Templating

Creating UI elements

SoftUni Team
Technical Trainers

Software University

SoftUni Foundation

Software University
http://softuni.bg

Table of Contents

1. Templating Concepts
2. Simple Templating
3. Templating Engines
4. Handlebars Overview

Have a Question?

SoftUni Foundation

sli.do

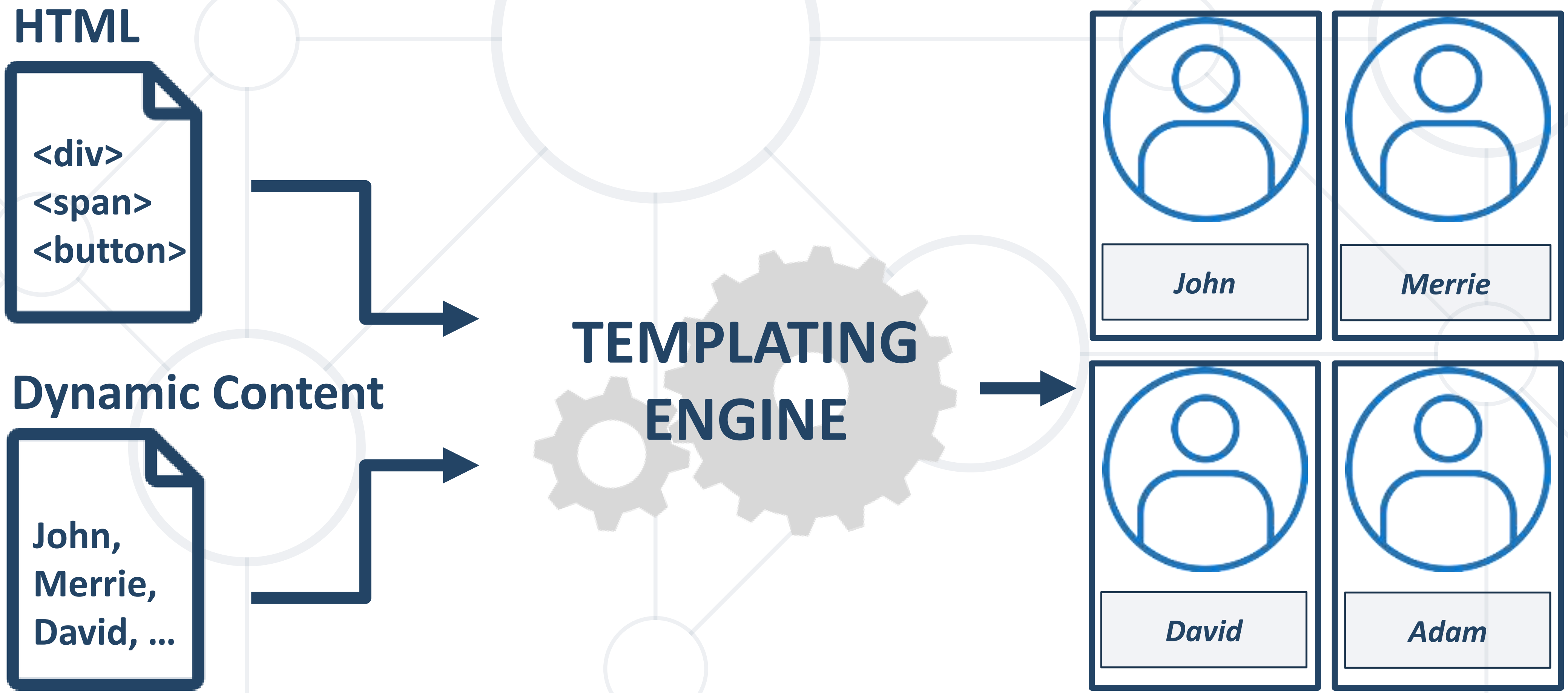
#JS-CORE

Templating Concepts

Definition and Uses

What is Templating?

