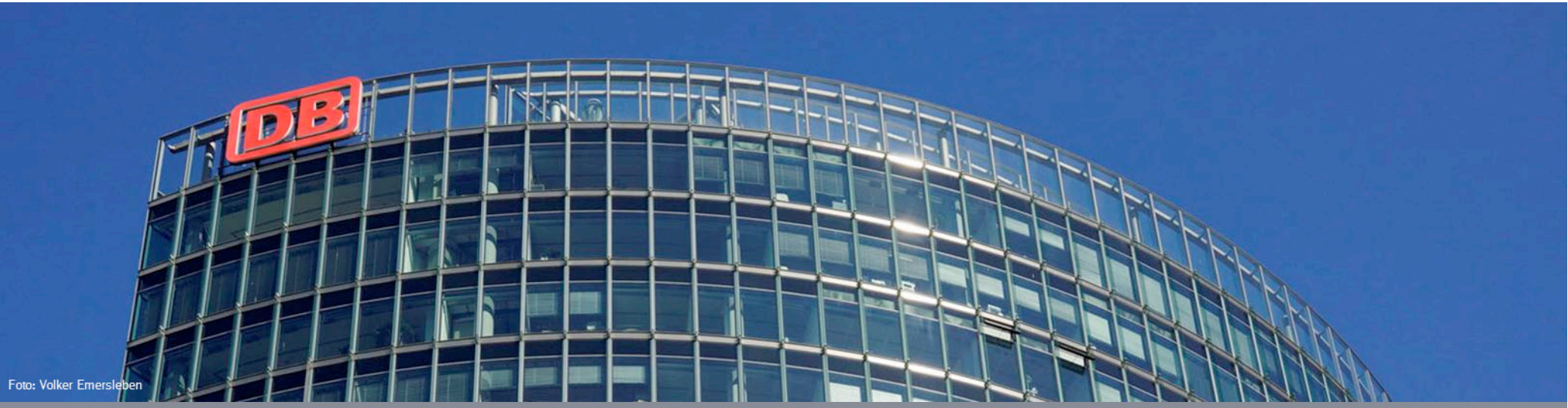


Foto: Volker Emersleben

Automating Security in CI/CD Pipelines

Our OWASP ZAP Journey

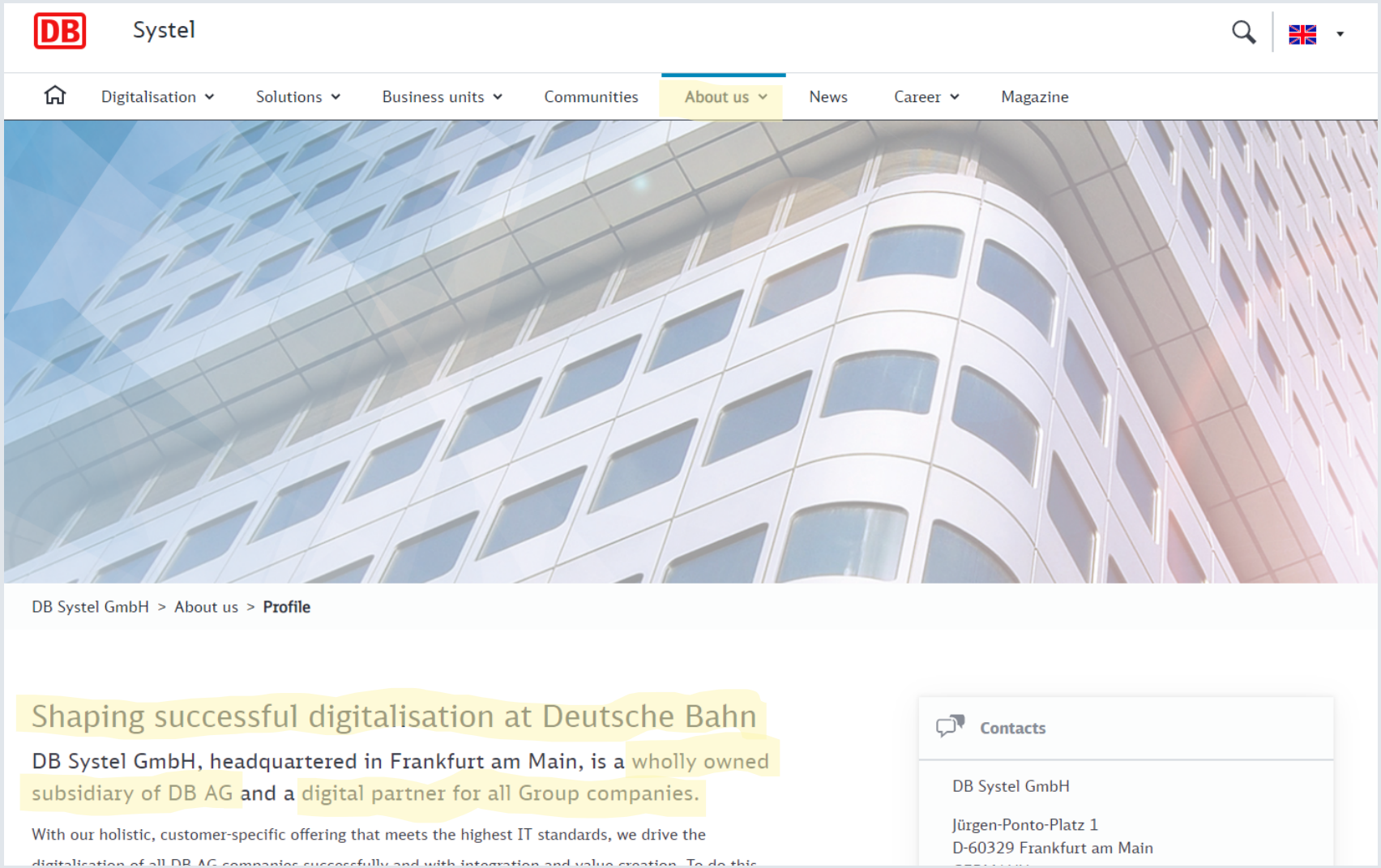
DB Systel
Moving the digital future. Together.

