Hackathon Kosice – Gigabit Challenge

Phillip Möller, Thea John
23.11.2024

Who are we?

Phillip Möller
Software Engineer Lead
Phillip.Moeller@t-systems.com

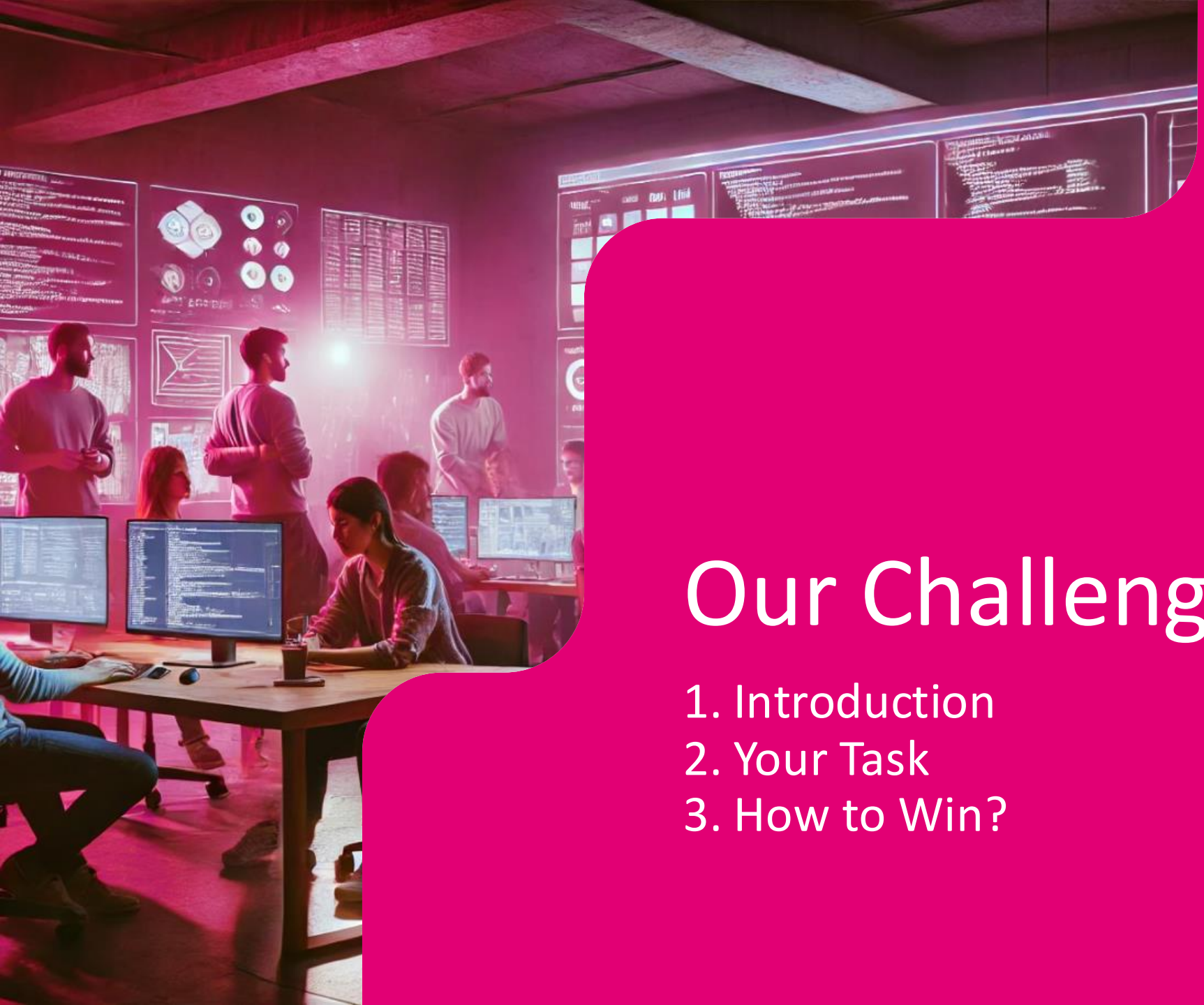
Thea John
Software Engineer Lead
Thea.John@telekom.de

Gigabit – About us

WELCOME  TO GIGABIT!

A HUGE OPPORTUNITY FOR TELEKOM AND GERMANY

WE WON'T STOP UNTIL EVERYONE IS CONNECTED!



Our Challenge

1. Introduction
2. Your Task
3. How to Win?

Vulnerabilities

Definition

A weaknesses in software, hardware, or operational processes that can be exploited by a threat actor to gain unauthorized access, cause disruptions, or compromise data integrity

Occurrences

- Application
- Operating System
- Dependencies

Examples

SQL Injection: Attackers exploit improperly sanitized user input to execute malicious SQL queries.

