



# Development of an Automated Integration System for SAPUI5 Web Applications

By: Eman Belal

Supervisor: Prof. Dr. Christian Baun

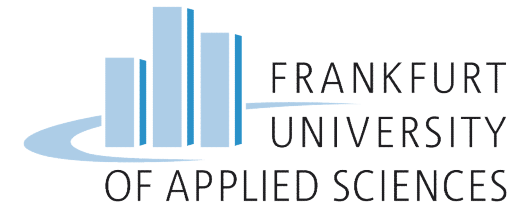
Co-Supervisor: Prof. Dr. Thomas Gabel

PUBLIC

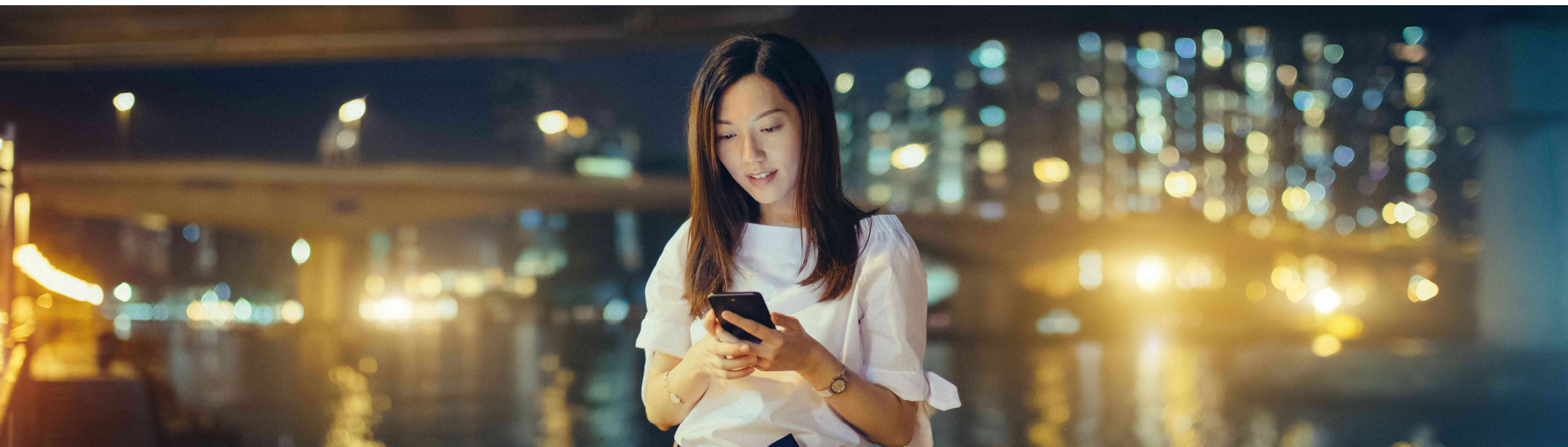


# Agenda

- Introduction
  - Web Applications
  - SAPUI5
- Problem Description
- Goal
- Solution Approach
  - State of the Art
  - Design
  - Implementation
- Evaluation
- Demo
- Summary & Outlook

