
# Cloud-Native Security

https://github.com/andifalk/cloud-security-workshop

Andreas Falk

# Vorstellung

## Andreas Falk
Novatec Consulting

andreas.falk@novatec-gmbh.de / @andifalk

https://www.novatec-gmbh.de/beratung/agile-security

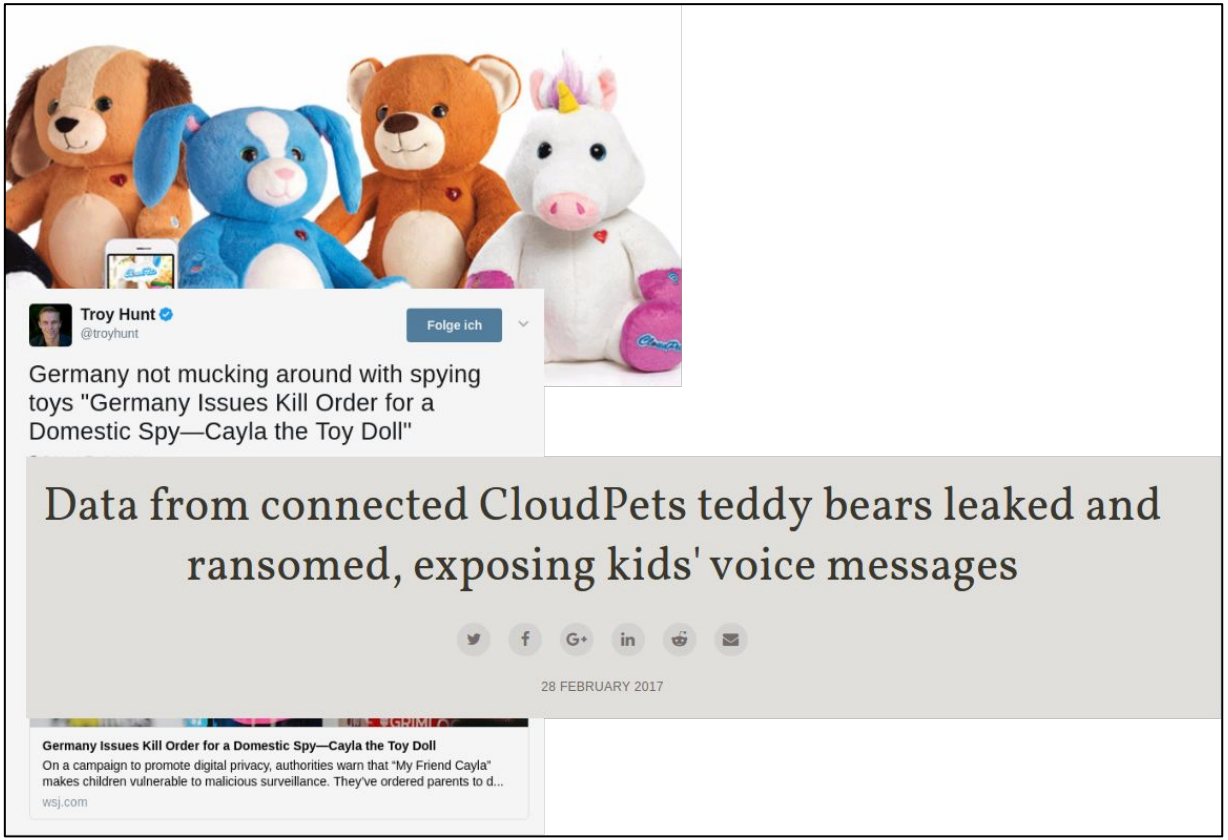
# Agenda

1. Intro:
   - Why Security?
   - What's Cloud-Native Security?
2. OWASP & OWASP Top 10
3. Client Security
4. Server Security
5. Security in the Cloud
6. Practice

# Why Security ?

#securityfails

# Cayla Doll - The Spy Toy






NOVATEC

# Ashley Madison - Serious Life Changer




NOVATEC

# Facebook Leak - Big Data Exposure

## Over 540 million Facebook records found on exposed AWS servers

Leak originated at two third-party companies that had collected Facebook data on their own servers.

By Catalin Cimpanu for Zero Day | April 3, 2019 -- 18:32 GMT (19:32 BST) | Topic: Security

**MORE FROM CATALIN CIMPANU**

Windows 10
**Microsoft changes how Windows 10 disconnects USB storage devices**

Developer
**After Chrome, Firefox will also support off-screen image lazy loading**

Security
**FBI criticized for delaying breach notifications, including insufficient details**

Security
**IoT botnet targeting your enterprise? Nope. Just a kid with an ExploitDB account**

**NEWSLETTERS**

**ZDNet Security**

Your weekly update on security around the globe, featuring research, threats, and more.



NOVATEC

# EasyJet Hacked For Four Months, Data On 9 Million Customers And 2,000 Credit Cards Stolen

**Thomas Brewster** Forbes Staff

Cybersecurity

*Associate editor at Forbes, covering cybercrime, privacy, security and surveillance.*




NOVATEC

# Crypto Mining Via K8s Dashboard

Source: https://blog.heptio.com

## On Securing the Kubernetes Dashboard

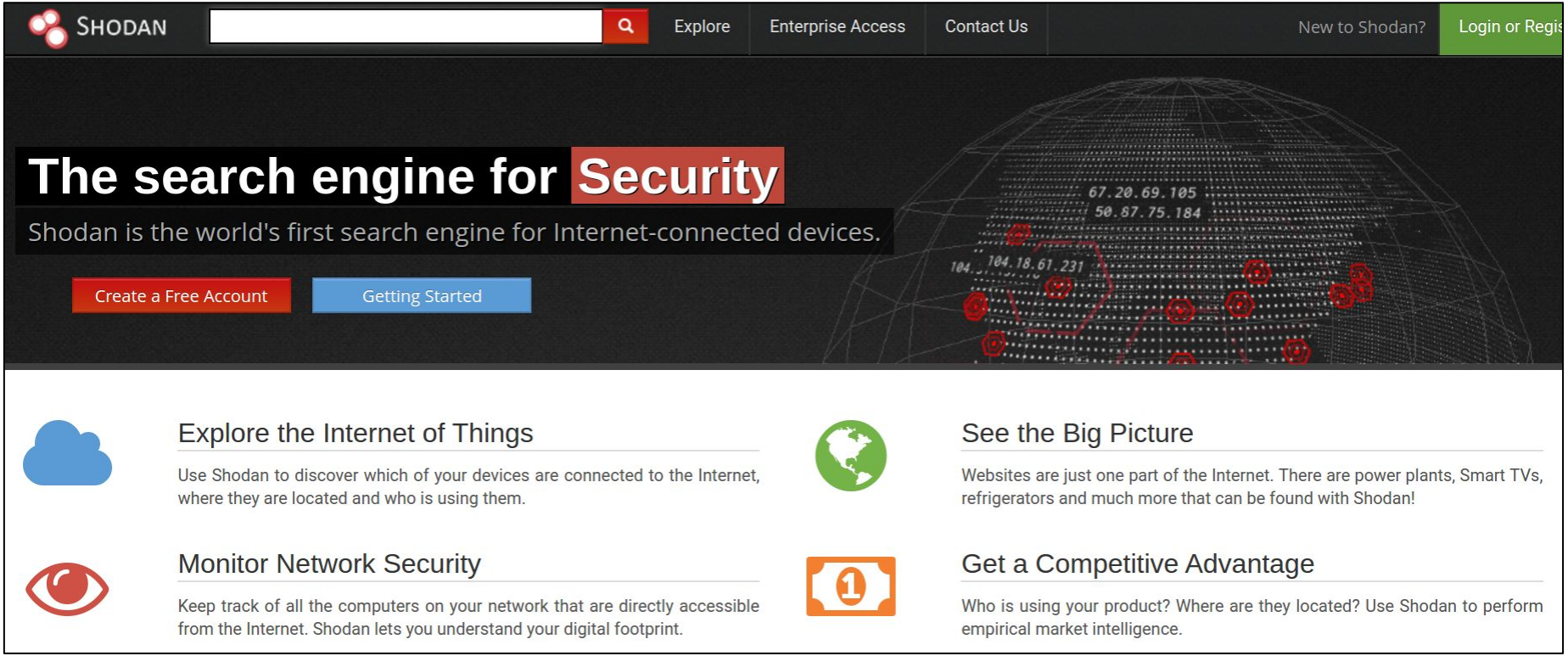
Joe Beda Follow

Feb 28, 2018 · 13 min read

Recently Tesla (the car company) was alerted, by security firm RedLock, that their Kubernetes infrastructure was compromised. The attackers were using Tesla's infrastructure resources to mine cryptocurrency. This type of attack has been called "cryptojacking".

The vector of attack in this case was a Kubernetes Dashboard that was exposed to the general internet with no authentication and elevated privileges. Not only this, but core AWS API keys and secrets were visible. How do you prevent this from happening to you?

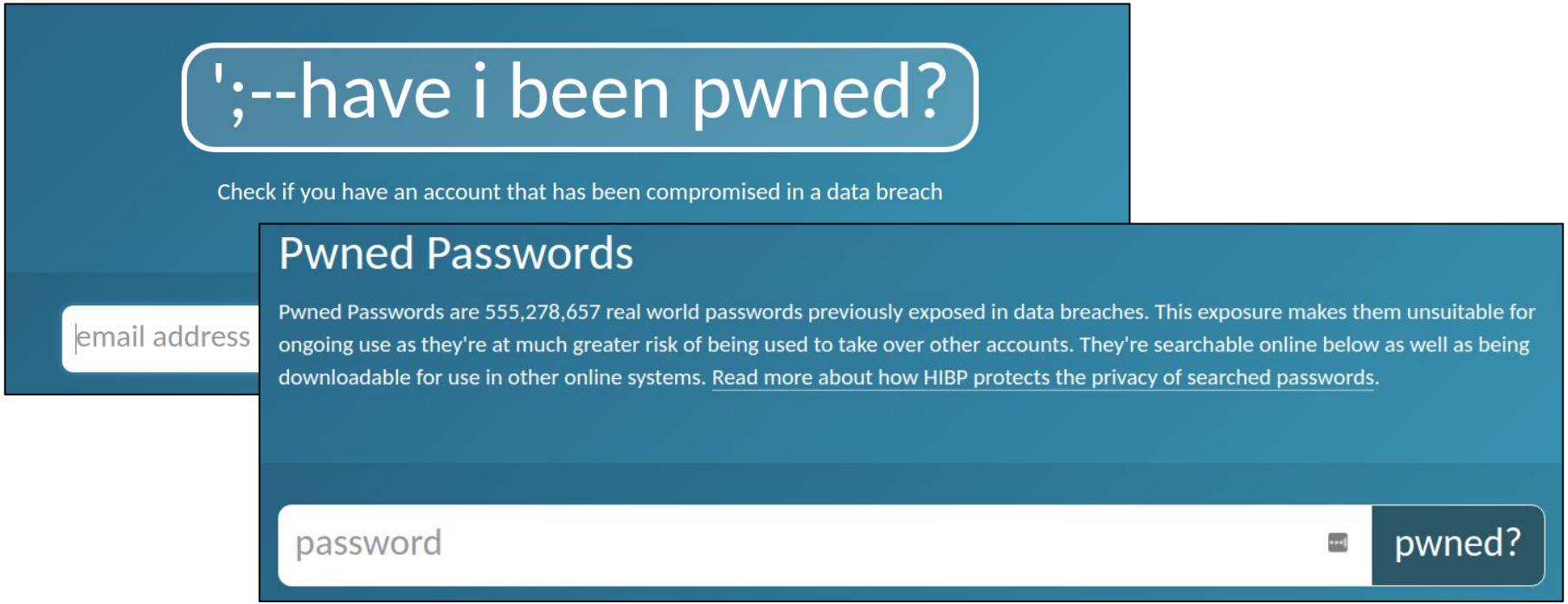
# Shodan.io - Google for “Hackers”






NOVATEC

# haveibeenpwned.com - Exposed Accounts & Passwords






NOVATEC

# Pay attention on what you copy from stackoverflow.com

This is some what simple

```
string inp = "hai";
StringBuilder strb = new StringBuilder();
foreach (char s in inp)
{
    int sin = s + 5;
    char newch = (char)sin;
    strb.Append(newch);
}
string output = strb.ToString();
```

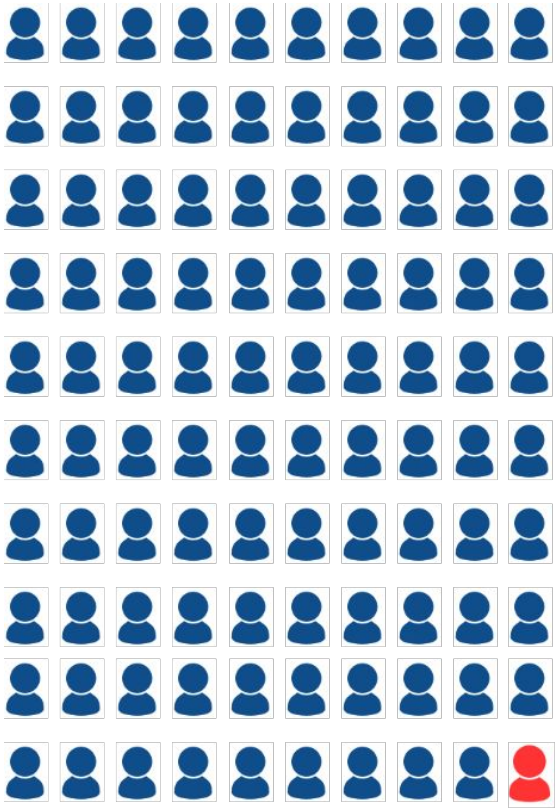
Now the output contains the encrypted string "mfn" (ie., 5 letters away from the original )in it....

# Security is a really comprehensive topic



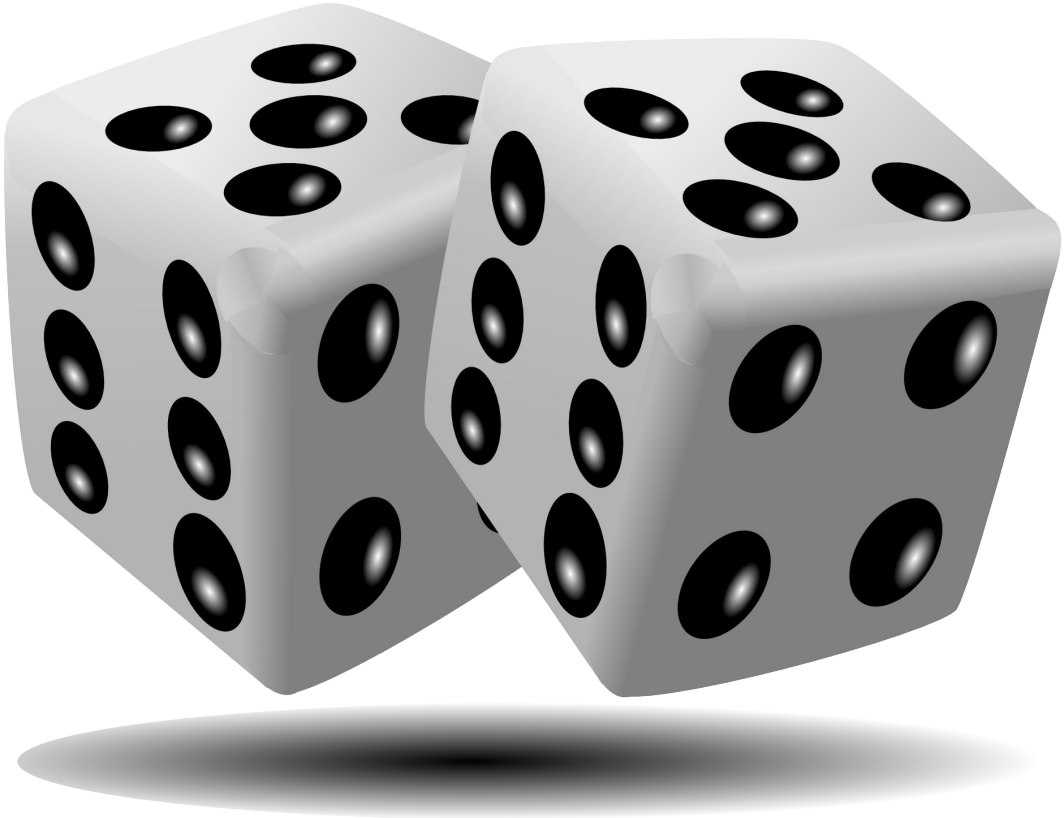

NOVATEC

# 1 Security-Professional for 100 Developers



Source:
https://www.sonatype.com/devops-survey-report

# Security as a game of chance?





NOVATEC

# Open ~~Web~~ Application Security Project

https://www.owasp.org

# OWASP Top 10 - 2017


| OWASP Top 10 - 2013 | | OWASP Top 10 - 2017 |
|---|---|---|
| A1 – Injection | → | A1:2017-Injection |
| A2 – Broken Authentication and Session Management | → | A2:2017-Broken Authentication |
| A3 – Cross-Site Scripting (XSS) | ↘ | A3:2017-Sensitive Data Exposure |
| A4 – Insecure Direct Object References [Merged+A7] | U | A4:2017-XML External Entities (XXE) [NEW] |
| A5 – Security Misconfiguration | ↘ | A5:2017-Broken Access Control [Merged] |
| A6 – Sensitive Data Exposure | ↗ | A6:2017-Security Misconfiguration |
| A7 – Missing Function Level Access Contr [Merged+A4] | U | A7:2017-Cross-Site Scripting (XSS) |
| A8 – Cross-Site Request Forgery (CSRF) | ☒ | A8:2017-Insecure Deserialization [NEW, Community] |
| A9 – Using Components with Known Vulnerabilities | → | A9:2017-Using Components with Known Vulnerabilities |
| A10 – Unvalidated Redirects and Forwards | ☒ | A10:2017-Insufficient Logging&Monitoring [NEW,Comm.] |

https://github.com/OWASP/Top10

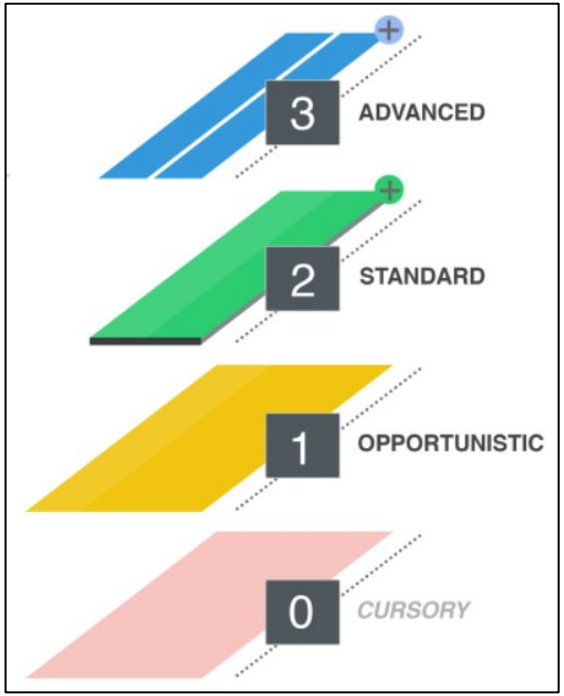


NOVATEC

# Other OWASP Offerings

https://www.owasp.org/index.php/OWASP_Proactive_Controls

## Application Security Verification Standard




https://github.com/OWASP/ASVS



NOVATEC

# OWASP Zap - Open Source Penetration Testing Tool


https://www.zaproxy.org/

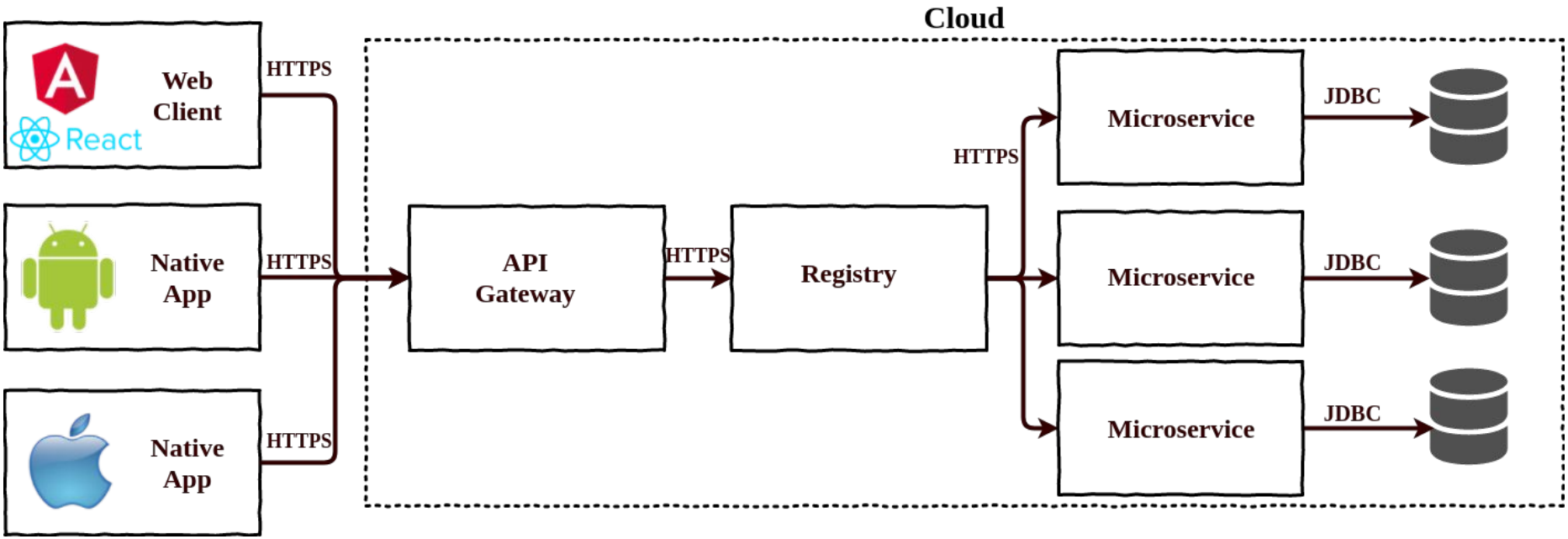


NOVATEC

# Cloud-Native Security

Cloud and Platforms (CloudFoundry / Kubernetes)

