



APPSEC
EUROPE

Building secure cloud-native apps with spring boot & security

Andreas Falk

About me



Andreas Falk / Germany
NovaTec Consulting GmbH
andreas.falk@novatec-gmbh.de

 **@Agile_Security**

Agile Web DevOps TDD SSO
Threat Modeling Spring Java OAuth2
Architecture Java EE
Clean Code Cloud BDD Scrum OWASP
Security IoT SAML
Code Review
Static Analysis Kanban Microservices



Developers vs. Security



Developers vs. Security

Java8

IoT

Microservices

DevOps

Cloud



NoSQL

Single Page Apps

BigData

Testing

Cross-Functional

Agile

Security?



Secure Web Application in 5 minutes

SPRING INITIALIZR bootstrap your application now

Generate a with Spring Boot

Project Metadata

Artifact coordinates

Group

com.example

Artifact

demo

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Web Security, JP... Actuator, ...vtools

Selected dependencies

Generate Project alt + ↵

Don't know what to look for? Want more options? [Switch to the full version.](#)

Live Coding Demo



Cloud Native



Cloud Native

Culture

DevOps

Process

Continuous Delivery

Architecture

Microservices

Technology

Containers



JP Morgan Chase's Cloud Native MM

Level 3

Cloud Native

12 Factor App

One Code Base

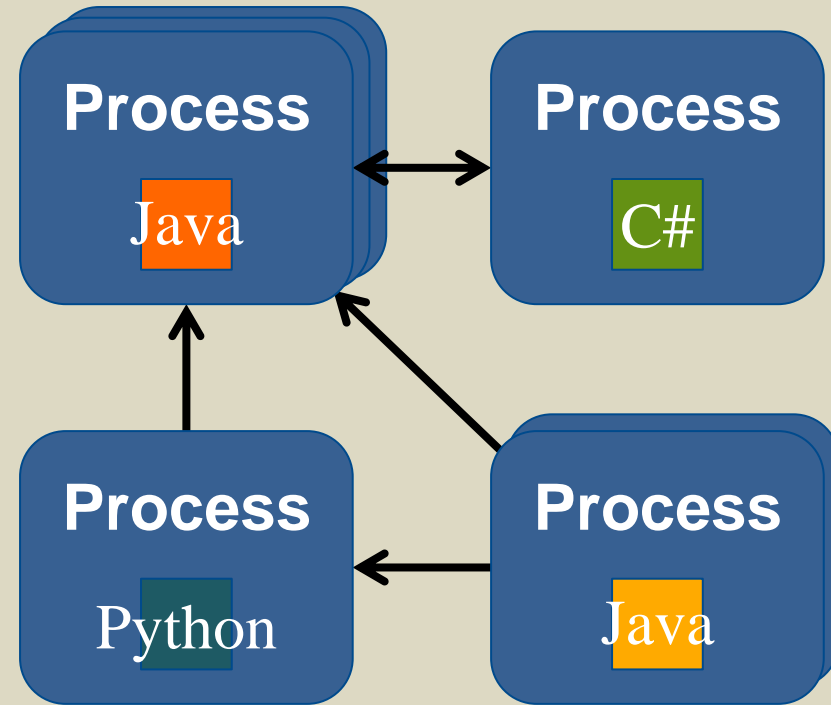
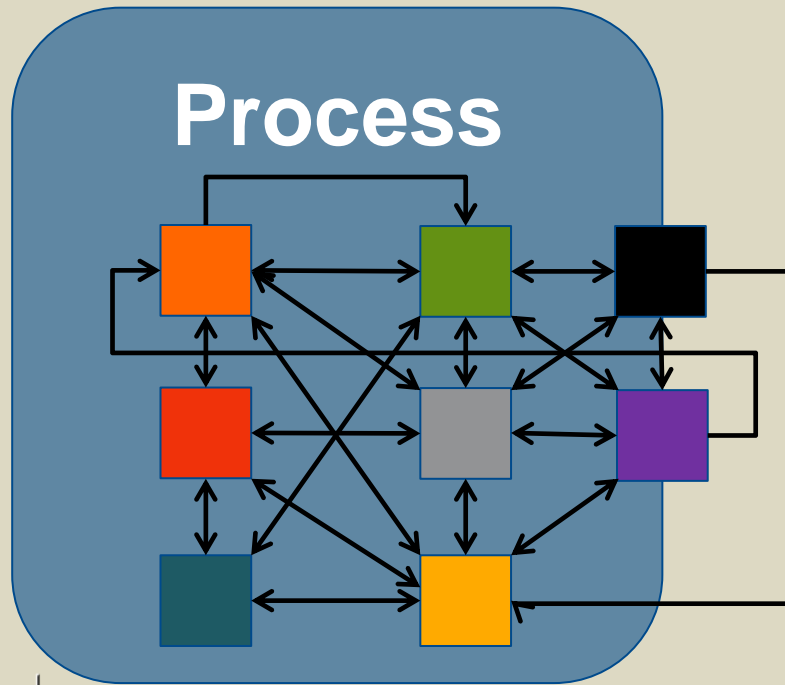
Externalize Configuration