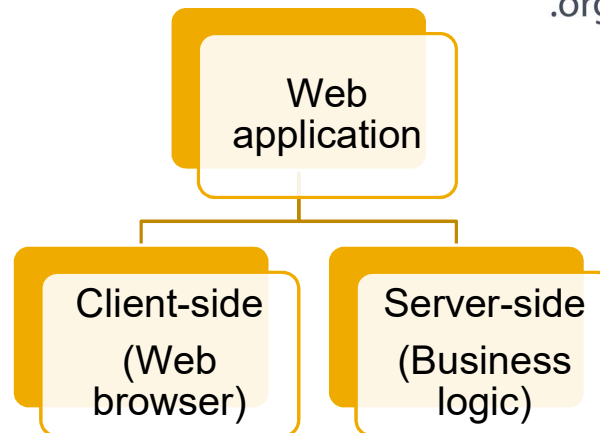


# Introduction

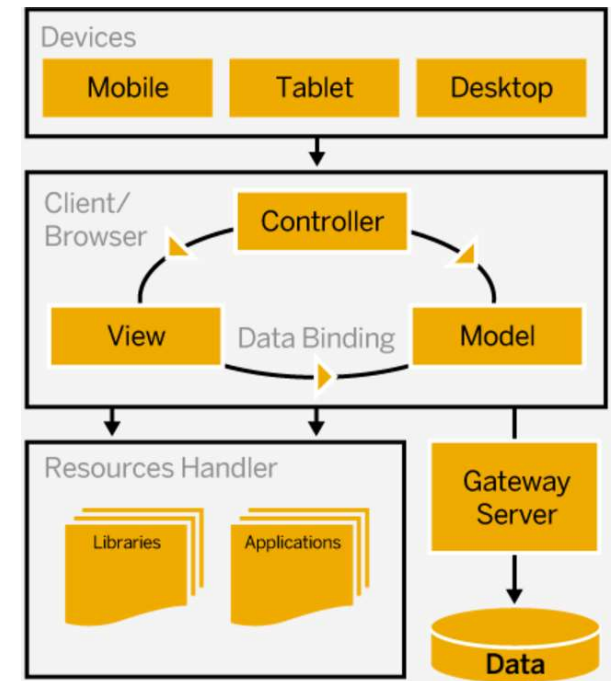


# Web Applications

- Instant easy Usage (web browser)
- High flexibility and large number of open frameworks
- More and more replacing native apps
- JavaScript is commonly used in the development of web applications

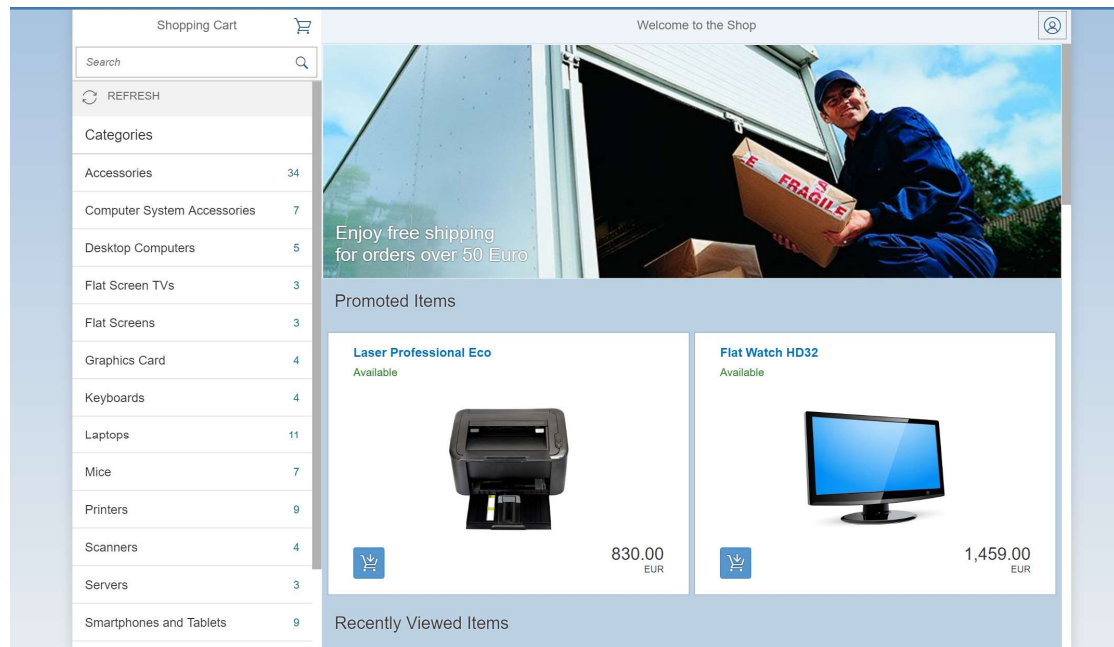


- JavaScript UI technology framework developed by SAP
- Cross platform
- Different UI control libraries
  - ex: sap.m, sap.ui.commons...
- Model-View-Controller (MVC) design pattern
- Supports Standard JavaScript libraries
  - ex: JQuery