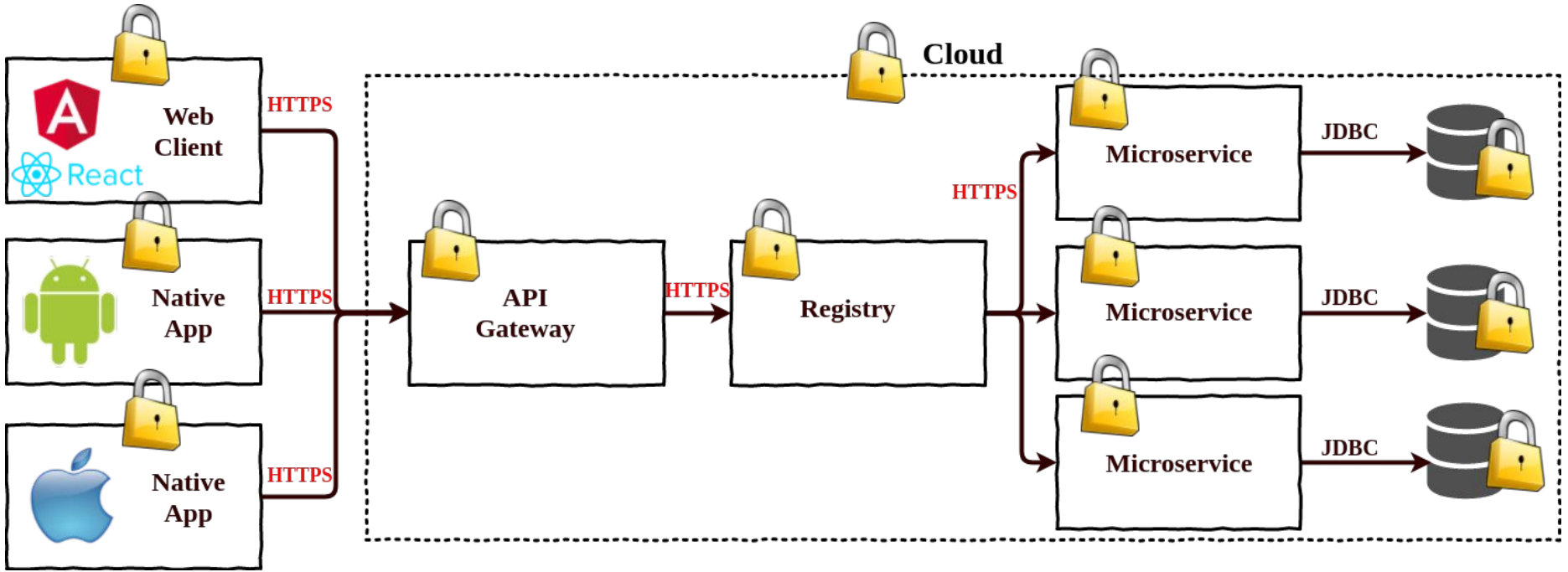
# Cloud-Native Architecture






NOVATEC

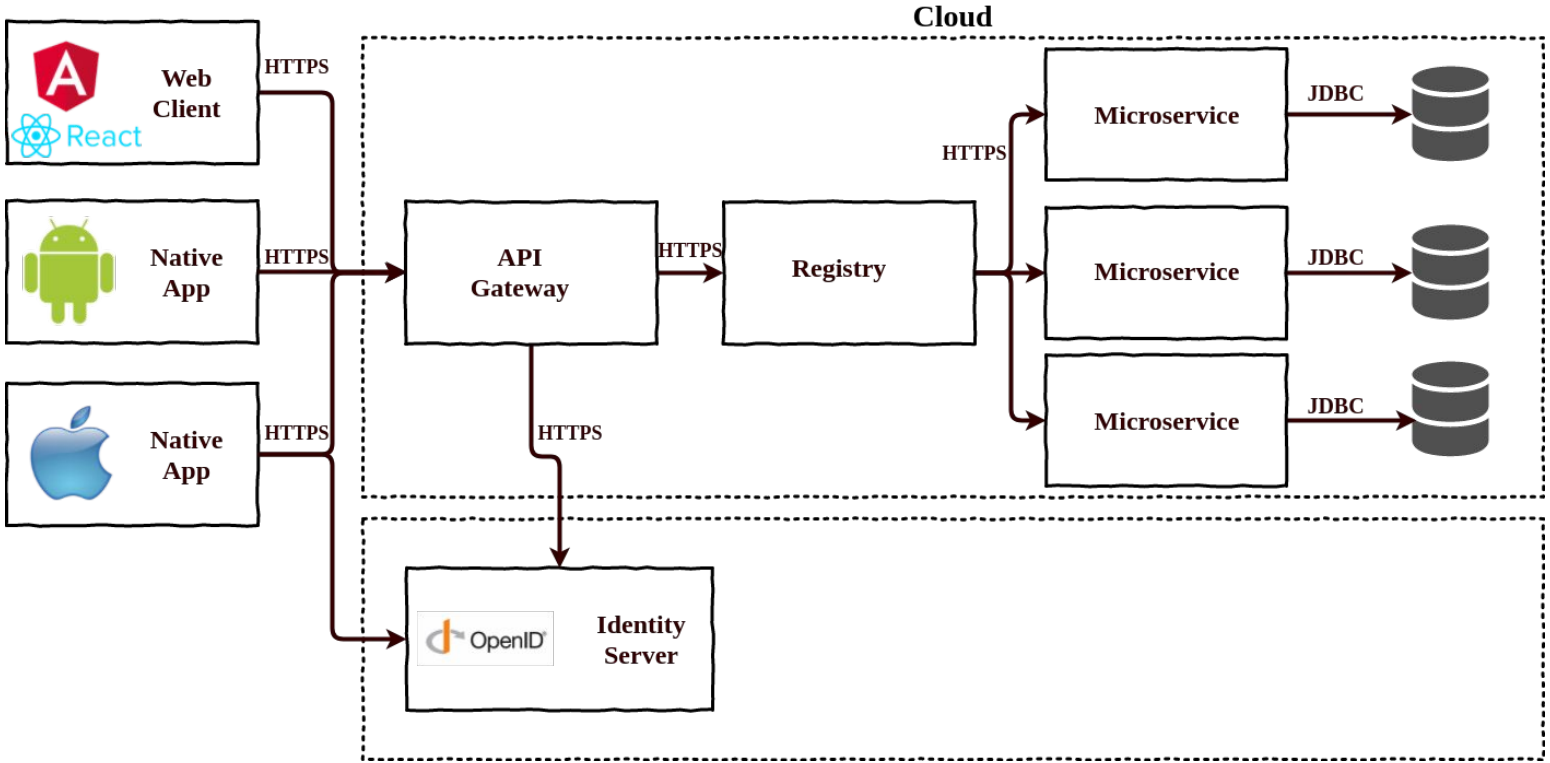
# Secure Cloud-Native Architecture






NOVATEC

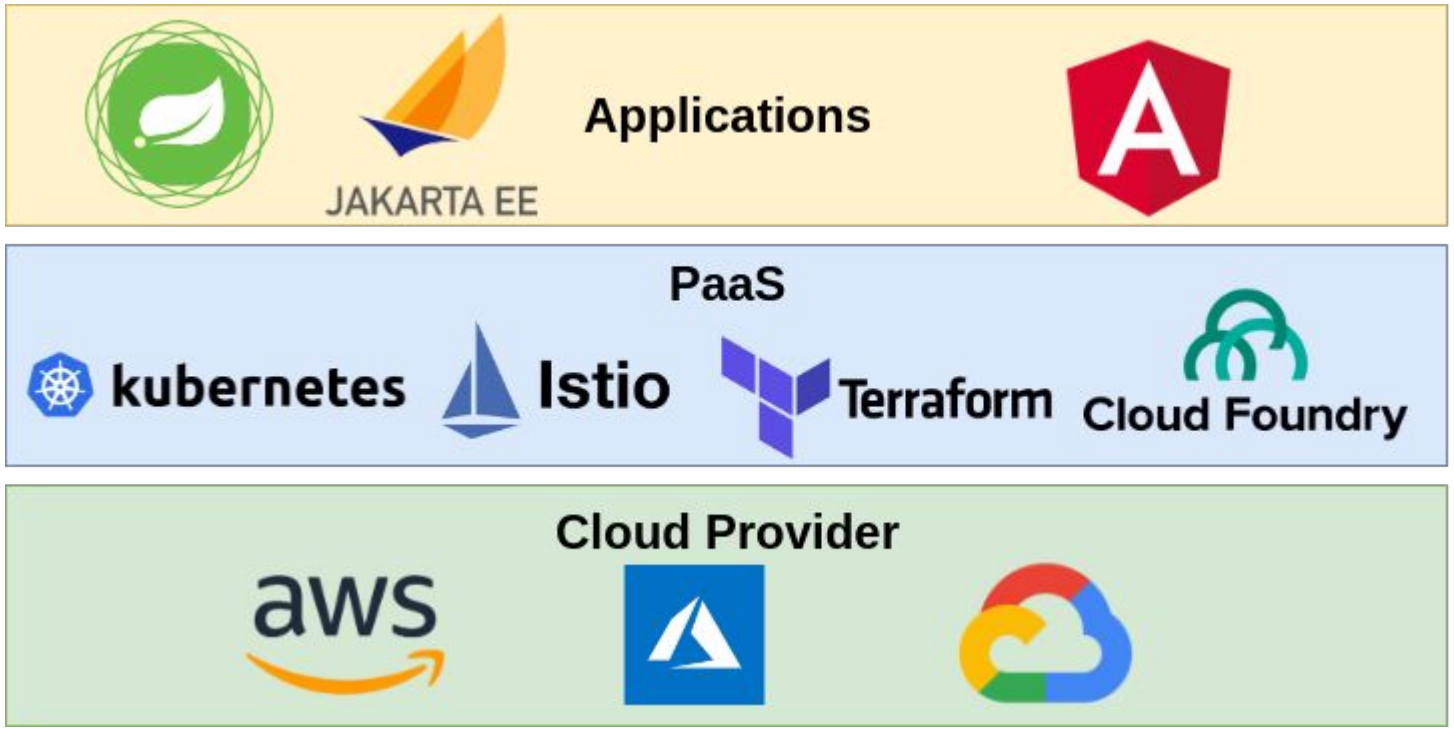
# Cloud-Native Architecture with Identity-Management



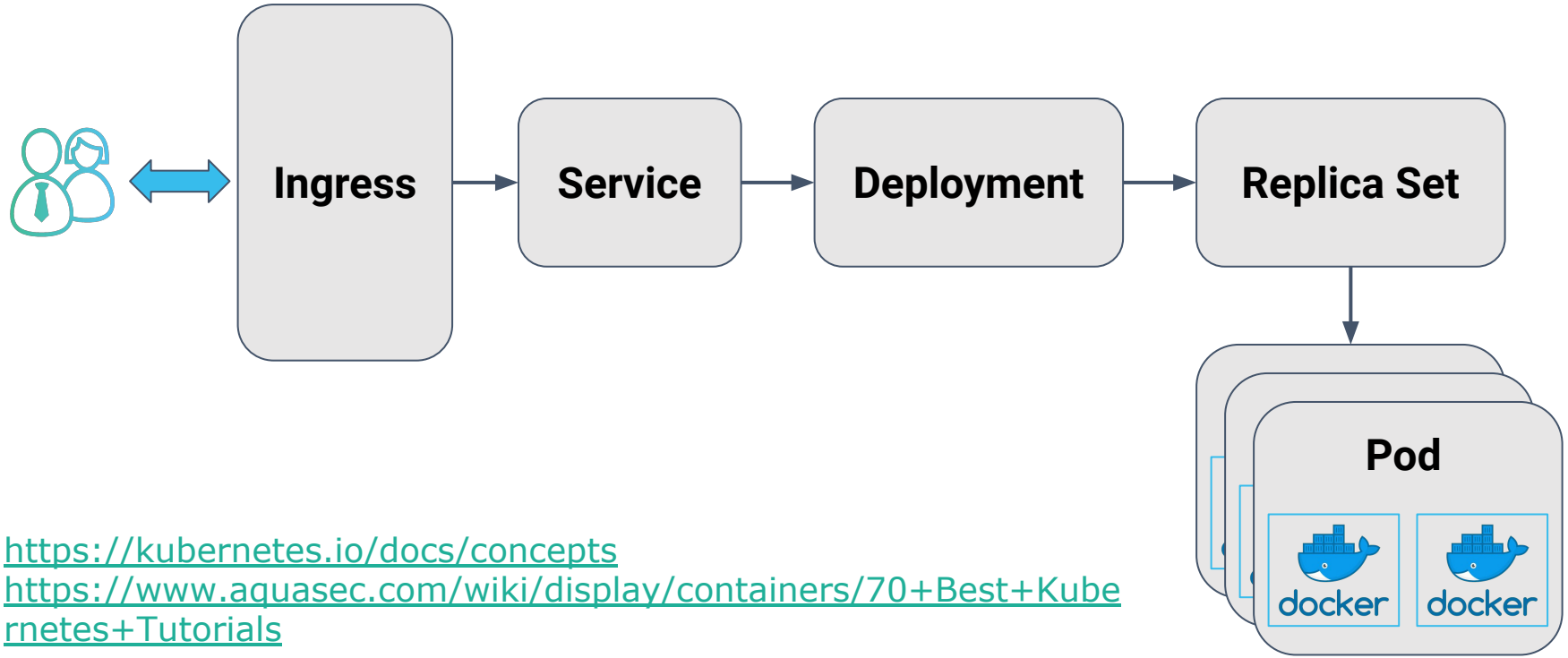



NOVATEC

# Technology Stacks

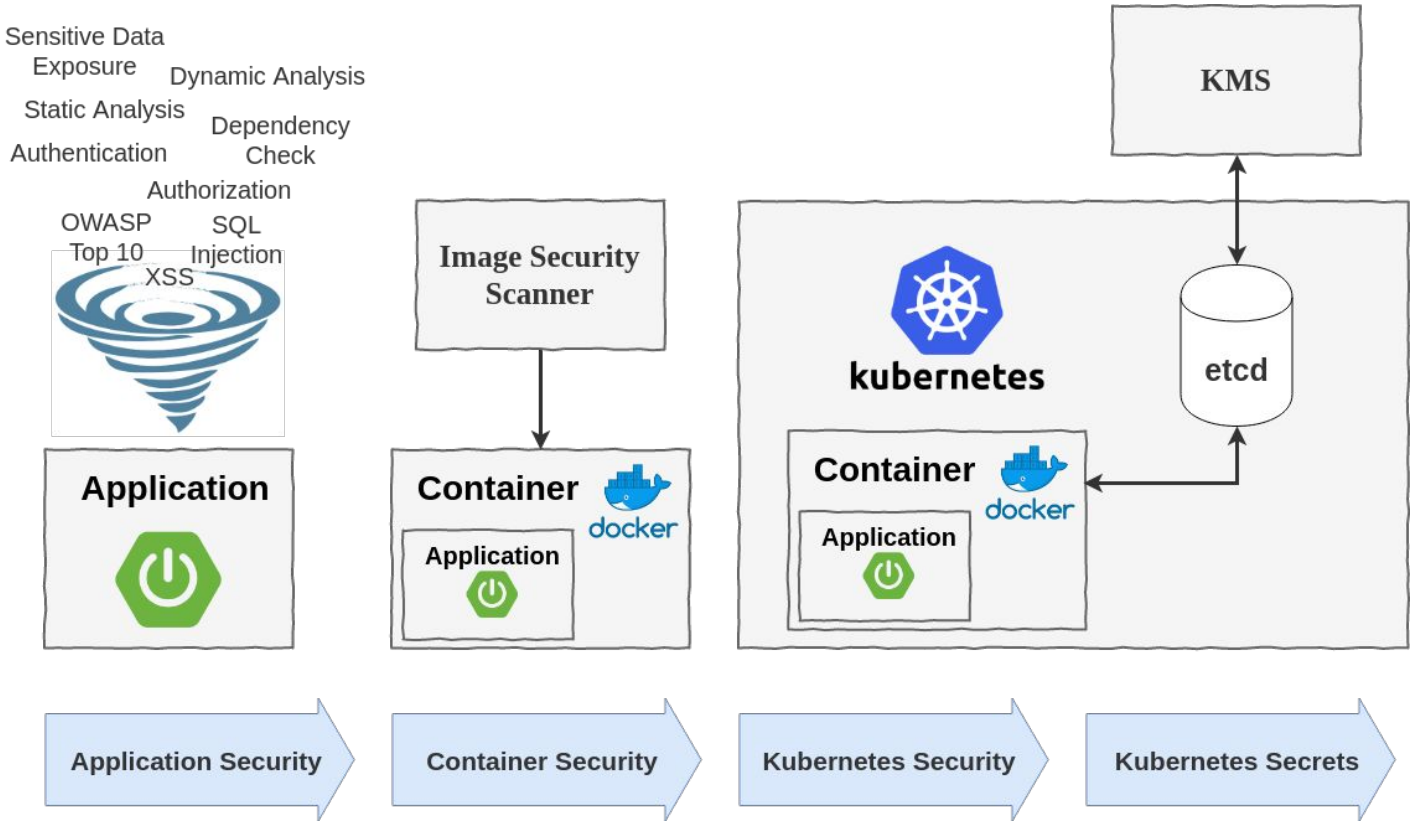


# Kubernetes Basics




https://kubernetes.io/docs/concepts
https://www.aquasec.com/wiki/display/containers/70+Best+Kubernetes+Tutorials