...

<http://12factor.net>



From Monolith To Microservices



Microservice = Spring Boot

Standalone Spring Apps

Auto Configuration

Embedded Servlet Container

„Make JAR Not WAR“

Production-Ready Features

```

      .
      /\ /  _\   /  _\   /  _\   /  _\   /  _\   /  _\   /  _\   /  _\   /  _\   /  _\
     ( )\  /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\
    \  \ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\
     '  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\ /  _\
     =====|=====|=====|=====|=====|=====|=====|=====|=====|=====|
    :: Spring Boot ::                (v1.4.0.M3)

Registering beans for JMX exposure on startup
Tomcat started on port(s): 8080 (http)
Started SpringBootTestApplication in 13.081 seconds

```

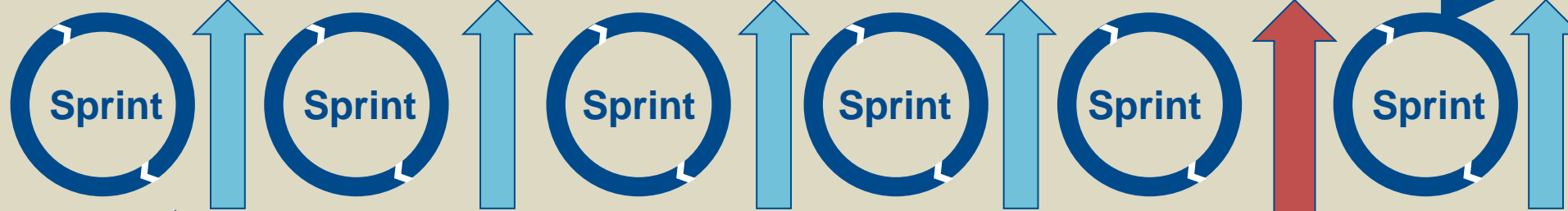


Secure Cloud Native



Secure Continuous Delivery ?

Attacks (24x7)