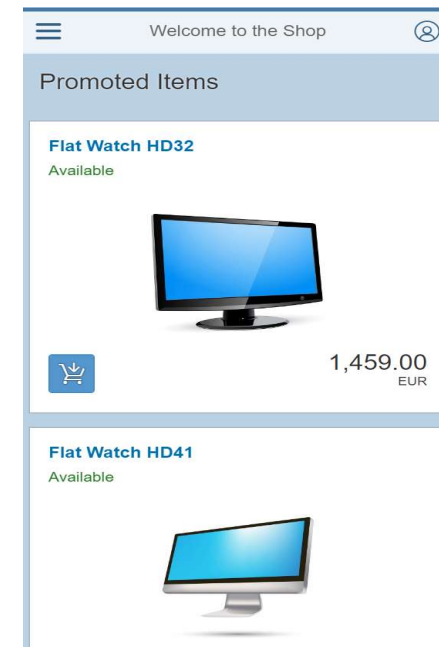




# SAPUI5 App Example



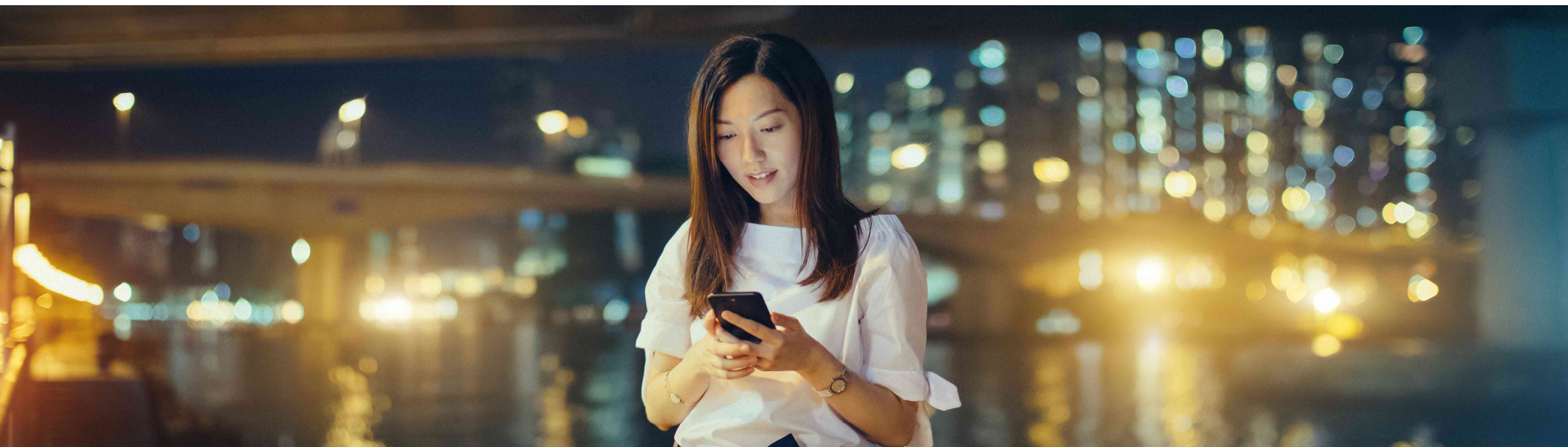
Desktop



Mobile

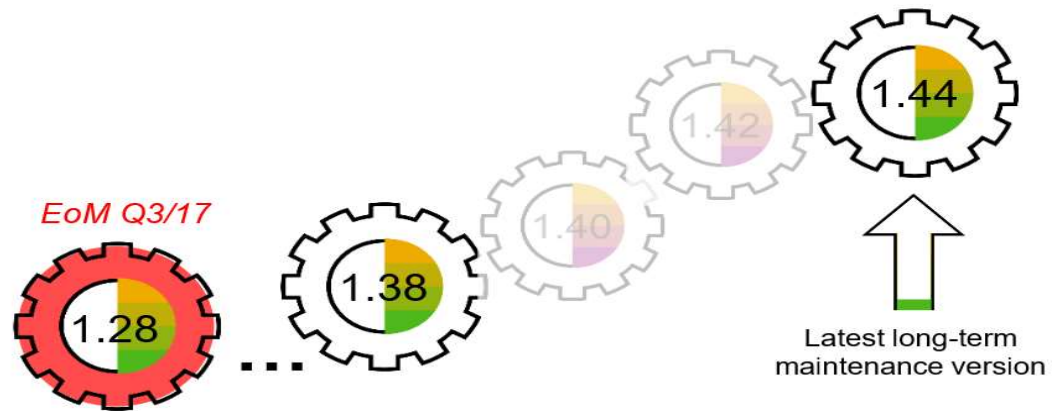
**WHY?**

# Problem Description



# Software Maintenance

- Frequent updates and versions for SAPUI5 released by SAP, impacting existing applications
- Potential Benefits:
  - Performance enhancements
  - Bug fixes
  - New features
- Potential Issues:
  - Compatibility
  - Regression effects



