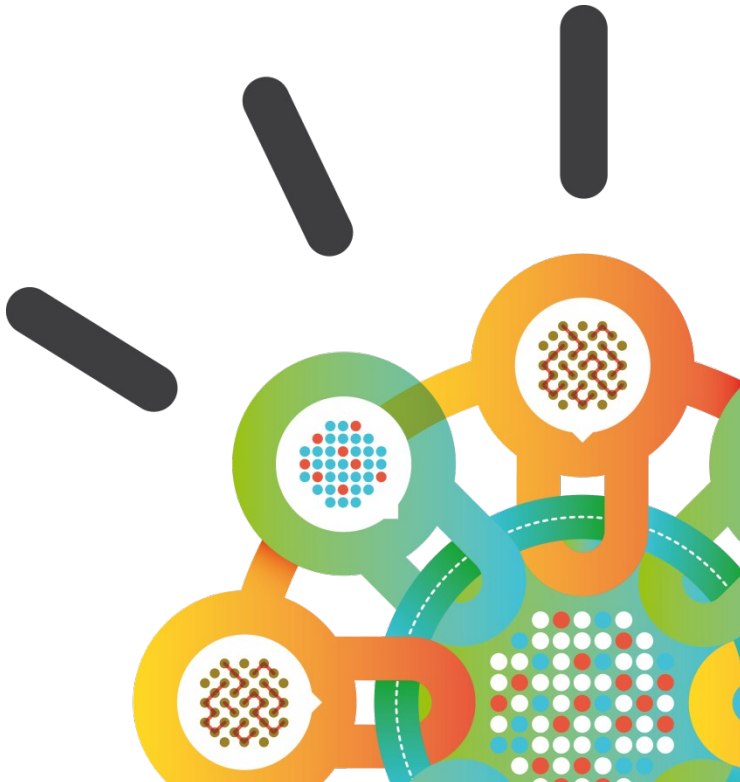
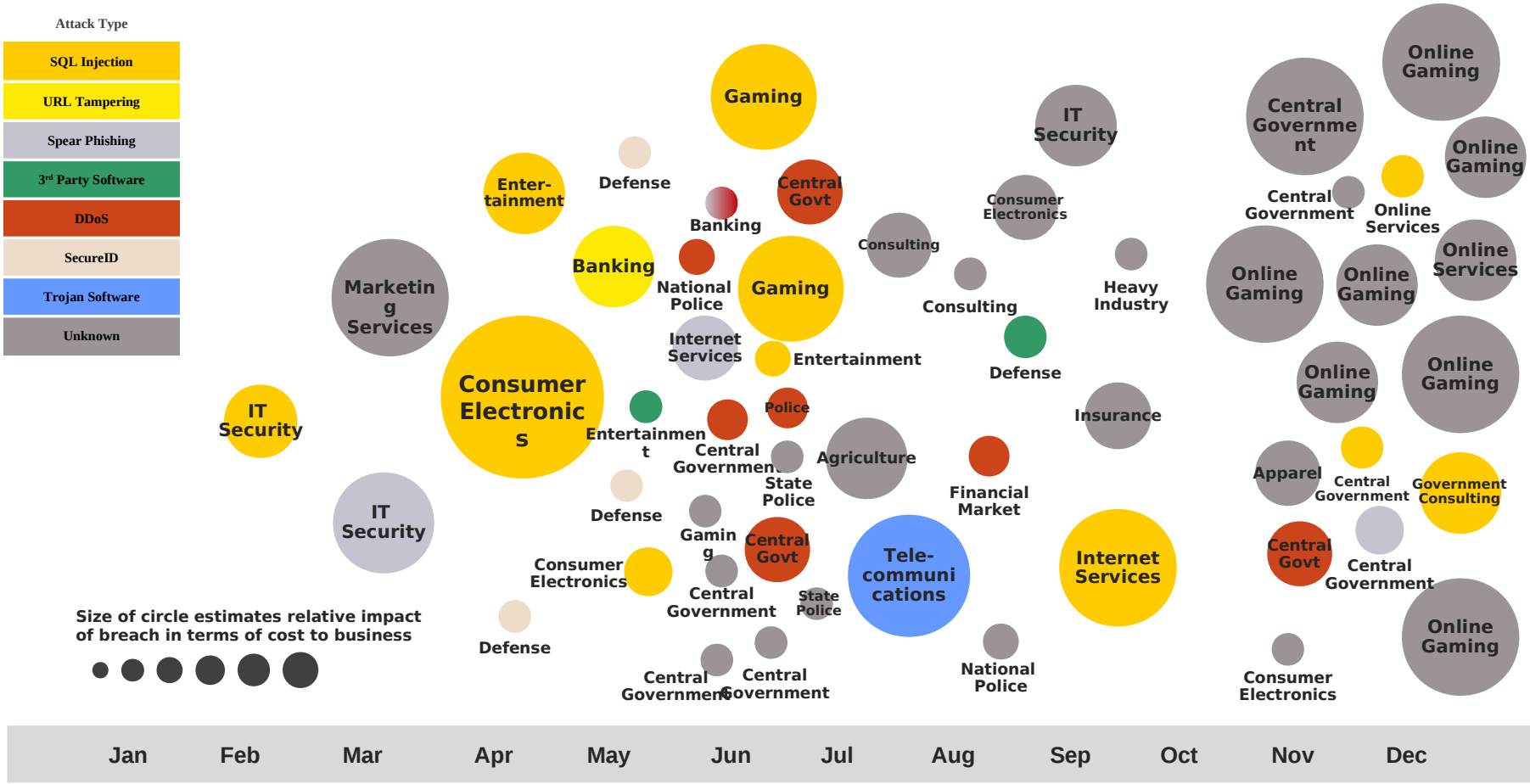
AppScan 9.0 Security for Applications

