CISPA
Center for IT-Security, Privacy and Accountability

EC SPRIDE
EUROPEAN CENTER FOR SECURITY AND PRIVACY BY DESIGN

jÄk: Using Dynamic Analysis to Crawl and Test Modern Web Applications

Giancarlo Pellegrino[1], Constantin Tschürtz[2], Eric Bodden[2], and Christian Rossow[1]

18th International Symposium on Research in Attacks, Intrusions and Defenses

November 3rd, Kyoto, Japan

[1] CISPA, Saarland University, Germany
[2] Fraunhofer SIT / TU Darmstadt, Germany

Web Application Scanners

- (Semi-)automated security testing tools
- Follow a dynamic and black-box testing approach

Web Application Scanners

- (Semi-)automated security testing tools
- Follow a dynamic and black-box testing approach

Architecture

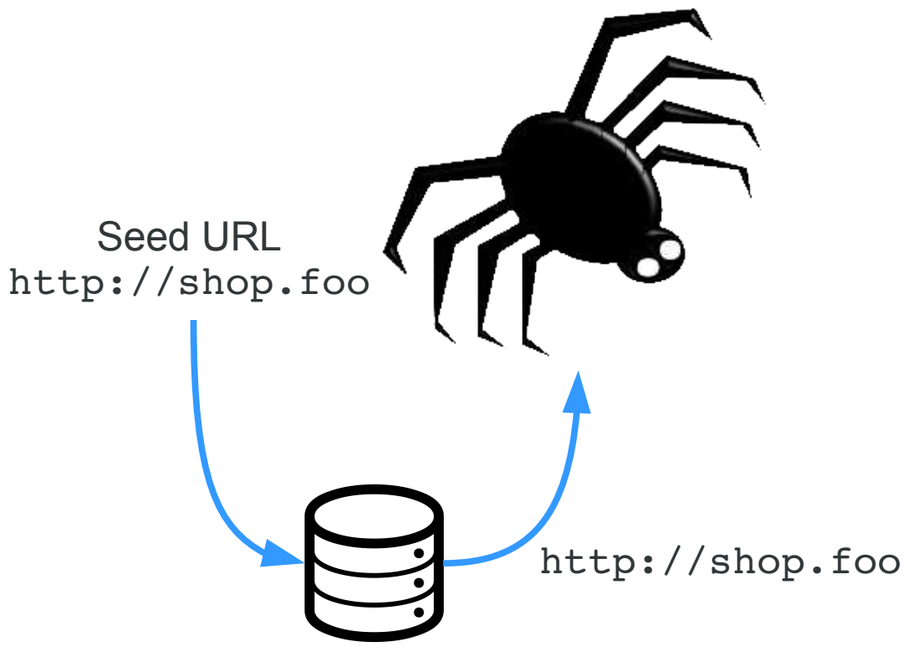
CISPA Center for IT-Security, Privacy and Accountability

Crawler Module

Attacker Module

Analysis Module

Crawler



Crawler

CISPA Center for IT-Security, Privacy and Accountability

http://shop.foo


```
<html>
    <head>
        <title>Online shopping</title>
    </head>
    <body>
        <a href="/contacts">Contacts</a>
        <form action="/search">
            <input type="text" name="q"/>
            <input type="submit"/>
        </form>
    </body>
</html>
```

Crawler

CISPA Center for IT-Security, Privacy and Accountability

http://shop.foo


```
<html>
    <head>
        <title>Online shopping</title>
    </head>
    <body>
        <a href="/contacts">Contacts</a>
        <form action="/search">
            <input type="text" name="q"/>
            <input type="submit"/>
        </form>
    </body>
</html>
```

New URL

Crawler

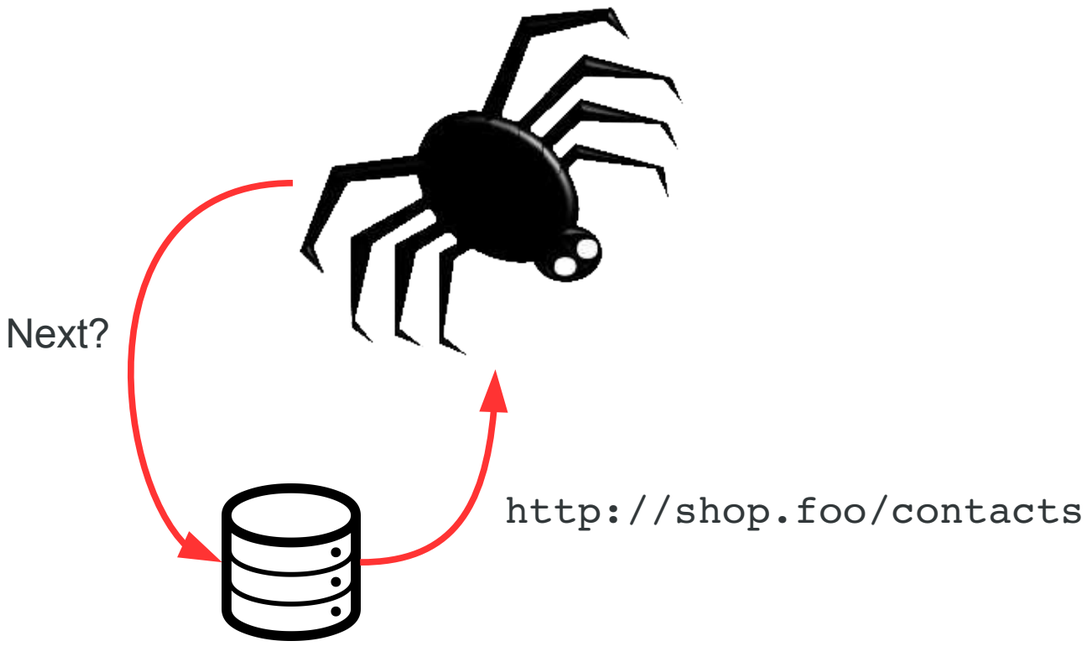
CISPA Center for IT-Security, Privacy and Accountability

http://shop.foo


```
<html>
    <head>
        <title>Online shopping</title>
    </head>
    <body>
        <a href="/contacts">Contacts</a>
        <form action="/search">
            <input type="text" name="q"/>
            <input type="submit"/>
        </form>
    </body>
</html>
```