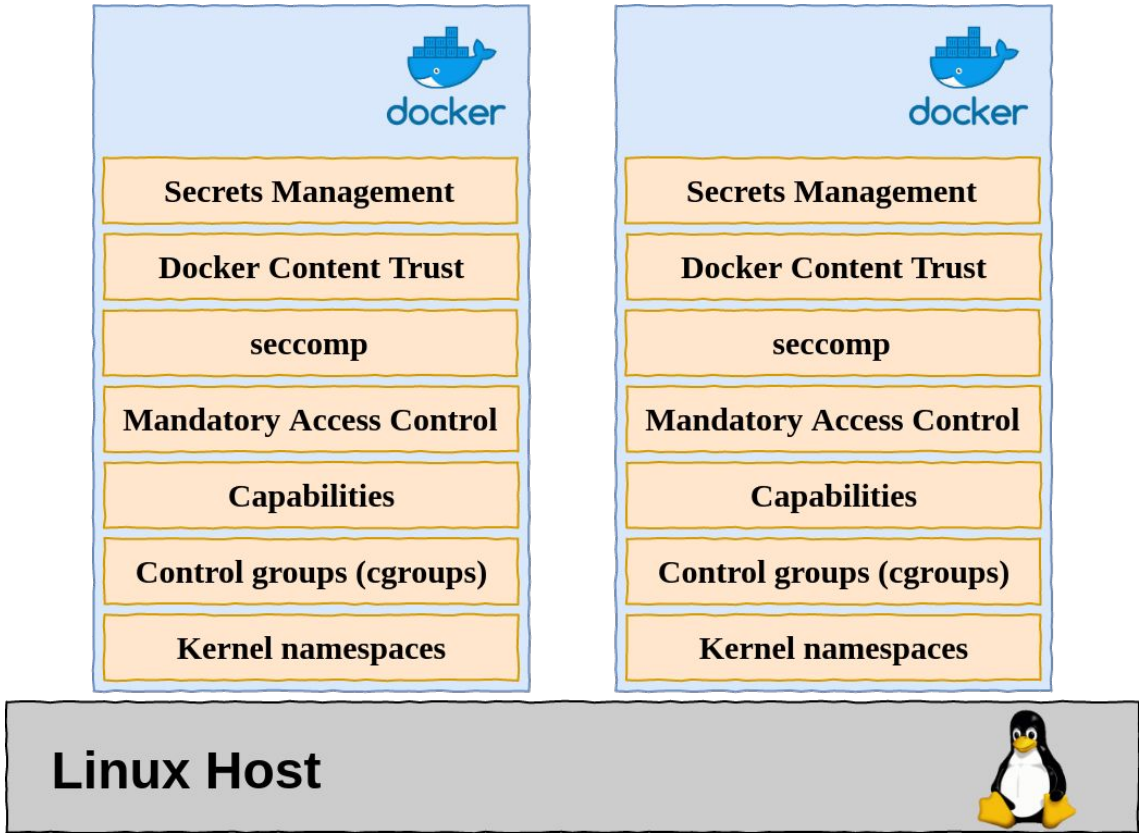
# Cloud-Native Security in Kubernetes






NOVATEC

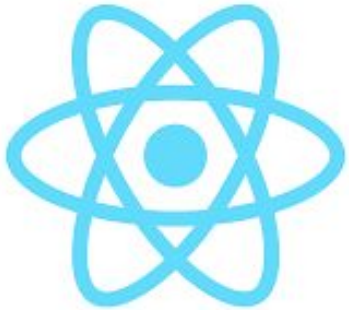
# Docker Security Basics



# Client Side Security

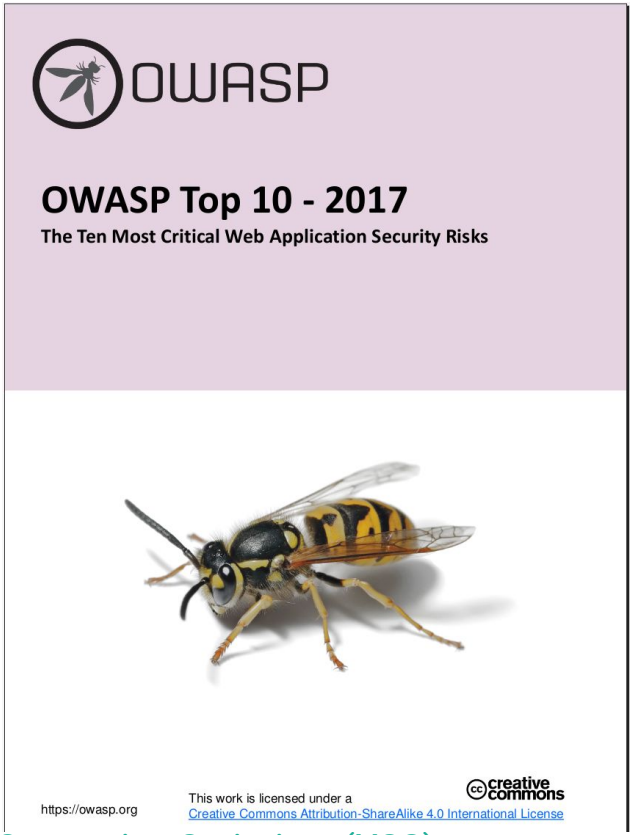
Frontends

# Cloud Side Security (Single Page Applications)



https://angular.io
https://vuejs.org
https://reactjs.org

# OWASP Top A7: Cross-Site Scripting (XSS)




https://owasp.org/index.php/Cross-site_Scripting_(XSS)



NOVATEC

# Cross-Site Scripting (XSS)

- Injection of malicious scripts into web sites
  - Reflected XSS
  - Persistent XSS
  - DOM based XSS

```
<script>alert(123)</script>

"><script>alert(document.cookie)</script>
```

# Cross-Site Scripting Defenses

- Output Escaping (Angular, React.js, Vue.js, JSF, Thymeleaf)
- Sanitizing (Angular, DOMPurify)
- Protect Sessions (e.g. in web.xml)

```
<session-config>
    <cookie-config>
        <http-only>true</http-only>
        <secure>true</secure>
    </cookie-config>
    <tracking-mode>COOKIE</tracking-mode>
</session-config>
```

# Angular

Angular.js = “Legacy” Angular 1.x versions

Angular = Current Angular version (latest is V8)

# Angular Security

“...The basic idea is to implement automatic secure escaping for all values that can reach the DOM...
By default, with no specific action for developers, Angular apps must be secure…”

https://github.com/angular/angular/issues/8511

# Angular XSS Protection

ANGULAR TEMPLATE = SAFE

INPUT VALUES = UNSAFE

# Angular Component (Typescript)

```
@Component({
    selector: 'app-root',
    templateUrl: 'app.component.html',
    styleUrls: ['app.component.css']
})
export class AppComponent {
    untrustedHtml:string =
                    '<em><script>alert("hello")</script></em>';
}
```

# Angular Template

HTML BINDINGS:

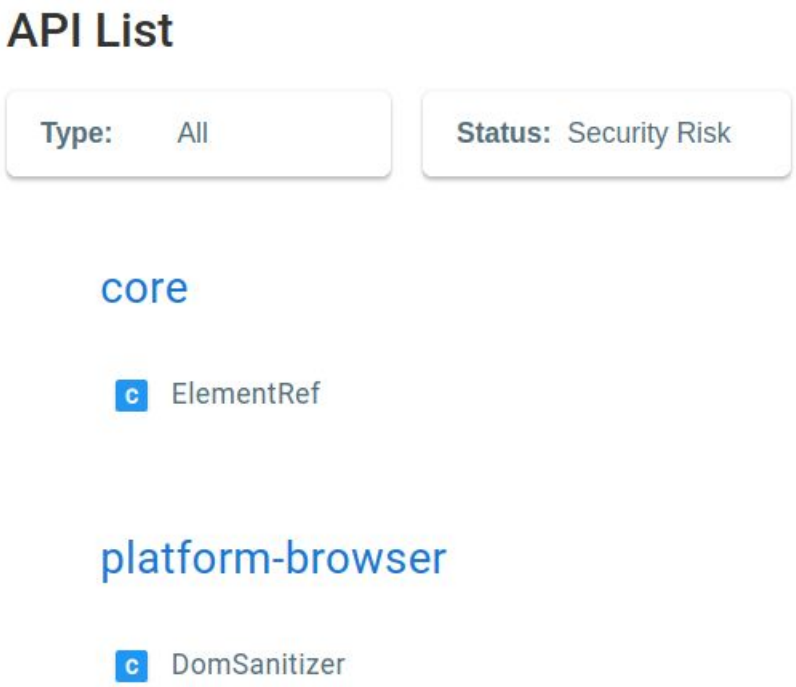
```
...
<h3>Encoded HTML snippet</h3>
<h3 class="trusted">{{untrustedHtml}}</h3>
...
<h3>Sanitized HTML snippet</h3>
<h3 class="trusted" [innerhtml]="untrustedHtml"></h3>
...
```

# Unsafe Angular APIs

- ElementRef:
  Direct access to DOM!
- DomSanitizer:
  Deactivates XSS-Protection!

**Do NOT use!**

https://angular.io/docs






NOVATEC

# Security Comparison Angular vs. React

Source: https://snyk.io/blog/javascript-frameworks-security-report-2019

| Item | Angular | React |
|---|---|---|
| Security page | ✓ https://angular.io/guide/security | ✗ React's website (https://reactjs.org) does not mention any security guidelines, except for the dangerouslySetInnerHTML function reference in the DOM Elements section of the API Reference documentation. |
| Security contact | ✓ security@angular.io | ✗ No security contact |
| Responsible disclosure policy | ✓ Backed by the internal security teams at Google and based on Google security philosophy. Reference: https://www.google.com/about/appsecurity/ | ✗ No responsible disclosure policy |
| Examples of vulnerable projects | ✓ https://angular.io/generated/live-examples/security/stackblitz.html | ✗ No references to any examples of vulnerable projects |
| Built-in sanitization | ✓ DomSanitizer provides a built-in sanitization function for untrusted values. Reference: https://angular.io/api/platform-browser/DomSanitizer#sanitize | ✗ Potentially malicious input sanitization is at the users' discretion to be implemented via 3rd-party libraries, such as DOMPurify. Reference: https://github.com/cure53/DOMPurify |
| Content Security Policy (CSP) | ✓ CSP compatibility for Angular v1.x directives. Reference: https://docs.angularjs.org/api/ng/directive/ngCsp | ⊖ Not relevant for React |
| Cross-Site Request Forgery (CSRF) | ✓ CSRF built-in support through Angular's HttpClient service. Reference: https://angular.io/guide/http and https://docs.angularjs.org/api/ng/service/$http | ⊖ Not relevant for React as a view library. This is up to the developers to handle using custom code or community modules. |

# Server Side Security

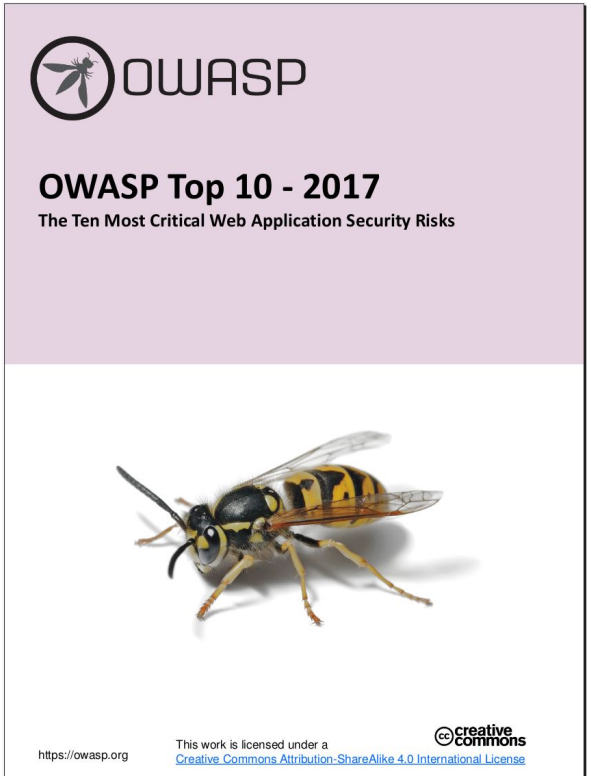
Java/Kotlin Backends

# Java / Kotlin Server-Side Applications



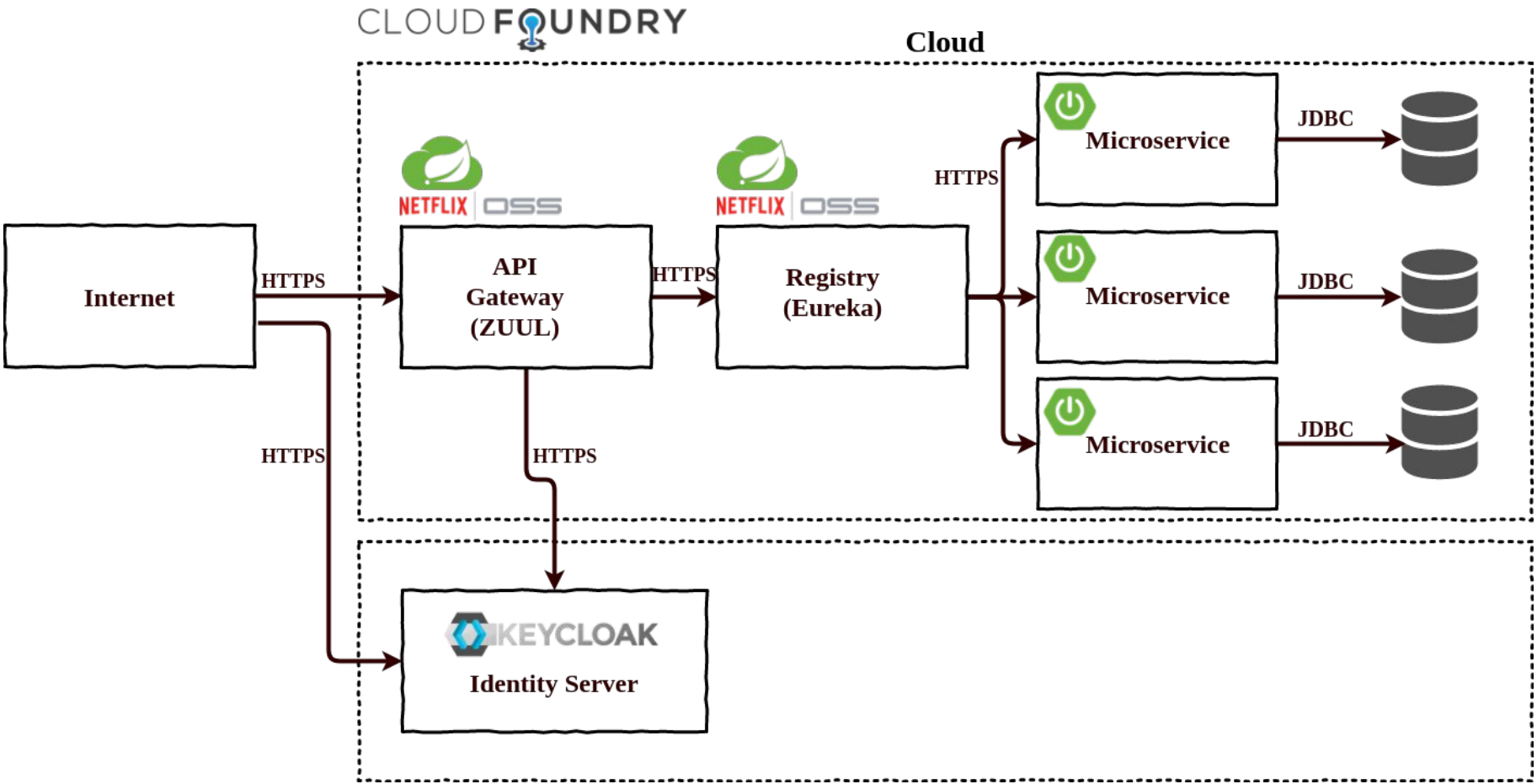
# OWASP Top 10 on Server-Side

A1: Injection
A2: Broken Authentication
A3: Sensitive Data Exposure
A5: Broken Access Control
A6: Security Misconfiguration
A9: Using Components With Known Vulnerabilities



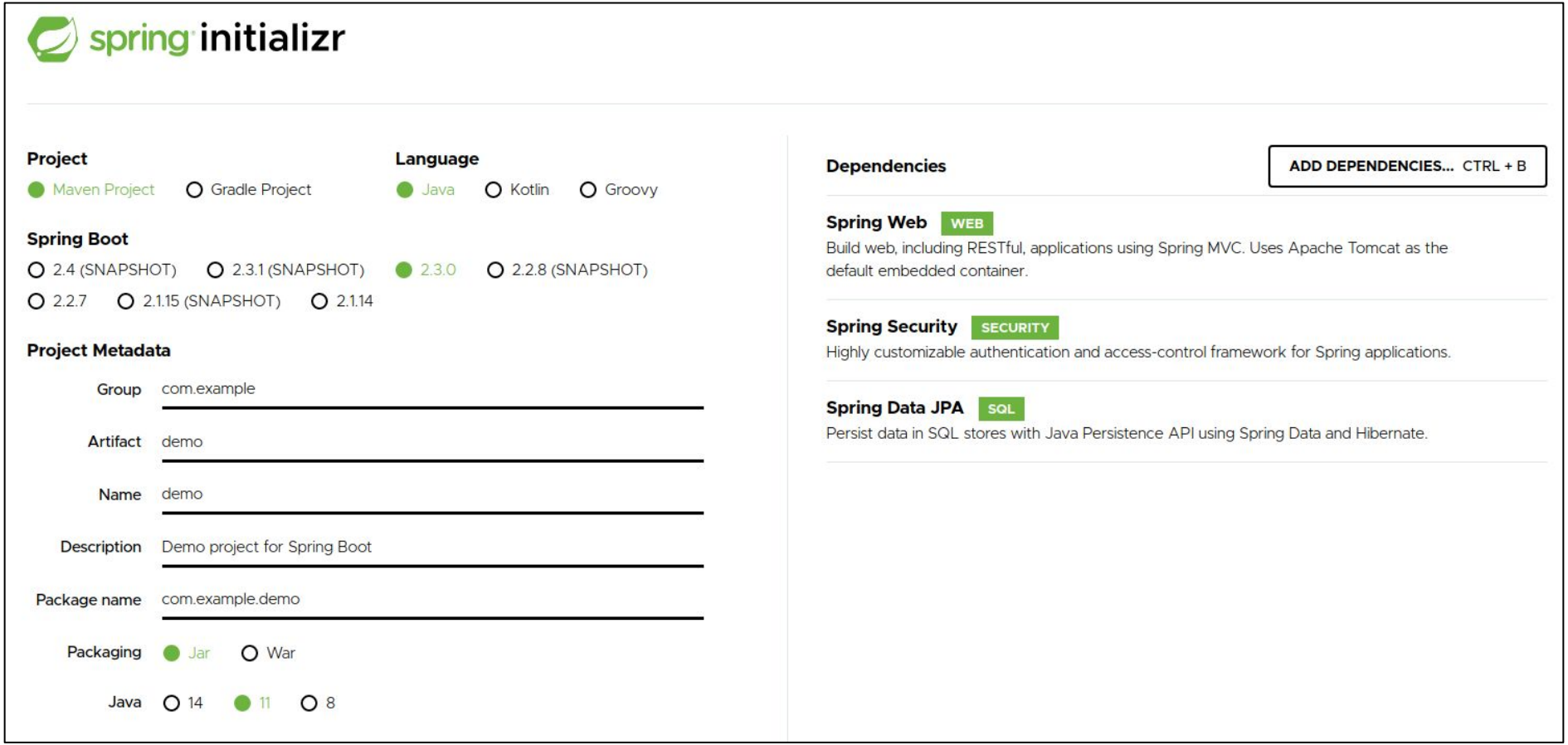
# The Spring Platform


https://spring.io


CLOUD FOUNDRY
Cloud
NETFLIX OSS
NETFLIX OSS
Internet
HTTPS
API
Gateway
(ZUUL)
HTTPS
Registry
(Eureka)
HTTPS
Microservice
JDBC
Microservice
JDBC
Microservice
JDBC
HTTPS
HTTPS
KEYCLOAK
Identity Server