Stefan.Mandl@de.ibm.com



2011 Sampling of Security Incidents by Attack Type, Time and Impact



Source: IBM X-Force® Research 2011 Trend and Risk Report

und im Jahr 2012, …

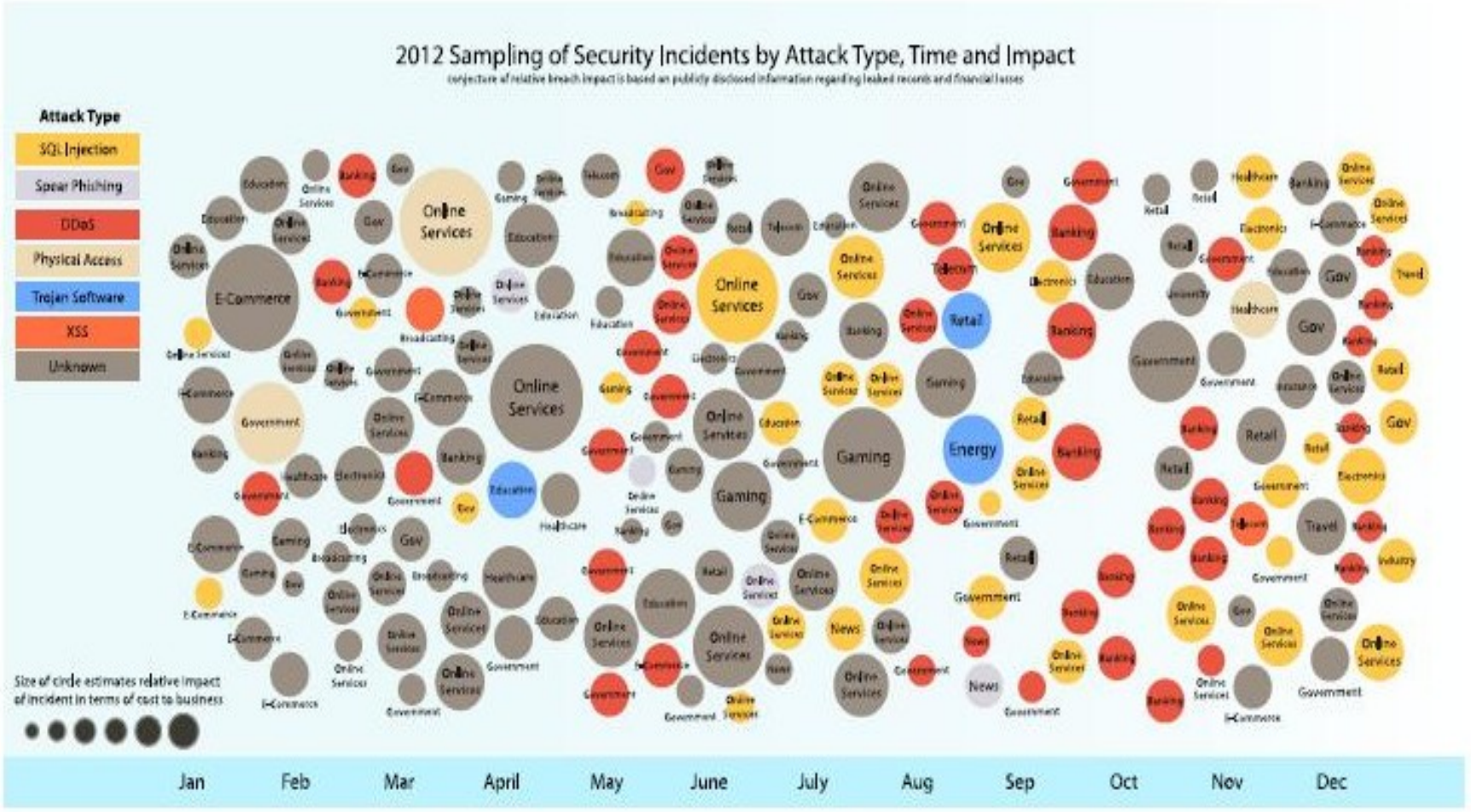
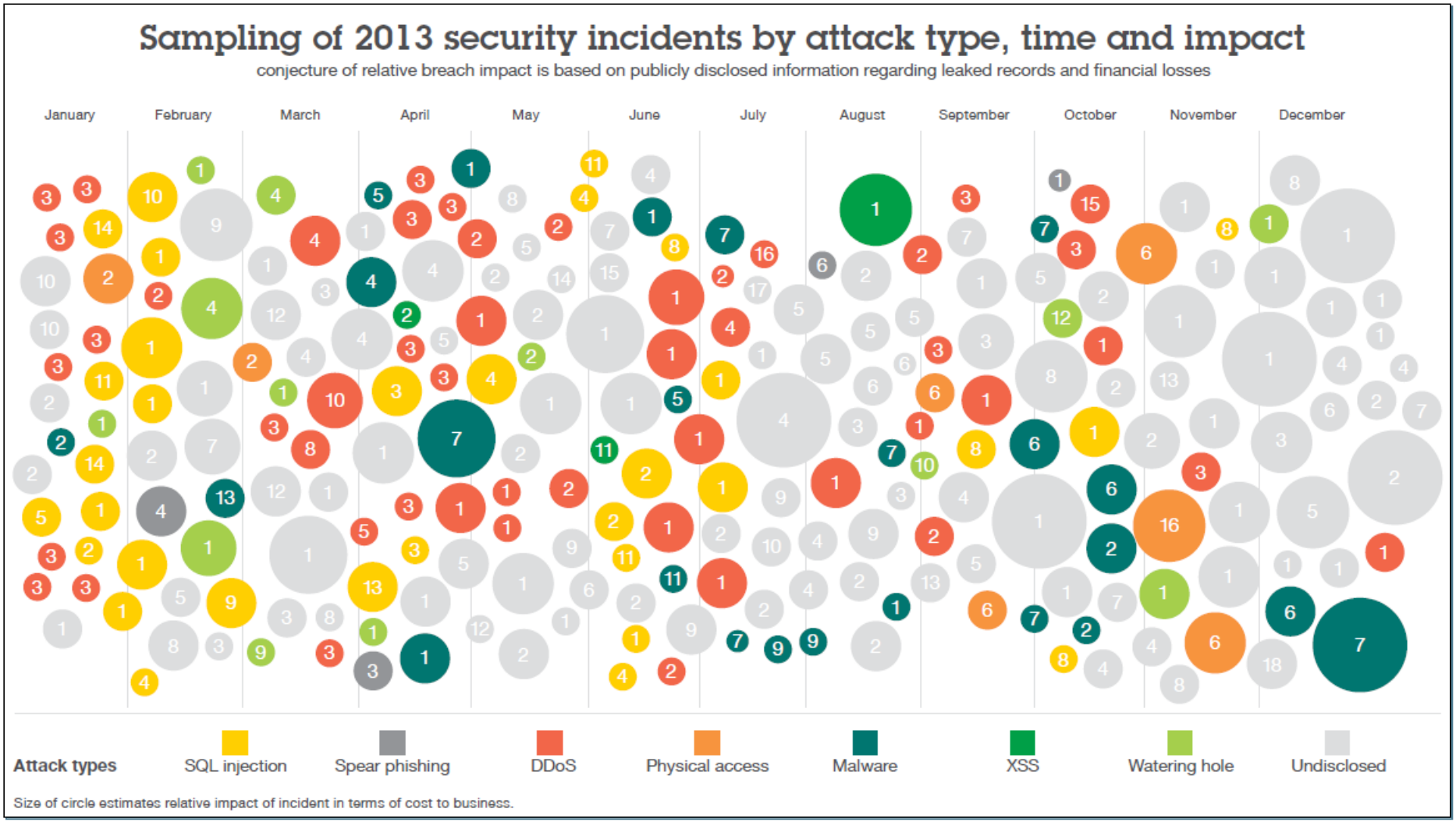