New search HTML form

Crawler



Crawler

CISPA Center for IT-Security, Privacy and Accountability

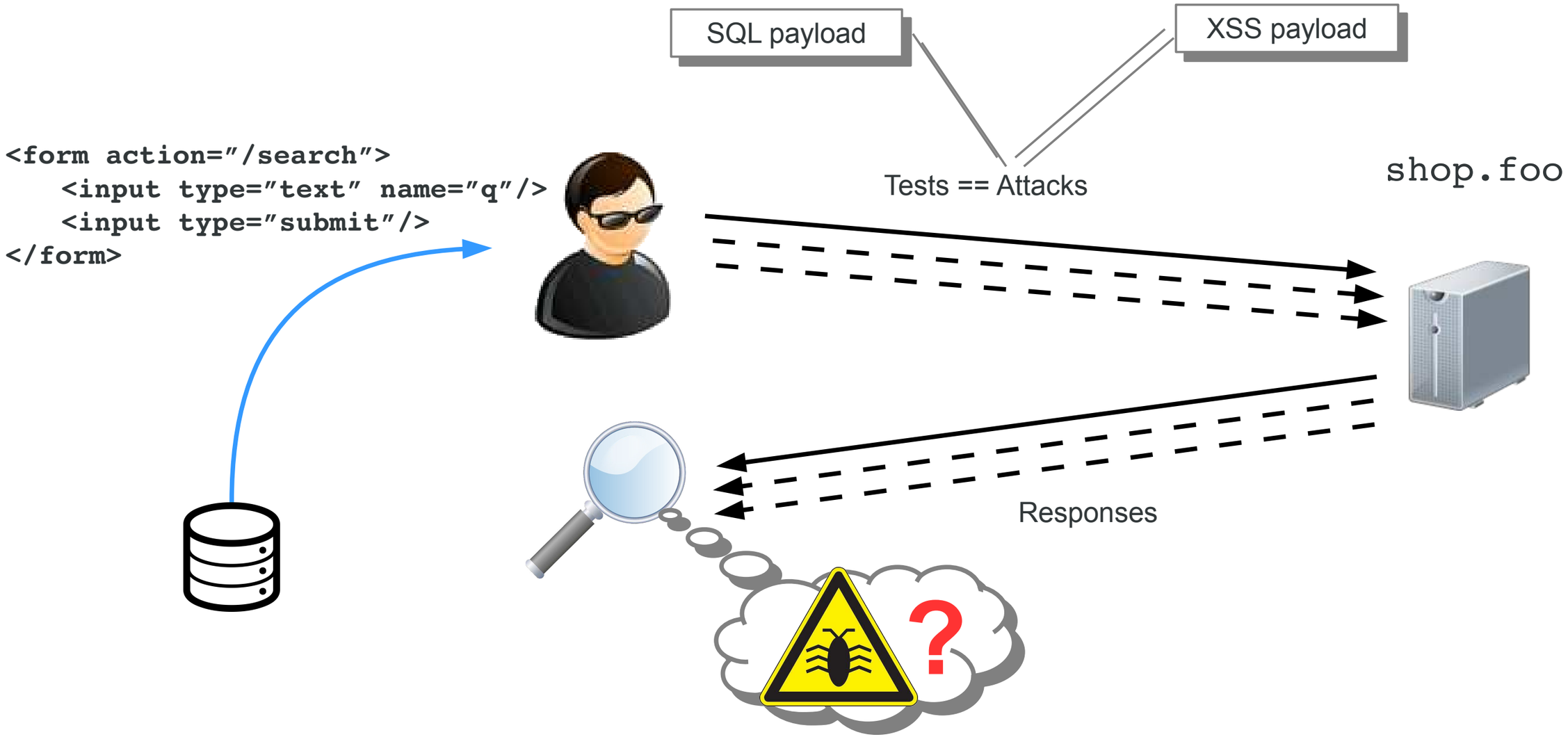
http://shop.foo/contacts


```
<html>
   <head>
      <title>Contact Page</title>
   </head>
   <body>
      <form action="/comments">
        <input type="text" name="msg"/>
        <input type="submit"/>
      </form>
   </body>
</html>
```

New HTML form

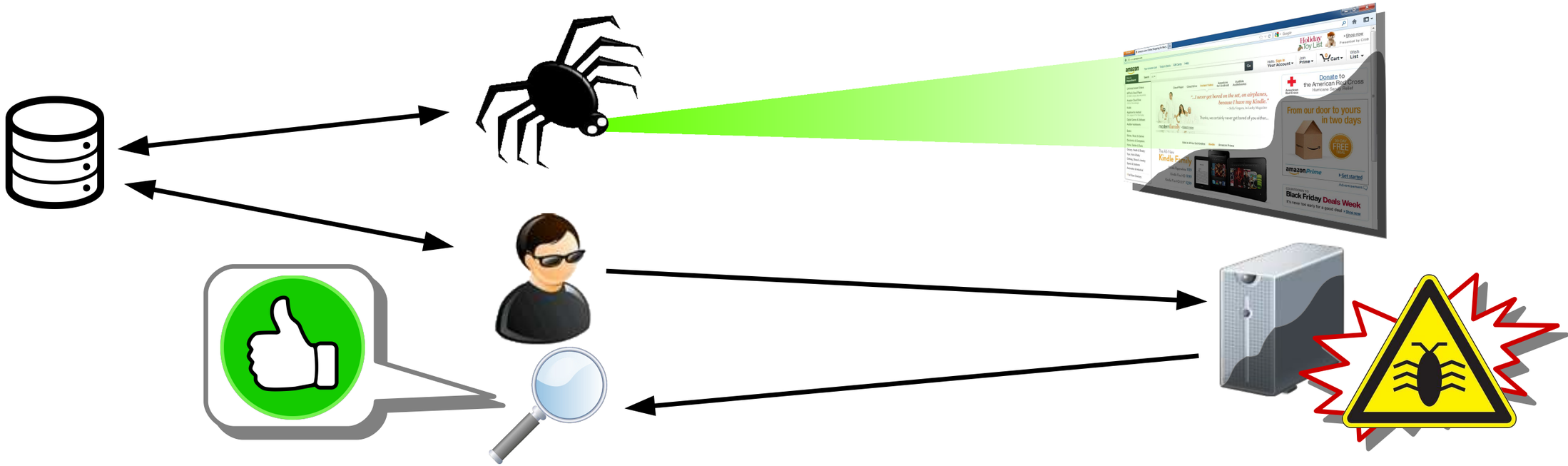
Security Testing

CISPA Center for IT-Security, Privacy and Accountability



Crawler Critical for Coverage

CISPA Center for IT-Security, Privacy and Accountability



- Crawler explores the Web application attack surface
  - Missing parts → missing possible vulnerabilities
- Existing crawlers based on:
  - HTML parsing and pattern matching to extract URLs
  - “clickable” areas to further explore the surface

Crawler and Modern Web Applications

CISPA Center for IT-Security, Privacy and Accountability

- Complexity of client side has dramatically increased (i.e., stateful JS programs)

Crawler and Modern Web Applications

CISPA Center for IT-Security, Privacy and Accountability

- Complexity of client side has dramatically increased (i.e., stateful JS programs)
- Links and forms can be built and inserted in the webpage at run-time

```
var url = scheme() + '://' + domain() + '/' + endpoint();
document.getElementByID('myLink').href = url;
```

➔ **HTML parsing and pattern matching no longer sufficient**

Crawler and Modern Web Applications

CISPA Center for IT-Security, Privacy and Accountability

- Complexity of client side has dramatically increased (i.e., stateful JS programs)
- Links and forms can be built and inserted in the webpage at run-time

```
var url = scheme() + '://' + domain() + '/' + endpoint();
document.getElementById('myLink').href = url;
```

➔ **HTML parsing and pattern matching no longer sufficient**

- JS is an event-driven language

  - Functions executed upon events

➔ **Lack of support of event-based execution model**

Crawler and Modern Web Applications

CISPA Center for IT-Security, Privacy and Accountability

- Complexity of client side has dramatically increased (i.e., stateful JS programs)
- Links and forms can be built and inserted in the webpage at run-time

```
var url = scheme() + '://' + domain() + '/' + endpoint();
document.getElementById('myLink').href = url;
```

➔ **HTML parsing and pattern matching no longer sufficient**

- JS is an event-driven language

  - Functions executed upon events

➔ **Lack of support of event-based execution model**

Large part of web applications remain unexplored!