DB Systel GmbH | Sebastian Biehler, André König, Tommaso Nuccio |
Team Application Security Specialists | OpenRheinMain 2019 | 2019-09-13

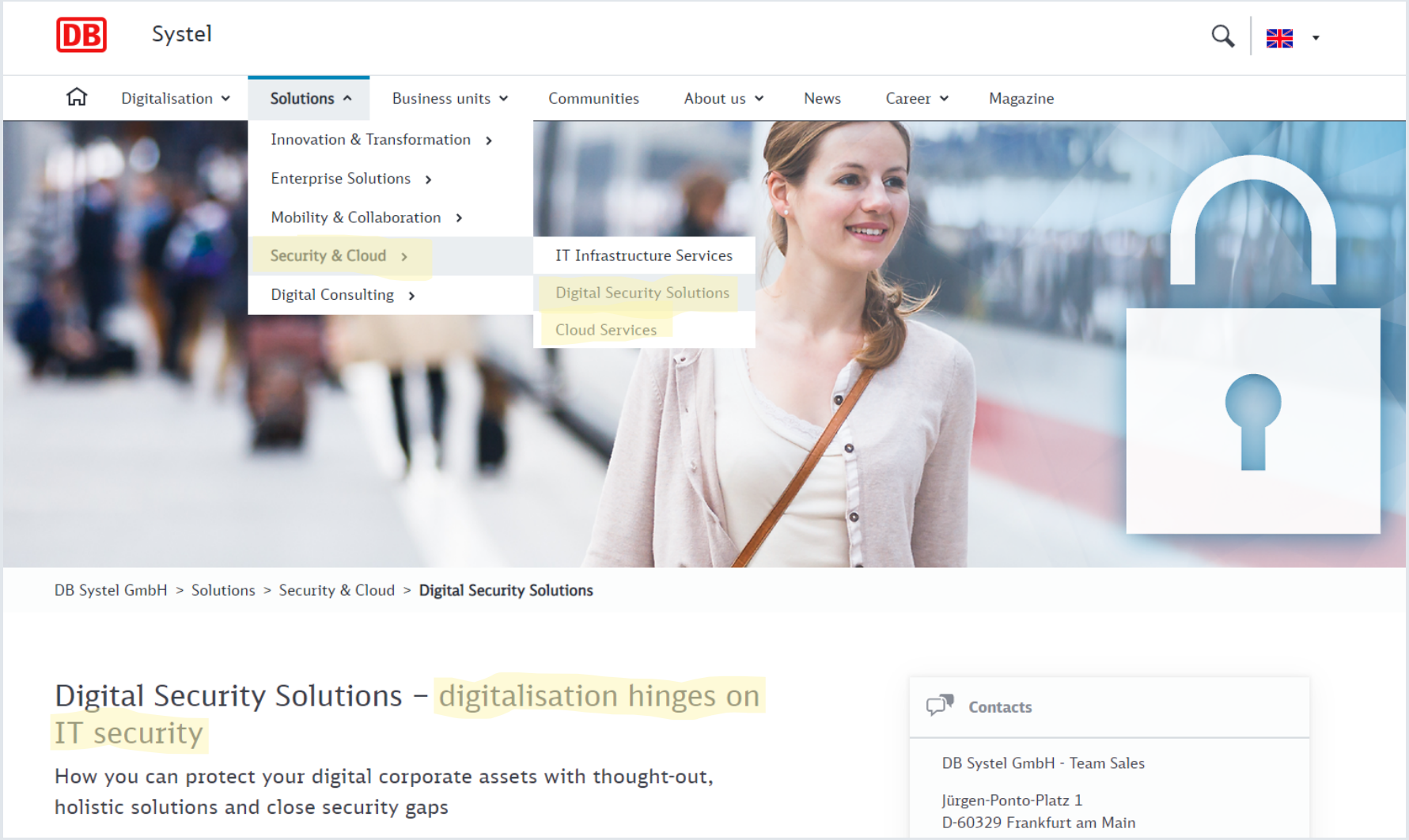
Agenda

1. Who we are and what we do
2. A CI/CD pipeline and its security challenges
3. Dynamic Application Security Testing (DAST)

Who is DB Systel?



Security & Cloud are currently two of our main topics



But the cloud is just a tool. You have to use it the right way.