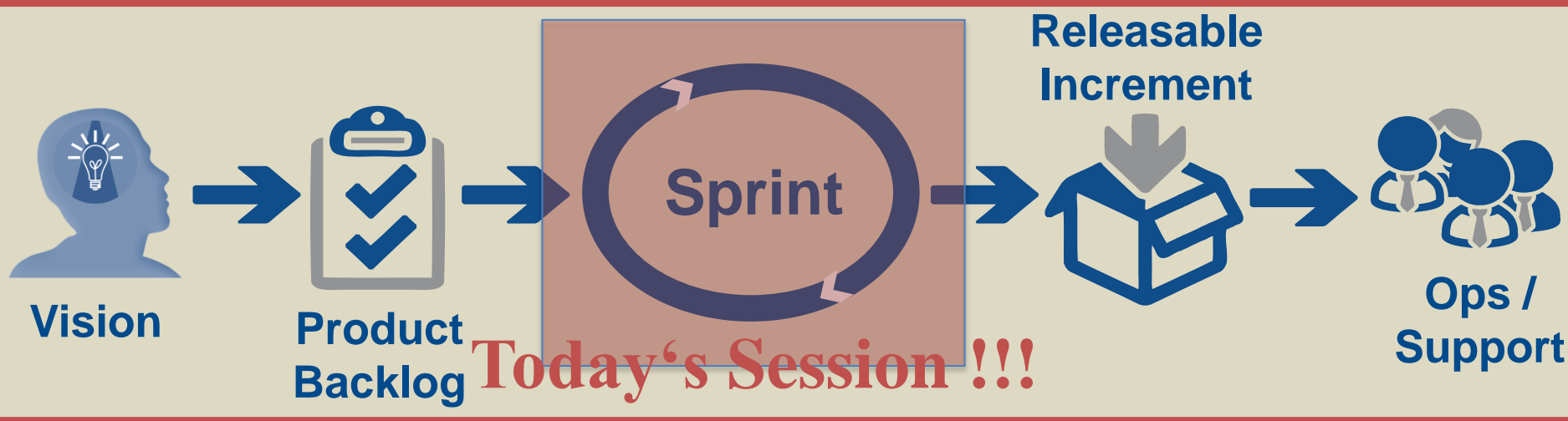
Deployments

Penetration Test



Agile Security / SecDevOps

Continuous Delivery



+ Security

Building secure cloud-native apps with spring boot & security



Secure Cloud-Native Applications

$$\begin{array}{ccc} \text{Secure} & & \text{Spring Security} \\ + & = & + \\ \text{Cloud-Native} & & \text{Spring Boot} \end{array}$$



Spring Security

„Secure By Default“ Configuration

Authentication / Authorization

Secure Password Encoding

Testing Support



„Secure By Default“ Configuration

Require Authentication for all URLs: On

Session Fixation Protection: On

Session Cookie (HttpOnly, Secure): On

CSRF Attack Protection: On

Security Response Headers: On



Security Response Headers



Cache Control



X-Content-Type-Options



X-Frame-Options



X-XSS-Protection



HTTP Strict Transport Security (SSL)



