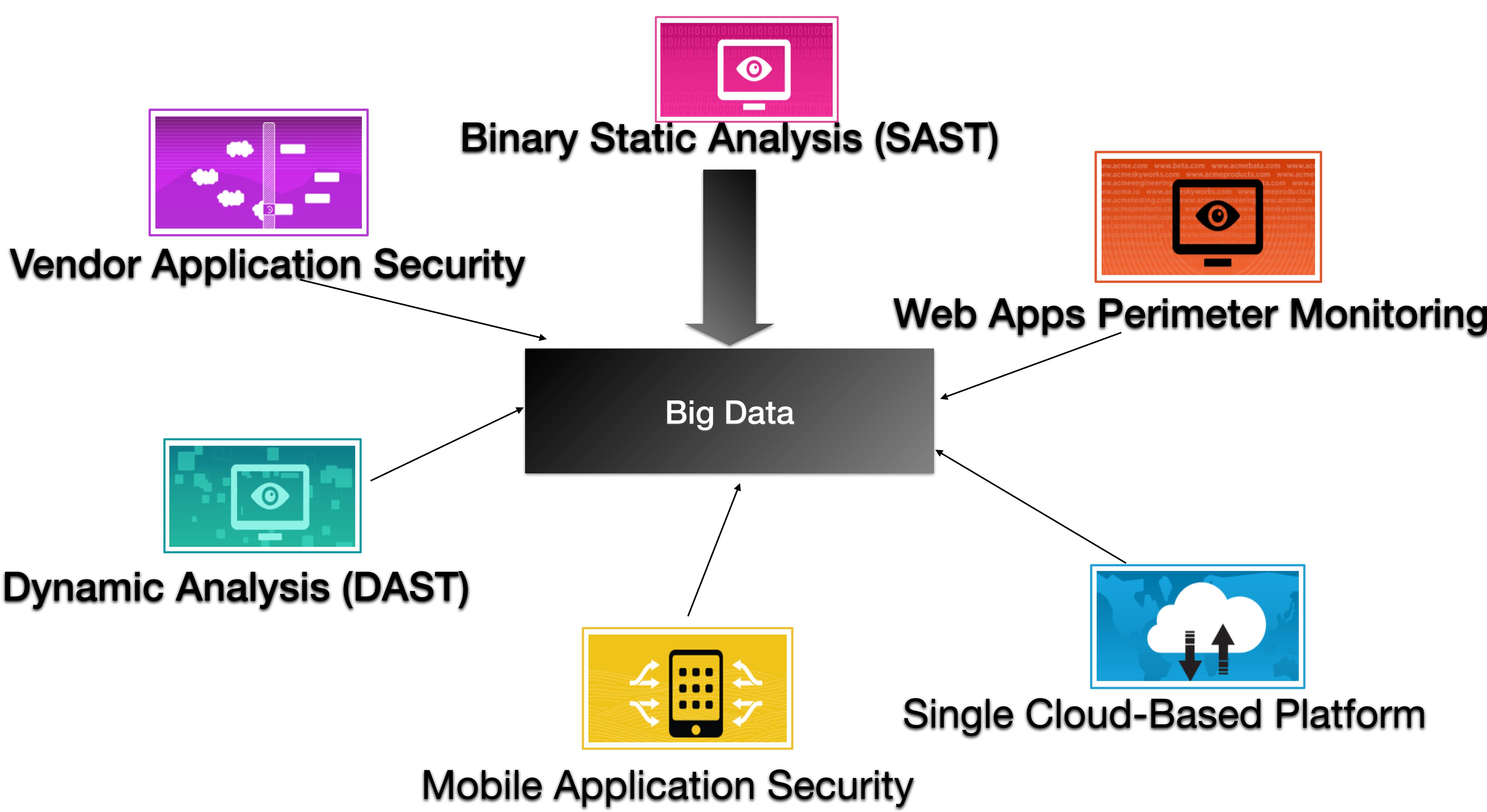


# Building Security Analytics using Native XML Databases

Mansi Sheth

Email: [msheth@veracode.com](mailto:msheth@veracode.com)  
Twitter: @1MansiS





The screenshot shows an IDE interface with two main panes. The left pane displays the Java code for the `Login` class, which extends `ExampleSupport`. It includes methods for executing actions based on user input validation and returning success or input results. The right pane displays the `example.xml` configuration file for Struts, defining a package named `example` with various actions like `*`, `HelloWorld`, and `Login!*`.

```
package example;

public class Login extends ExampleSupport {

    public String execute() throws Exception {
        if (isValid(username)) return INPUT;
        if (isValid(password)) return INPUT;
        return SUCCESS;
    }

    private boolean isValid(String value) {
        return (value == null || value.length() == 0);
    }

    private String username;

    public String getUsername() {
        return username;
    }
}

<struts>
<package name="example" namespace="/example" extends="struts-default">
    <action name="*" class="example.ExampleSupport">
        <result>/example/{1}.jsp</result>
    </action>
    <action name="HelloWorld" class="example.HelloWorld">
        <result>/example/HelloWorld.jsp</result>
    </action>
    <action name="Login!*" class="example.Login" method="{1}">
        <result name="input"/>/example/Login.jsp
        </result>
        <result type="redirect-action">Menu</result>
    </action>
</package>
</struts>
```