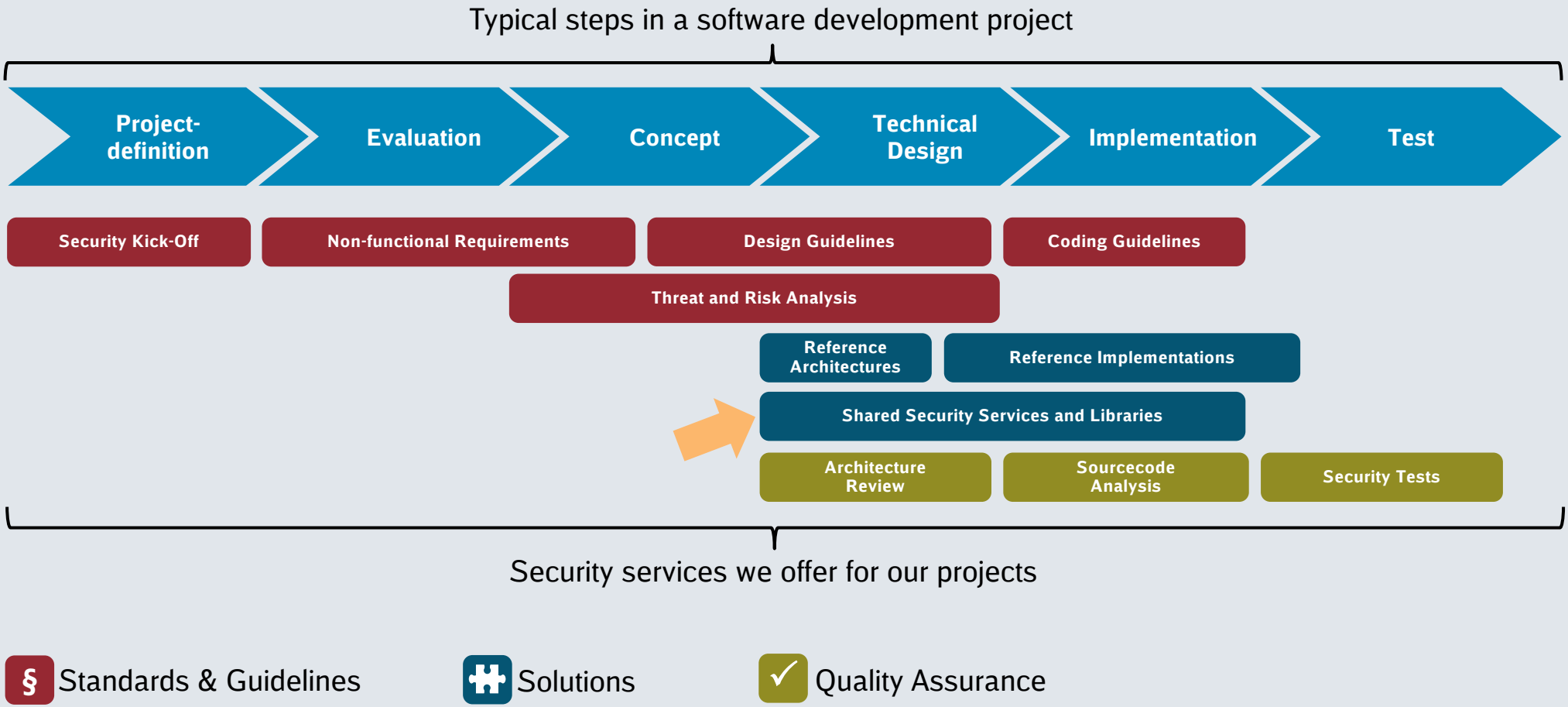
DB Systel on its way to the agile working world

The core elements of this dynamic development are modern, flexible IT structures and a new, active working environment thanks to agile, independent teams. At the moment, we are creating a major shift from traditional working and organisational structures to self-organisation and enterprise-wide networks. Our goal is to support our customers and their increasingly variable requirements in the best possible way with flexible ways of working and agile methods.

= DevOps teams using CI/CD pipelines

Who is Team Application Security Specialists @ DB Systel? What we do...

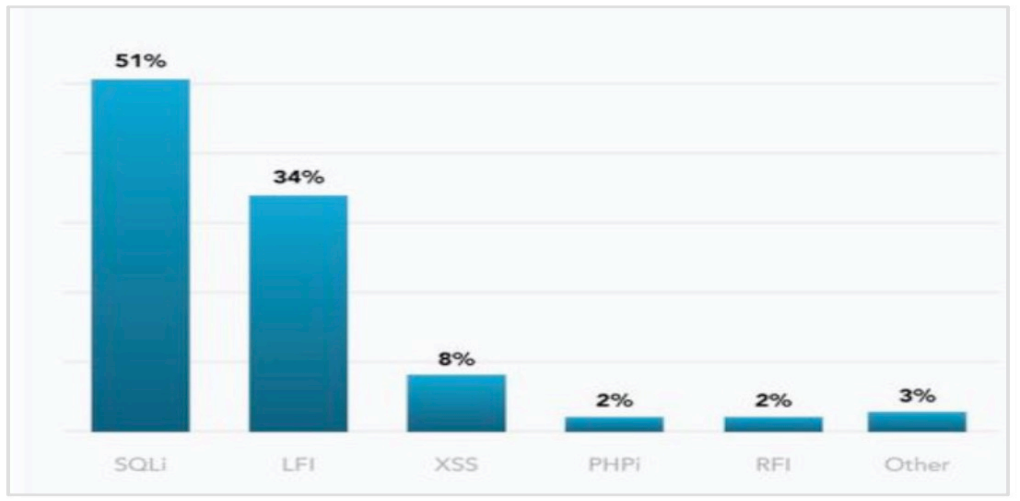
Young & agile team implementing a ***security development lifecycle*** at DB Systel