Figure 3: 2012 Sampling of Security Incidents by Attack Type, Time and Impact

Source: IBM X-Force® Research 2012 Trend and Risk Report

und 2013, …



Source: IBM X-Force® Research 2013 Trend and Risk Report

Der Mythos: “Wir sind sicher”

Wir haben Firewalls

Wir lassen jedes Jahr ein Audit durchführen

Wir haben SSL Verschlüsselung

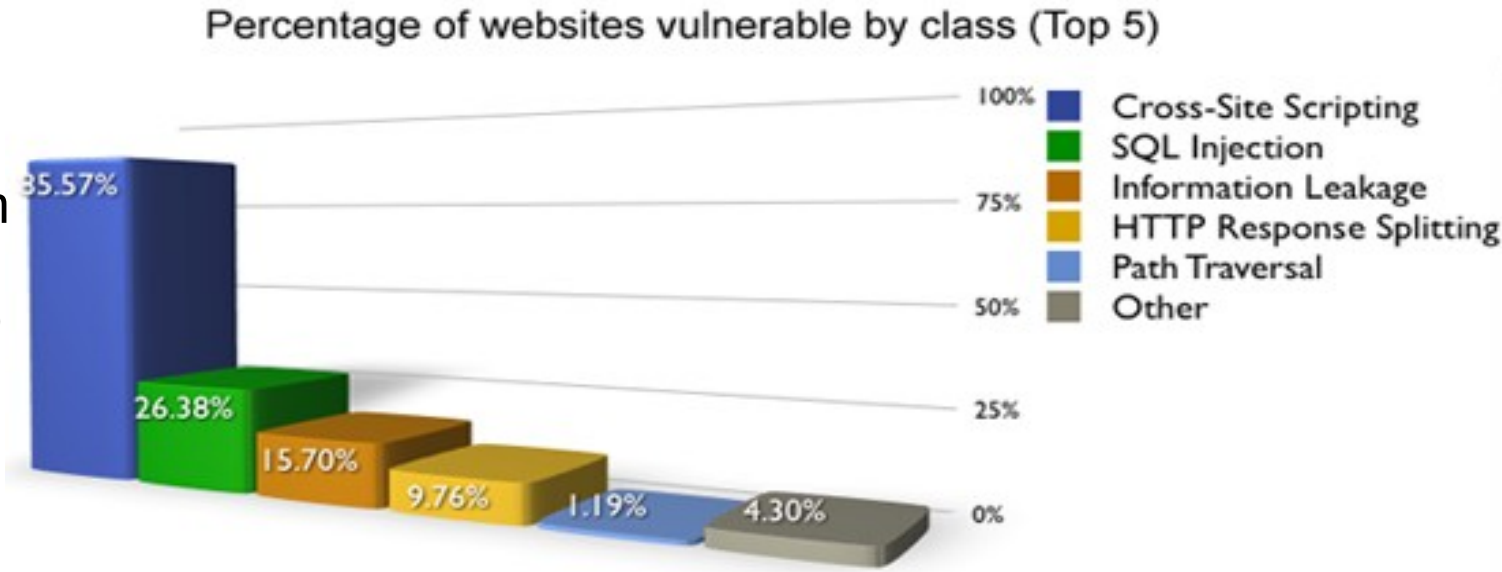
Wir haben Netzwerk-Scanner

… und die Security Lücken

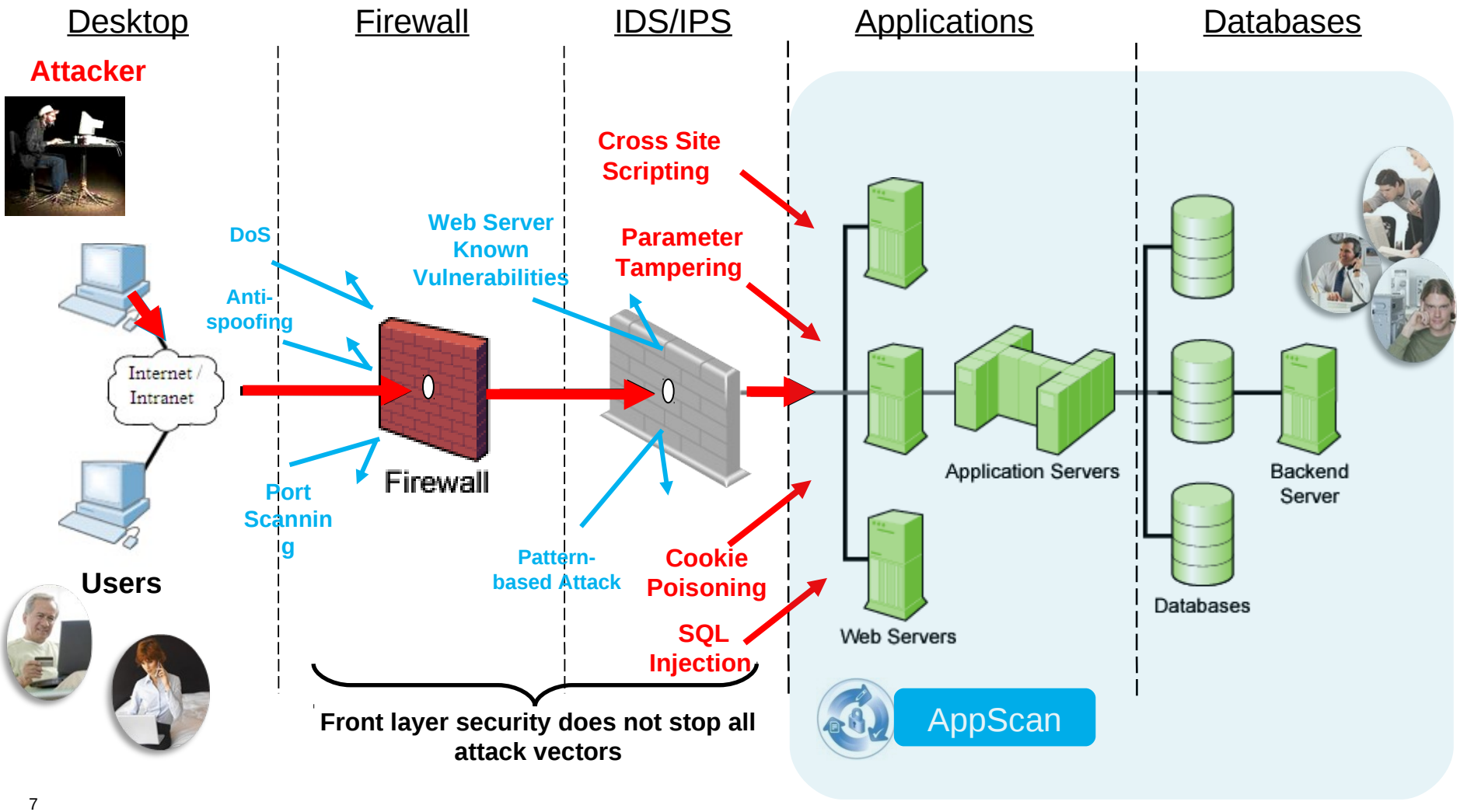
Top Ten des "Open Web Application Security Project" (OWASP)