Crawler and Modern Web Applications

CISPA Center for IT-Security, Privacy and Accountability

- Complexity of client side has dramatically increased (i.e., stateful JS programs)
- Links and forms can be built and inserted in the webpage at run-time

```
var url = scheme() + '://' + domain() + '/' + endpoint();
document.getElementById('myLink').href = url;
```

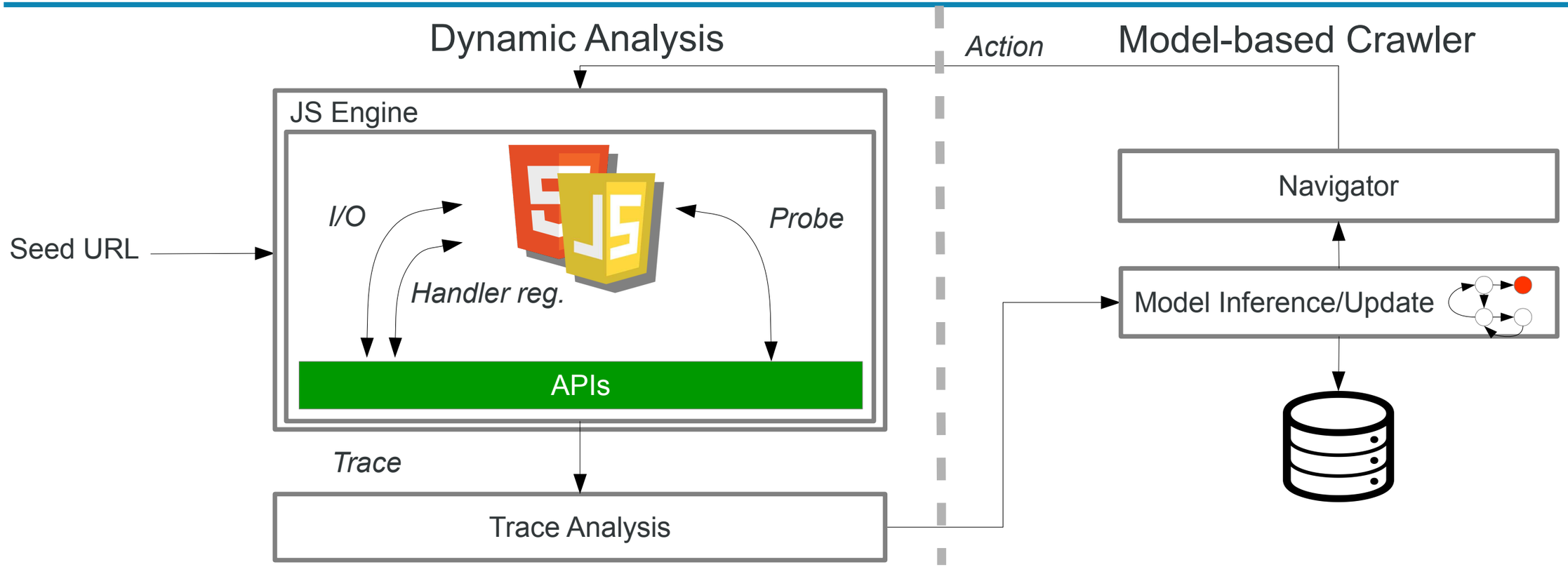
- We addressed the coverage problem with
  - JavaScript client side dynamic analysis
  - Model-based Crawler
- Build a tool: **jÄk**

- Functions executed upon events

➔ **Lack of support of event-based execution model**

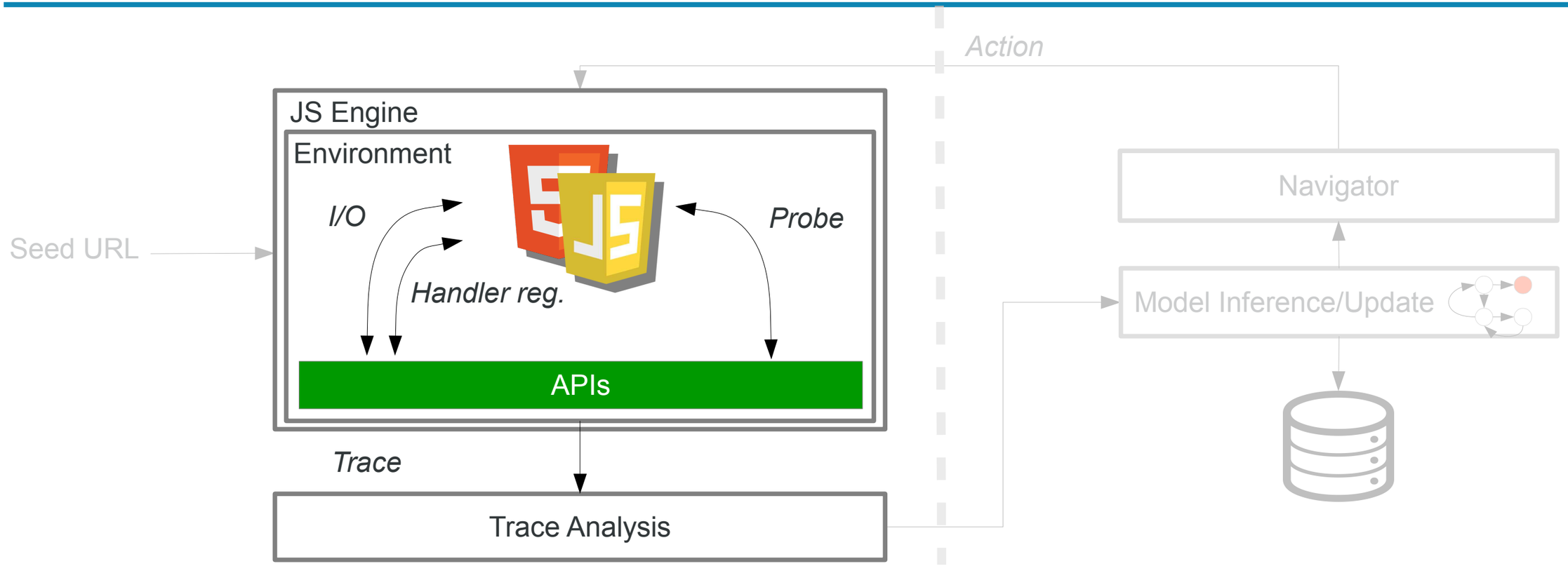
Large part of web applications remain unexplored!