Buffer Overflow: Occurs when software writes more data to a memory buffer than it can hold, potentially allowing attackers to overwrite adjacent memory and execute arbitrary code.

Finding Vulnerabilities

01

CVE

A CVE (**Common Vulnerabilities and Exposures**) is a standardized identifier assigned to a publicly known cybersecurity vulnerability or exposure.

02

OWASP

Open Web Application Security Project provides resources, tools, standards, and best practices to help build secure applications and defend against potential threats.

03

CWE

Common Weakness Enumeration is a standardized system for categorizing and describing software weaknesses. Unlike **CVE** (which focuses on specific vulnerabilities), CWE focuses on the underlying **weaknesses** in software design, architecture, or implementation that can lead to vulnerabilities.

04

Scanner Tools

The **Gitlab SAST (Static Application Security Testing) analyzer** performs cross-function and cross-file taint analysis to detect complex vulnerabilities.

The **Gemnasium Dependency Scanner** is a tool designed to analyze an application's dependencies for known vulnerabilities (CVEs).

Needs triage Detected · 1 day ago in pipeline 36541918

Status Needs triage

Improper neutralization of special elements in data query logic

Description

Untrusted user input in findOne() function can result in NoSQL Injection.

Severity: Critical

Project:

Tool: SAST

Scanner: Semgrep

Location

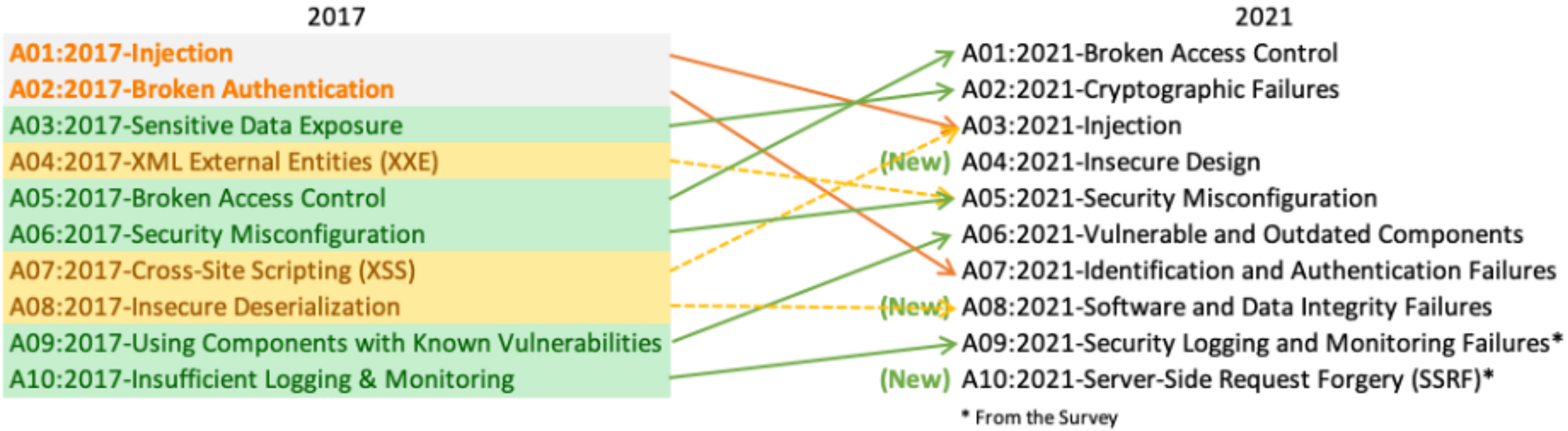
File: routes/address.ts:18

```
18    const address = await AddressModel.findOne({ where: { id: req.params.id, UserId: req.body.UserId } })
```

Identifiers

- A1:2017 - Injection
- nodejs_scan.javascript-database-rule-node_nosqli_injection
- NodeJS Scan ID javascript-database-rule-node_nosqli_injection
- CWE-943
- A03:2021 - Injection