Templates allow similar content to be **replicated** in a web page, **without repeating** the corresponding markup everywhere



Templating Concepts

- **Templates** are a method of **separating HTML** structure from the **content** contained within
- Templating systems generally introduce some **new syntax** but are usually very simple to work with
- Typically token replacement is used to indicate part, which must be replaced - (**{{ ... }}, <%...%> etc.**)

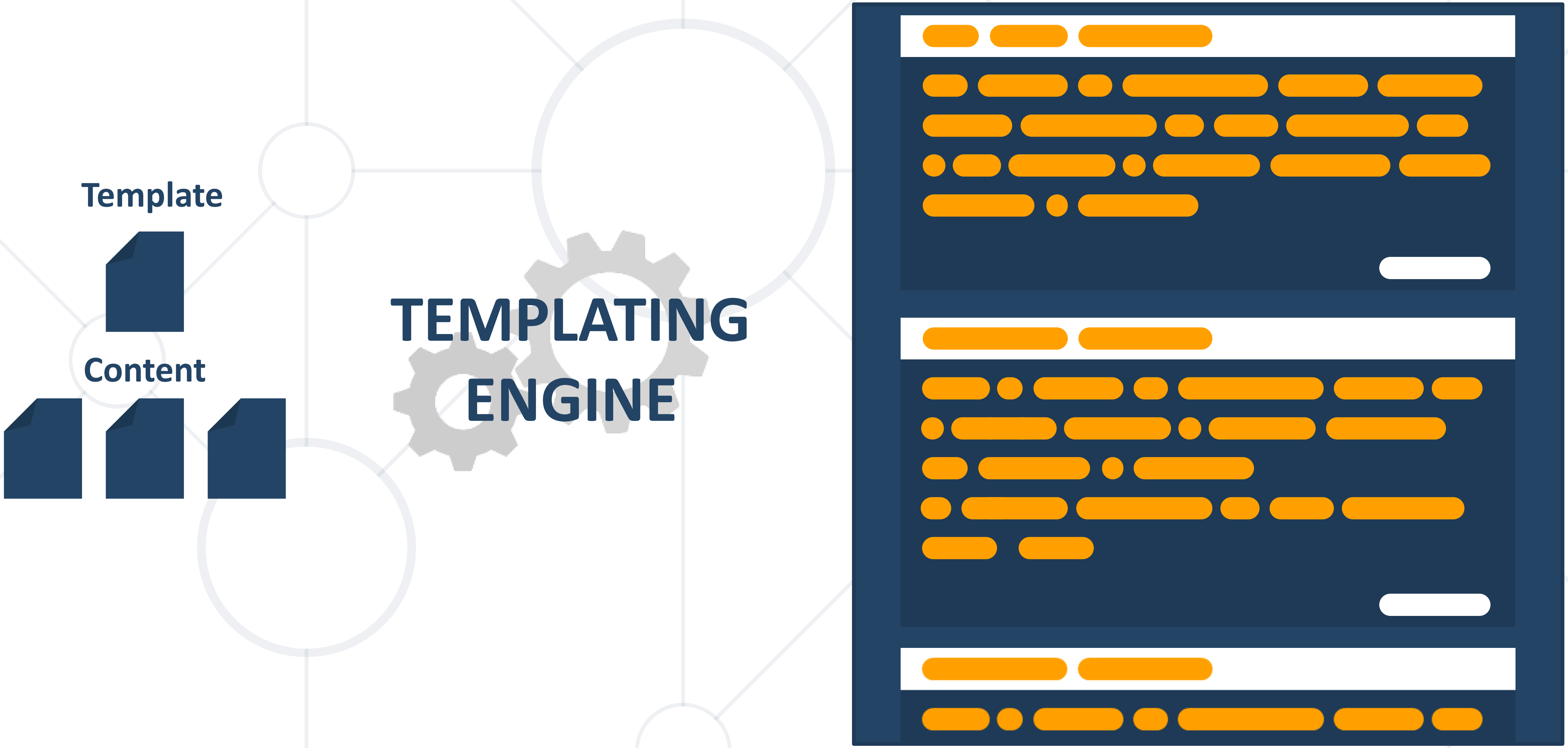
Templating Concepts

- The **static parts** of a webpage are stored as **templates**
- The **dynamic content** is kept separately (e.g. in a **database**)
- A **templating engine** combines the two
- Benefits:
  - **Productivity** - avoid writing the same markup over and over
  - **Save bandwidth** - send the HTML once, fill in any content
  - **Composability** - a single element can be used on multiple pages