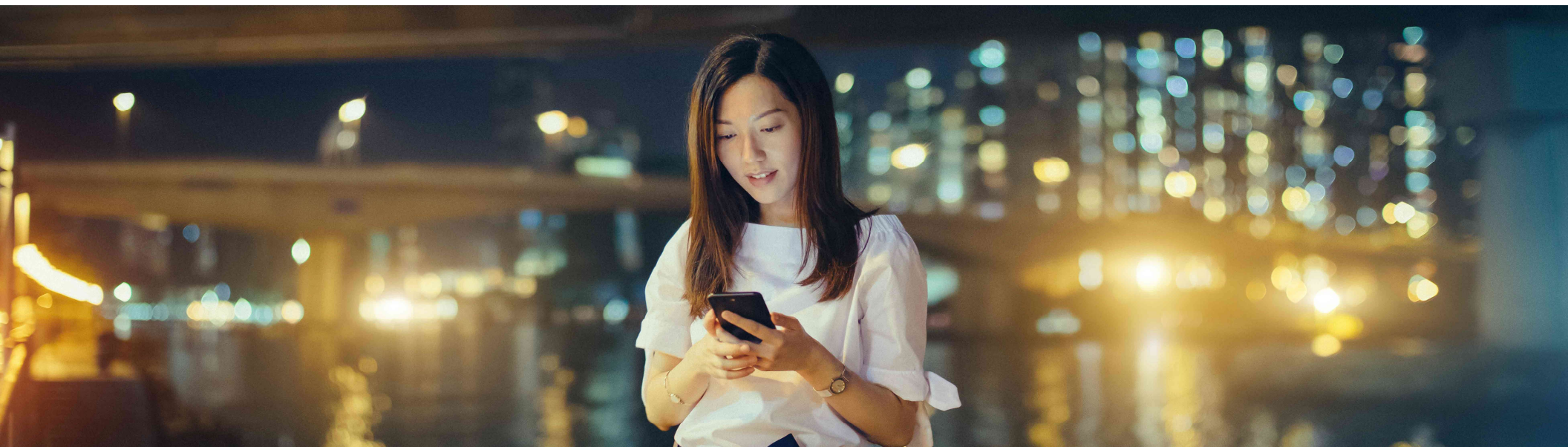


# JavaScript Challenges

Static Analysis of web applications is a challenging problem when it comes to the use of JavaScript Language due to the following features of JavaScript:

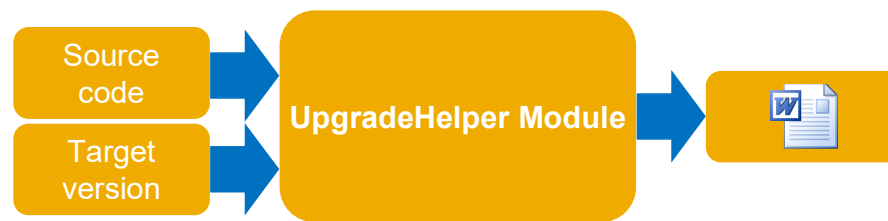
- Dynamic
- Interpreted
- Weakly-typed
- Prototype-based

# Goal



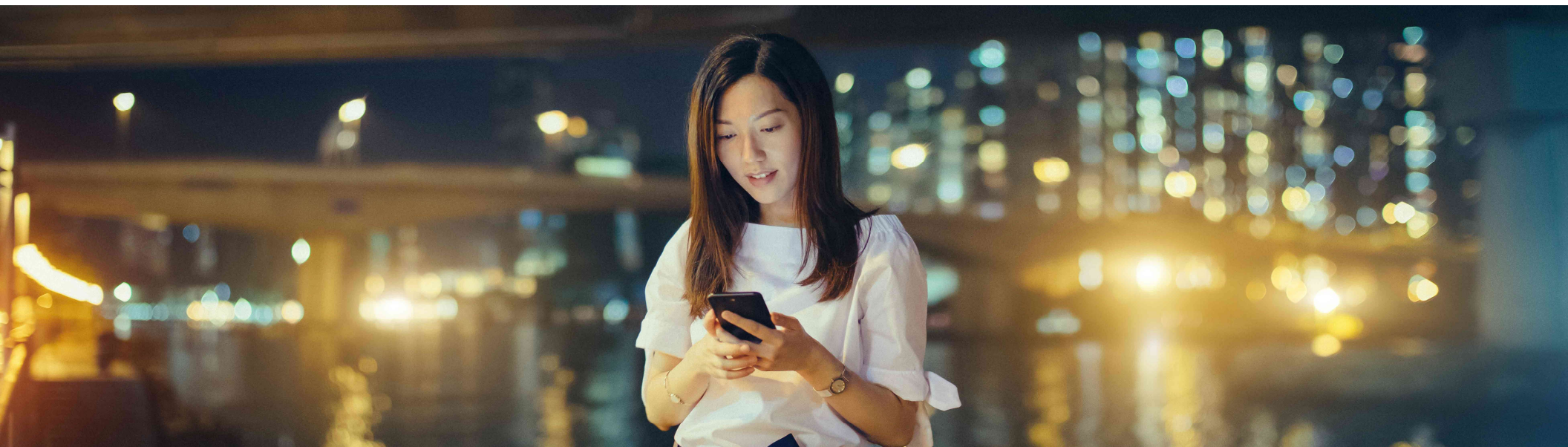
# Goal of the Thesis

- Implementation of a core parser tool for SAPUI5 web applications
- Analyzing the JavaScript source code of the SAPUI5 application
- Checking the upgrade compatibility of SAPUI5 applications
- Parser can be reused in further modules