§ Standards & Guidelines Solutions ✓ Quality Assurance

…and why we do it:
Attacks on web applications are on top of the list

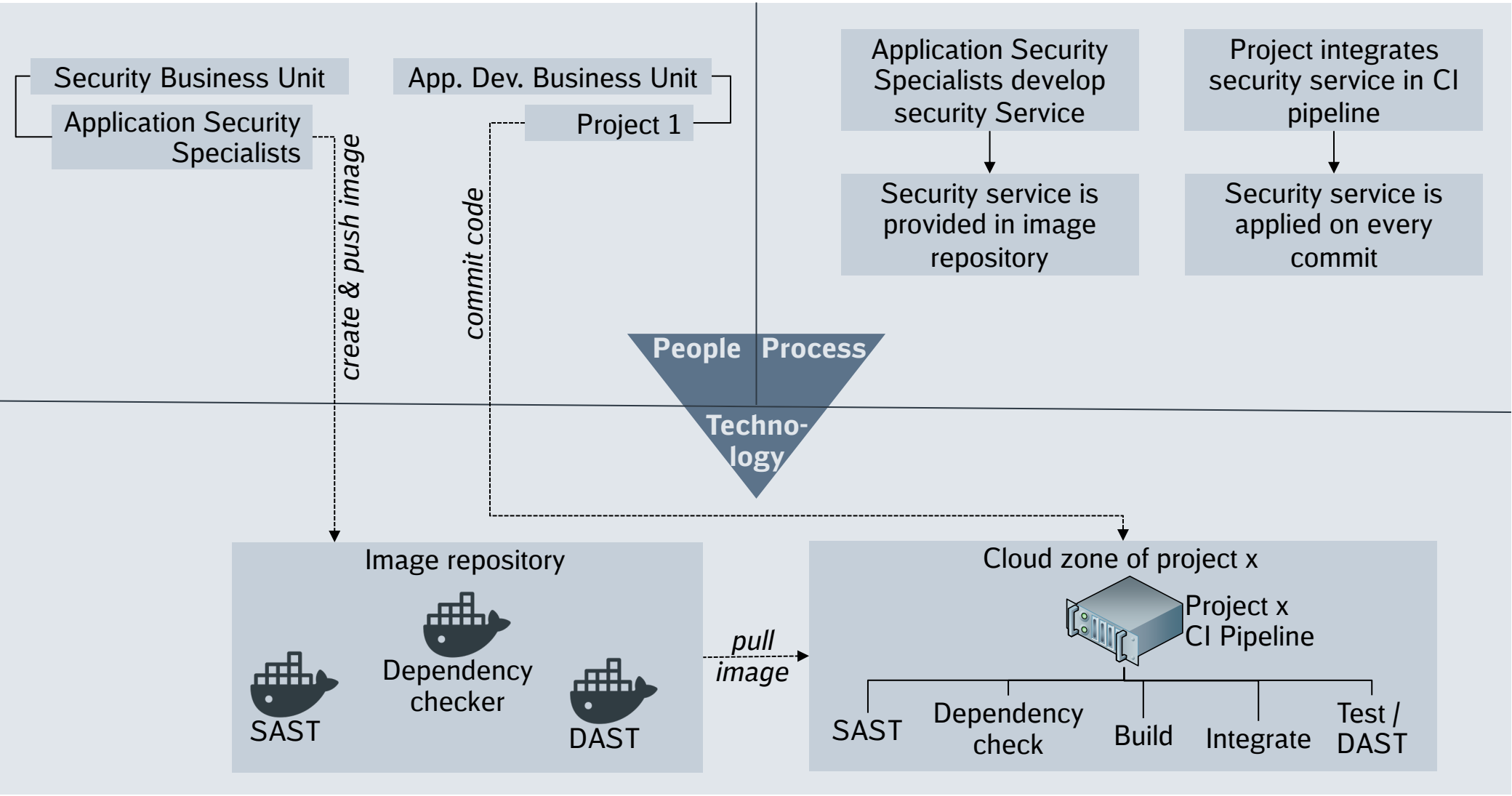
Top Threats 2018	Assessed Trends 2018	Change in ranking
1. Malware	→	→
2. Web Based Attacks	↑	→
3. Web Application Attacks	→	→
4. Phishing	↑	→
5. Denial of Service	↑	↑
6. Spam	→	↓
7. Botnets	↑	↑
8. Data Breaches	↑	↑
9. Insider Threat	↓	→
10. Physical manipulation/ damage/ theft/loss	→	→
11. Information Leakage	↑	↑
12. Identity Theft	↑	→
13. Cryptojacking	↑	NEW
14. Ransomware	↓	↓
15. Cyber Espionage	↓	→



SQLi: SQL Injection	Injecting database commands through e.g. unsanitized input fields on websites
LFI: Local File Inclusion	Accessing confidential server-side files like /etc/passwd through e.g. directory traversal
XSS: Cross-Site Scripting	Injecting script-code through e.g. unsanitized input fields on websites
RFI: Remote File Inclusion	Injecting remote files into the application through e.g. unsanitized input
PHPi: PHP Injection	Injecting PHP objects through unsanitized input to deserialization function

Source: ENISA Threat Landscape 2017

...and how we do it:
We provide easy to integrate services for our project teams

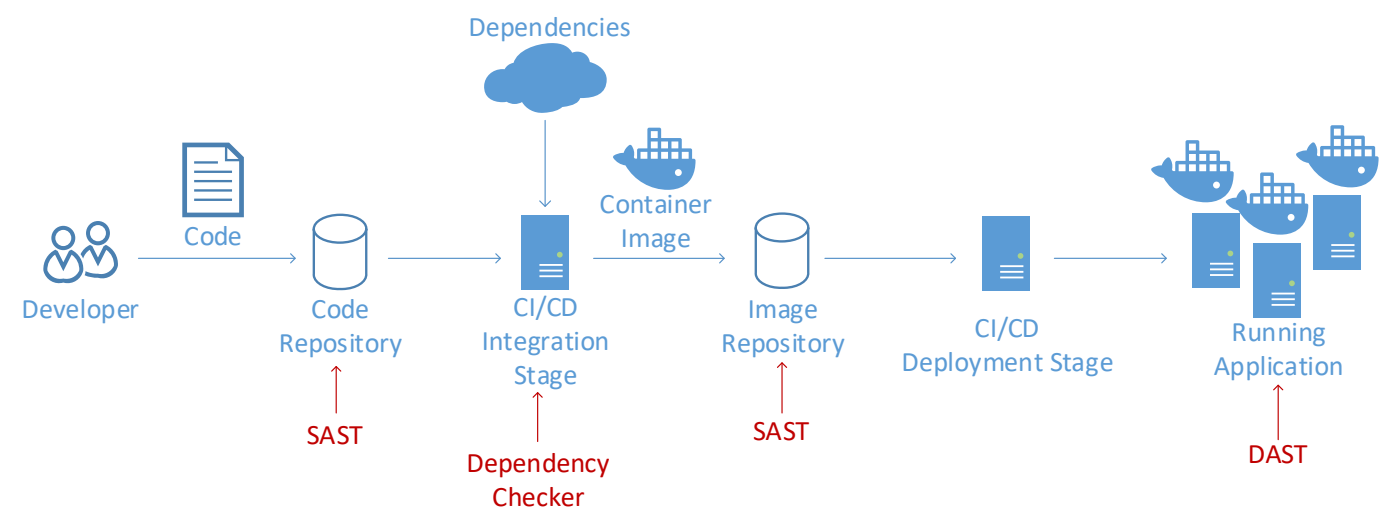


Agenda

1. Who we are and what we do
2. A CI/CD pipeline and its security challenges
3. Dynamic Application Security Testing (DAST)

