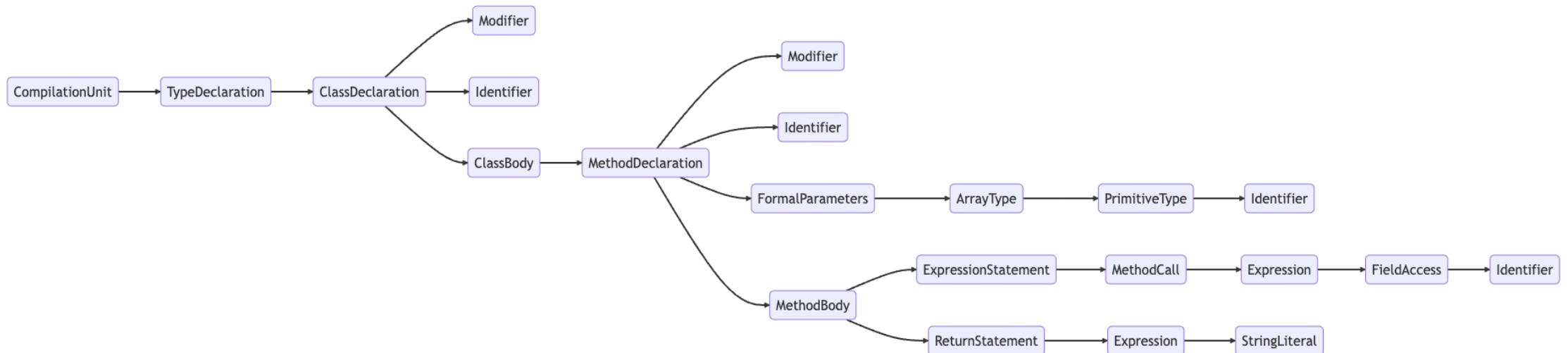
VS

Binary / Bytecode Analysis



Abstract Syntax Tree (AST) [Models]