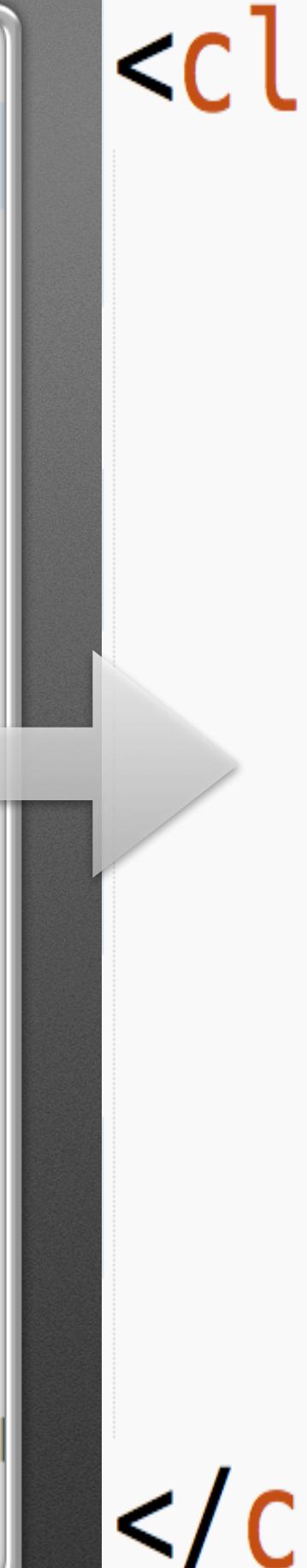
# #XML Metadata

```
>Login.class ✘  
  
package example;  
  
public class Login extends ExampleSupport {  
  
    public String execute() throws Exception {  
  
        if (isValid(username)) return INPUT;  
  
        if (isValid(password)) return INPUT;  
  
        return SUCCESS;  
    }  
  
    private boolean isValid(String value) {  
        return (value == null || value.length() == 0);  
    }  
}
```



```
<class name="example.Login">  
    <inheritance  
        name="example.ExampleSupport"/>  
    <function name="isValid"/>  
    <function name="<init>">  
        <apiCalls  
            name="example.ExampleSupport:<init>"/>  
    </function>  
    <function name="execute">  
        <apiCalls  
            name="example.Login:getUsername"/>  
        <apiCalls  
            name="example.Login:isValid"/>  
        <apiCalls  
            name="example.Login:getPassword"/>  
        <apiCalls  
            name="example.Login:isValid"/>  
    </function>  
</class>
```

```
<struts>
  <package
    name="example"
    namespace="/example"
    extends="struts-default">
    <action
      name="*"
      class="example.ExampleSupport">
      <result>/example/{1}.jsp</result>
    </action>
    <action
      name="HelloWorld"
      class="example.HelloWorld">
      <result>/example/HelloWorld.jsp</result>
    </action>
```



```
<class name="WEB-INF/example.xml">
  <function name="action/@name">
    <apiCalls
      name="HelloWorld"/>
    <apiCalls
      name="Login!*"/>
    <apiCalls
      name="*"/>
  </function>
</class>
```

```
<archive name="struts2-blank-2.0.1.war">
    <class name="example.Login">
        <inheritance name="example.ExampleSupport"/>
        <function name="isValid"/>
        <function name="<init>">
            <apiCalls name="example.ExampleSupport:<init>"/>
        </function>
        <function name="execute">
            <apiCalls name="example.Login:getUsername"/>
            <apiCalls name="example.Login:isValid"/>
            <apiCalls name="example.Login:getPassword"/>
            <apiCalls name="example.Login:isValid"/>
        </function>
    </class>
    <class name="WEB-INF/src/java.example.xml">
        <function name="action/@name">
            <apiCalls name="HelloWorld"/>
            <apiCalls name="Login!*>"/>
            <apiCalls name="*"/>
        </function>
    </class>
    <class name="WEB-INF.web.xml">
        <function name="filter-class">
            <apiCalls name="org.apache.struts2.dispatcher.FilterDispatcher"/>
        </function>
    </class>
</archive>
</Archives>
```

# Trusted by the world's largest enterprises

## Healthcare + Insurance



## Financial Services



## Technology



## Public Sector



## Telco + Media



## Major Brands



$$10\text{MB} * 300 \approx 3\text{GB}$$

$$3\text{GB} * 5 * 52 \approx 1\text{TB}$$

$$1\text{TB} * 7 \approx 7\text{TB}$$

Clients using insecure api: org.insecured:f\_insecure()

Top 10 most prevalent frameworks across all our clients ?