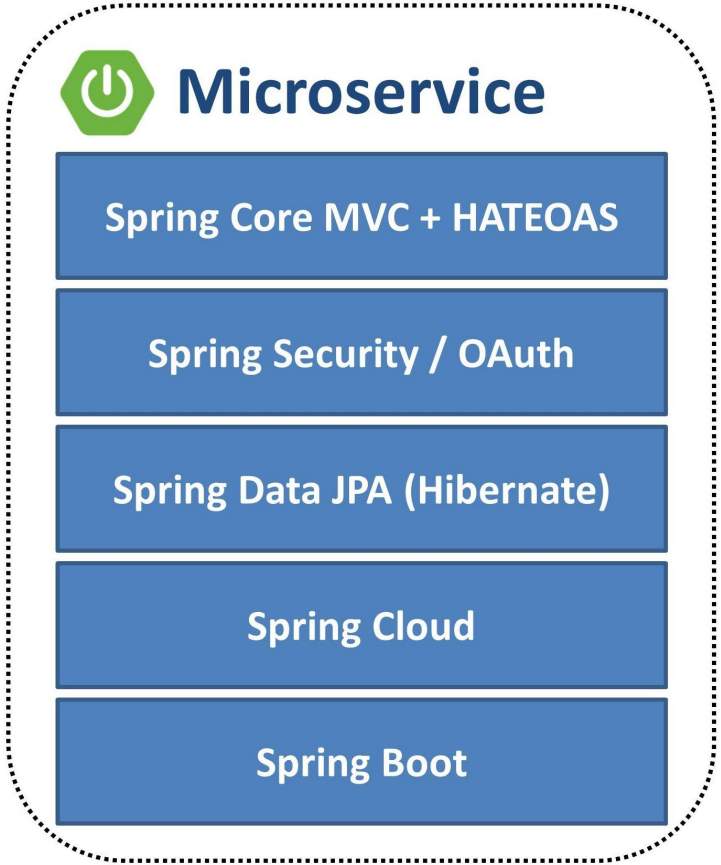
NOVATEC



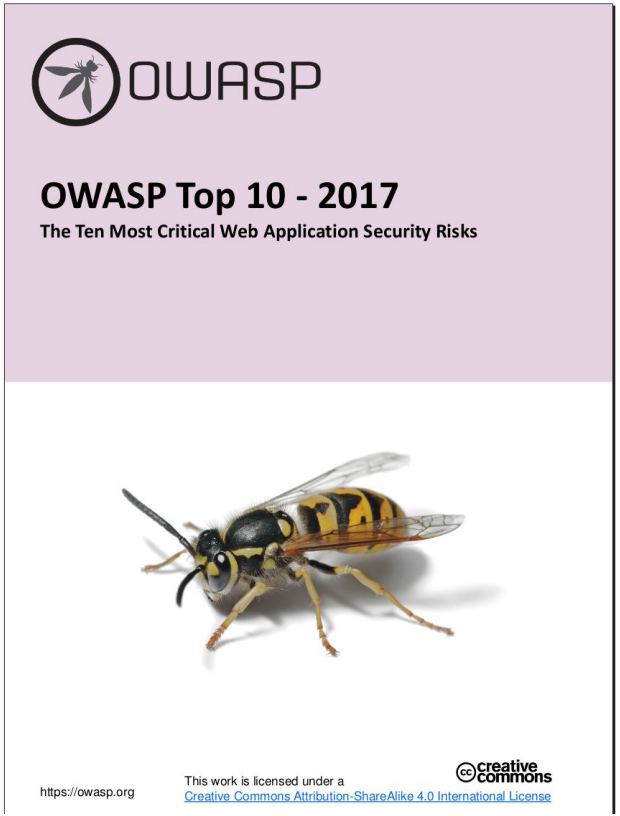



https://start.spring.io

# The Spring Microservice Tech Stack



# A2: Broken Authentication






NOVATEC

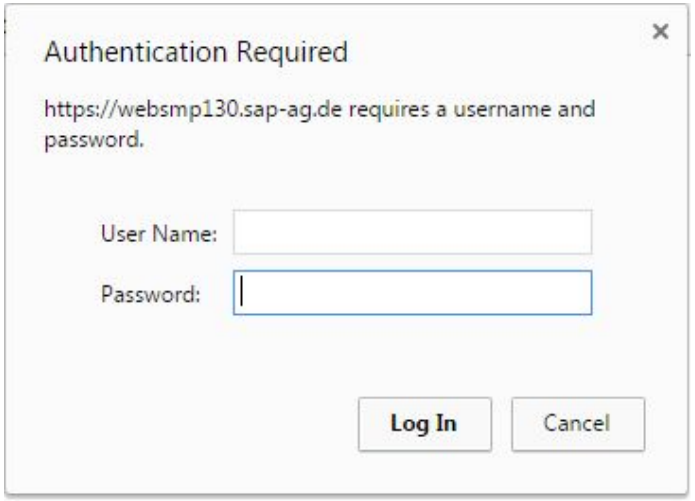
# Authentication - Who am I?

HTTP 401 - UNAUTHORIZED

# Authentication Mechanisms

- Basic Authentication / Digest Access Authentication
- Form-based Authentication (i.e. using Session Cookies)
- Client-Certificates (MTLS)
- Kerberos Tickets
- SAML Assertion Tokens
- JSON Web Tokens
- OAuth 2.0 & OpenID Connect
- Proprietary mechanisms like API-Tokens, Siteminder etc.

# Basic Authentication

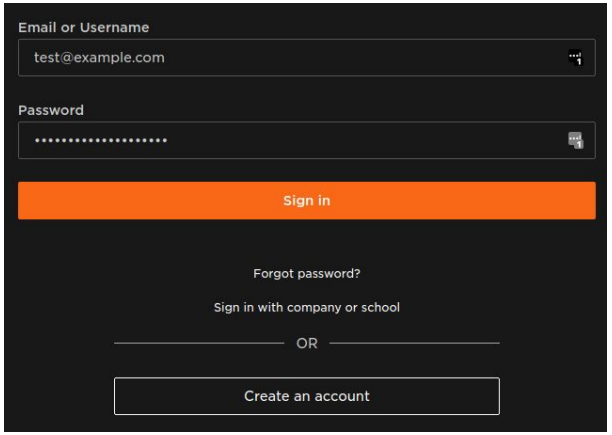



GET / HTTP/1.1
Host: localhost:8080

HTTP/1.1 401
WWW-Authenticate: Basic realm="hello"

GET / HTTP/1.1
Host: localhost:8080
Authorization: Basic dXNlcjpzZWNyZXQ=

# Form-Based Authentication




```
POST /login HTTP/1.1
Host: localhost:8080
Content-Type: application/x-www-form-urlencoded
Cookie: JSESSIONID=14965E3A995DA1973F42F308D59727D4
username=user&password=secret&submit=Login

HTTP/1.1 302
Set-Cookie: JSESSIONID=49C63238780031602 1BE;Path=/; HttpOnly

GET / HTTP/1.1
Host: localhost:8080
Cookie: JSESSIONID=49C63238780031602 1BE
```

# Bearer Token Authentication

- Used for
  - OAuth 2.0
  - OpenID Connect

```
GET / HTTP/1.1
Host: localhost:8080
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiI
```

# Authentication - Stateful vs. Stateless

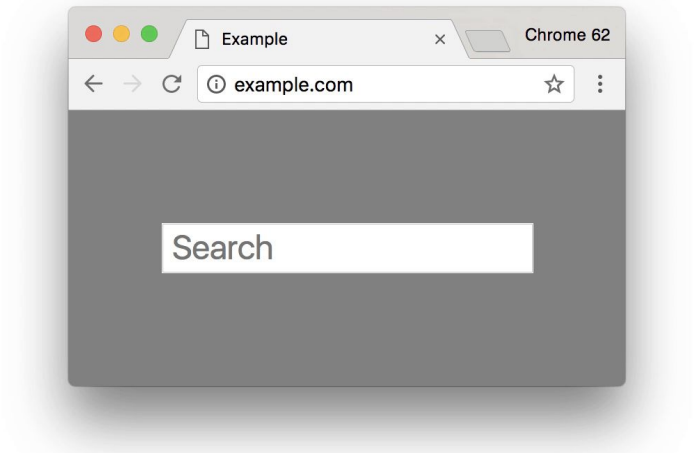
| Session Cookie | Token |
| --- | --- |
| With each Request (on same domain) | Manually set as Header |
| Potential CSRF! | No CSRF possible |
| One domain | Cross domain (Cross-Origin Resource Sharing) |
| Sensitive Info (HTTPS) | Sensitive Info (HTTPS) |

# HTTPS (SSL/TLS)

- Validation (Destination Server)
- Data Confidentiality (Encryption)
- Data Integrity (Hashing)