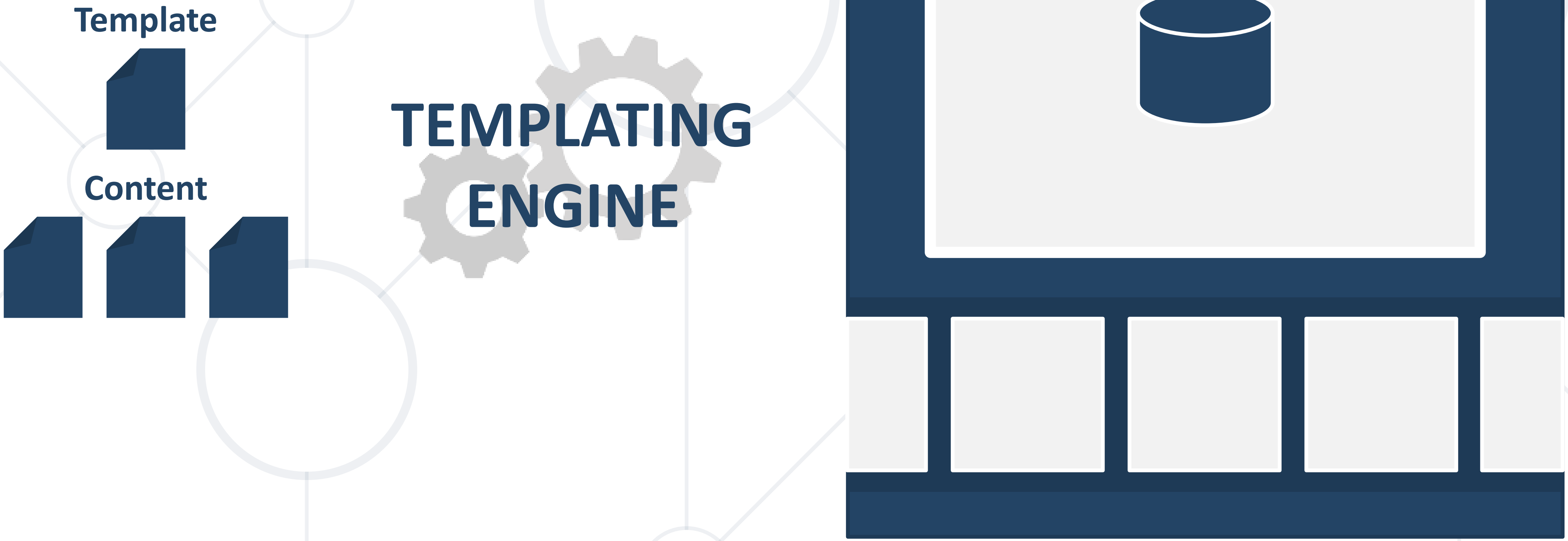
When should we use JS Templating?

- As soon as we find ourselves including **HTML inside JavaScript strings**
- **Separation of concerns** is of utmost importance when building a maintainable codebase