Our Approach



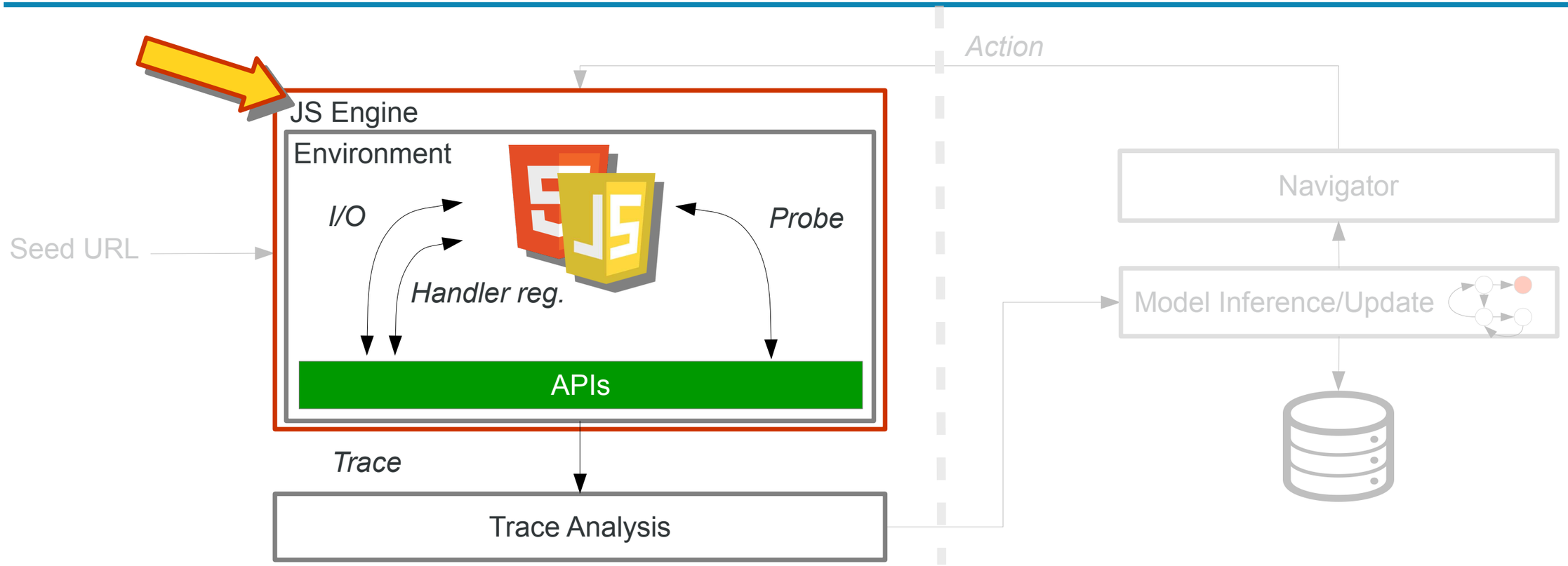
- Combine dynamic analysis with model-based crawler
  - Dynamic analysis monitors client side program execution
  - Crawler builds, maintains, uses a model of the visited attack surface

Dynamic Analysis



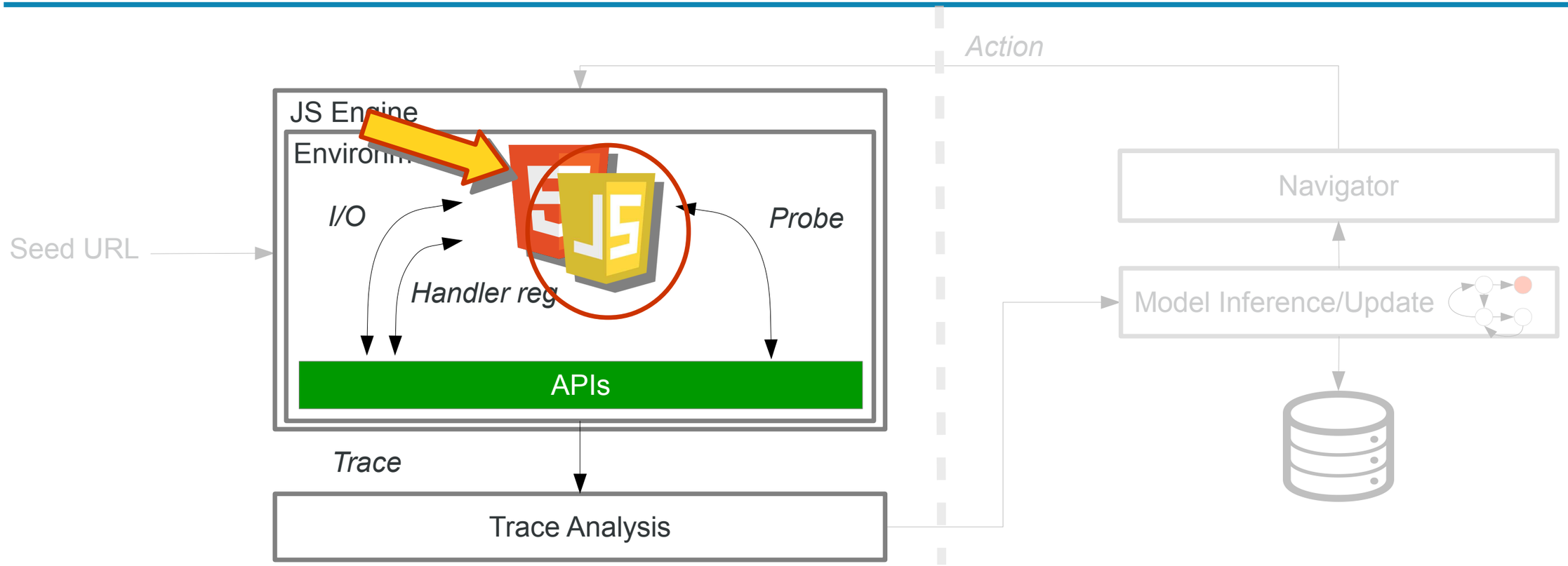
- Different approaches:

Dynamic Analysis



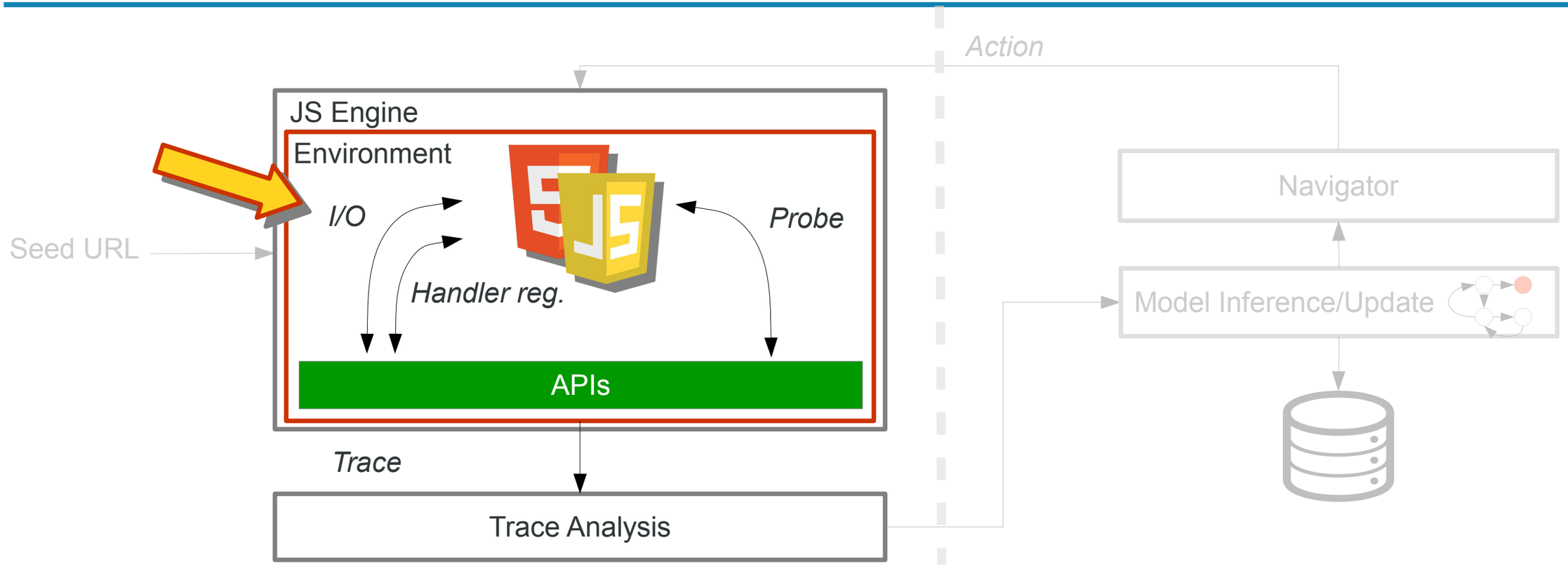
- Different approaches:
  1) JS engine instrumentation → laborious task, engine-dependent