# HTTPS (SSL/TLS)

- Let's Encrypt
- CloudFlare
- HTTP/2






NOVATEC


# OAuth 2.0 (RFC 6749)

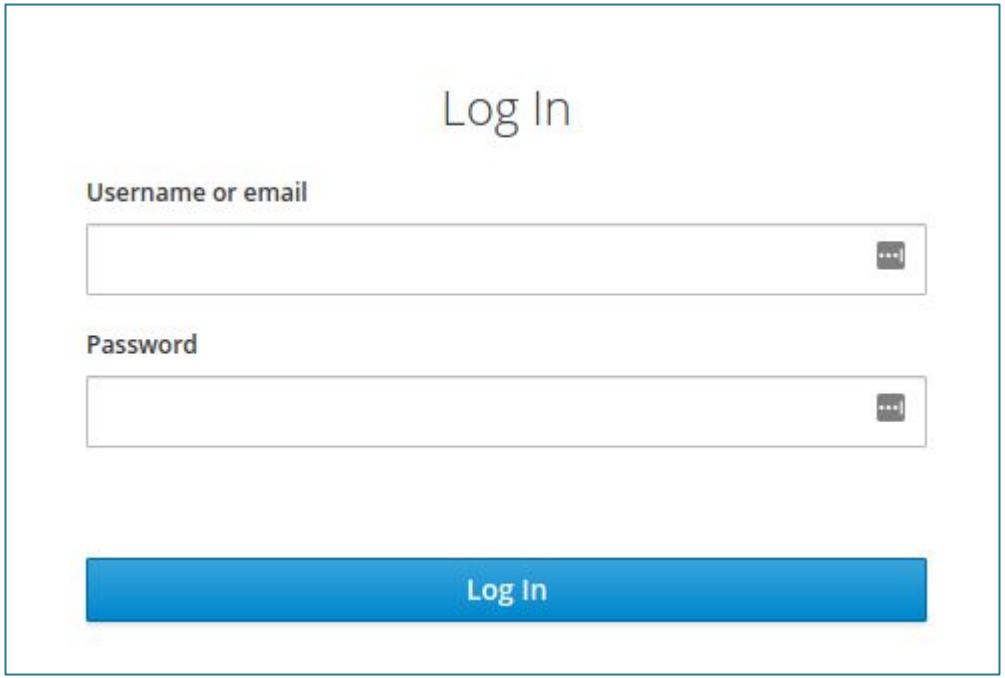
https://tools.ietf.org/html/rfc6749

# Do you like implementing your own Authentication?

**Different Clients**

**Brute Force Prevention**

**Reset Password Process**

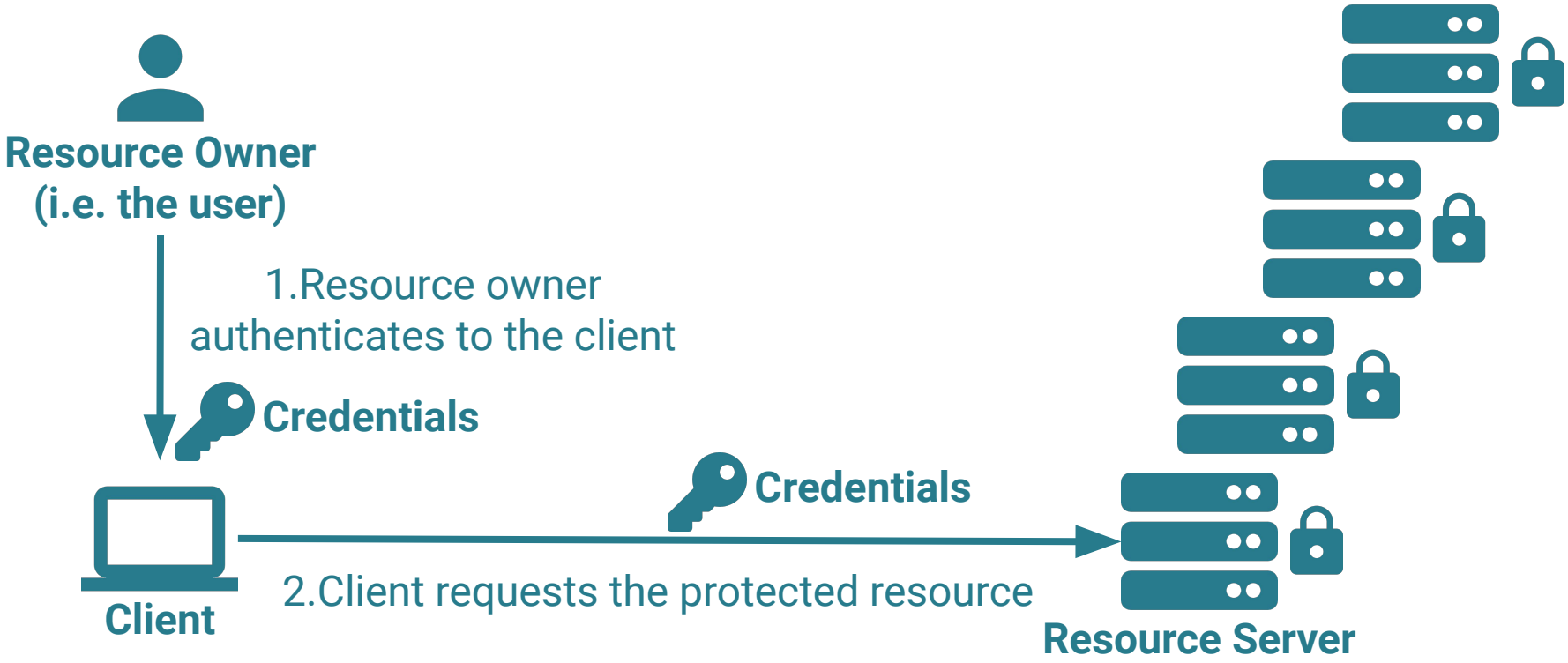



**Password Policies**

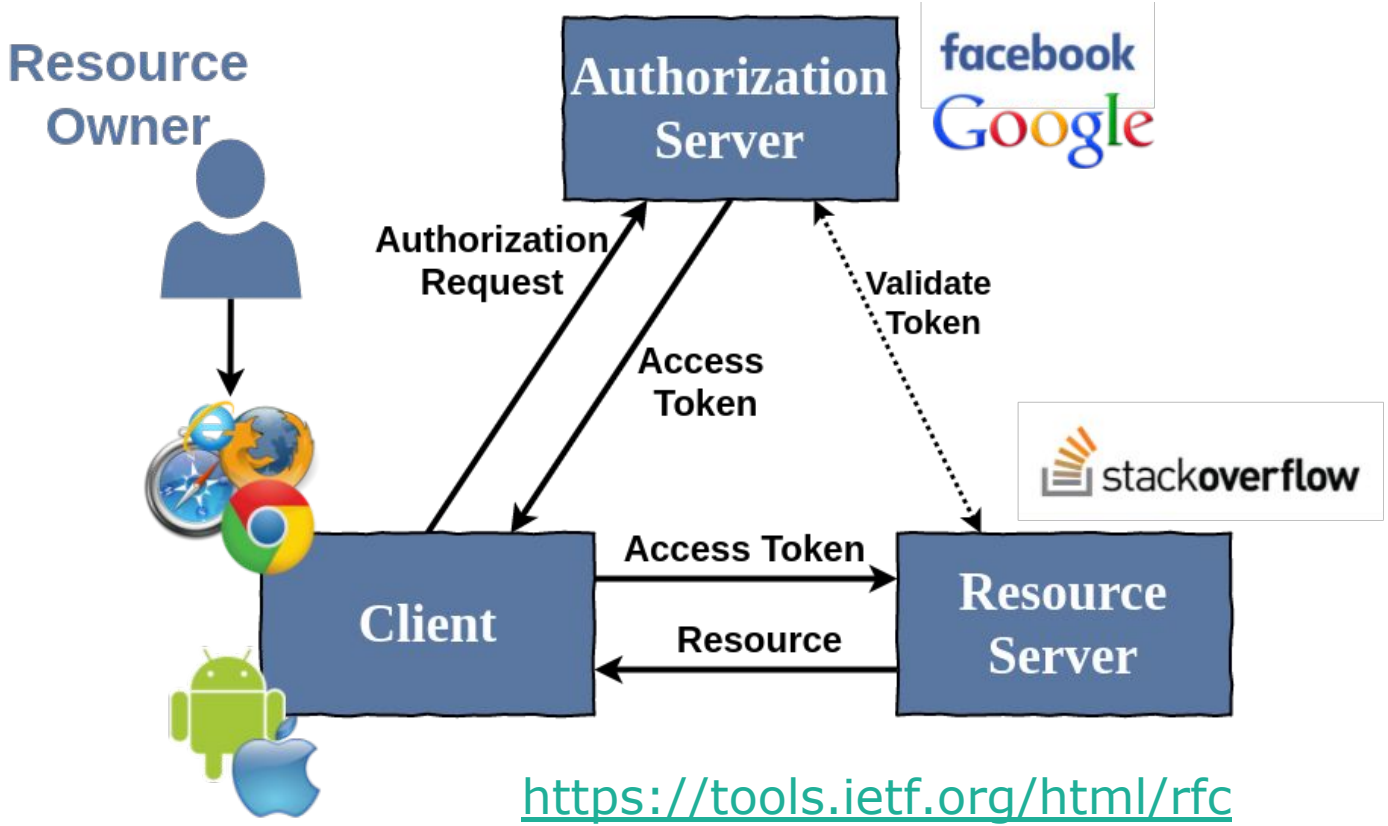
**MFA**

**Secure Password Storage**

# Spreading Credentials before OAuth 2.0

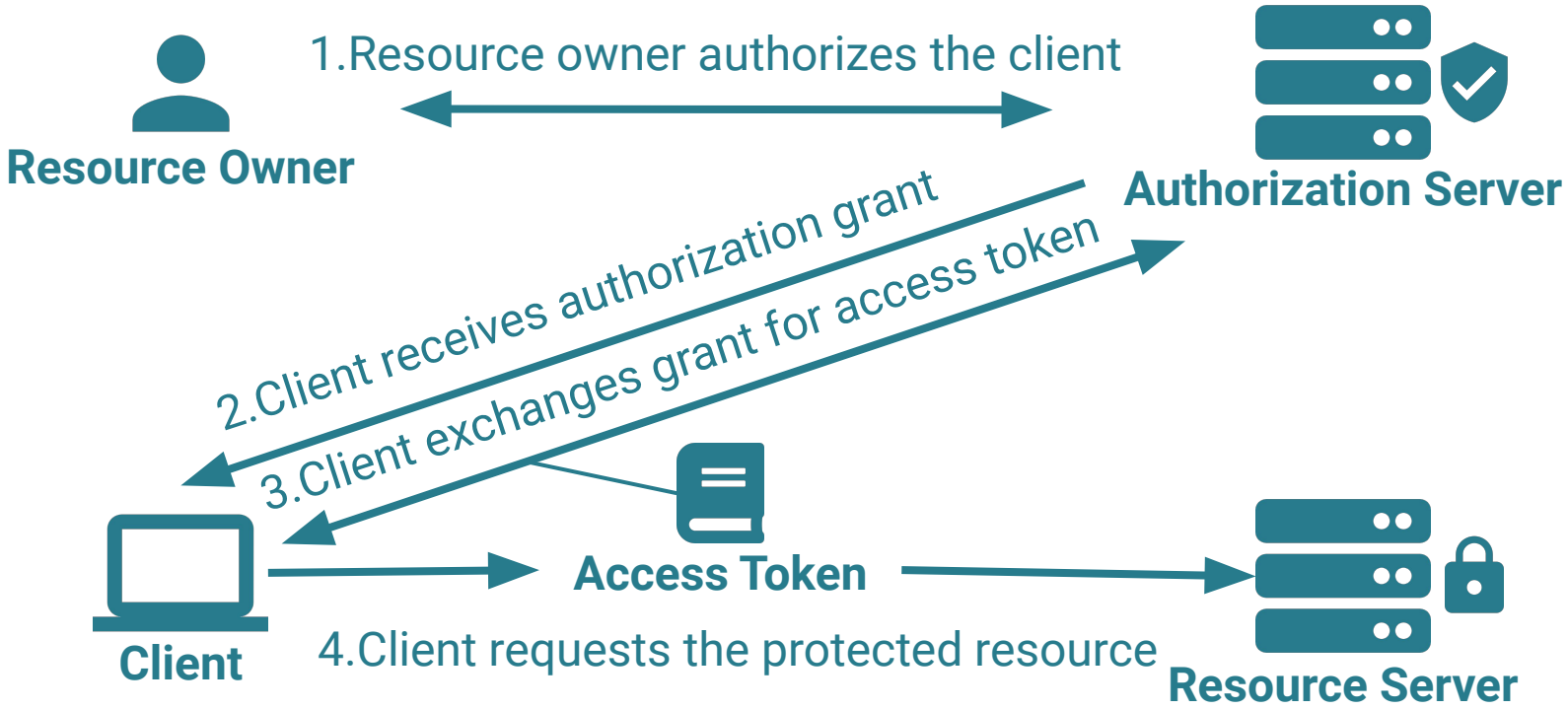


# OAuth 2.0 Roles

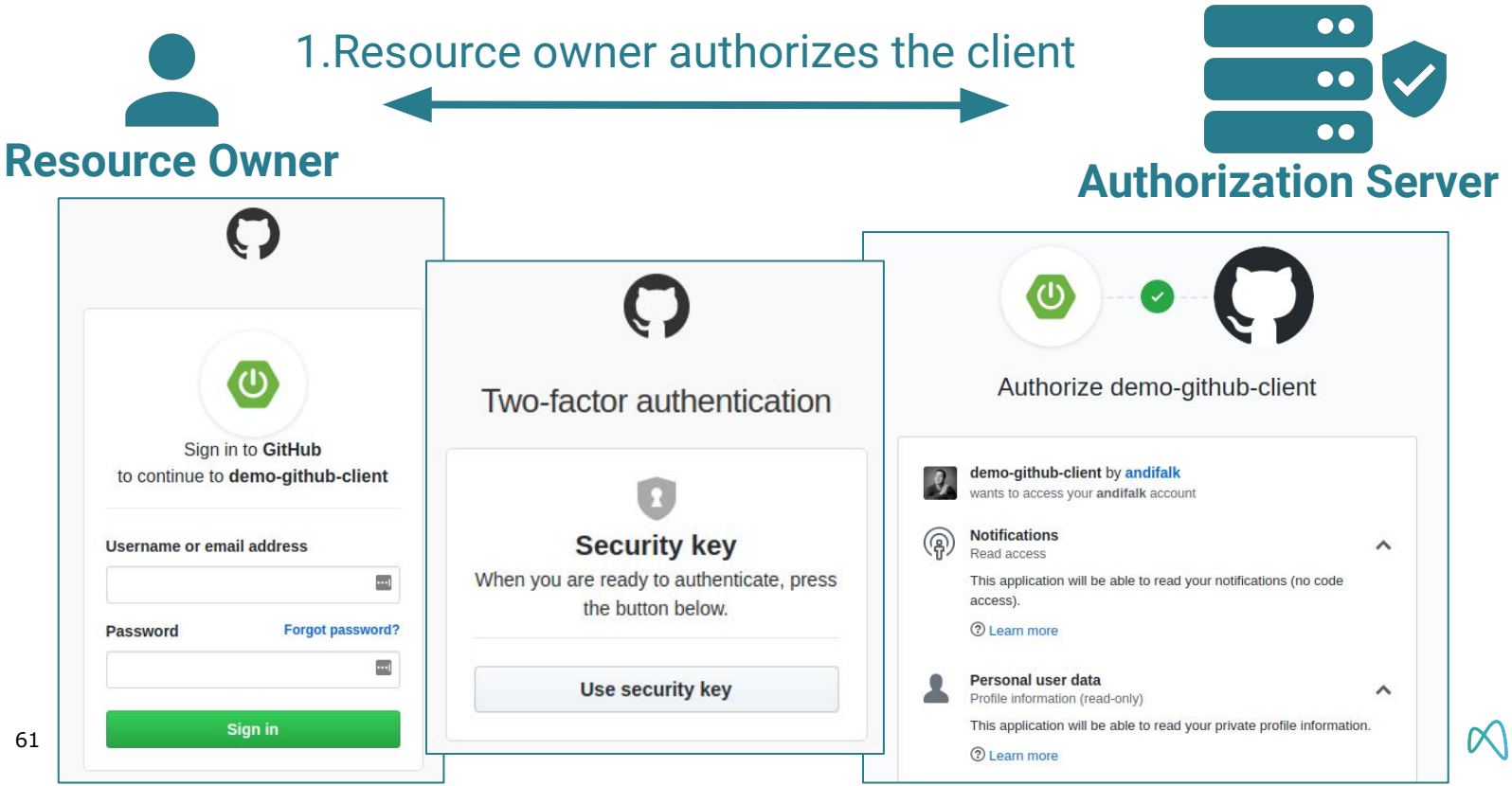



https://tools.ietf.org/html/rfc6749

# Basic OAuth 2.0 Protocol Flow



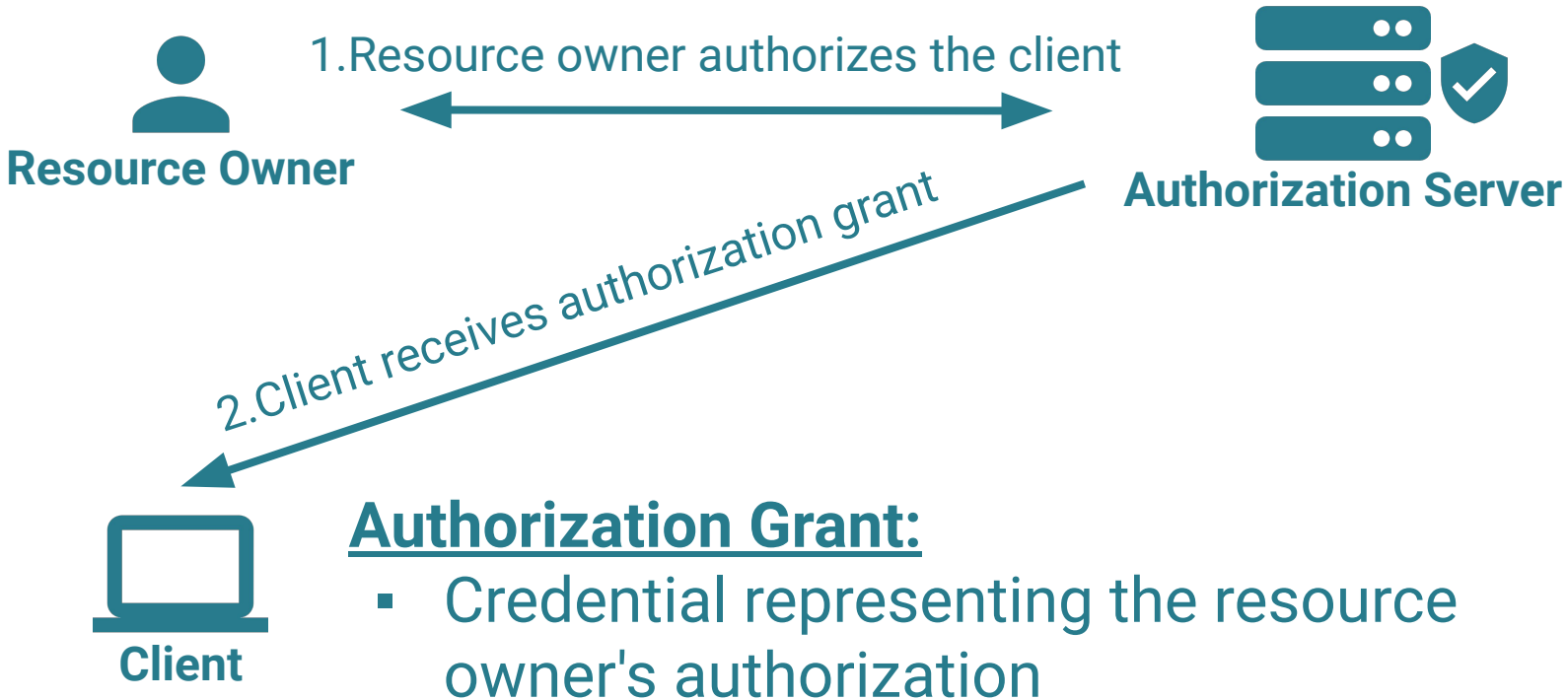
# Protocol Flow (1): Resource owner authorizes client






NOVATEC

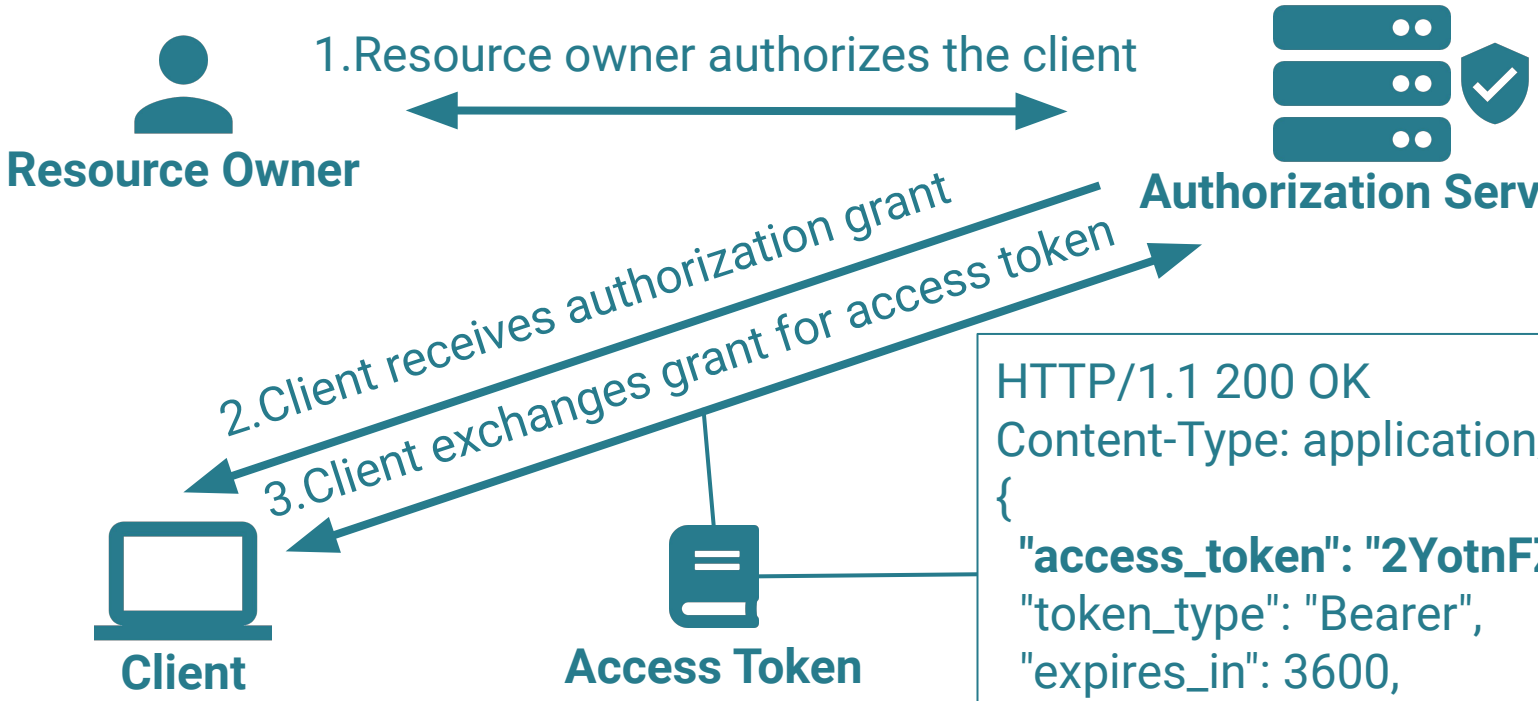
# Protocol Flow (2): Client receives authorization grant




**Authorization Grant:**

- Credential representing the resource owner's authorization
- Exchange for access token

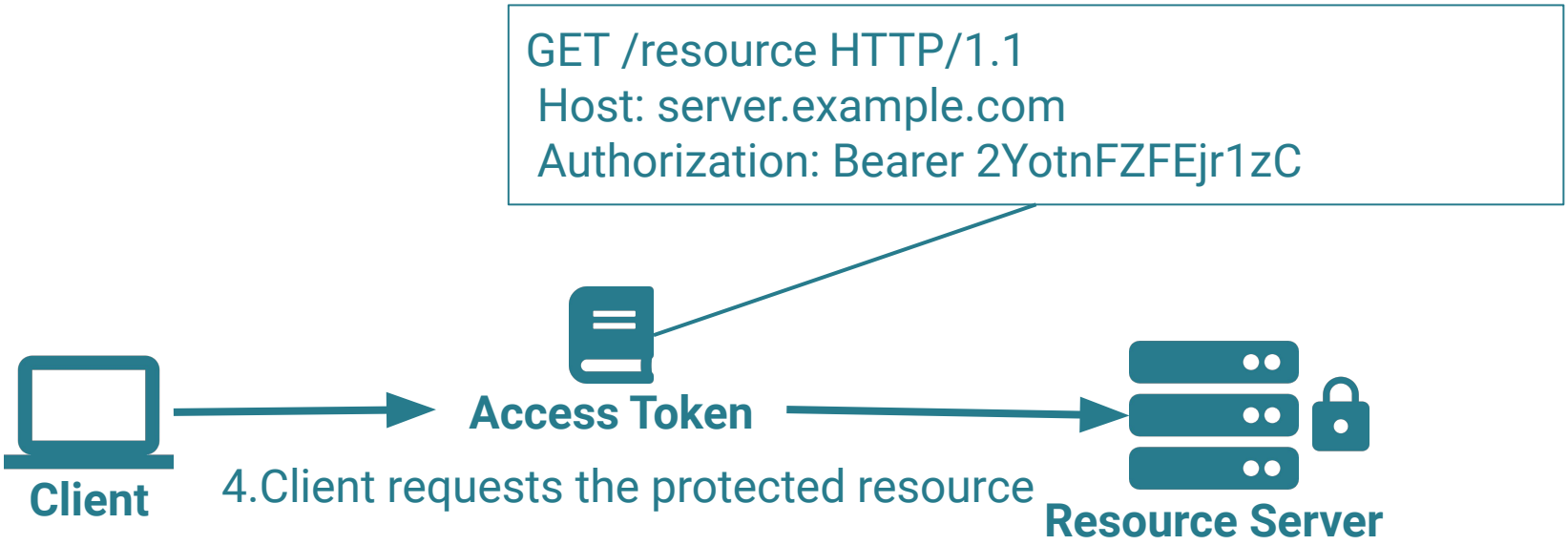
# Protocol Flow (3): Client exchanges grant for access token






NOVATEC

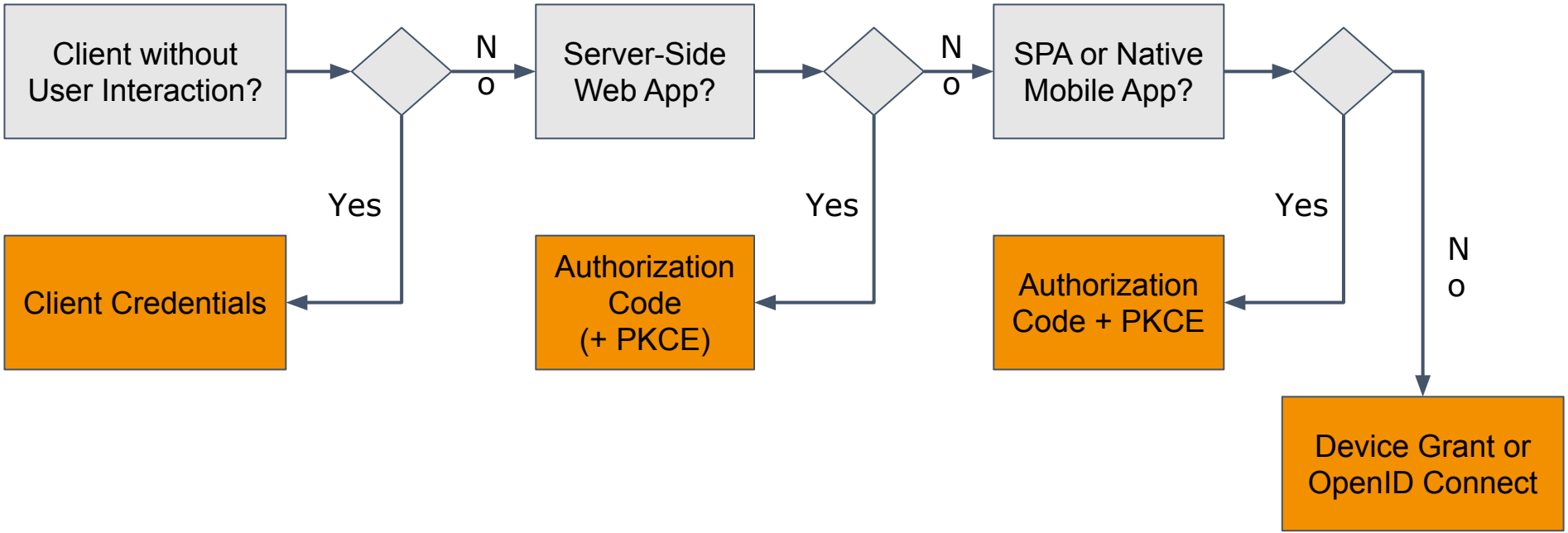
# Protocol Flow:  Client requests the protected resource