Secure Password Encoding

Encoder Implementations

BCryptPasswordEncoder

SCryptPasswordEncoder

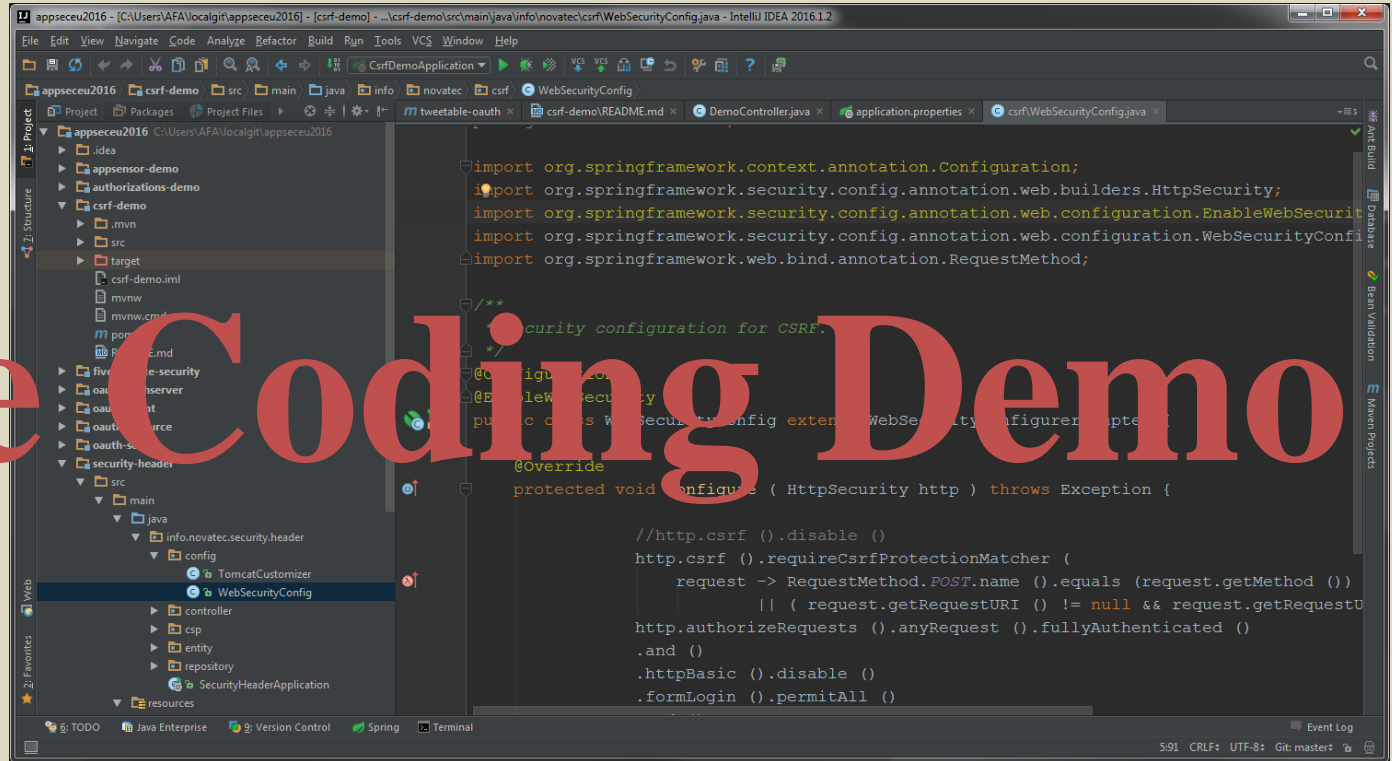
Pbkdf2PasswordEncoder

BytesEncryptor (implementation for BouncyCastle)



„Secure By Default“ Conventions

Live Coding Demo



```
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.web.bind.annotation.RequestMethod;

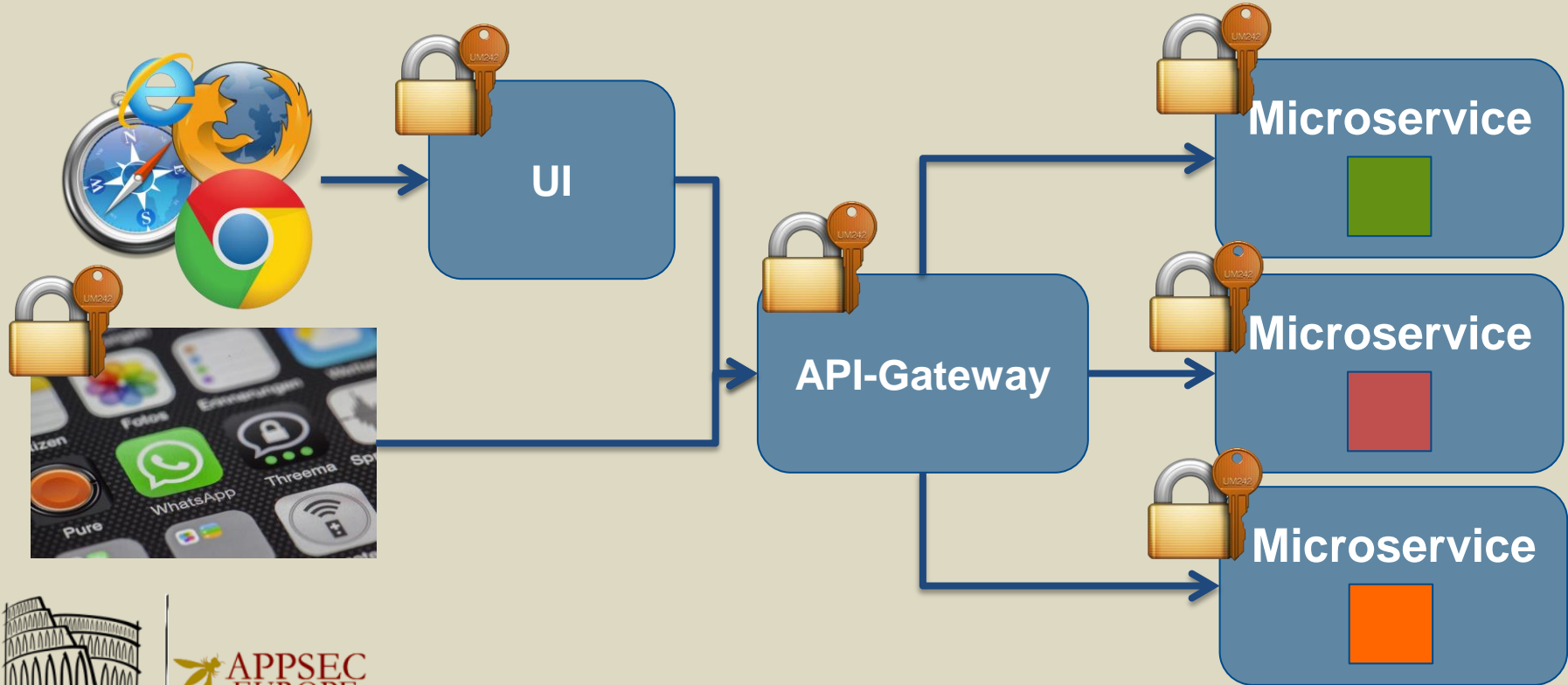
/**
 * Security configuration for CSRF.
 */
@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {

    @Override
    protected void configure ( HttpSecurity http ) throws Exception {

        //http.csrf ().disable ()
        http.csrf ().requireCsrfProtectionMatcher (
            request -> RequestMethod.POST.name ().equals (request.getMethod ())
                || ( request.getRequestURI () != null && request.getRequestURI ().contains ("/api") )
        )
        http.authorizeRequests ().anyRequest ().fullyAuthenticated ()
            .and ()
            .httpBasic ().disable ()
            .formLogin ().permitAll ()
    }
}
```