1. **"Cross-Site Scripting" (XSS)**
2. "SQL Injection"
3. Verstecktes Ausführen einer Datei
4. Unsichere "Direct Object Reference"
5. Verfälschung eines "Cross-Site Requests"
6. Informationsverlust und unsaubere Fehlerbehandlung
7. Broken Authentication & Session Management
8. Unsichere Kryptografie Speicherung
9. Unsichere Kommunikation
10. Fehlerhafte Abwehr von URL Zugriffen

Schwachstellen Statistik (31.373 Seiten)



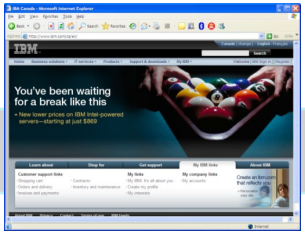
Wie “lösen” Unternehmen das Problem?



Analyse der Applikation

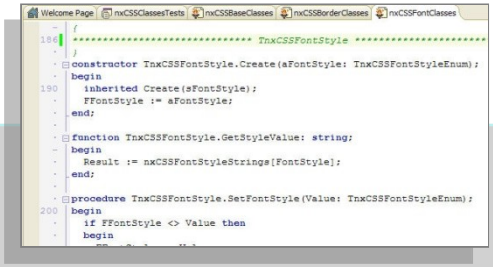
Dynamische Analyse = Blackbox

Scanning nach Security Schwachstellen auf Basis der compilierten Applikation



Statische Code Analyse = Whitebox

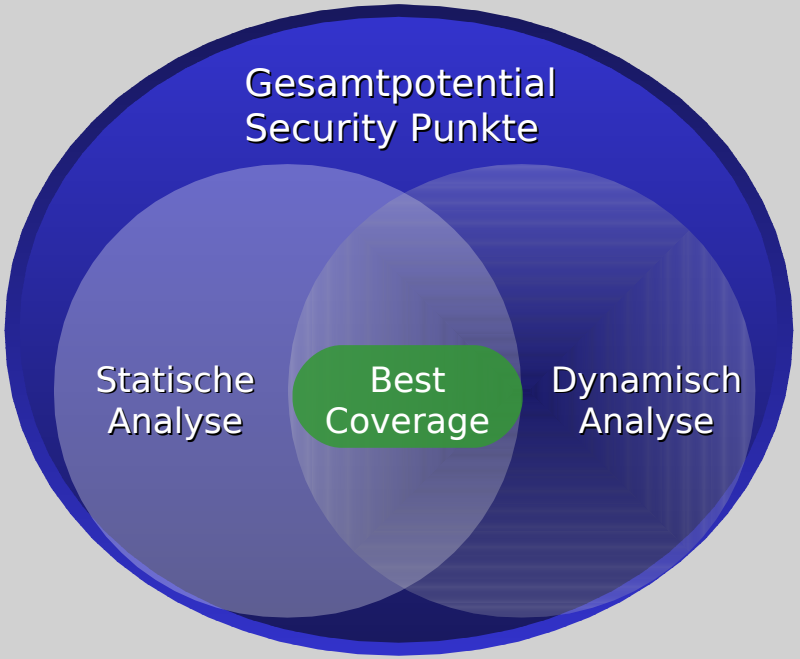
Scanning nach Security Schwachstellen auf Source Code Basis



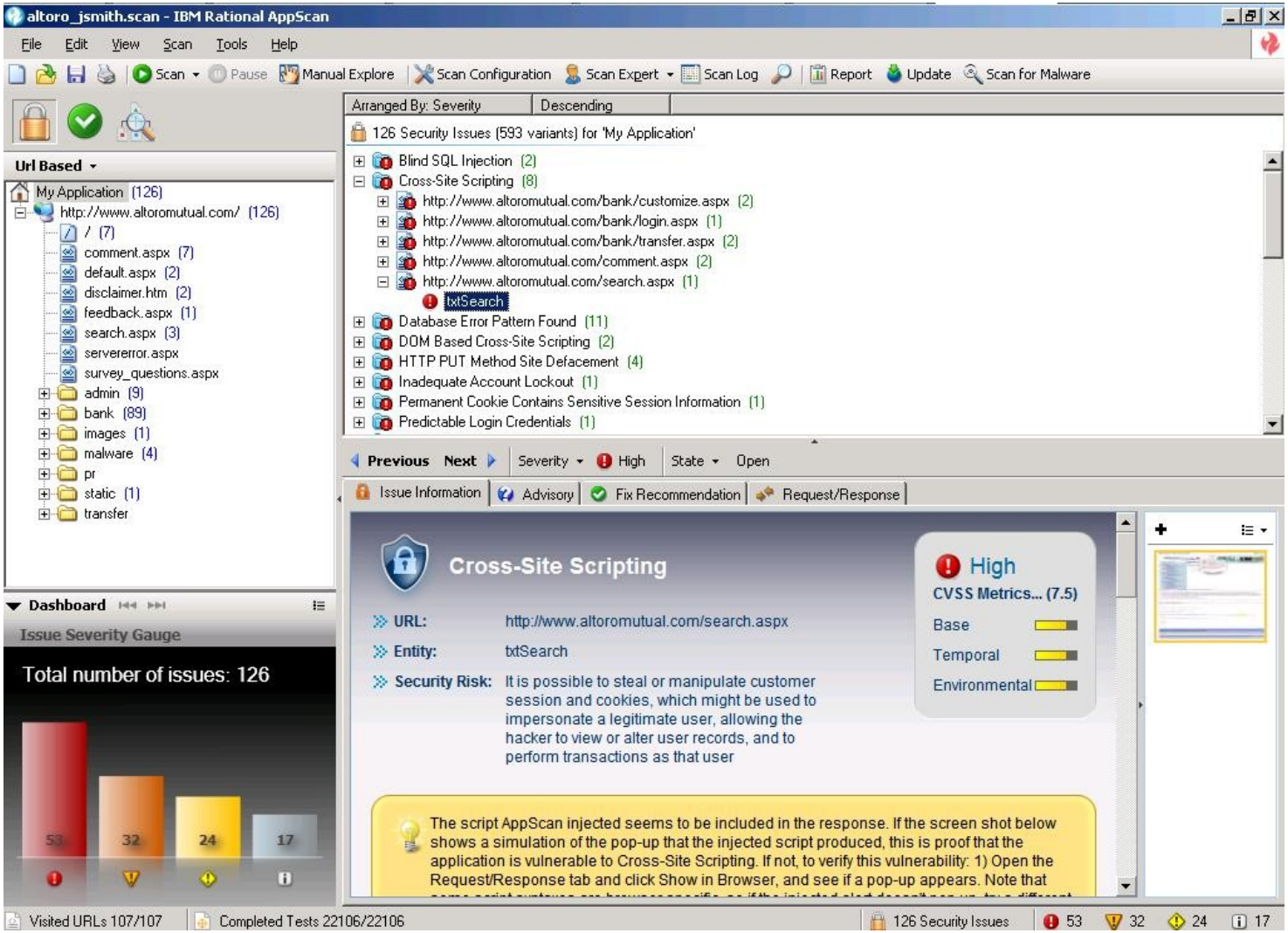
```
Welcome Page | nxCSSClassesTests | nxCSSBaseClasses | nxCSSBorderClasses | nxCSSFontClasses
{
********************************* TnxCSSFontStyle *********************************
}
constructor TnxCSSFontStyle.Create(aFontStyle: TnxCSSFontStyleEnum);
begin
  inherited Create(sFontStyle);
  FFontStyle := aFontStyle;
end;

function TnxCSSFontStyle.GetStyleValue: string;
begin
  Result := nxCSSFontStyleStrings[FontStyle];
end;

procedure TnxCSSFontStyle.SetFontStyle(Value: TnxCSSFontStyleEnum);
begin
  if FFontStyle <> Value then
  begin
```