# HOW?

## **Solution Approach**



# Approach

## 1. Check State of the Art

- Static analysis approach
- Current existing solutions

## 2. Design

- Requirements analysis
- Solution architecture

## 3. Implementation

- PoC
- Flow chart
- Solution implementation



# 1. Check State of the Art

## Current Existing Solutions

	Static Analysis	Dynamic Analysis
<b>Operation</b>	Without running the program	Program should be running
<b>Execution path</b>	All execution paths	One or more execution paths
<b>Precision</b>	Lower precision	Higher precision
<b>Usage</b>	Ensuring code quality	Detecting bugs
<b>Speed</b>	Faster	slower
<b>Example</b>	ESLint, JSLint	Code testing techniques

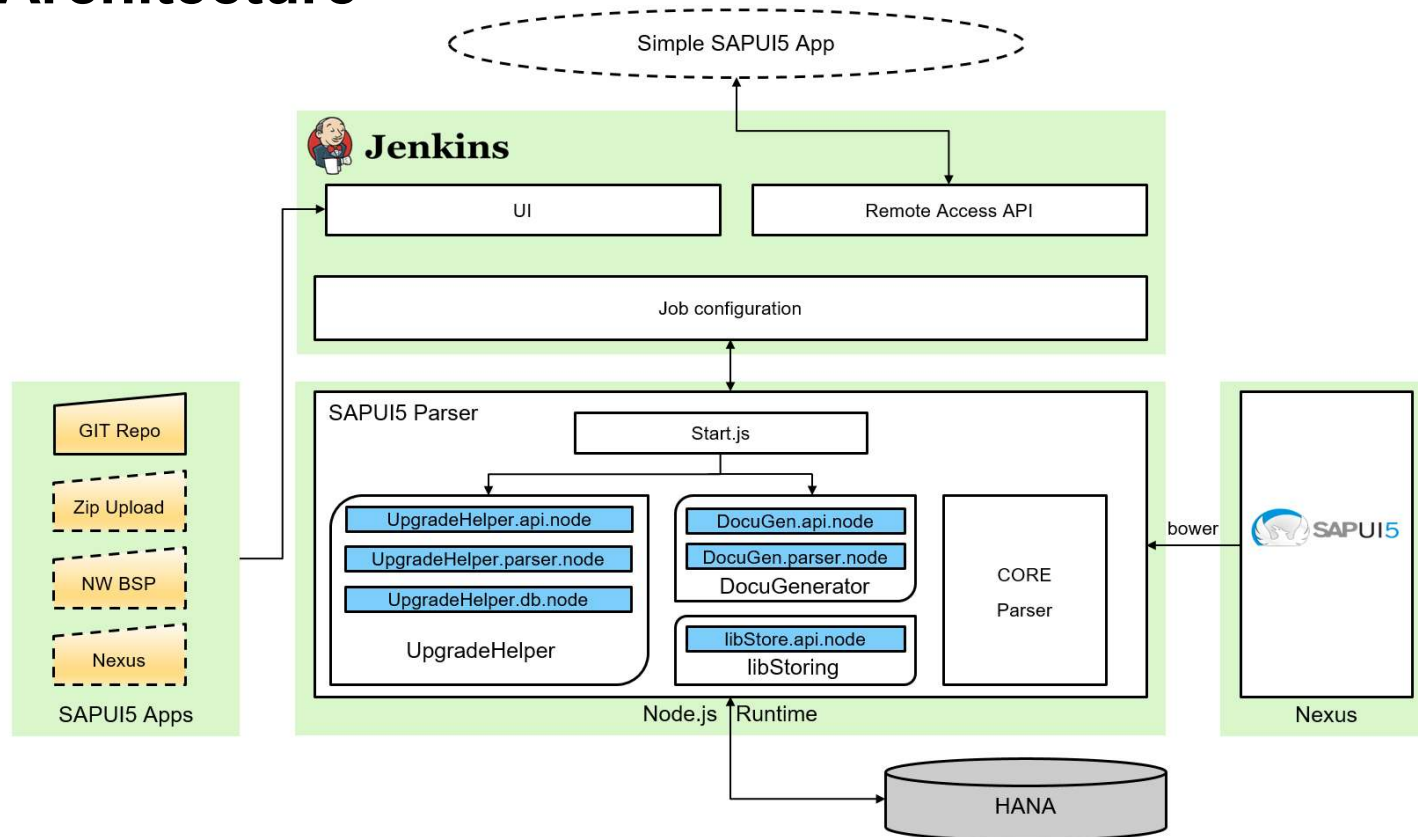
## 2. Design

### Requirements Analysis

1. Source code analysis of any SAPUI5 based application.
2. Source code analysis of the target SAPUI5 library version to be upgraded to
3. List of deprecated standard UI controls and properties within the app
4. Amount of changes made between current and target version of the SAPUI5 library.
5. Output report document showing the upgrade impact.