Dynamic Analysis



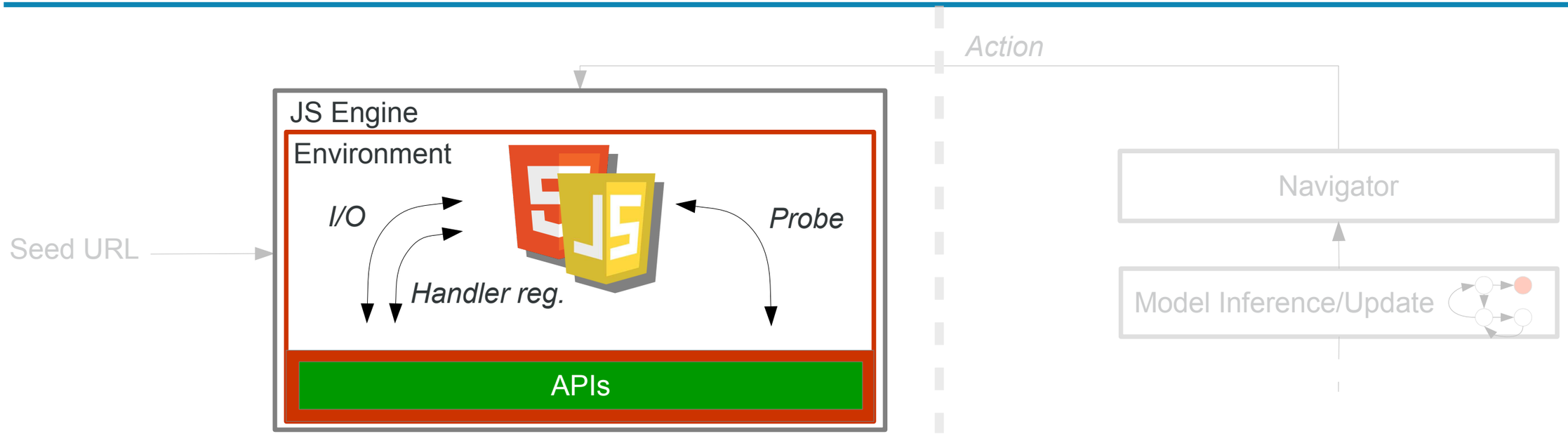
- Different approaches:
  1) JS engine instrumentation → laborious task, engine-dependent
  2) JS program instrumentation → JS code is not entirely available

Dynamic Analysis



- Different approaches:
  1) JS engine instrumentation → laborious task, engine-dependent
  2) JS program instrumentation → JS code is not entirely available
  3) Modification of execution environment

Dynamic Analysis



- Modify execution environment via **function hooking**:
  - Intercept API calls (e.g., network I/O and event handler registration)
  - Object manipulations (i.e., object properties)
  - Schedule DOM inspections
- Hooks installed by injecting own JS code:
  - Function redefinition
  - Set functions

Function Redefinition

CISPA Center for IT-Security, Privacy and Accountability

Application JS code

```
function handler() {
   alert("hello world");
}

el = document.getElementByID('img')
el.addEventListener("click", handler);
```

Function Redefinition

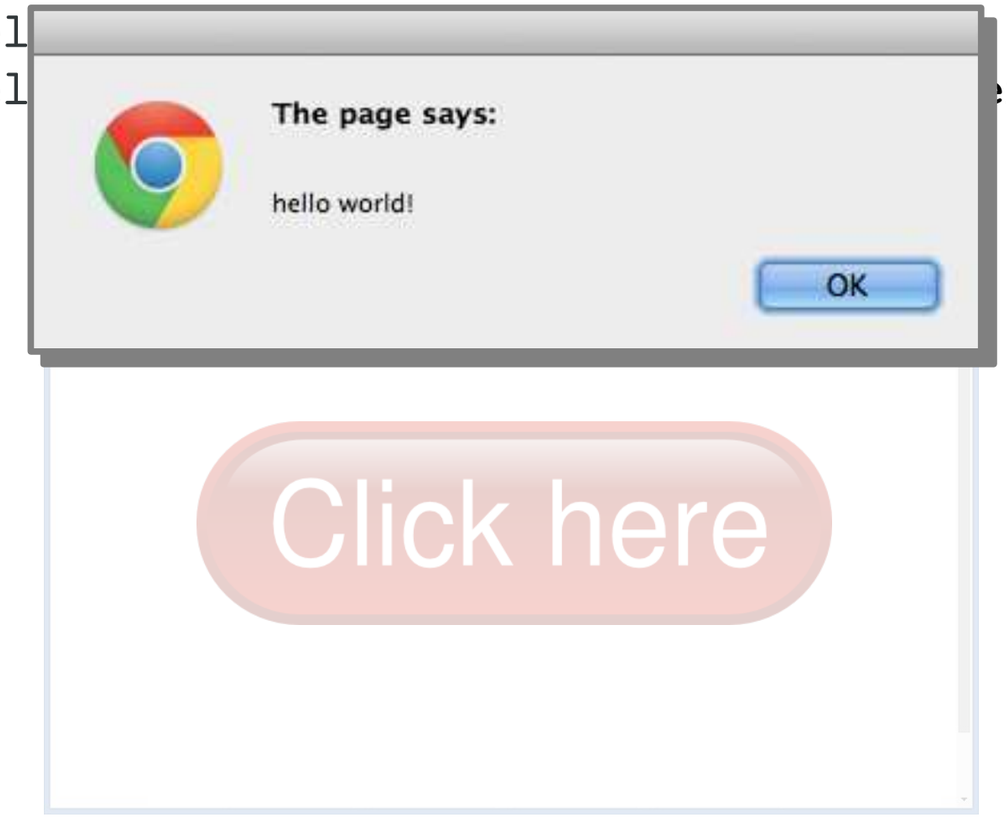
CISPA Center for IT-Security, Privacy and Accountability

```
function handler() {
   alert("hello world");
}

el = document.getElementByID('img')
el.addEventListener("click", handler);
```

Function Redefinition

CISPA Center for IT-Security, Privacy and Accountability

```
function handler() {
   alert("hello world");
}

el
el                                          er);
```



Function Redefinition

CISPA Center for IT-Security, Privacy and Accountability

```
function handler() {
   alert("hello world");
}

el = document.getElementByID('img')
el.addEventListener("click", handler);
```

API

```
Element.prototype.addEventListener = function(e, h) {
    […]
    listeners[e].append(h);
}
```

Function Redefinition

CISPA Center for IT-Security, Privacy and Accountability

```
function handler() {
   alert("hello world");
}

el = document.getElementByID('img')
el.addEventListener("click", handler);
```

API

```
Element.prototype.addEventListener = function(e, h) {
    […]
    listeners[e].append(h);
}
```

Intercept!