AppScan Standard Edition



Deployed Application

Dynamic Analysis

AppScan Standard (Enterprise)

(DAST)

Deployed Application

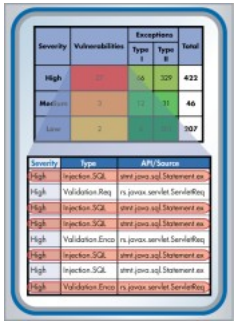
Dynamic Analysis

AppScan Standard

(DAST)

CODE BASE

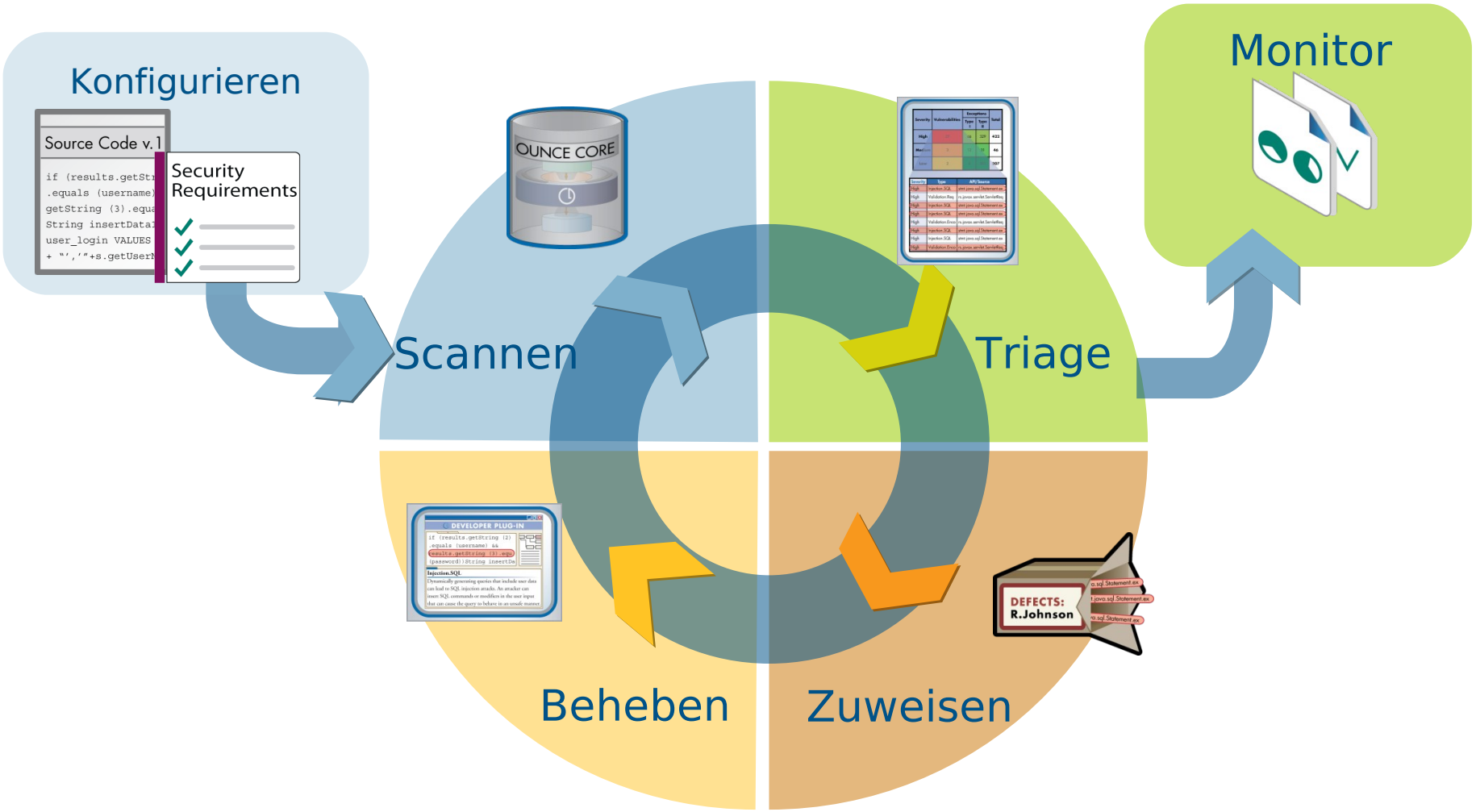
Static Analysis



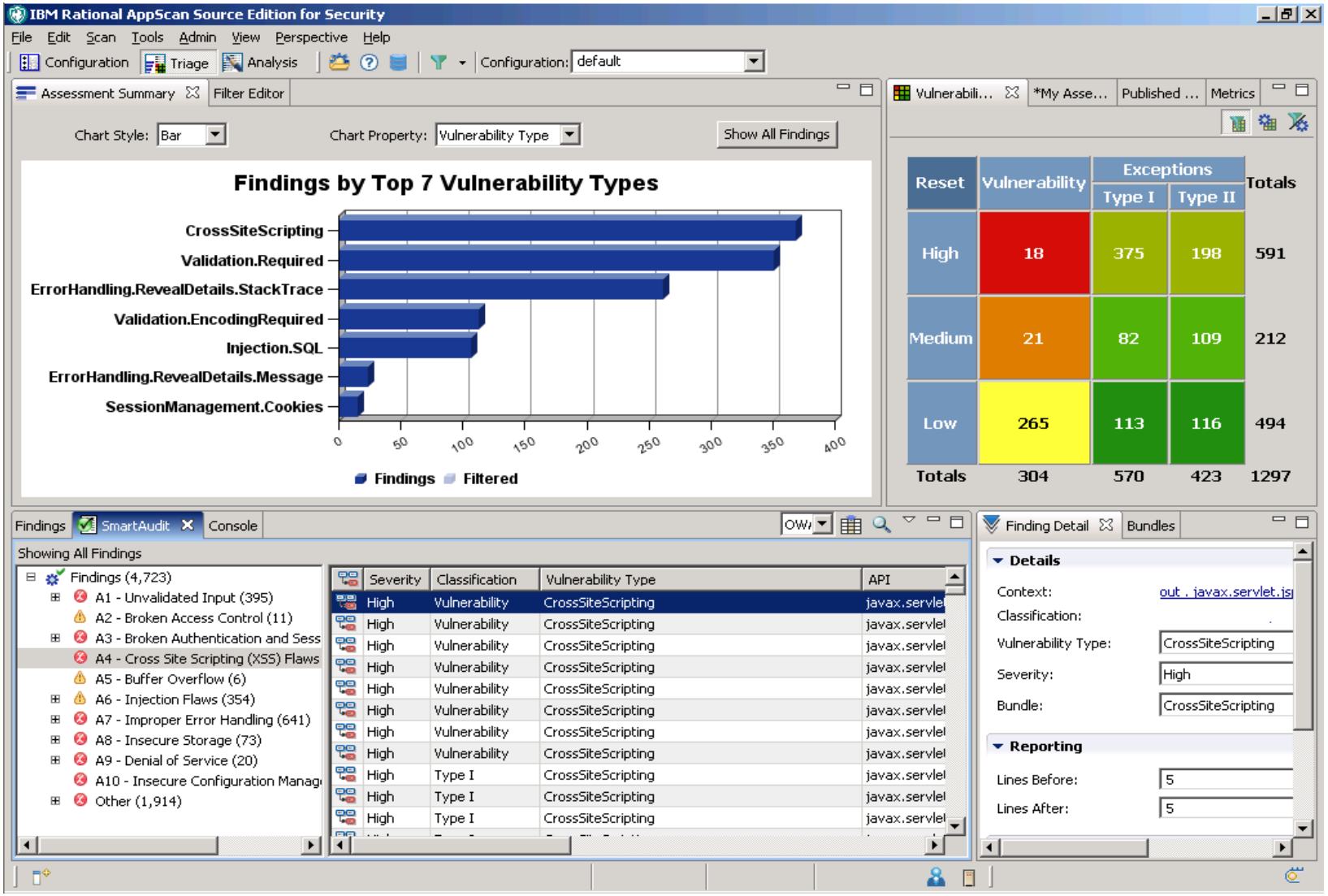
AppScan Source

(SAST)