## 2. Design

### Solution Architecture

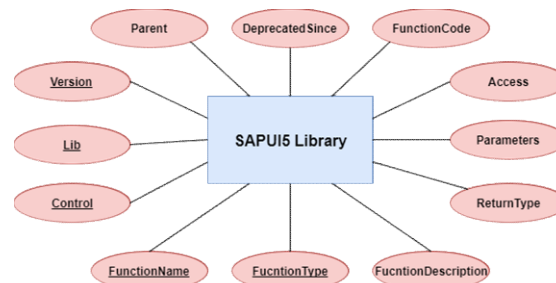


# 3. Implementation

## Indexing & Maintaining SAPUI5 library (LibStoring Module)

- SAPUI5 library is downloaded via „**Bower**“ into „bower\_components“ folder
- Model definition for SAPUI5 library
- Maintaining the library on SAP HANA database

```
1  {  
2    "name": "sapui5Parser",  
3    "version": "0.0.0",  
4    "dependencies": {  
5      "sapui5-1.44.8": "http://nexusrel.wdf.sap.corp:8081/nexus/service/local/repositories/build.milestones.hosted/content/com/sap/ui5/  
6      "sapui5-sdk-1.38.9": "http://nexusrel.wdf.sap.corp:8081/nexus/service/local/repositories/build.milestones.hosted/content/com/sap/  
7      "sapui5-1.28.9": "http://nexusrel.wdf.sap.corp:8081/nexus/service/local/repositories/build.milestones.hosted/content/com/sap/ui5/  
8    },  
9    "private": true  
10 }
```



SAPUI5Lib	
Version	NVARCHAR(7) PK
Lib	NVARCHAR(256) PK
Control	NVARCHAR (256) PK
FunctionName	NVARCHAR (256) PK
FunctionType	NVARCHAR(256) PK
FunctionDescription	NVARCHAR(5000)
ReturnType	NVARCHAR(256)
Parameters	NVARCHAR(1024)
Access	NVARCHAR(256)
FunctionCode	NCLOB
DeprecatedSince	NVARCHAR(1024)
Parent	NVARCHAR(256)

## 3. Implementation

### SAPUI5 App Parser

- General information about the SAPUI5 app
- Information about used views and controllers
- Information about used standard UI controls and their properties
- Occurrence of each UI control within the application



## 3. Implementation

### Upgrade Helper Module

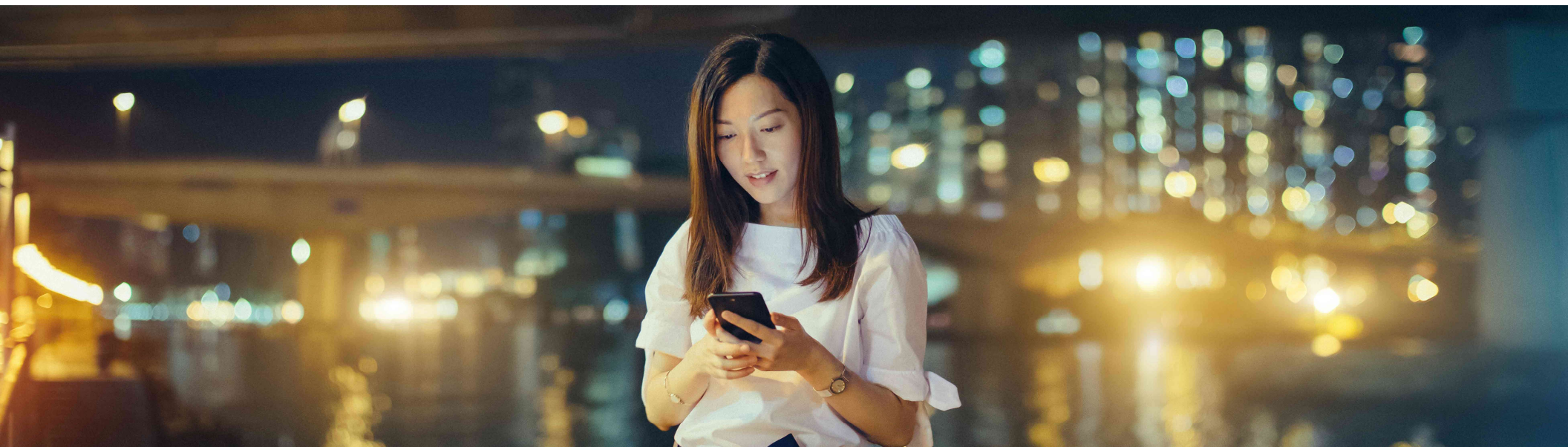
- Checks Deprecated standard controls and properties
- Compares between current and target SAPUI5 versions
- Percentage of the amount of changes made between both versions

$$\text{Changes \%} = \frac{\text{Total number of changed lines}}{\text{Total number of old version lines}} \times 100$$