Secure Cloud Architectures

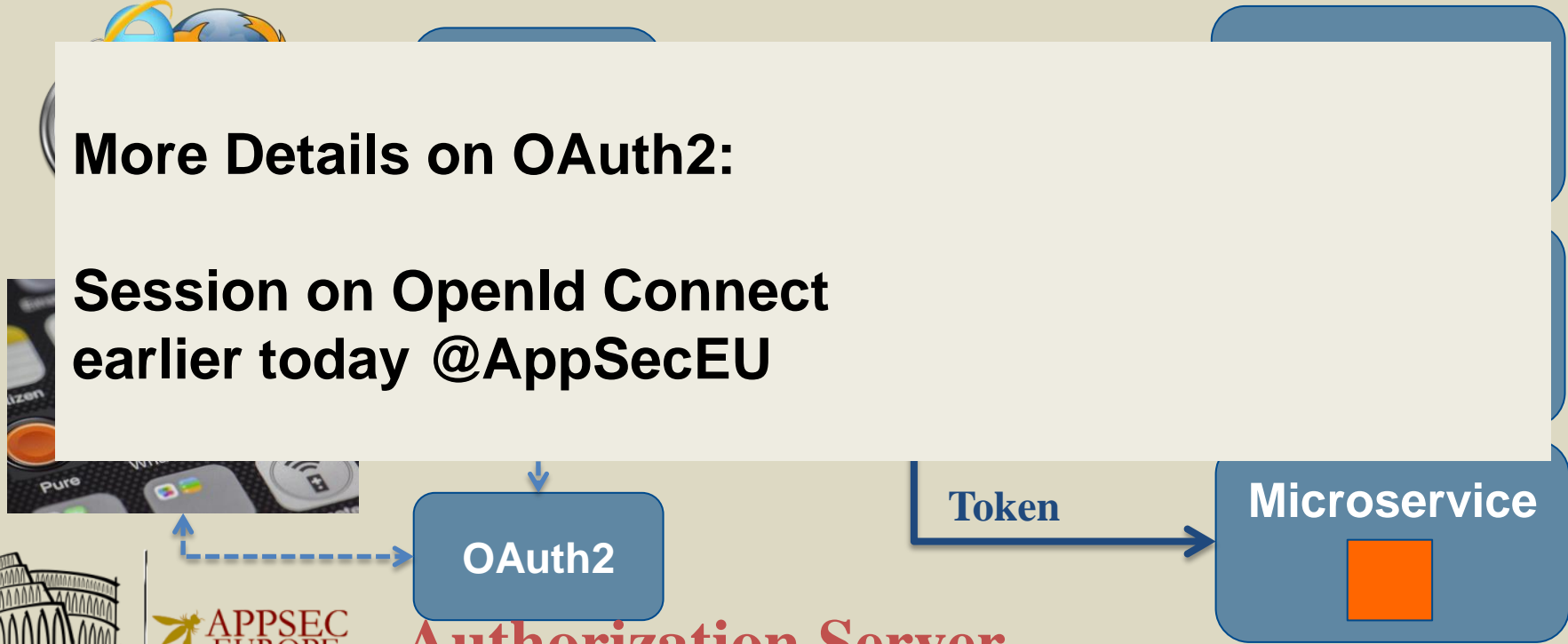


Secure Cloud with OAuth2

Resource Server

More Details on OAuth2:

Session on OpenId Connect
earlier today @AppSecEU



Authorization Server

Building secure cloud-native apps with spring boot & security



Tweetable OAuth2 Application



 **Dave Syer**
@david_syer

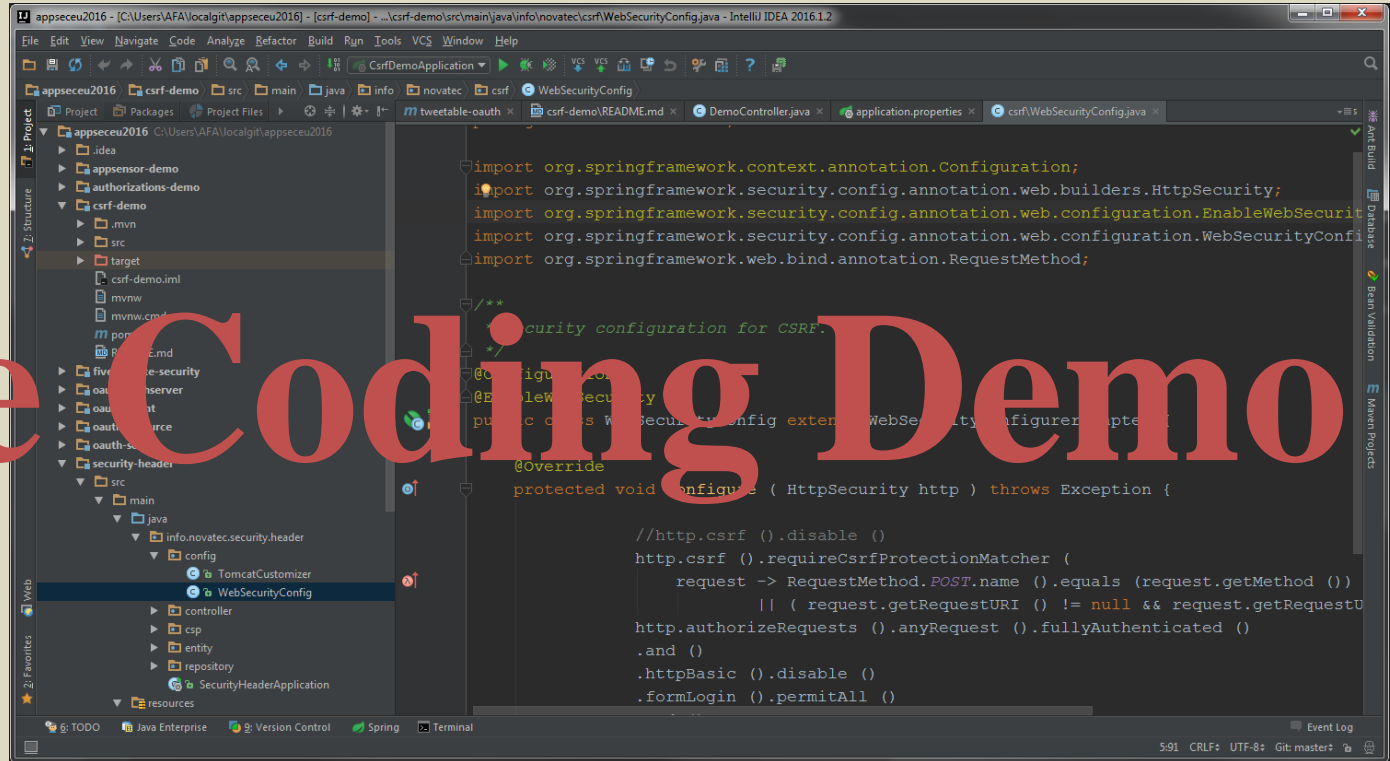
 [Folge ich](#)

@andifalk @springboot tweetable secure #oauth2 app:
@EnableAuthorizationServer
@EnableResourceServer
class AuthServer {}