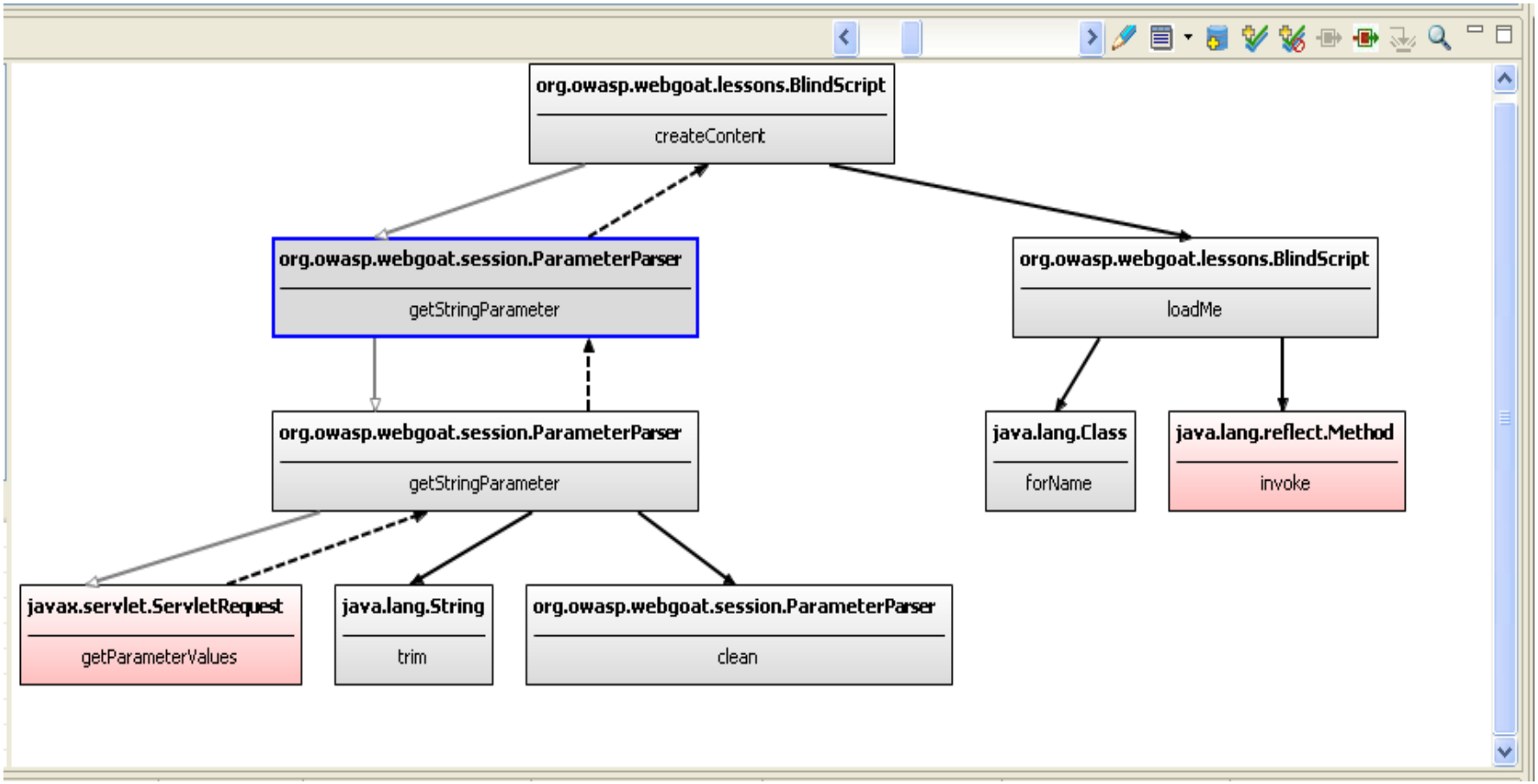
AppScan Source Edition - Workflow



AppScan Source Edition

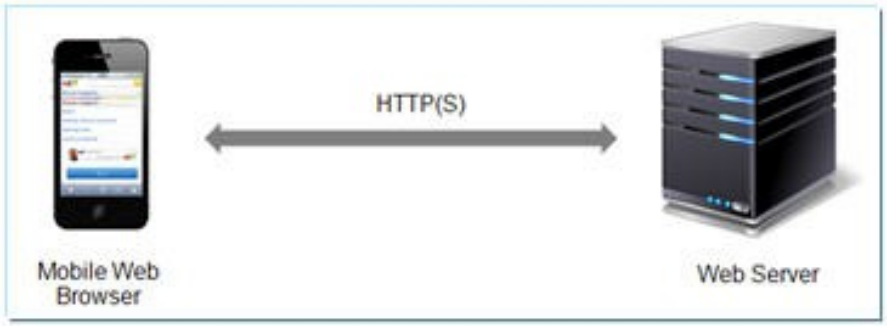


Datenflussmodell in AppScan Source Edition



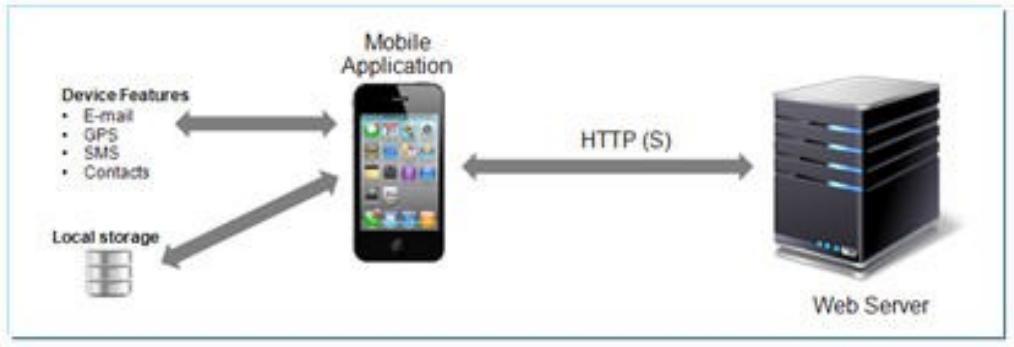