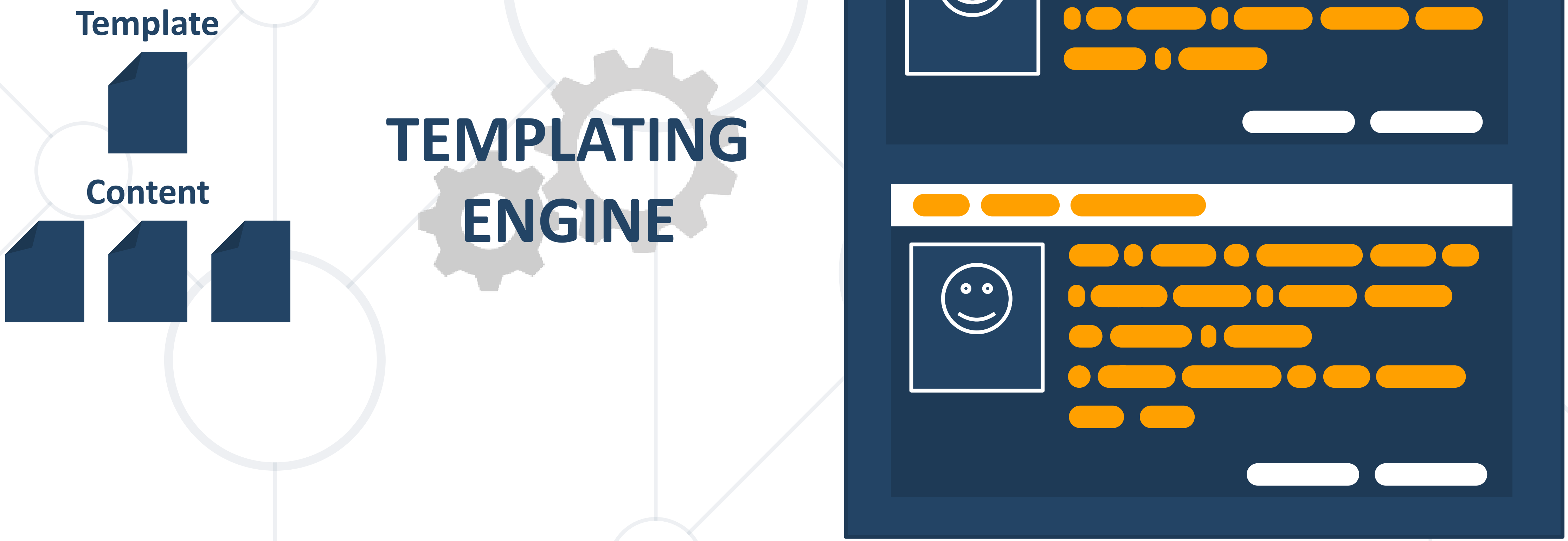
Display Articles in Blog



Display a Photo Gallery



Display User Profiles

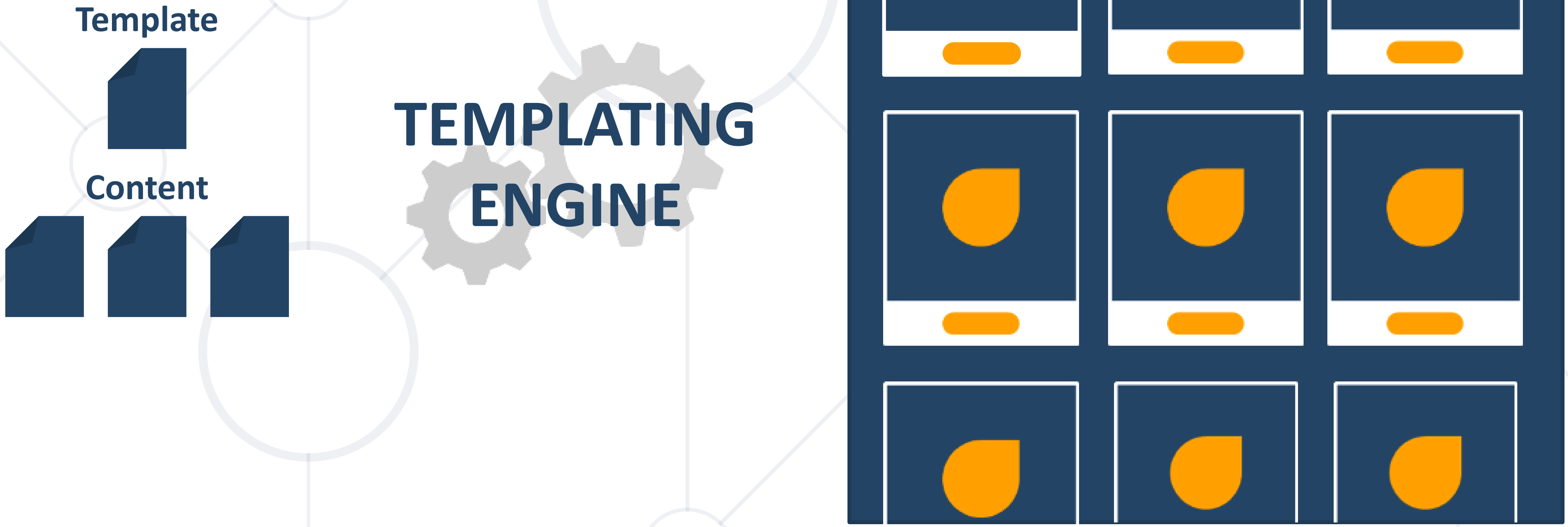


Template

Content

TEMPLATING ENGINE

Display Items in a Catalog



Live Demo

Simple HTML Templating

Templating Engines

Overview of Popular JS Libraries

Templating Engines

Pug

Vue

jQuery Templates

Underscore Templates

Mustache

Handlebars

Mustache Overview

Mustache is a multi-language, logic-less templating system

- Key Points
  - 9kb file size (small)
  - Simple
  - No dependencies
  - No logic
  - No precompiled templates