Secure Microservices With OAuth2

Live Coding Demo



The screenshot shows an IDE window with the following code in `WebSecurityConfig.java`:

```
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.web.bind.annotation.RequestMethod;

/**
 * Security configuration for CSRF.
 */
@Configuration
@EnableWebSecurity
@ExtendWith({CsrfDemoApplication.class})
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {

    @Override
    protected void configure ( HttpSecurity http ) throws Exception {

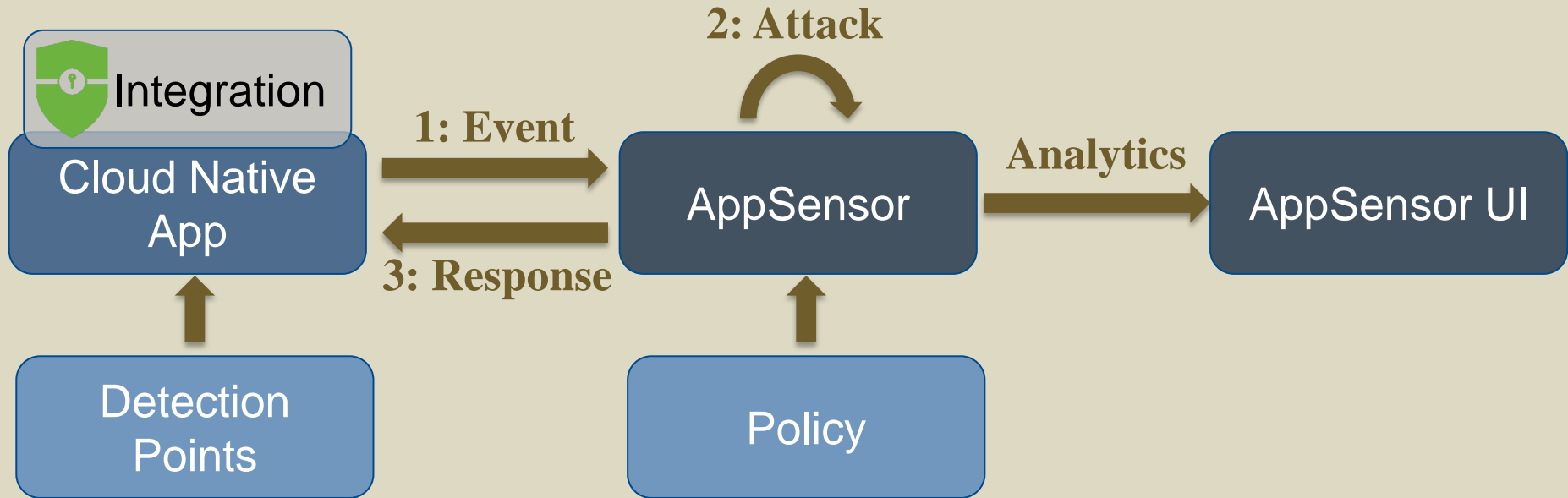
        //http.csrf ().disable ()
        http.csrf ().requireCsrfProtectionMatcher (
            request -> RequestMethod.POST.name ().equals (request.getMethod ())
                || ( request.getRequestURI () != null && request.getRequestURI ().contains ("/api") )
        )
        http.authorizeRequests ().anyRequest ().fullyAuthenticated ()
            .and ()
            .httpBasic ().disable ()
            .formLogin ().permitAll ()
    }
}
```



Runtime Application Self-Protection



RASP With AppSensor



<http://appsensor.org>

<https://github.com/jtmelton/appsensor>

AppSensor UI

Most Active Users

Last Refresh: 2016-06-28T08:26:57+02:00

Activity Log (most recent)

Type	Category	From	To	Timestamp
Event	AE4 (Authentication)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-06-28T07:29:51.263Z
Event	RE3 (Request)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-06-28T07:29:49.970Z
Event	IE2 (Input Validation)	frank (10.10.10.1) (37.596758 / -121.647992)	myclientgeoapp3 (10.10.10.7) (59.164625 / 123.96234)	2016-06-28T07:29:49.732Z
Event	IE1 (Input Validation)	stephen (10.10.10.3) (29.66889 / -8.576706)	myclientgeoapp3 (10.10.10.7) (59.164625 / 123.96234)	2016-06-28T07:29:49.620Z
Event	IE1 (Input Validation)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-06-28T07:29:48.397Z
Attack	IE1 (Input Validation)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-06-28T07:29:48.397Z
Response	logout	myclientapp (no IP Address) (no geo)	bob (10.10.10.1) (37.596758 / -121.647992)	undefined

Input Validation: 0
Request: 0

0 ● ——— 08:00

2016/06/28, 09:08:24)

disableComponentForSpecificUser (33 minutes)
(bob) (client app: "myclientapp") (started at:
"2016/06/28, 09:08:55")