- Risk factor for each used control

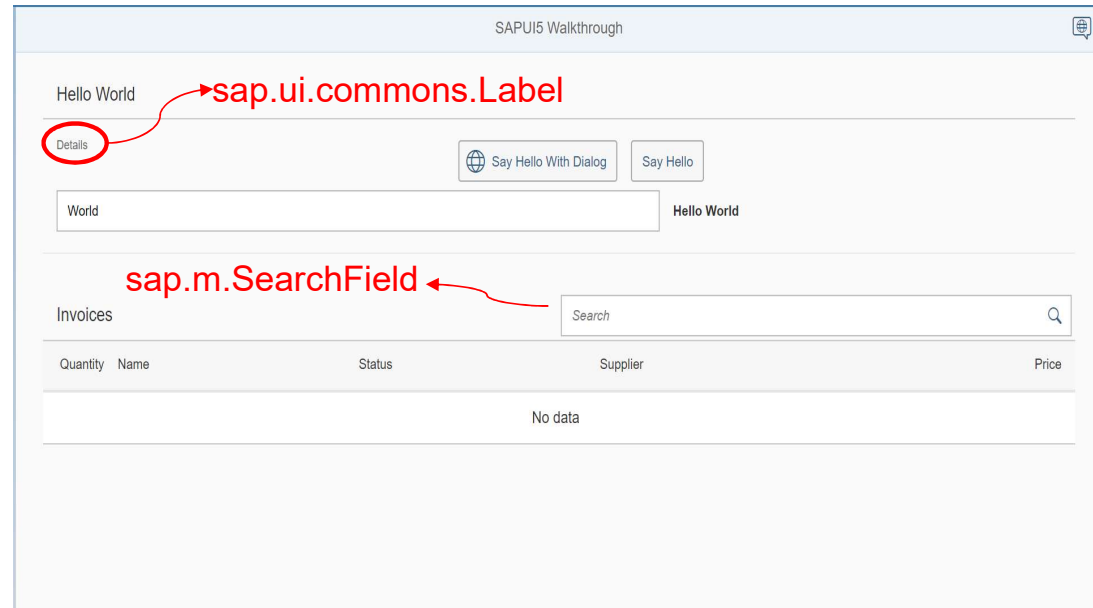
$$\text{Risk Factor} = \left( \frac{\text{Changes \%}}{10} \right) \times (\text{Control occurrence})$$

# Evaluation



# Evaluation

- One of the SAPUI5 applications analyzed by the tool:
  - SAPUI5 version: **1.30.0**
  - Target version: **1.44.8**
  - Deprecated UI control: **sap.ui.commons.Label**
  - Deprecated property: **selectOnFocus** for sap.m.SearchField
- The higher the risk factor, the more chance of getting compatibility issues



```
* @author SAP SE
* @version 1.30.0
*
* @constructor
* @public
* @alias sap.ui.commons.Label
* @ui5-metamodel This control/element also will be described in the UI5 (legacy) designtime metamodel
*/
var Label = Control.extend("sap.ui.commons.Label", /** @lends sap.ui.commons.Label.prototype */ { metadata : {
```