A CI/CD pipeline and its security challenges

Stage	Security challenge	Security control
Code Repository	Vulnerable code (see previous slide)	Static Application Security Testing (SAST) on application level
Integration	Vulnerabilities in dependent libraries (same)	Dependency Checker
Image Repository	Vulnerable platform components (OS, Application Server, Database, ...)	Static Application Security Testing (SAST) on platform level
Running Application	False negatives from all above, vulnerable configurations, ...	Dynamic Application Security Testing (DAST)



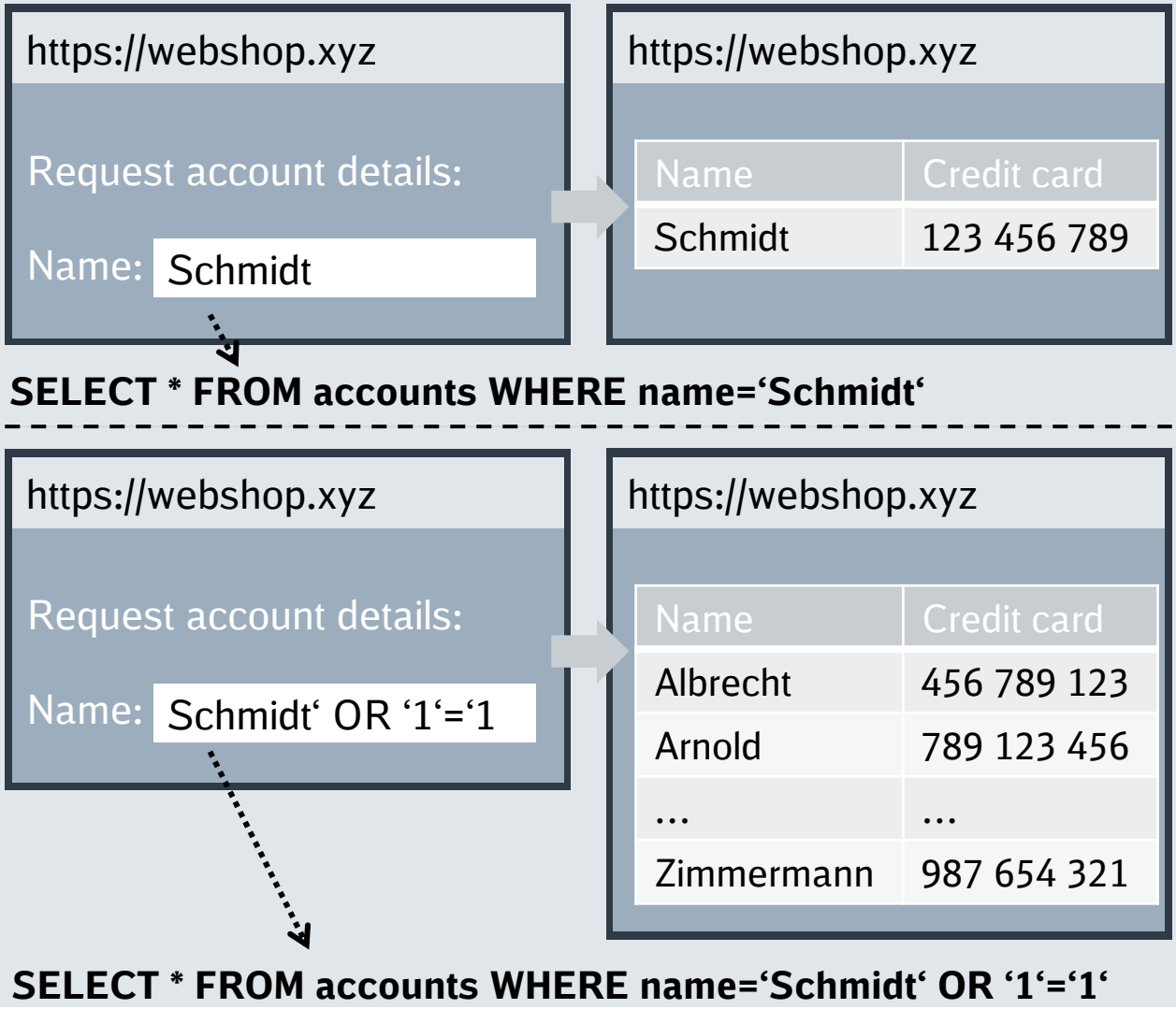
Agenda

1. Who we are and what we do
2. A CI/CD pipeline and its security challenges
3. Dynamic Application Security Testing (DAST)

Dynamic Application Security Testing (DAST)

A DAST Tool

- can perform automated security checks on a running application (i.e., in its dynamic context)
- uses well-known attack patterns for automated tests (e.g. SQL injection strings) in requests to the application
- compares the response of the application for different requests (attack patterns)
- deduces successful attacks from changes in the responses (e.g., output length)
- can also act as passive scanner for existing test sets, as intercepting proxy for manual pentesting, ...