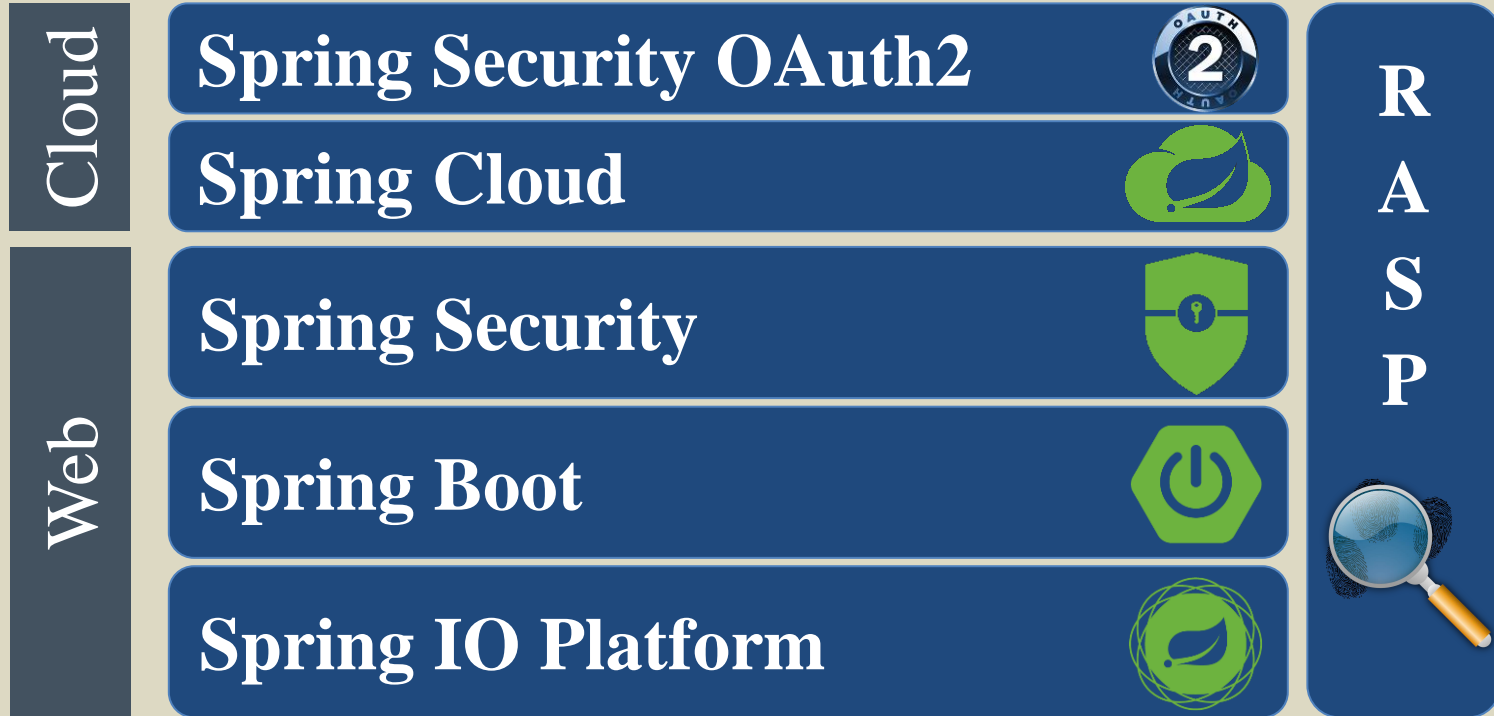
2016/06/28, 09:09:24)

ableComponentForSpecificUser (33 minutes)
(bob) (client app: "myclientapp") (started at:
2016/06/28, 09:08:55")

ableComponentForSpecificUser (33 minutes)
(bob) (client app: "myclientapp") (started at:
2016/06/28, 09:09:52")



Wrap Up: Secure Cloud-Native Apps



Wrap Up: Secure Cloud-Native Apps



„Secure By Default“ Conventions !!

„Secure By Default“ Developer API's !!



Questions?

Andreas Falk
NovaTec Consulting GmbH
andreas.falk@novatec-gmbh.de

 **@Agile_Security**

<https://github.com/andifalk/appseceu2016>

<http://projects.spring.io/spring-security>

<http://projects.spring.io/spring-security-oauth>