Most inherited entry points from framework f ?

# Approaches Considered

Bash Automated Script

Relational Database

Hadoop + Mahout

Hadoop + AVRO + HIVE

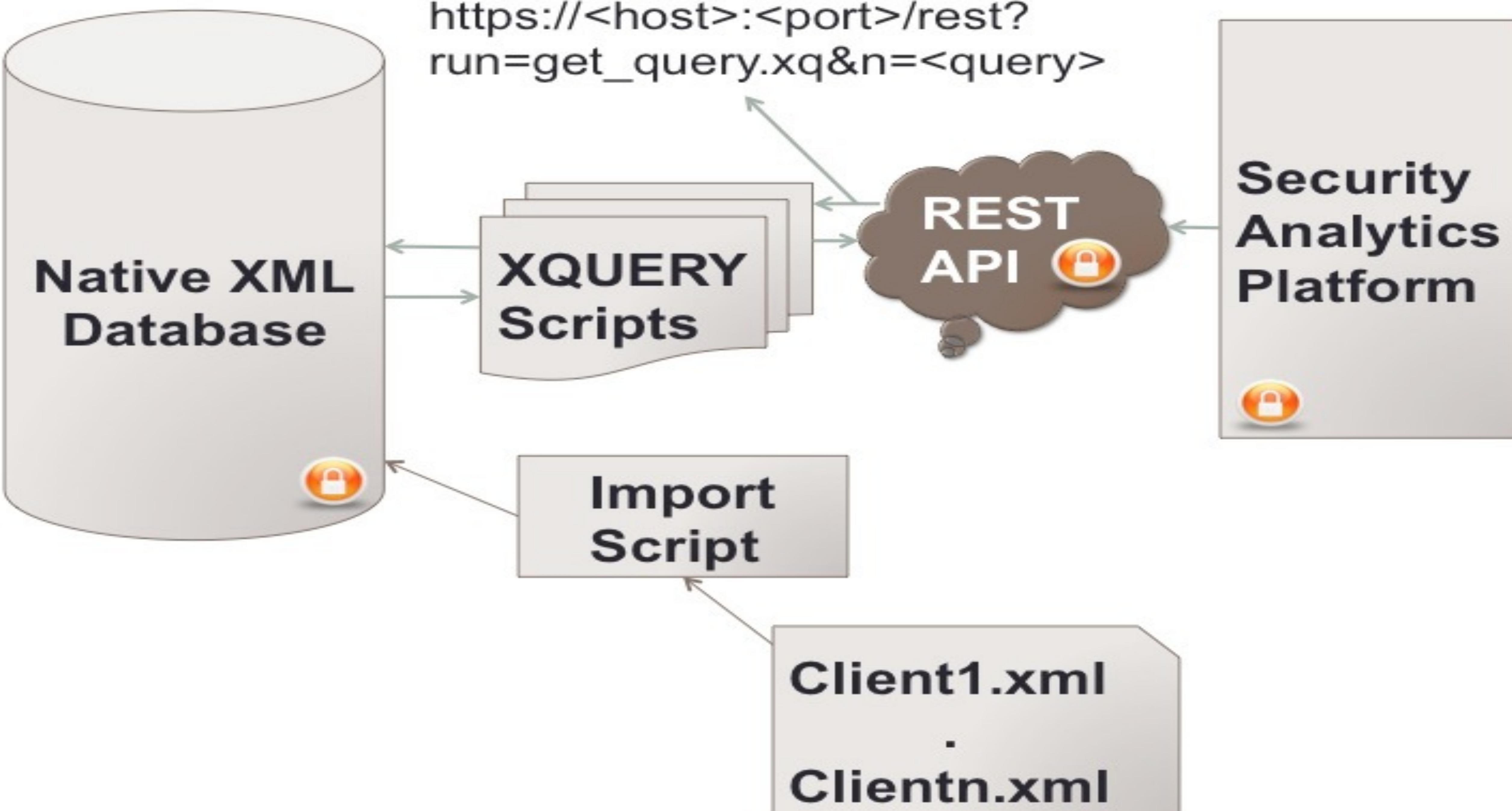
NoSQL

# Wish List

Store Parsed XML

Use XML Query technologies

Results in reasonable timeframe



```
$n = <query>
```

```
results =
```

```
for all docs in NXD:  
    return eval($n)
```

```
return unique(result)
```

# Exploring Struts 2 OGNL Vulnerability CVE-2013-2251

## Languages

- [English](#)
- [Espanol](#)

**1TB**

**14 million**

**70 min**

# Credits

Chris Eng

Veracode Management

BaseX Community

XMLPrague Committee

Photo Credits: Kenneth Rougeau

# Building Security Analytics using Native XML Databases

## Mansi Sheth

Email: [msheth@veracode.com](mailto:msheth@veracode.com)

Twitter: [@1MansiS](#)

Blog: <http://www.veracode.com/blog>



2010 © Kenneth Rougeau - <http://synchronicity313.etsy.com>