The OWASP Zed Attack Proxy (ZAP)

OWASP Zed Attack Proxy Project

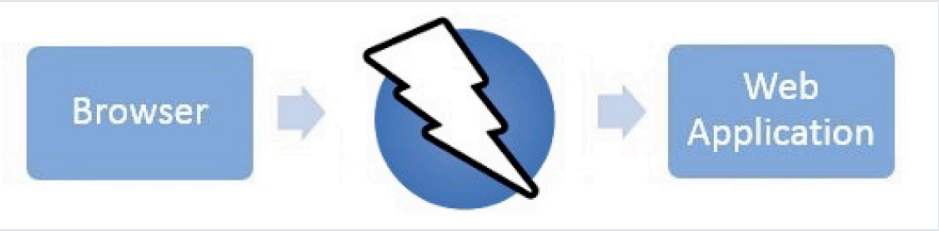
Main | Screenshots | Talks | News | ZAP Gear | Supporters

Some of ZAP's functionality:

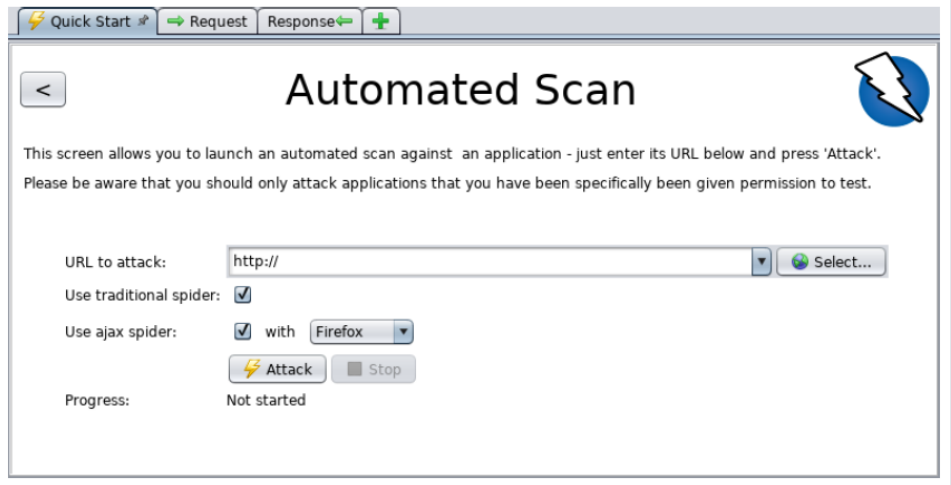
- Man-in-the-middle Proxy
- Traditional and AJAX spiders
- Automated scanner
- Passive scanner
- Forced browsing
- Fuzzer
- Dynamic SSL certificates
- Smartcard and Client Digital Certificates support
- Web sockets support
- Support for a wide range of scripting languages
- Plug-n-Hack support
- Authentication and session support
- Powerful REST based API
- Automatic updating option
- Integrated and growing marketplace of add-ons