# Demo Time

## Authorization Code Grant Flow in Action

# Which Authorization Grant should I Use ?

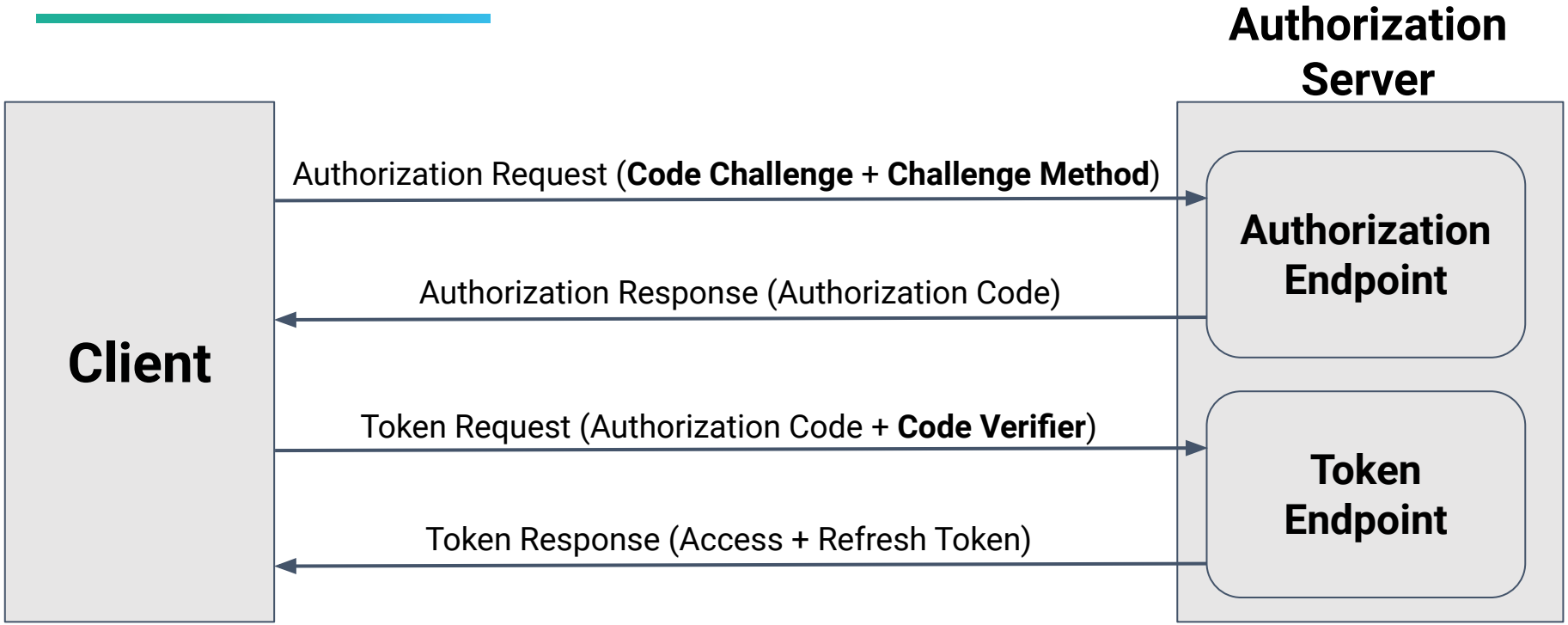


# OAuth 2.1 Authorization Grant Types

| Resource Owner | Client Type | Authorization Grant | Refresh Token |
|---|---|---|---|
| ✓ | Web Client (Confidential) | Authorization Code + PKCE | ✓ |
| ✓ | Mobile Client (Public) | Authorization Code + PKCE | ✓ |
| ✓ | SPA Client (Public) | Authorization Code + PKCE | ✓ |
| ✗ | Resource Owner = Client (Confidential) | Client Credentials | ✗ |

https://datatracker.ietf.org/doc/draft-ietf-oauth-security-topics
https://datatracker.ietf.org/doc/draft-parecki-oauth-v2-1

# Authorization Code + PKCE

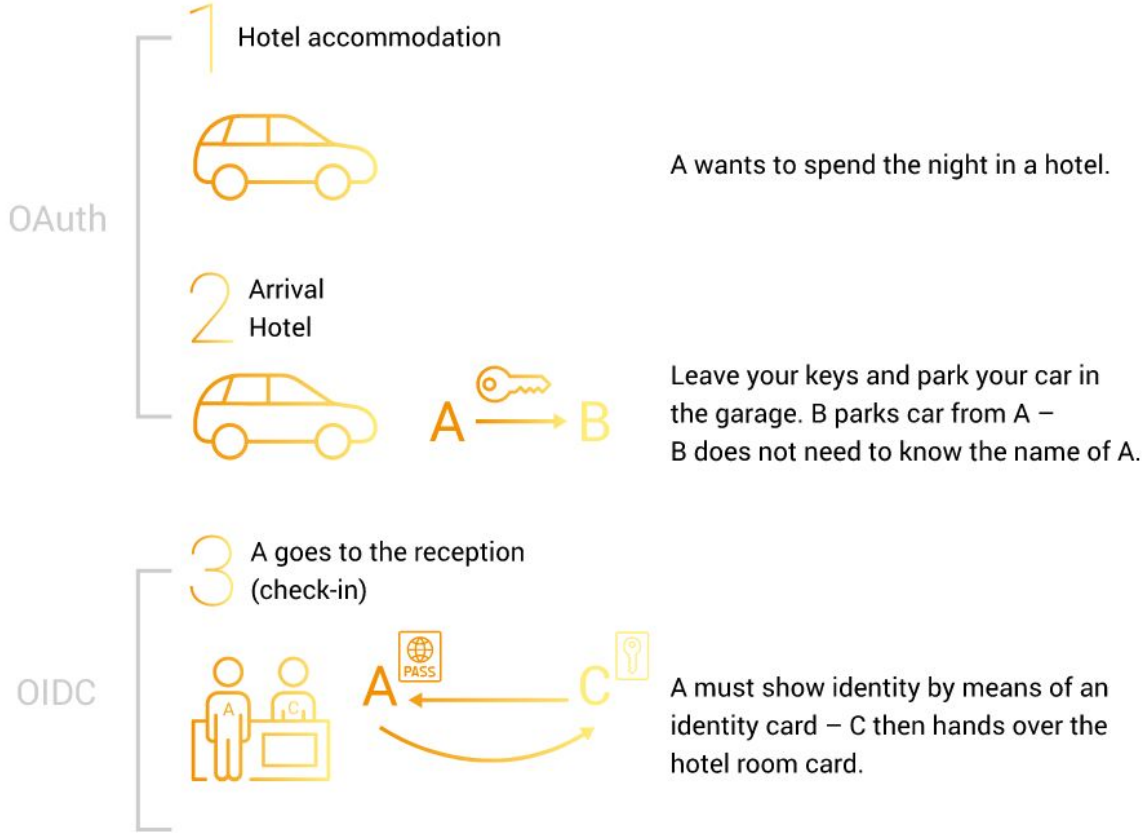



https://tools.ietf.org/html/rfc7636


# OpenID Connect 1.0

https://openid.net/specs/openid-connect-core-1_0.html

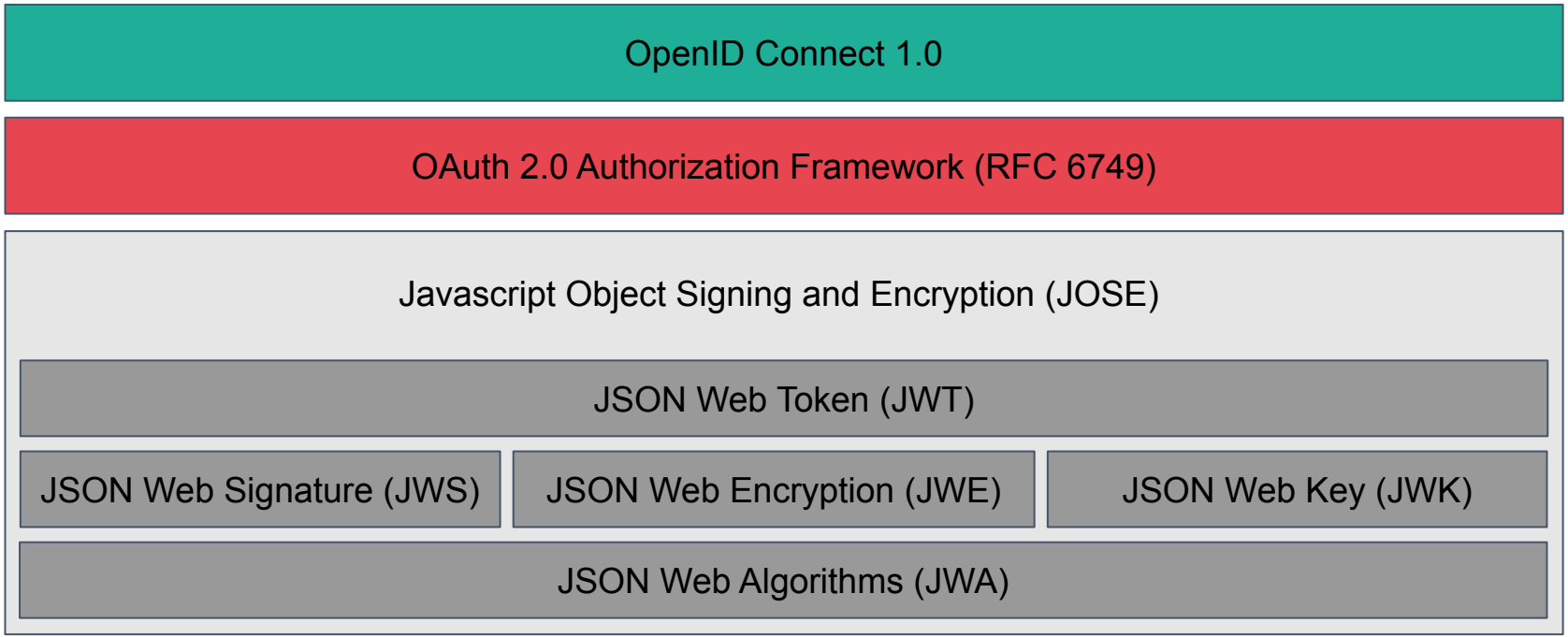
# OAuth 2.0 is NOT an Authentication Protocol!






NOVATEC

# OpenID Connect 1.0 Standards Layer



# JSON Web Algorithms (JWA)

- Cryptographic algorithms and identifiers for JWS, JWE, and JWK specifications
- Digital Signatures and MACs
- Algorithms for Key Management
- Algorithms for Content Encryption
- Algorithms for Keys

https://tools.ietf.org/html/rfc7518

# JSON Web Key (JWK)


```
{"keys": [
       {"kty":"EC",
        "crv":"P-256",
        "x":"MKBCTNIcKUSDii11ySs3526iDZ8AiTo7Tu6KPAqv7D4",
        "y":"4Etl6SRW2YiLUrN5vfvVHuhp7x8PxltmWWlbbM4IFyM",
        "use":"enc",
        "kid":"1"},
       {"kty":"RSA",
        "n": "0vx7agoebGcQSuuPiLJXZptN9nn...",
        "e":"AQAB",
        "alg":"RS256",
        "kid":"2011-04-29"}]}
```


https://tools.ietf.org/html/rfc7517

# JSON Web Signature (JWS)

- JSON Web Signature (JWS) represents content secured with digital signatures or Message Authentication Codes (MACs) using JSON-based data structures
- A document using JWS can answer two questions about the JSON payload:
  - Has the JSON object been altered after creation?
  - Who created this JSON object?

https://tools.ietf.org/html/rfc7515

# JSON Web Encryption (JWE)

- Data structure representing an encrypted and integrity-protected message
- As of July 2019 only identity server of PingIdentity supports JWE
- NOT supported by Spring Security 5.x (See github issue 4435) !

https://tools.ietf.org/html/rfc7516
https://github.com/spring-projects/spring-security/issues/4435

# JSON Web Token (JWT)

- JSON Web Tokens consist of three parts separated by dots (“.”), which are:
  - Header
  - Payload
  - Signature
- Each part is Base64Url encoded
- Signature supports symmetric or asymmetric algorithms (e.g. HMAC or RSA)
- Signature = HMACSHA256(
  base64UrlEncode(header) + "." + base64UrlEncode(payload), secret)

https://tools.ietf.org/html/rfc7519
https://tools.ietf.org/html/draft-ietf-oauth-jwt-bcp
https://tools.ietf.org/html/draft-ietf-oauth-proof-of-possession

# JSON Web Token (JWT) - Decoded Form

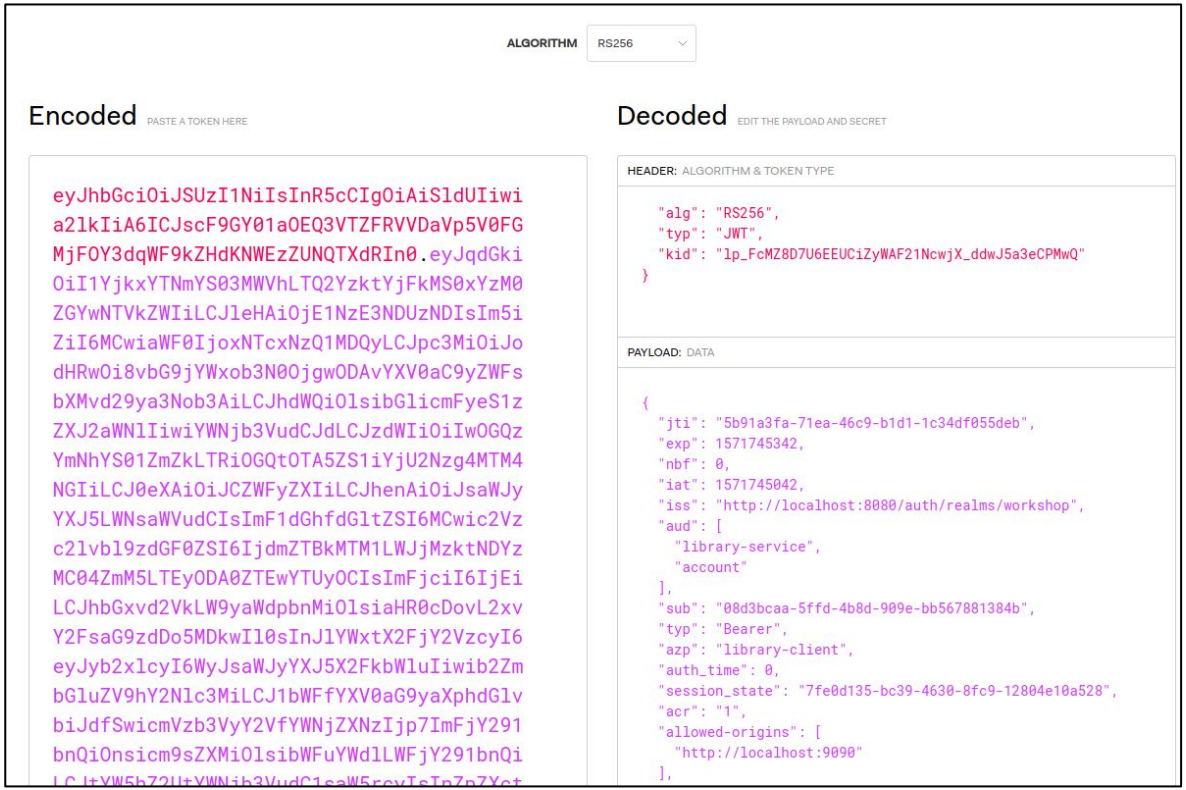

```
{
  "alg": "RS256",
  "typ": "JWT",
  "kid": "lp_FcMZ8D7U6EEUCiZyWAF21NcwjX_ddwJ5a3eCPMwQ"
}
```


HEADER


```
{
  "exp": 1571745342,
  "iat": 1571745042,
  "iss": "http://localhost:8080/auth/realms/workshop",
  "aud": ["library-service","account"],
  "sub": "08d3bcaa-5ffd-4b8d-909e-bb567881384b"
}
```