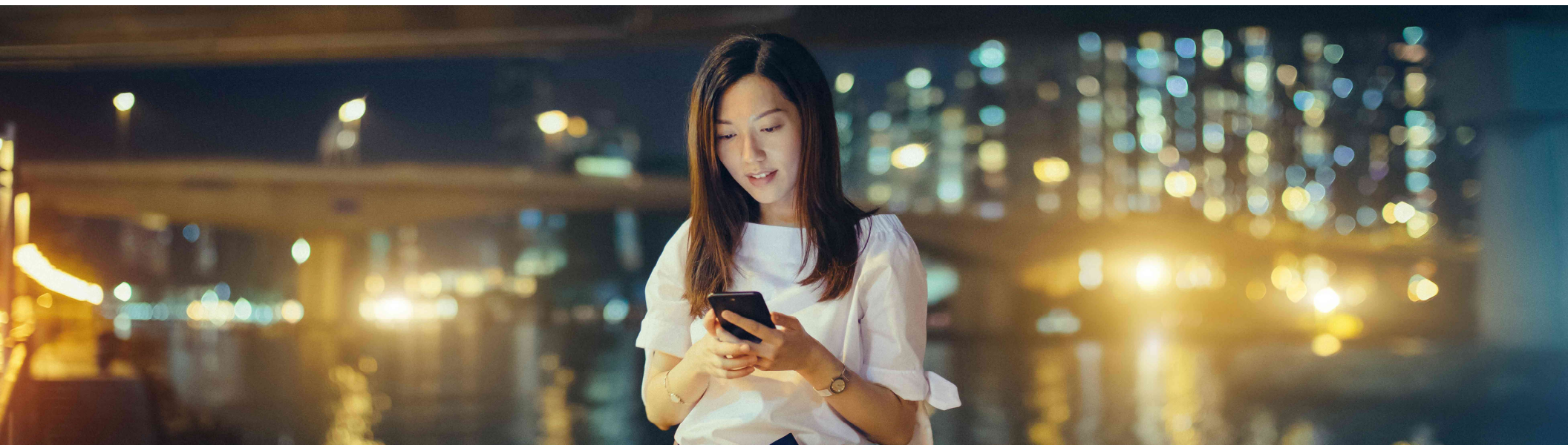
**1.30.0**



```
* @author SAP SE
* @version 1.44.7
*
* @constructor
* @public
* @deprecated Since version 1.38. Instead, use the <code>sap.m.Label</code> control.
* @alias sap.ui.commons.Label
* @ui5-metamodel This control/element also will be described in the UI5 (legacy) designtime metamodel
*/
var Label = Control.extend("sap.ui.commons.Label", /** @lends sap.ui.commons.Label.prototype */ { metadata : {
```

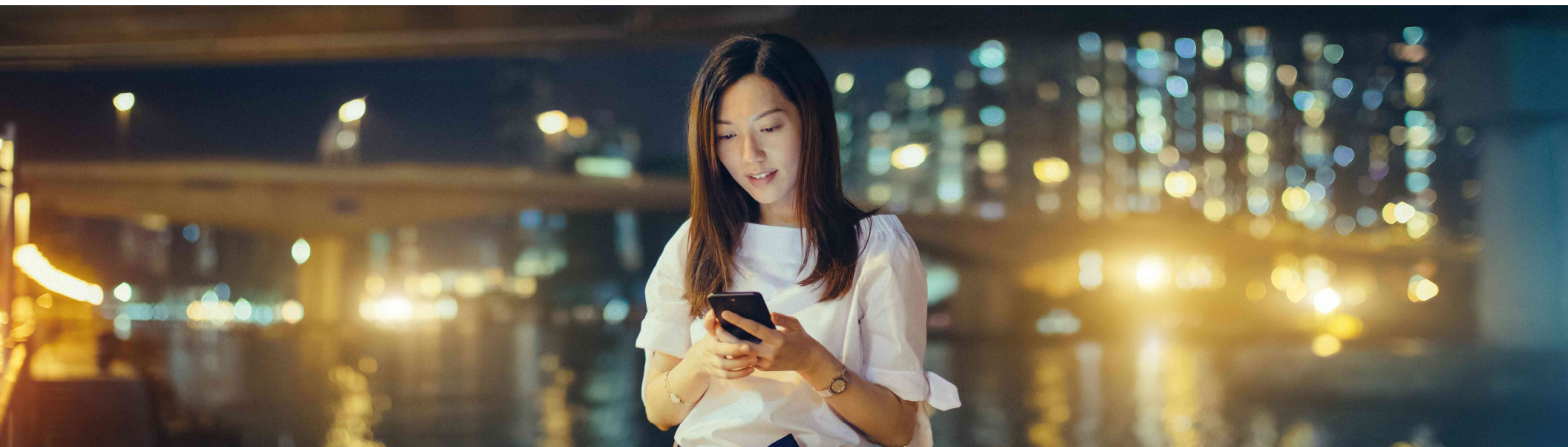
**1.44.8**

# DEMO





# Summary & Outlook





# Summary

- Prototype of an automated integration system for SAPUI5 web applications
- User assistance in knowing the risk impact on upgrading their SAPUI5 apps by:
  - List of deprecated standard controls and properties
  - The amount of changes made to the control between the two versions
  - Analysis of the occurrence of each control within the given app
  - Risk factor for each control used within the app

# Outlook – Room for Improvement

- Extending the indexing methods for SAPUI5 source code
- Further modules and use cases
- Remote triggering of Jenkins via SAPUI5 app
- Parsing additional file types
- Automatic indexing of the SAPUI5 library

# Thank you

# Questions?

**By: Eman Belal**

**Supervisor: Prof. Dr. Christian Baun**

**Co-Supervisor: Prof. Dr. Thomas Gabel**