Source: owasp.org

High level setup:
ZAP sits between browser and Web Application

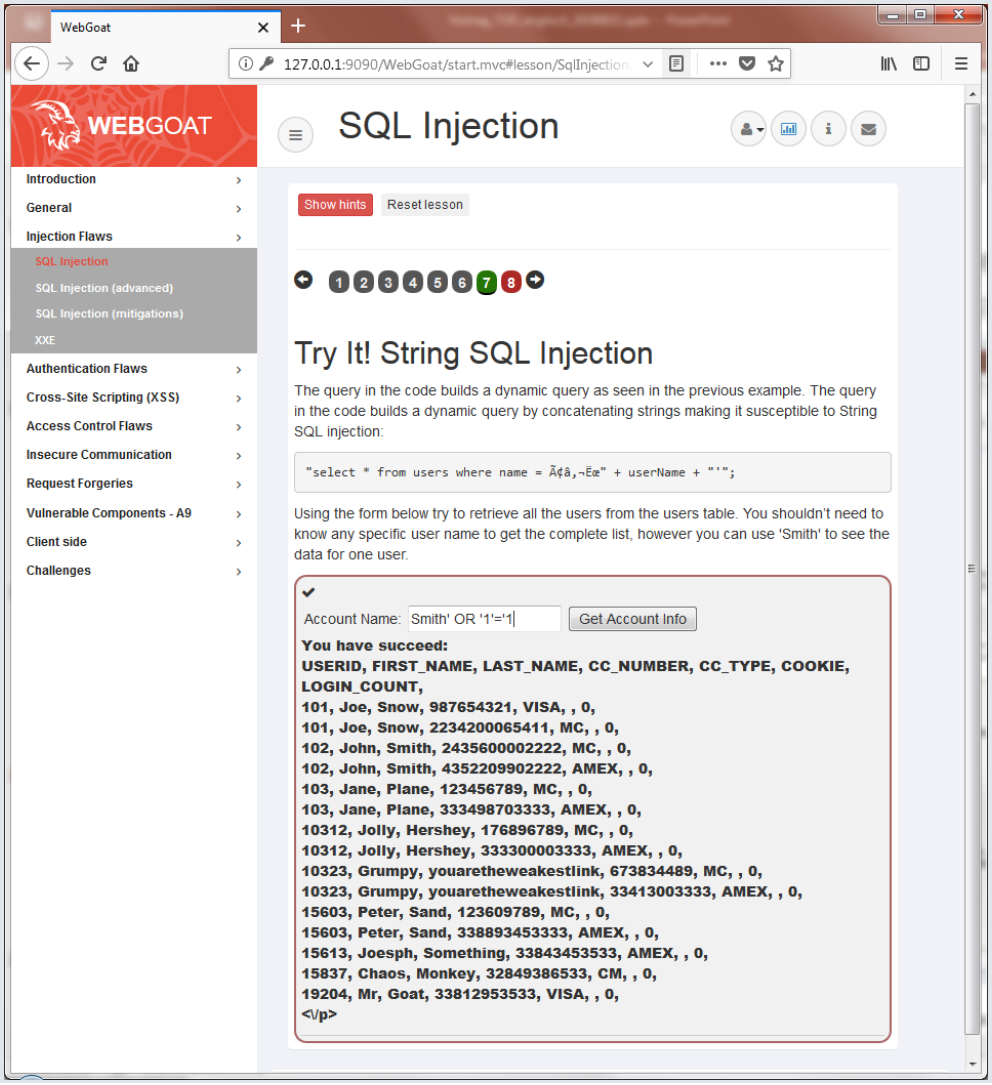


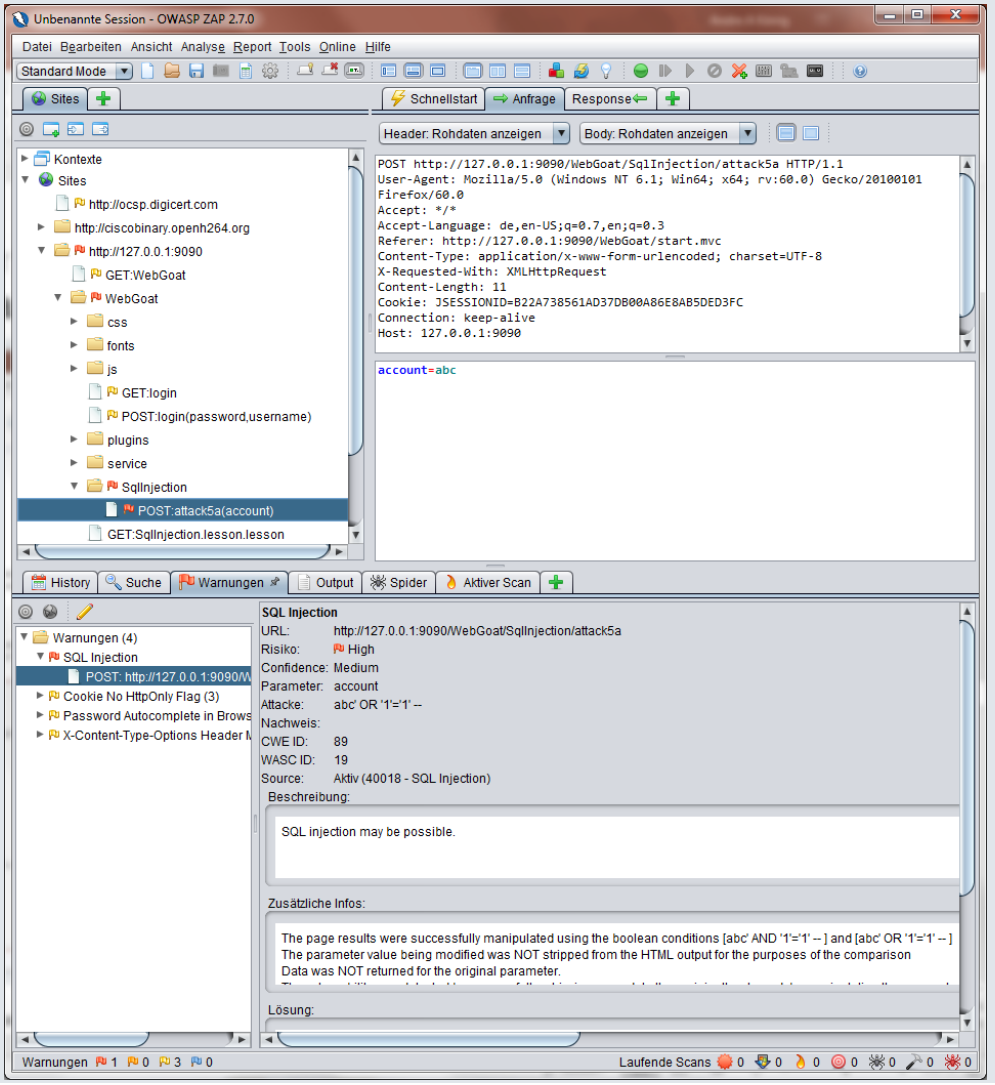
Automated scanning can be as easy as this:



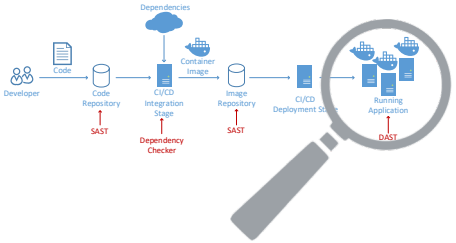
But: We typically need an authenticated context
But: ...