Security Risk a Function of Mobile Application Type

Mobile Web applications

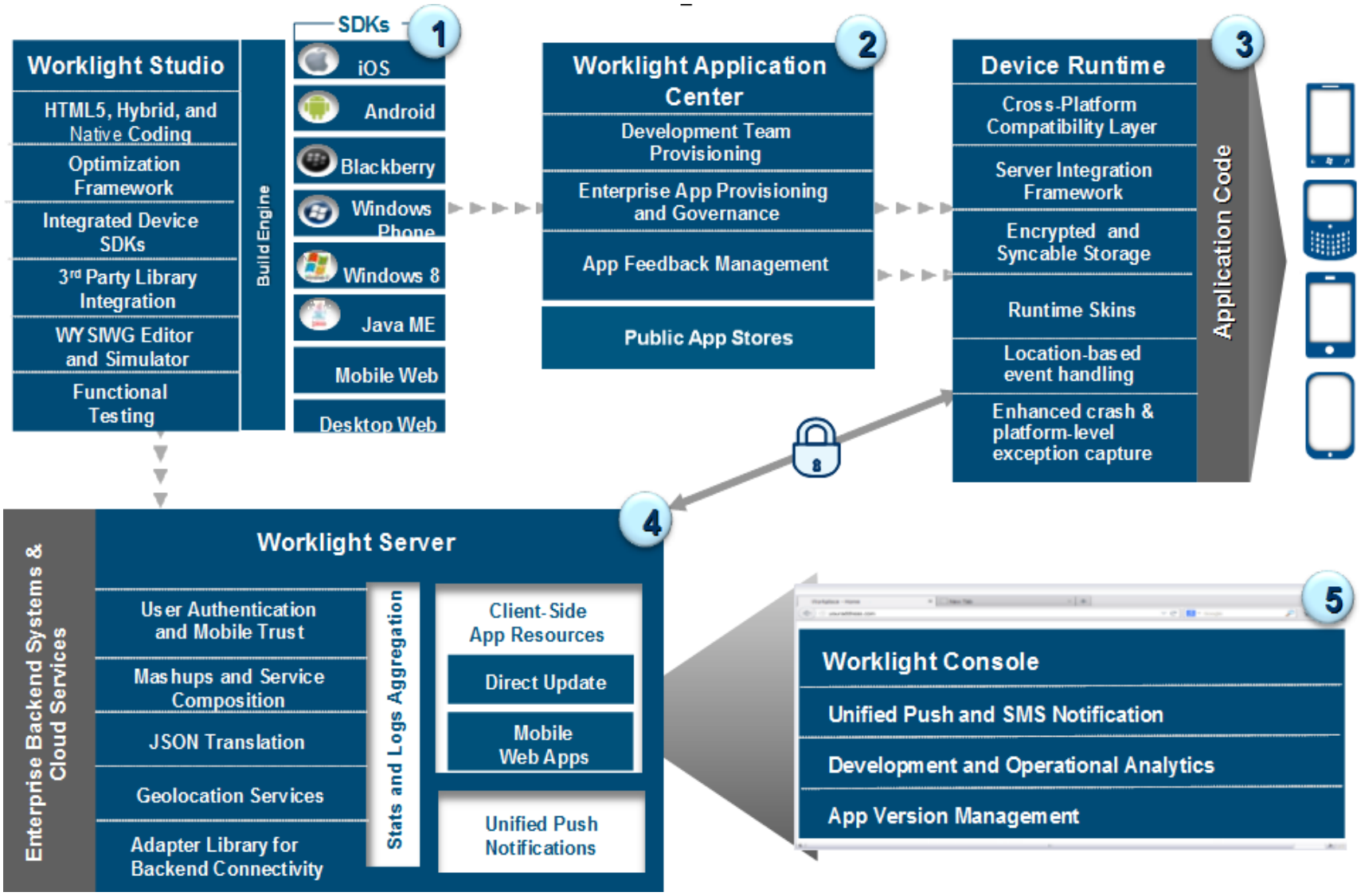


Mobile Native applications

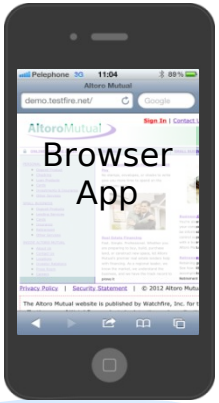
Mobile Hybrid applications



Was ist IBM Worklight ?