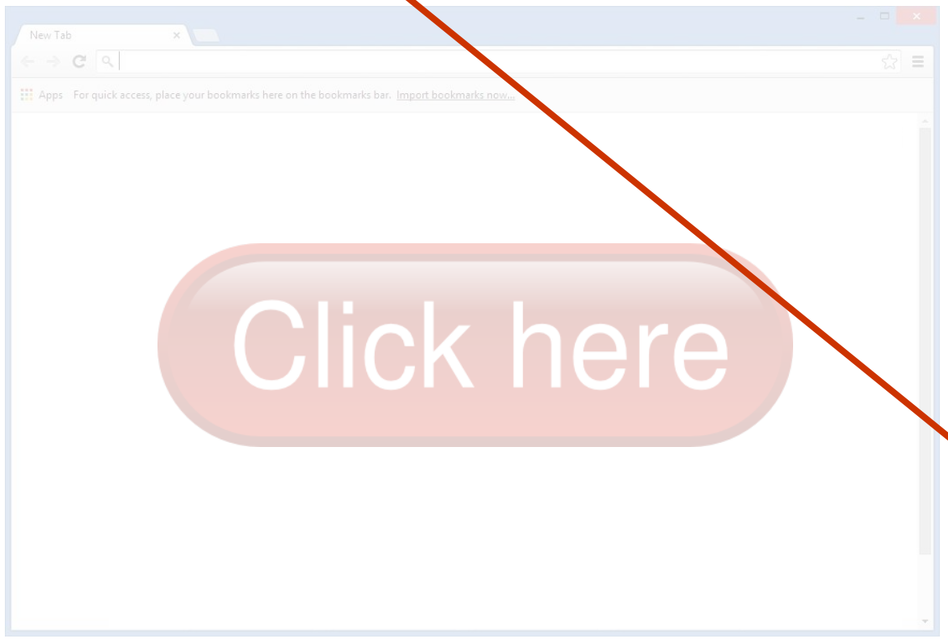
Function Redefinition

CISPA Center for IT-Security, Privacy and Accountability

preamble

Application JS code

```
function handler() {
   alert("hello world");
}

el = document.getElementByID('img')
el.addEventListener("click", handler);
```



PREAMBLE

```
var orig_f = Element.prototype.addEventListener;

Element.prototype.addEventListener = function(){
          console.log("new handler registration");
          return orig_f.apply(this, argument);
      };
```

Function Redefinition

CISPA Center for IT-Security, Privacy and Accountability

preamble

Application JS code

```
function handler() {
    alert("hello world");
}

el = document.getElementByID('img')
el.addEventListener("click", handler);
```

API

```
Element.prototype.addEventListener = function(e, h) {
    […]
    listeners[e].append(h);
}
```


PREAMBLE

```
var orig_f = Element.prototype.addEventListener;

Element.prototype.addEventListener = function(){
          console.log("new handler registration");
          return orig_f.apply(this, argument);
      };
```

Function Redefinition

CISPA Center for IT-Security, Privacy and Accountability

preamble

Application JS code

```
function handler() {
    alert("hello world");
}

el = document.getElementByID('img')
el.addEventListener("click", handler);
```

API

```
Element.prototype.addEventListener = function(e, h) {
    […]
    listeners[e].append(h);
}
```


PREAMBLE

```
var orig_f = Element.prototype.addEventListener;

Element.prototype.addEventListener = function(){
        console.log("new handler registration");
        return orig_f.apply(this, argument);
    };
```

Function Redefinition

CISPA Center for IT-Security, Privacy and Accountability

preamble

Application JS code

```
function handler() {
   alert("hello world");
}

el = document.getElementByID('img')
el.addEventListener("click", handler);
```

API

```
Element.prototype.addEventListener = function(e, h) {
    […]
    listeners[e].append(h);
}
```


PREAMBLE

```
var orig_f = Element.prototype.addEventListener;

Element.prototype.addEventListener = function(){
          console.log("new handler registration");
          return orig_f.apply(this, argument);
      };
```

Function Redefinition

CISPA Center for IT-Security, Privacy and Accountability

preamble

Application JS code

```
function handler() {
   alert("hello world");
}

el = document.getElementByID('img')
el.addEventListener("click", handler);
```


API

```
Element.prototype.addEventListener = function(e, h) {
    […]
    listeners[e].append(h);
}
```

PREAMBLE

```
var orig_f = Element.prototype.addEventListener;

Element.prototype.addEventListener = function(){
          console.log("new handler registration");
          return orig_f.apply(this, argument);
       };
```

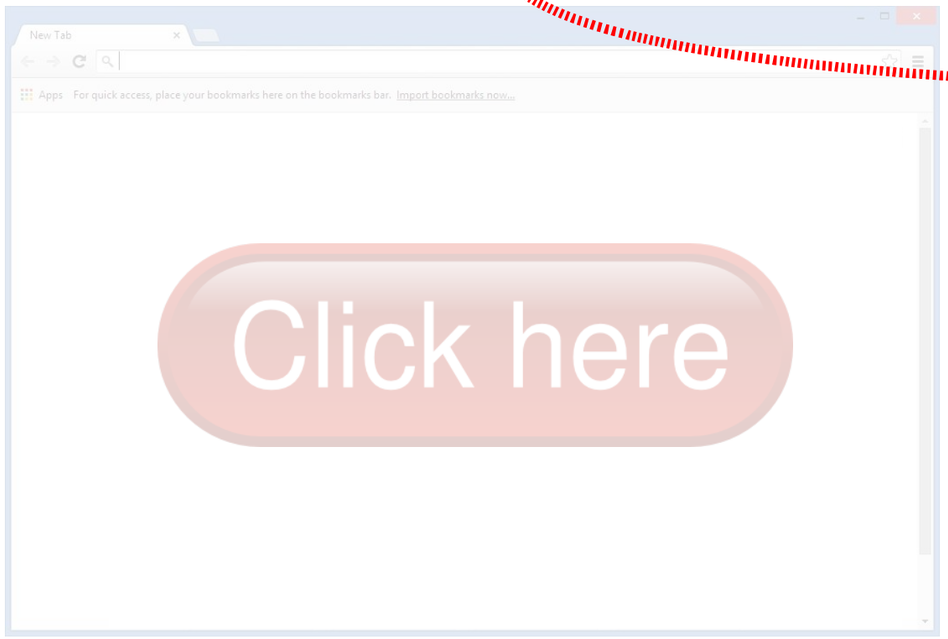
Function Redefinition

CISPA Center for IT-Security, Privacy and Accountability

preamble

Application JS code

```
function handler() {
   alert("hello world");
}

el = document.getElementById('img')
el.addEventListener("click", handler);
```