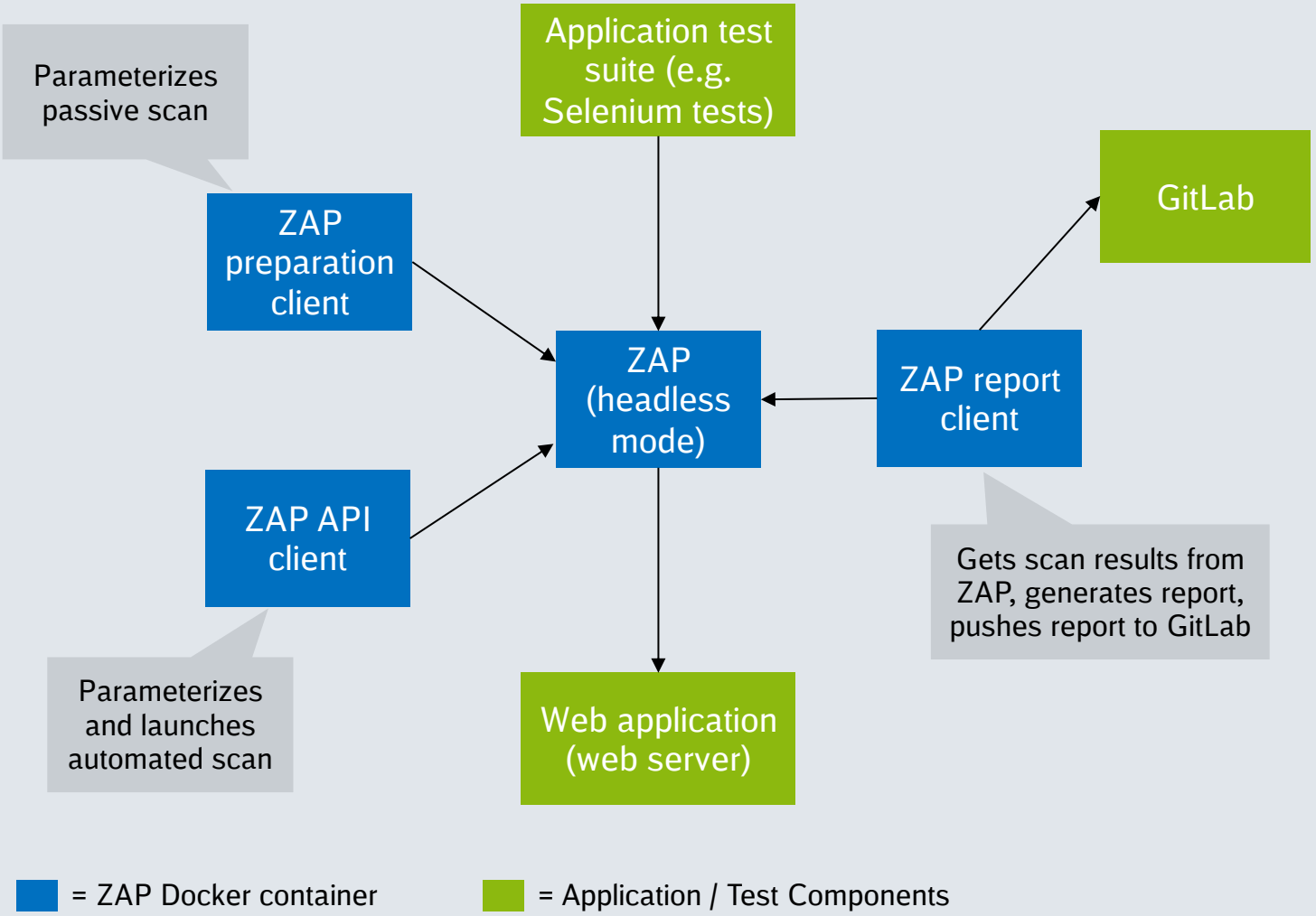
Performing an automated scan on Webgoat using ZAP





A deeper look on integrating ZAP in a CI/CD pipeline





DAST process:

- Perform passive scan to learn site tree
  - Parameterize passive scan with ZAP preparation client
  - Launch regular application test suite
- Perform automated scan
  - Parameterize and launch scan with ZAP API client
- Generate report with ZAP report client

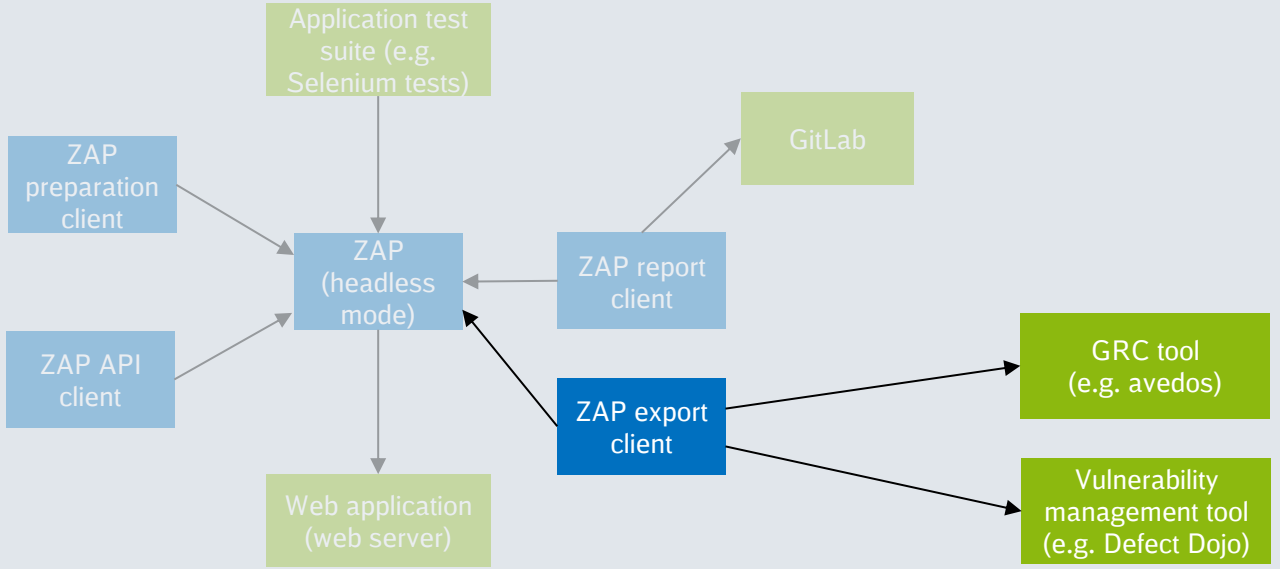
Outlook: Integration of DAST results in development and governance processes and tools

Performing security tests and pushing reports to Git does not fix the problems

- Development teams have to work with the results and must close vulnerabilities
- Security governance must manage risks and initiate measures on enterprise scale

We, thus, are building interfaces to integrate DAST results in

- Vulnerability management tools of development teams
- Risk / compliance management tools of CISO teams



Thank you for your attention

Please feel free to contact us:
sebastian.biehler@deutschebahn.com
andre.a.koenig@deutschebahn.com
tommaso.nuccio@deutschebahn.com
app.sec.specialists@deutschebahn.com