PAYLOAD

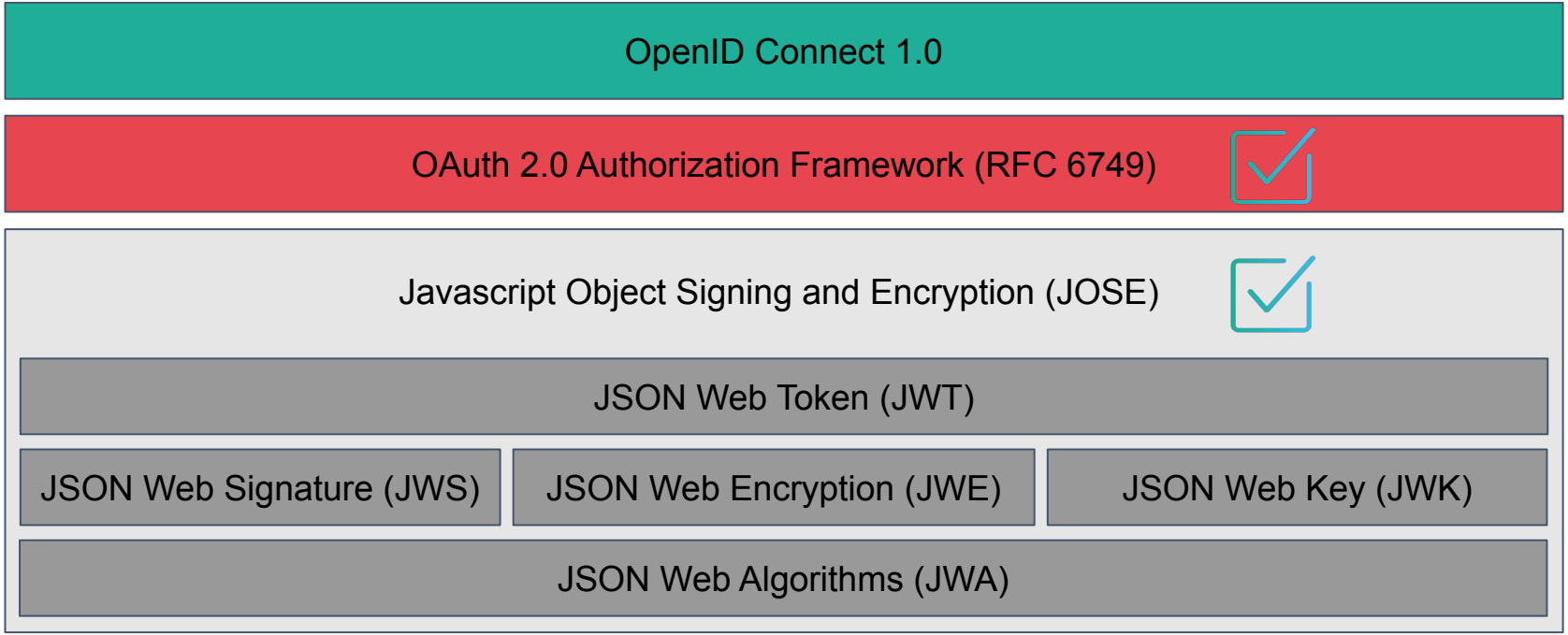
# https://jwt.io






NOVATEC

# OpenID Connect 1.0 Standards Layer



# OpenID Connect 1.0 (OIDC)

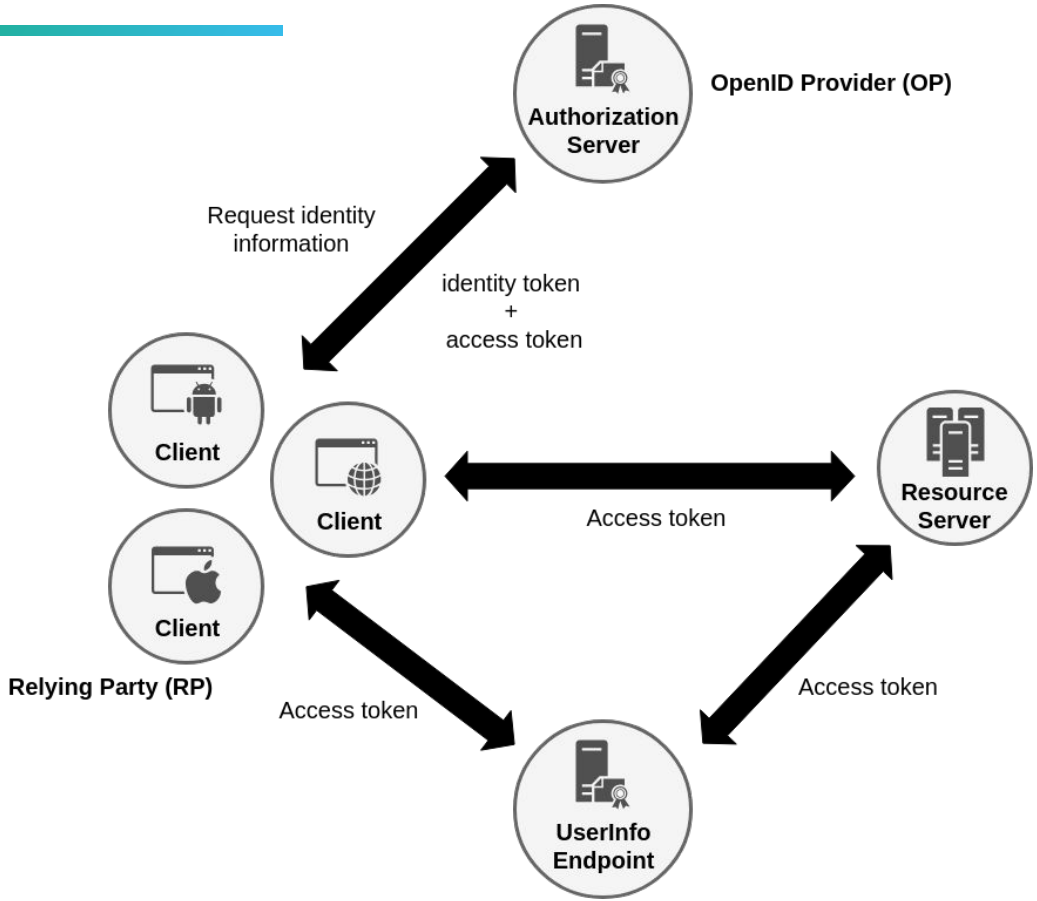
- Based on OAuth 2.0
- Additions:
  - ID Token (JWT format is mandatory)
  - User Info Endpoint (Mandatory)
  - Hybrid Grant Flow (Mandatory)
  - OpenID Provider Configuration Information (Discovery, Optional)

https://openid.net/specs/openid-connect-core-1_0.html
https://openid.net/specs/openid-connect-registration-1_0.html
https://openid.net/specs/openid-connect-discovery-1_0.html

# OpenID Connect 1.0 Claims

| Claim | Required | Description |
|---|---|---|
| iss | ☑ | Issuer Identifier |
| sub | ☑ | Unique Subject Identifier |
| aud | ☑ | Target audience(s) of an ID Token |
| exp | ☑ | Expiration time |
| iat | ☑ | Time at which the JWT was issued |
| auth_time | ☑ | Time of End-User authentication |
| nonce | | Used to associate a client with an ID Token |

# OpenID Connect 1.0 Roles






NOVATEC

# OpenID Connect 1.0 Discovery


```
{
  "issuer": "https://access-me.eu.auth0.com/",
  "authorization_endpoint": "https://access-me.eu.auth0.com/authorize",
  "token_endpoint": "https://access-me.eu.auth0.com/oauth/token",
  "userinfo_endpoint": "https://access-me.eu.auth0.com/userinfo",
  "jwks_uri": "https://access-me.eu.auth0.com/.well-known/jwks.json",
  "scopes_supported": [
    "openid",
    "profile"
  ],...
}
```


https://access-me.eu.auth0.com/.well-known/openid-configuration

# OpenID Connect 1.0 User Info Endpoint

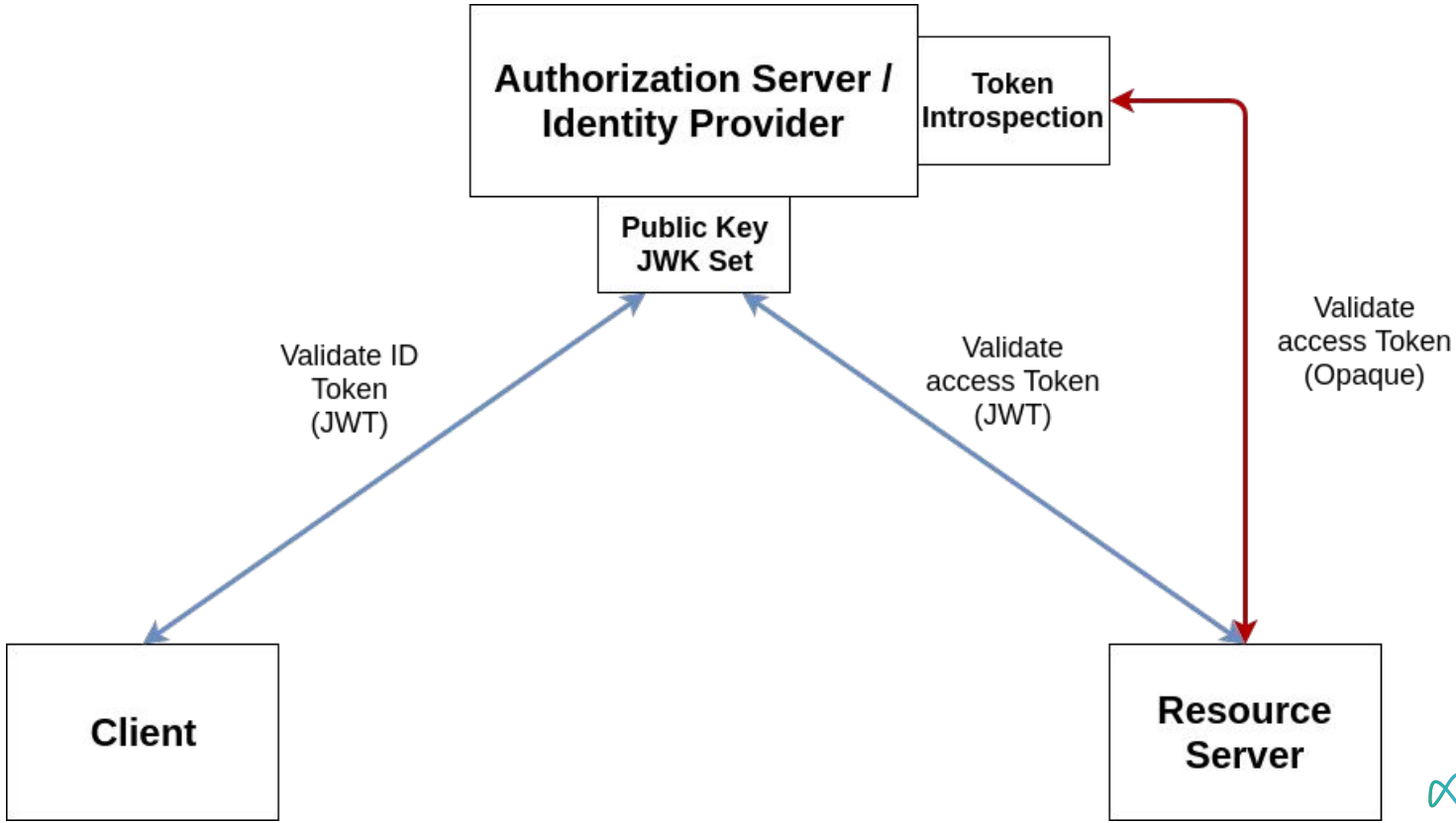
```
GET /userinfo HTTP/1.1
    Host: access-me.eu.auth0.com
    Authorization: Bearer eyJhbGciOiJSUzI1N...
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
   "email": "user@example.com",
   "email_verified": true,
   "sub": "auth0|5bc44fceb144eb0173391741"
}
```

# OpenID Connect 1.0: Access Token Types

| JWT Token (Self-contained) | Opaque Token (Reference) |
| --- | --- |
| ☑ Offline-Validation (Signature/Expiration) | ! Validation call to introspection-endpoint |
| ☑ Contains all required information | ! Additional call to get required information |
| ☑ Protocol agnostic | ! Bound to Http |
| ! Cannot be revoked | ☑ May be revoked |
| ! Mandatory for Id Tokens | ! Must not be used for Id Tokens |
| May be used for Access Tokens | May be used for Access Tokens |

# Token Validation






NOVATEC

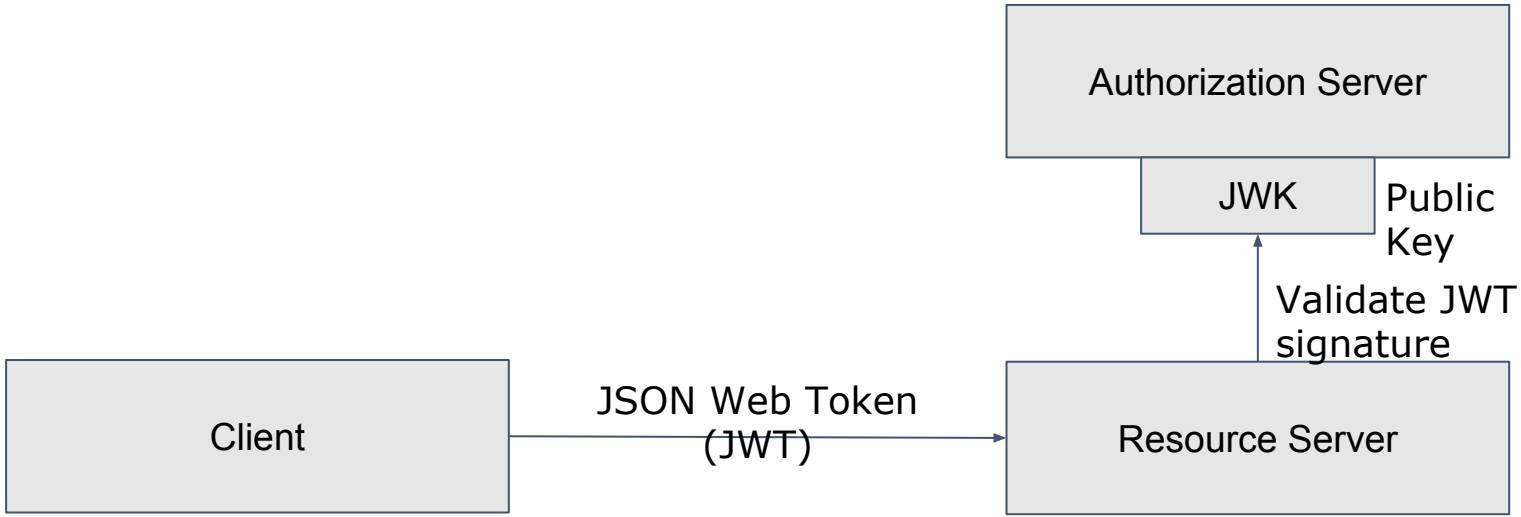
# OpenID Connect Identity Providers

- RedHat/JBoss Keycloak (https://www.keycloak.org)
- Auth0 (https://auth0.com)
- Okta (https://www.okta.com)
- ForgeRock (https://www.forgerock.com/platform/identity-management)
- CloudFoundry UAA (https://github.com/cloudfoundry/uaa)
- PingFederate (https://www.pingidentity.com/en/platform/single-sign-on/sso-overview.html)
- Azure Active Directory (https://azure.microsoft.com/en-us/services/active-directory)
- ...

See: https://openid.net/developers/certified/#OPServices

# OpenID Connect on the Server side (Resource Server)

# Authentication in a single Resource server




```
GET / HTTP/1.1
Host: localhost:8080
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N...
```

# OpenID Connect on the Client side

# OAuth 2.1 Authorization Grant Types

| Resource Owner | Client Type | Authorization Grant | Refresh Token |
|---|---|---|---|
| ✓ | Web Client (Confidential) | Authorization Code + PKCE | ✓ |
| ✓ | Mobile Client (Public) | Authorization Code + PKCE | ✓ |
| ✓ | SPA Client (Public) | Authorization Code + PKCE | ✓ |
| ✗ | Resource Owner = Client (Confidential) | Client Credentials | ✗ |

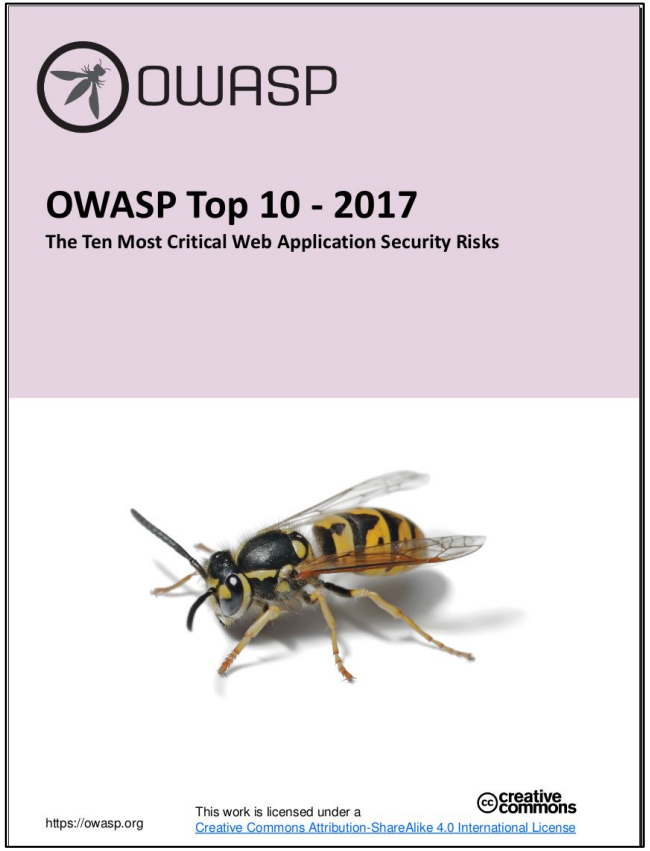
https://datatracker.ietf.org/doc/draft-ietf-oauth-security-topics
https://datatracker.ietf.org/doc/draft-parecki-oauth-v2-1

# OpenID Connect Libraries

- oidc-client (Javascript) https://github.com/IdentityModel/oidc-client-js
- angular-oauth2-oidc (Typescript) https://github.com/manfredsteyer/angular-oauth2-oidc
- angular-auth-oidc-client (Typescript) https://github.com/damienbod/angular-auth-oidc-client
- IdentityModel.OidcClient (C#/.Net) https://github.com/IdentityModel/IdentityModel.OidcClient
- Nimbus OAuth 2.0 SDK (Java) https://connect2id.com/products/nimbus-oauth-openid-connect-sdk
- OIDC RP library (Python) https://github.com/openid/JWTConnect-Python-OidcRP
- ...

See: https://openid.net/developers/certified/#OPServices

# A5: Broken Access Control






NOVATEC

# Authorization - What can I access?

HTTP 403 - FORBIDDEN

# Authorization (Static Roles)

```
public class UserBoundaryService {

    @PreAuthorize("hasRole('ADMIN')")
    public List<User> findAllUsers() {...}
}
```

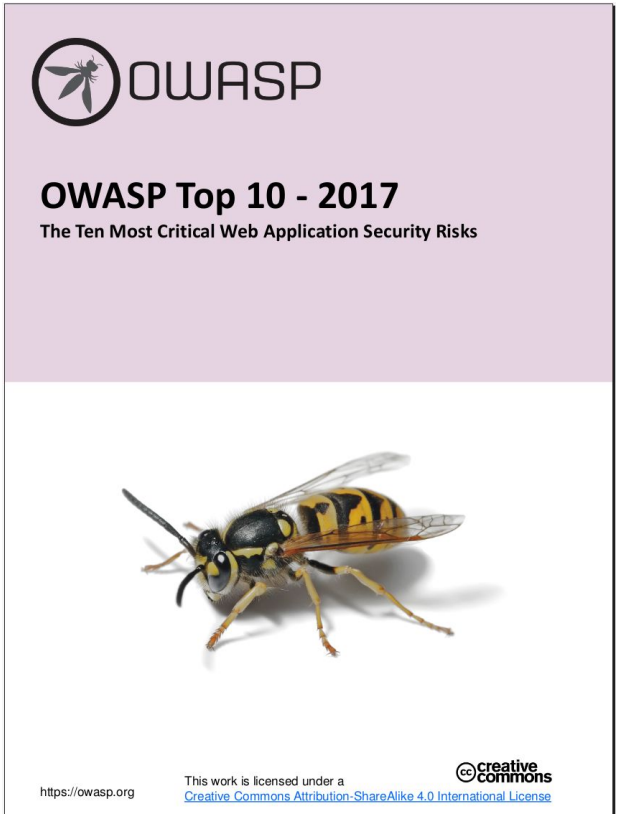
# Authorization (Dynamic Permissions)

```
public class TaskBoundaryService {

    @PreAuthorize("hasPermission(#taskId, 'TASK', 'WRITE')")
    public Task findTask(UUID taskId) {...}
}
```

# Automated Authorization Testing

```
public class AuthorizationIntegrationTest {

    @WithMockUser(roles = "ADMIN")
    @Test
    public void verifyFindAllUsersAuthorized() {...}

    @WithMockUser(roles = "USER")
    @Test(expected = AccessDeniedException.class)
    public void verifyFindAllUsersUnauthorized() {...}
}
```

# A1: Injection






NOVATEC

# SQL Injection

```
SELECT balance FROM user WHERE username =
            + "'" + request.getParameter('userName')
             + "'";
```

→ Parameter *userName*: Tom

```
SELECT balance FROM user WHERE username = 'Tom';
```

→ Parameter userName: Tom' or 1=1; --

```
SELECT balance FROM user WHERE username = 'Tom' or 1=1
```

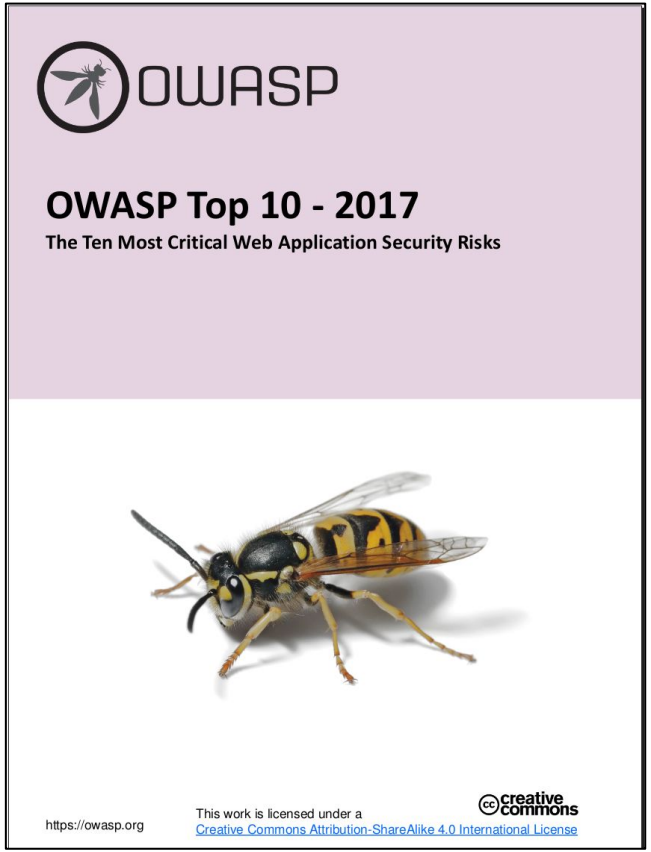
# SQL Injection Defensive - Use Input Validation

```
@Entity
public class Person extends AbstractPersistable<Long> {

    @NotNull
    @Pattern(regexp = "^[A-Za-z0-9- ]{1,30}$")
    private String lastName;

    @NotNull
    @Enumerated(EnumType.STRING)
    private GenderEnum gender;
    ...
}
```

# SQL Injection Defensive - Prepared Statements

```
@Query(
"select u from User u where u.username = "
+ " :username and u.password = :password")
User findByUsernameAndPassword(
     @Param("username") String username,
     @Param("password") String password);
```

# A9: Using Components With Known Vulnerabilities






NOVATEC

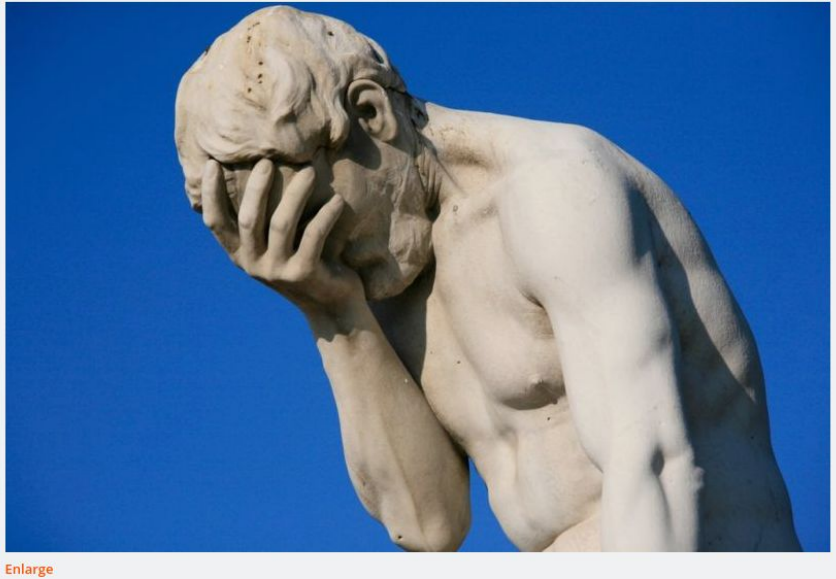
# Equifax Struts Vulnerability

BIZ & IT —

## Failure to patch two-month-old bug led to massive Equifax breach

Critical Apache Struts bug was fixed in March. In May, it bit ~143 million US consumers.