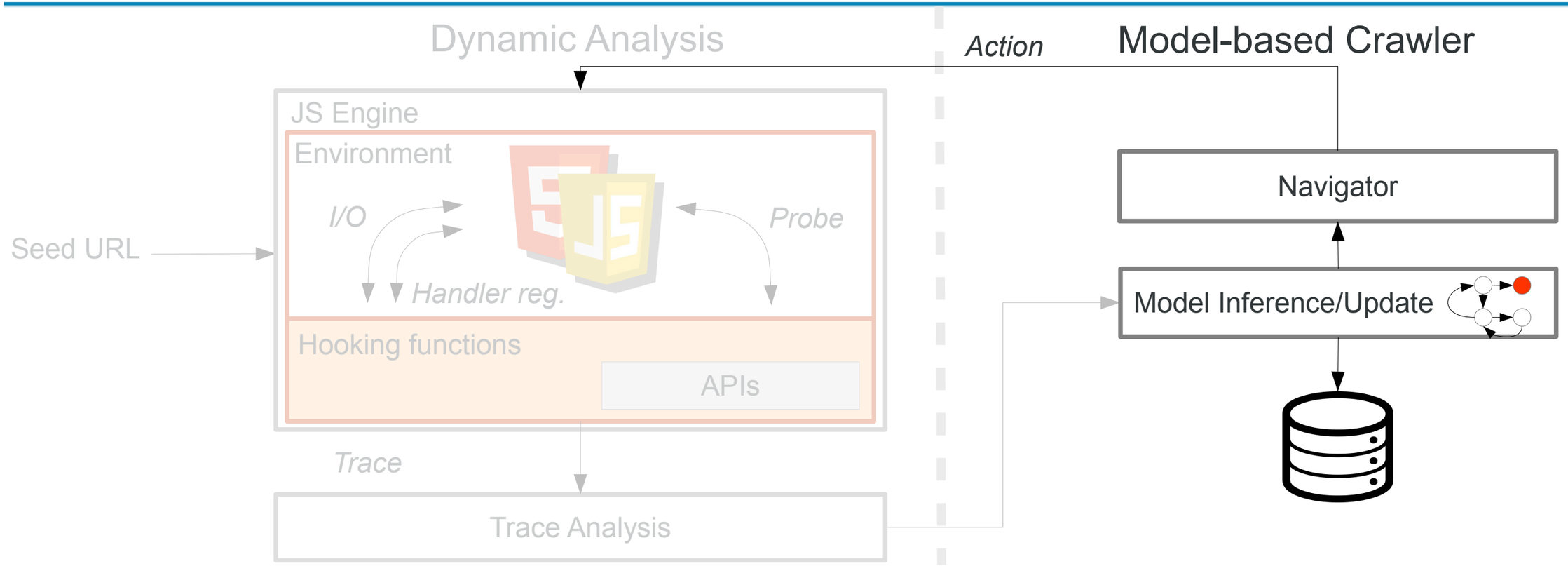
API

```
Element.prototype.addEventListener = function(e, h) {
    […]
    listeners[e].append(h);
}
```

PREAMBLE

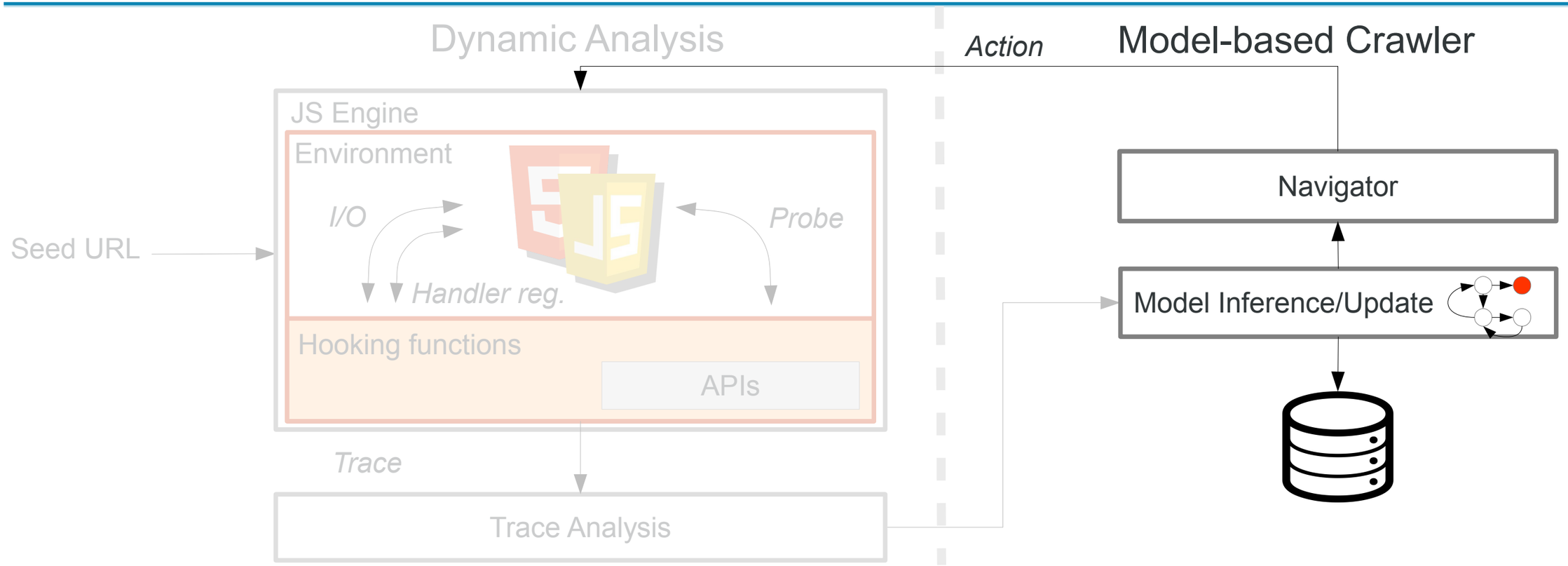
```
var orig_f = Element.prototype.addEventListener;

Element.prototype.addEventListener = function(){
          console.log("new handler registration");
          return orig_f.apply(this, argument);
       };
```

Model-based Crawler



- Creates and maintain a web application model
  - Oriented graph: nodes are page clusters and edges are URLs, HTML forms, or events
- Model used to decide on the next action
  - Priority: Events → high, URLs/forms → low