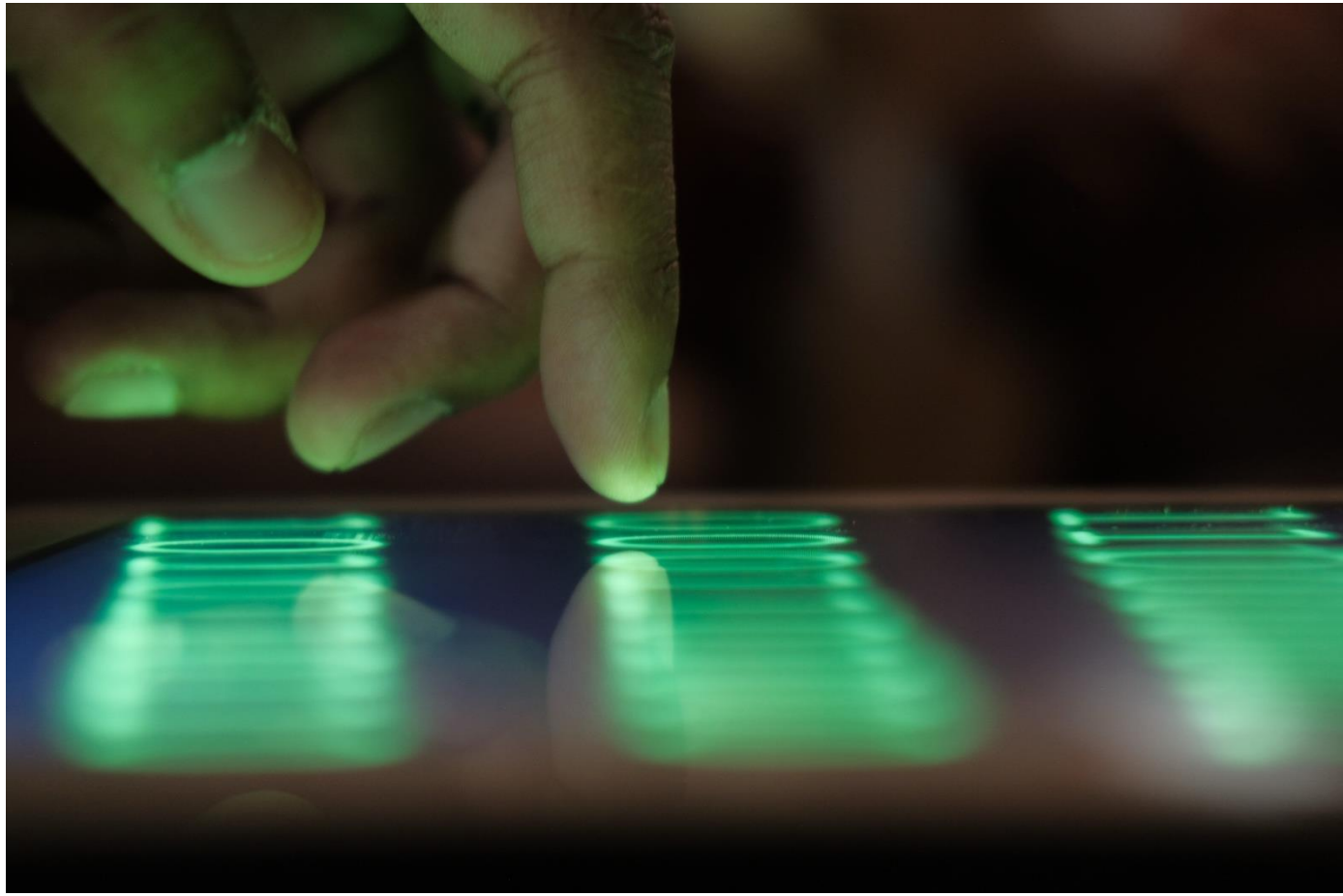
Top 10 Web Application Security Risks



- Source: https://owasp.org/www-project-top-ten/

Penetration Testing

- Penetration Testing, often referred to as pen testing, is a **simulated cyberattack** performed on a computer system, network, application, or other infrastructure to evaluate its security.
- The goal is to **identify vulnerabilities and weaknesses**.



Automated Pen Testing – Your Task

Input

- Service (GitHub Repo)
- List of its Vulnerabilities

Your App

- Automated exploiting of Vulnerabilities
- Recreation of Vulnerabilities

Output

- Your frontend with list of Vulnerabilities that were exploitable
- Live Demo

Recommended Technologies

Backend: Python, Go, TypeScript

Frontend: HTML, JavaScript, CSS, Angular, React, Vue

All Products

Apple Juice (1000ml)

1.99¤

Apple Pomace

0.89¤

Banana Juice (1000ml)

1.99¤

Best Juice Shop Salesman Artwork

5000¤

Carrot Juice (1000ml)

2.99¤

DSOMM & Juice

Winning Criteria

01

Amount of automated exploits multiplied with the Criticality of Exploit

How many weaknesses of our list were you able to exploit with your app? You will get most points for severity “critical”.