DAN GOODIN - 9/14/2017, 5:12 AM



wikimedia commons/alex t. tromos

Enlarge

237

The Equifax breach that exposed sensitive data for as many as 143 million US consumers was accomplished by exploiting a Web application vulnerability that had been patched more than two months earlier, officials with the credit reporting service said Thursday.



NOVATEC

# OWASP Dependency Check

- Detects Vulnerabilities in Project Dependencies
- Supports Java and .NET applications
- Experimental: Python, Ruby, PHP, Node.js
- Command line, Ant, Maven, Gradle, Jenkins, SBT

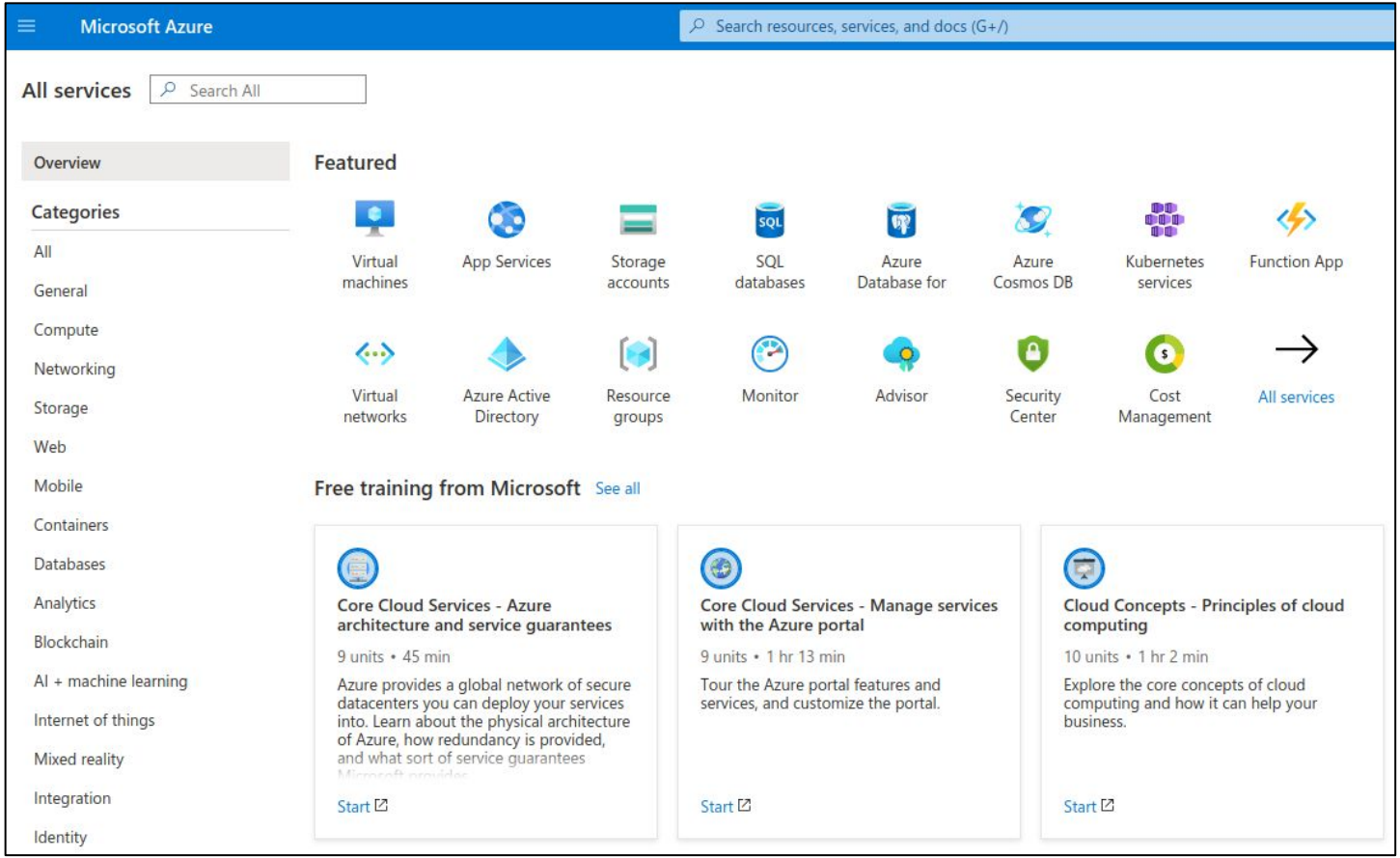
https://github.com/jeremylong/DependencyCheck

# What about the Cloud?

# “Good old friends” + more

- CSRF, XSS, SQL Injection, ...
- Distributed DoS
- Economic DoS

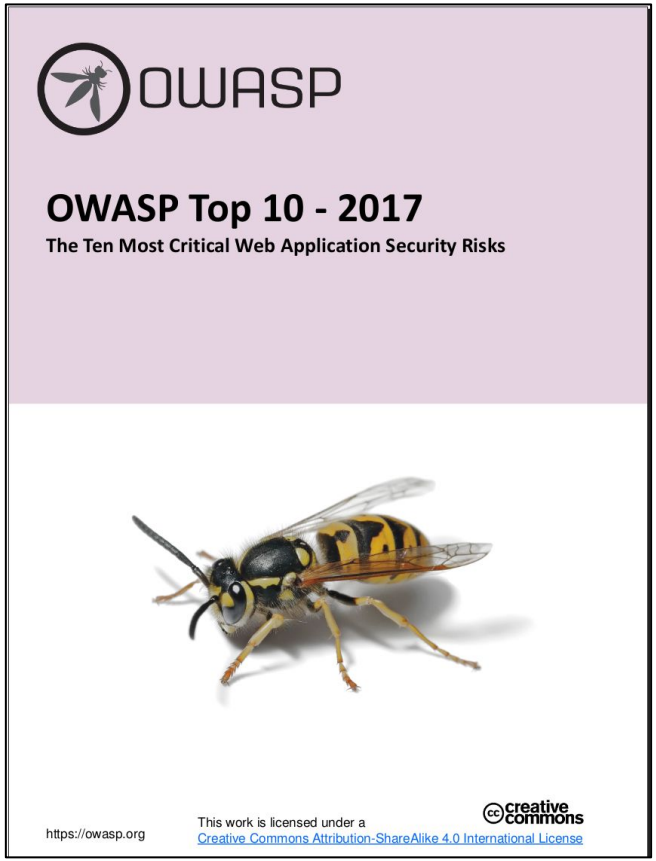
# Weak Passwords






NOVATEC

# A3: Sensitive Data Exposure






NOVATEC

# EU General Data Protection Regulation (GDPR)

“...should adopt internal policies and implement measures which meet in particular the principles of **data protection by design** and **data protection by default**”

http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679

# Key Management

# Key Management

- Azure Key Vault
- AWS Secrets Manager
- Google Cloud HSM
- CloudFoundry CredHub
- Hashicorp Vault



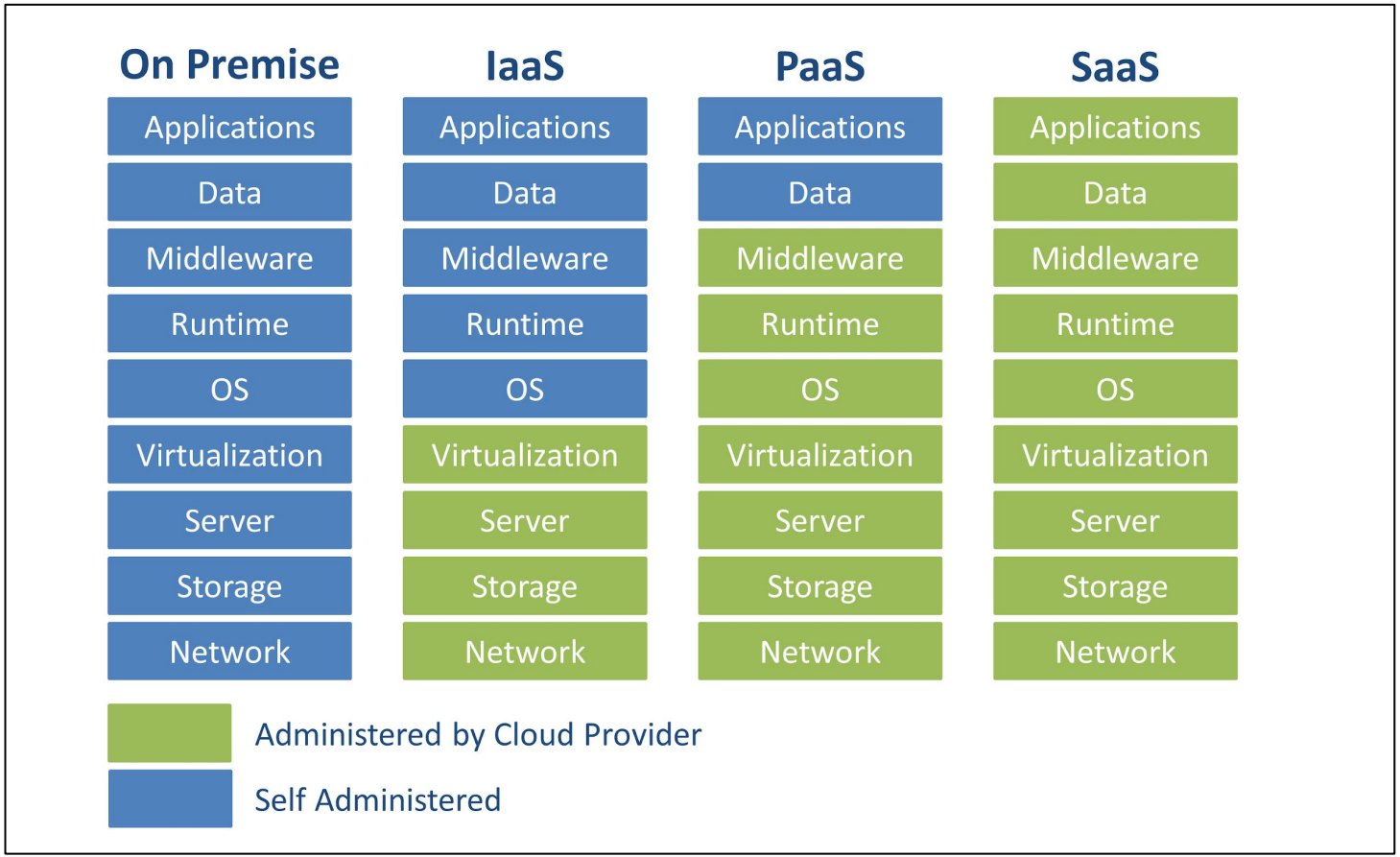


# Rotate, Repair, Repave

“What if every server inside my data center had a maximum lifetime of two hours? This approach would frustrate malware writers...”
Justin Smith (Chief Security Officer at Pivotal)

https://thenewstack.io/cloud-foundrys-approach-security-rotate-repair-repave

# Cloud Service Models



# Spring Security 5 Basics

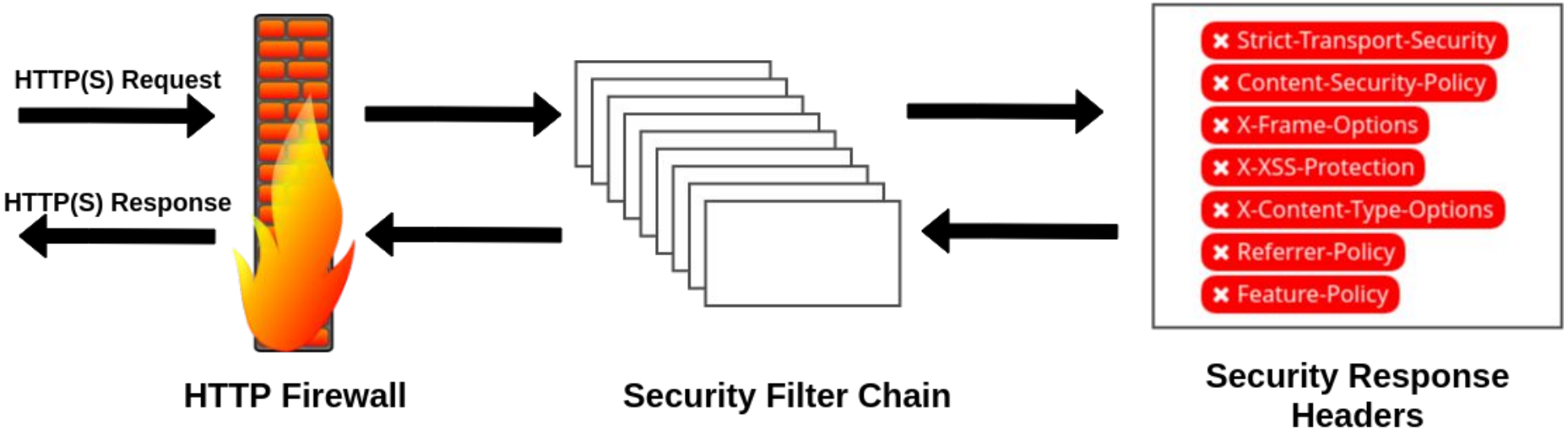
# Spring Security 5

- Authentication & Authorization
- Password Encoding
- Support for Servlet & Reactive Web Applications
- Support for OAuth 2.0
- Support for OpenID Connect 1.0, JWT and JOSE (JWS/JWE/JWK)
- Testing Support for Auth/Authz/JWT

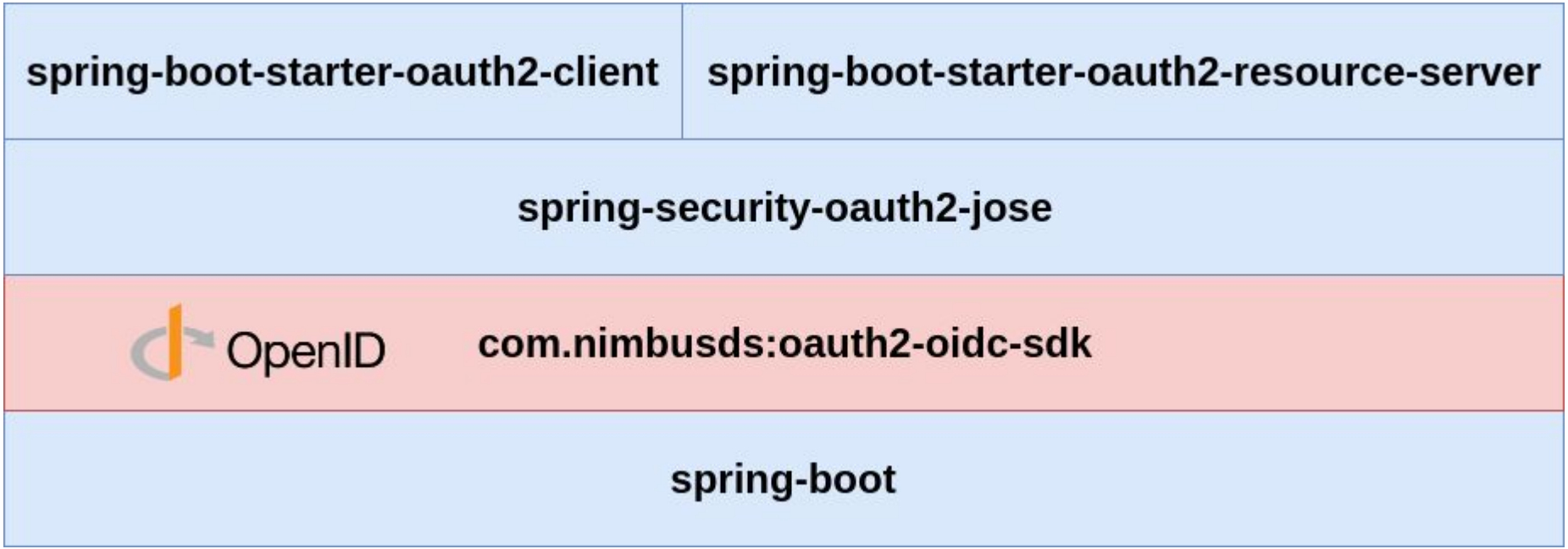
# Spring Security 5 - Secure by Default

- Authentication required for all HTTP endpoints
- Session Fixation Protection
- Session Cookie (HttpOnly, Secure)
- CSRF Protection
- Security Response Header

# Spring Security High Level View



# New Spring Security 5.x OIDC Technology Stack




https://connect2id.com/products/nimbus-oauth-openid-connect-sdk

# Practice

https://andifalk.gitbook.io/cloud-security-workshop

https://github.com/andifalk/cloud-security-workshop

**Andreas Falk**
Managing Consultant

Mobil: +49 151 46146778
E-Mail: andreas.falk@novatec-gmbh.de

**Novatec Consulting GmbH**
Dieselstraße 18/1
D-70771 Leinfelden-Echterdingen

T. +49 711 22040-700
info@novatec-gmbh.de
www.novatec-gmbh.de