Model-based Crawler

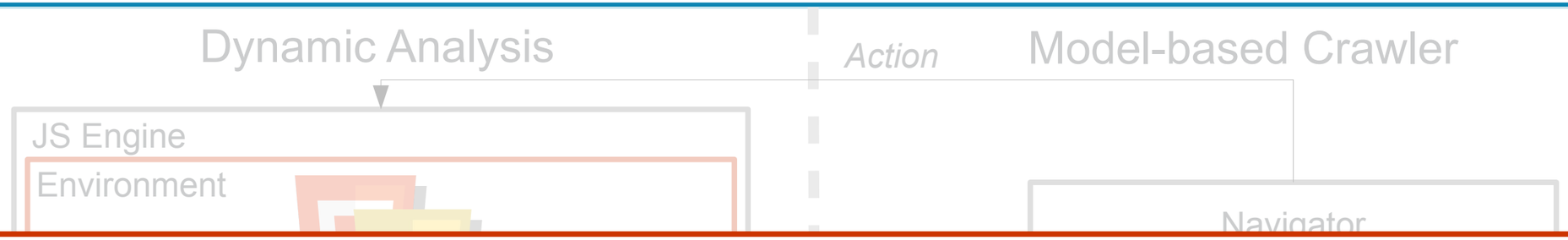


- Creates and maintain a web application model
  - Oriented graph: nodes are page clusters and edges are URLs, HTML forms, or events
- Model used to decide on the next action
  - Priority: Events → high, URLs/forms → low

Assessment

Our Tool

CISPA Center for IT-Security, Privacy and Accountability



- New tool: **jÄk** *[pron. Jack]*
- Source code on GitHub
  - `https://github.com/ConstantinT/jAEk`
- Free to run, copy, distribute, study, change and improve it
  - Free Software (GPL3) … and also free as free beer!

Experiments

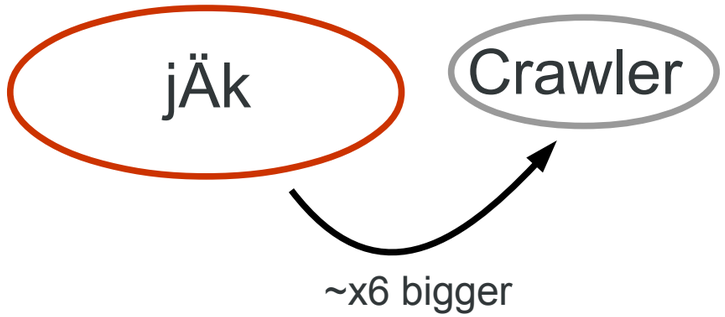
CISPA Center for IT-Security, Privacy and Accountability

- Comparative analysis
  - Skipfish, W3af, Wget, and Crawljax
- Case studies:
  - WIVET (Web Input Vector Extractor Teaser)
    - assess strength and limitations of existing crawlers
  - 13 web applications
    - studied coverage and vulnerability detection power

Coverage

CISPA Center for IT-Security, Privacy and Accountability

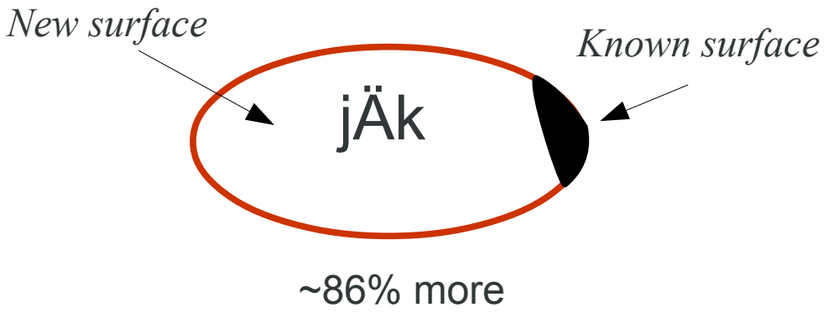
- Explored surface by jÄk (# of unique URL structs.)
  - x16 (Crawljax) to x2 (Skipfish) bigger



Coverage

CISPA Center for IT-Security, Privacy and Accountability

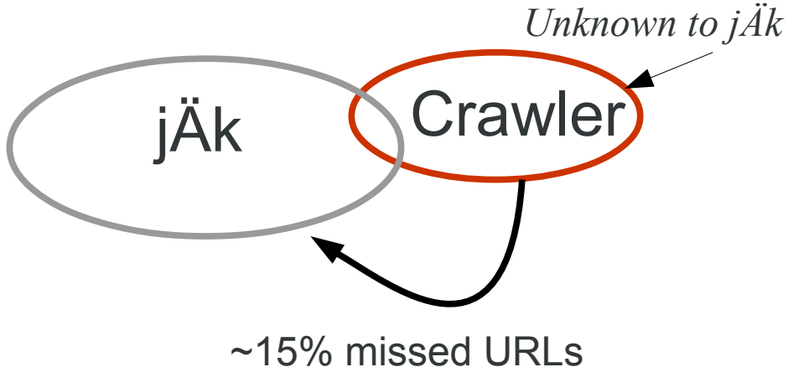
- Explored surface by jÄk (# of unique URL structs.)
  - x16 (Crawljax) to x2 (Skipfish) bigger
- Relative size of new surface:
  - From +70% (Wget) to +98% (Crawljax) of URLs are new



Coverage

CISPA Center for IT-Security, Privacy and Accountability

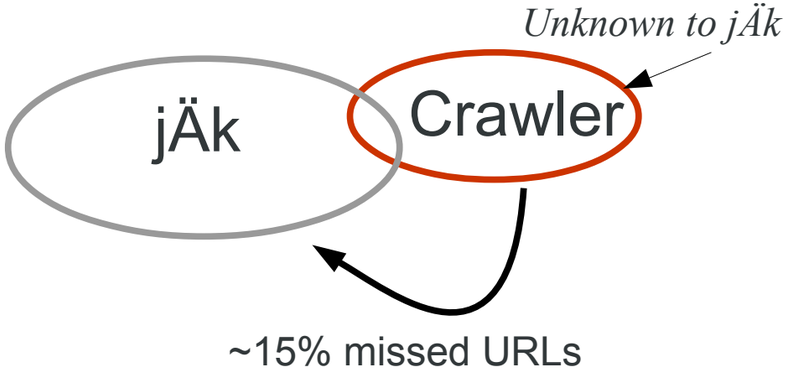
- Explored surface by jÄk (# of unique URL structs.)
  - x16 (Crawljax) to x2 (Skipfish) bigger
- Relative size of new surface:
  - From +70% (Wget) to +98% (Crawljax) of URLs are new
- Global surface missed by jÄk:
  - From 22% (Skipfish) to 0.5% (Crawljax) are missed



Coverage

CISPA Center for IT-Security, Privacy and Accountability

- Explored surface by jÄk (# of unique URL structs.)
  - x16 (Crawljax) to x2 (Skipfish) bigger
- Relative size of new surface:
  - From +70% (Wget) to +98% (Crawljax) of URLs are new
- Global surface missed by jÄk:
  - From 22% (Skipfish) to 0.5% (Crawljax) are missed
  - Further analysis:
    - 75% of missed are due to URL forgery
    - 25% to static resources, unsupported action, and others



Conclusion

Conclusion/Takeaway

CISPA Center for IT-Security, Privacy and Accountability

- Novel technique based on
  - **dynamic analysis** of JS program + **model-based crawling**
- Built **jÄk**, a tool implementing our approach
- Assessed against 13 web applications
- Our result show that jÄk explores a surface ~6x larger with +86% new URLs