02

Reusability of your automated exploits to unknown services

We will give you another service and its list of vulnerabilities. Can your app exploit that new service without big adjustments?

03

Reusability of your automated exploits to other CWEs/CVEs

We will give you more and different vulnerabilities. Can your app also test other vulnerabilities without big adjustments?

04

UX of your UI

Can I check easily on your UI how the exploit was performed? How is the traceability of those exploits? Can you show screenshots or logs of the performed steps?

Notes

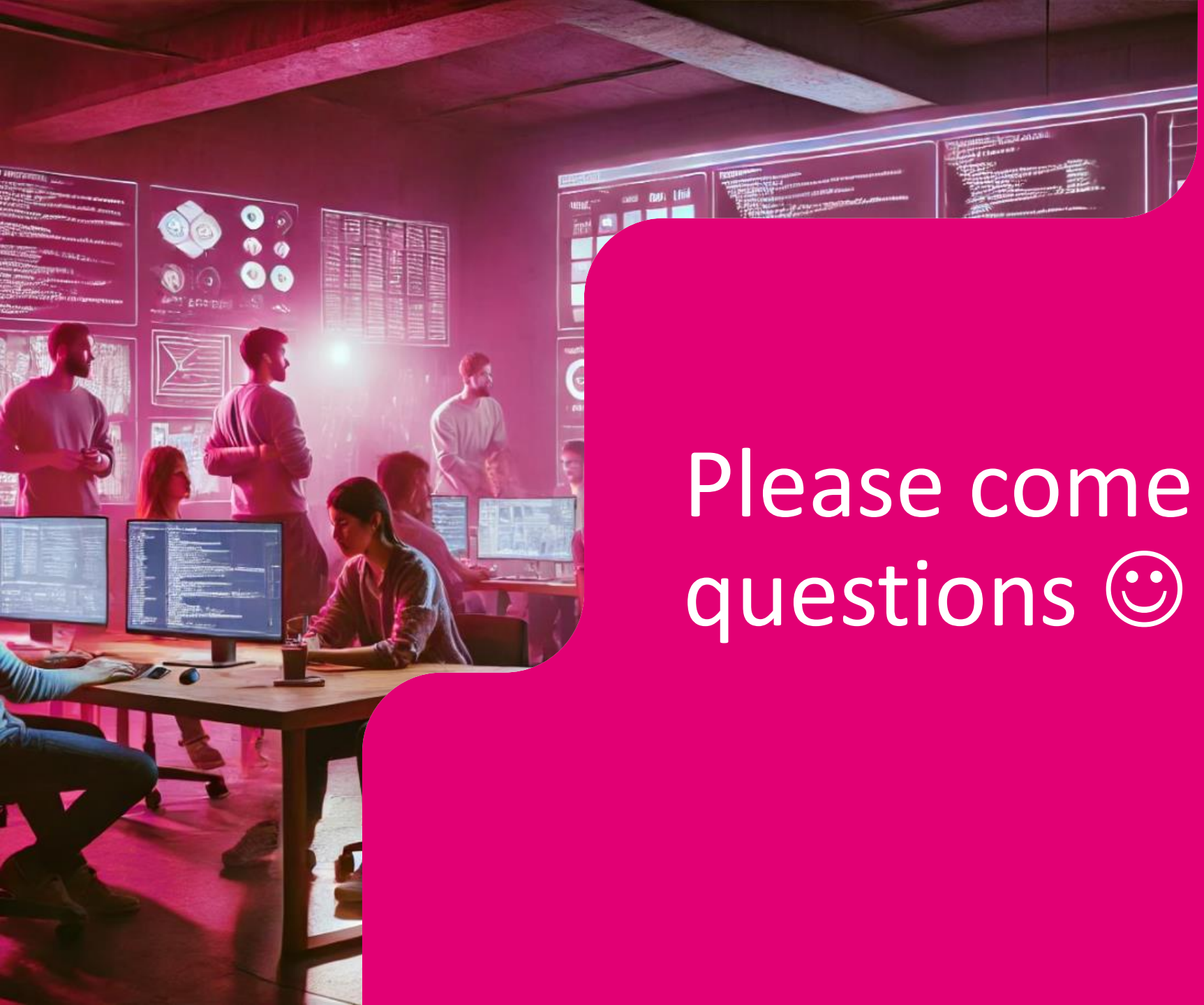
Must Have

- Running Code
- External dependencies need to be documented
- README for Getting Started (how to run your code)

Not Needed

- Running in Container
- Pipeline Setup

Good Luck!



Please come to us for questions ☺