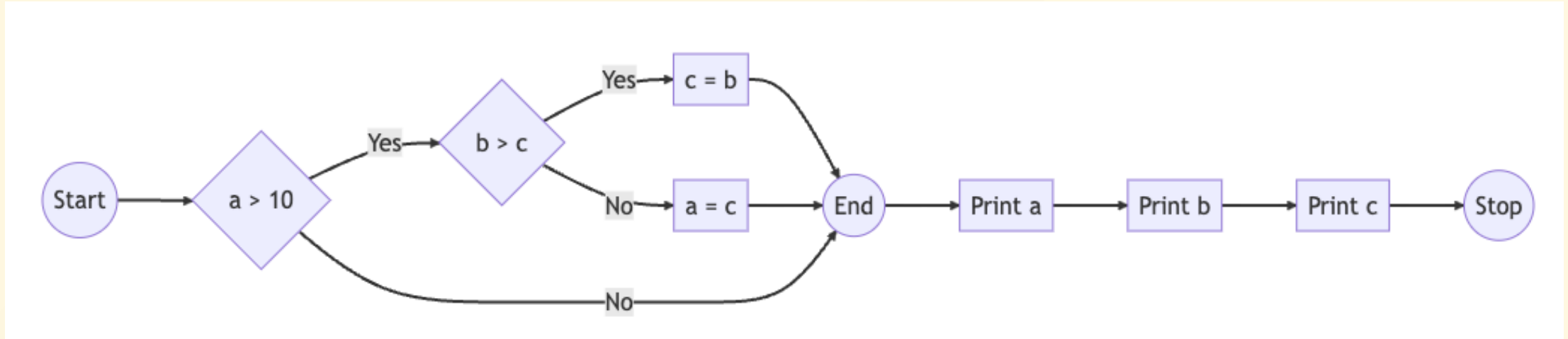
```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```



[Models] Control-flow graph

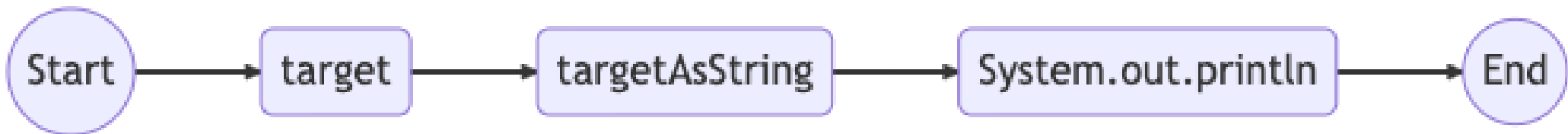
```
if (a > 10) {  
  if (b > c) {  
    c = b;  
  } else {  
    a = c;  
  }  
}  
System.out.println(a);  
System.out.println(b);  
System.out.println(c);
```

[Models] Control-flow graph



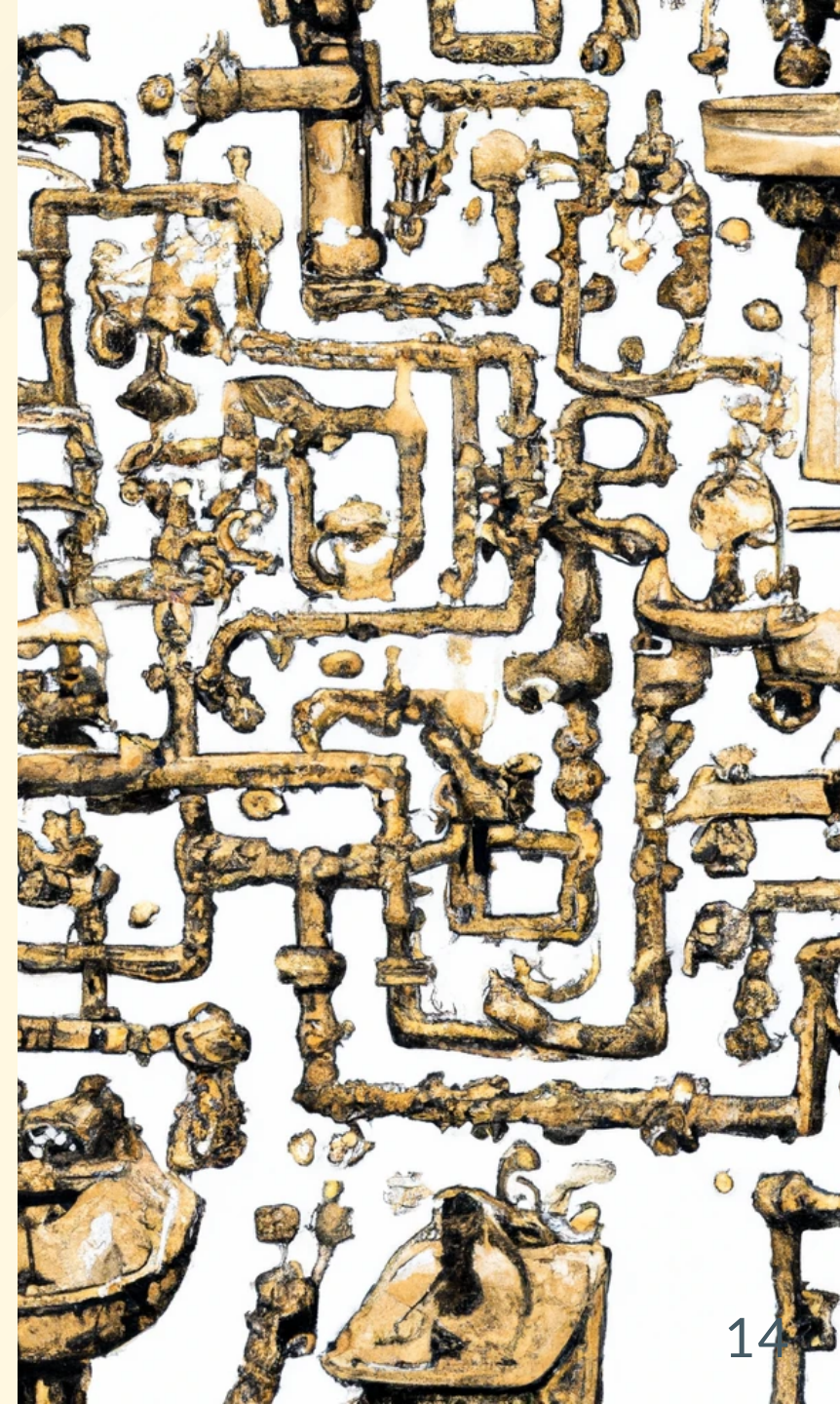
[Models] 🏗️ Data-flow graph

```
int target = input();  
String targetAsString = "Input: " + target  
System.out.println(targetAsString)
```



⚡ Taint Analysis

- Sources (user controlled inputs)
- Sinks (dangerous methods / assignments)
- Sanitizers (secures the user data)
- Validator
- Passthroughs (functions that track tainte- data)



Pattern 1/2

Using something insecure





- Configurations / Setting
- "Is debugging set to True?"

Pattern 2/2




 Data flows into somewhere insecure

User Input => [some other stuff] => `sql.execute(input)`

Results

-  Security Issues
 - SQL Injection, Cross Site Scripting, ...
-  Best Practices
 - Using Key Vaults, ...
-  Code Quality and Code Smells
 - Long Functions, Duplicated code, ...
-  Positive Results
 - Using appropriate hashing algorithm
 - automatic encoding, ...




Configuration

-  Configuration Rules (yaml, json, data structure...)
 - Simpler to write
 - Complex flows can be very hard to declare
-  Dynamic Queries ( programming like language)
 - Harder to learn and write
 - Complex flows are easier

Demo?



Conclusion

- Easy to configure
- False positive (Context)
-  An automated tool to analyse source code
 - Automate Code Review
-  Discover known security issues
-  Discover repetitive security issues
- Remediation
- Security analysis for Security Engineers / Researchers

Slides

[https://adrienpessu.github.io/slides/introduction to SAST/](https://adrienpessu.github.io/slides/introduction%20to%20SAST/)



<https://openfeedback.io/VWEMZHoBj0mPrdZ9Ippo>

Thanks to *@geekmasher*