AppScan Mobile Support: Server and Native



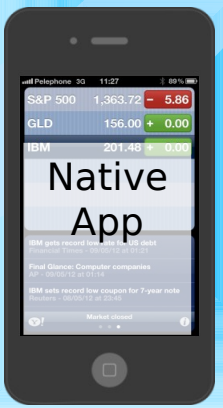
Mobile Web Apps

JavaScript / HTML5 hybrid analysis

Server Side Logic

SAST (source code)

DAST (web interfaces)

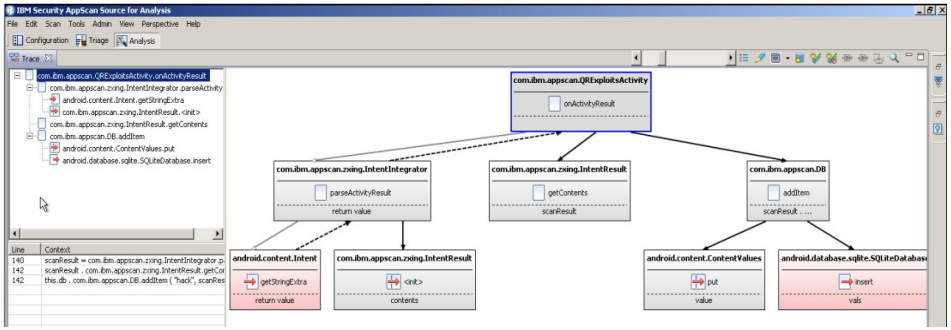


Native Apps

Android applications		*Static Analysis*
iOS applications		*Static Analysis*
JavaScript		*Static Analysis*

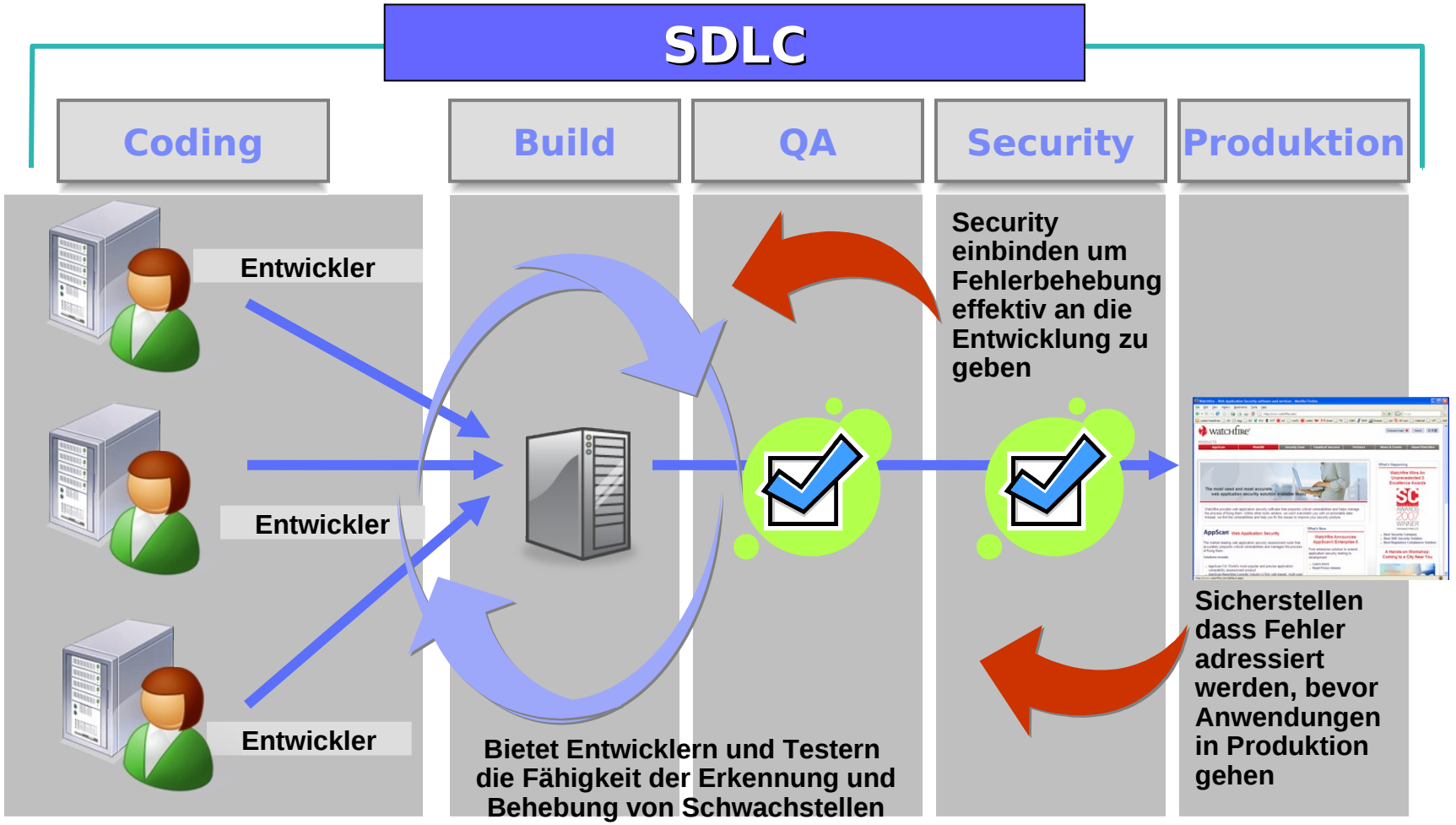
IBM AppScan Source – Native SAST Support

The most comprehensive mobile application security!



- Support for **Native Android** and **iOS** applications
- Security SDK research & risk assessment of over 40,000 APIs
- **Full** call and data flow analysis
  - Java
  - JavaScript
  - **Objective-C** (Mac OS X only)
- **Identify** where sensitive **data** is being **leaked**
- Ensures applications are not susceptible to malware

Sicherheitstests bereits in den Entwicklungsprozess einbinden



ibm.com/security