Mustache Installation and Using

- Download Mustache using the terminal
- Or download from https://mustache.github.io/
- Browser builds will be located in
- Use mustache from an online CDN
- Link it with a script tag

```
<script
  src="node_modules/mustache/mustache.js"
</script>
```

Mustache Overview

```
<script id="template" type="x-tmpl-mustache">
  <p>Use the <strong>{{power}}</strong>, {{name}}!</p>
</script>
```

```
var template = document.getElementById('template').innerHTML;
Mustache.parse(template);
var rendered = Mustache
       .render(template, {name: "Luke", power: "force"});

document.getElementById('target').innerHTML = rendered;
```

jQuery Overview

- jQuery Template is not as popular as mustache.js
- The templates are just plain HTML with no new syntax
- Key Points
  - 7kb file size (small)
  - Requires jQuery (+82kb)
  - Simple, but different to how Mustache and Handlebars.js work
  - No precompiled templates

jQuery Overview

```
<script id="template" type="text/html">
  <p>
    Use the <strong data-content="power"></strong>,
    <span data-content="name"></span>!
  </p>
</script>
```

```
$('#target').loadTemplate($('#template'), {
    name: "Luke",
    power: "force"
});
```

Templating with Handlebars

Syntax and Examples

Overview

- Based on the Mustache specification
- Adds helper functions and nested context paths
- Uses double curly brace notation {{ }}

```
<div class="entry">
  <h1>{{ title }}</h1>
  <div class="body">
    {{ body }}
  </div>
</div>
```

```
<div class="entry">
  <h1>My New Post</h1>
  <div class="body">
    This is my first post!
  </div>
</div>
```

Handlebars Installation and Using

- Download Handlebars using the terminal
- Or download from handlebarsjs.com
- Browser builds will be located in
- Use handlebars from an online CDN
- Link it with a script tag

```
<script
  src="node_modules/handlebars/dist/handlebars.js"
</script>
```

Expressions

- You can place your templates in a script element
  - Use type **text/x-handlebars-template**
  - Give the element an ID for easier use
- Anything inside double curly braces **{{ }}** will be evaluated

Compilation and Execution

- Once loaded, a template must be compiled

```
let source = $("#contact-template").html();
let template = Handlebars.compile(source);
```

- Compiled templates are functions, that can be executed with what ever variables we need

```
let context = {
  name: 'Ivan Ivanov',
  phone: '0888 123 456',
  email: 'i.ivanov@gmail.com'
};
let html = template(context);
```

Hello Handlebars

```html
<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
  <title>Hello Handlebars</title>
  <!-- Include Handlebars distribution -->
</head>
<body>
<div id="app"></div>
<script>
  let template = Handlebars.compile('<h1>Hello {{name}}</h1>');
  let container = document.getElementById('app');
  container.innerHTML = template({ name: 'Handlebars' });
</script>
</body>
</html>
```

Simple Identifiers

```
<script id="contact-template"
        type="text/x-handlebars-template">
  <article>
    <div class="title">
      {{name}}
      <button>&#8505;</button>
    </div>
    <div class="info">
      <span>&phone; {{phone}}</span>
      <span>&#9993; {{email}}</span>
    </div>
  </article>
</script>
```

For-Loops

- A template can be repeated for every entry in an array

```
<ul id="contacts">
  {{#each contacts}}
    <li>{{name}}: {{email}}</li>
  {{/each}}
</ul>
```

The expression inside the loop uses each **entry as context**

Conditional Statements

```
{{#if sunny}}
  The sky is clear
{{else}}
  The sky is overcast
{{/if}}
```

Variable to check for **truthiness**

```
<ul id="contacts">
  {{#each contacts}}
    <li>{{name}}: {{email}}</li>
  {{else}}
    <i>(empty)<i>
  {{/each}}
</ul>
```

Will be shown if the array is **empty**

Partials

- Partials are templates that can be inserted into other templates

```
let source = $("#contact-template").html();
Handlebars.registerPartial('contact', source);
```

Partial name

Template as string

```
<div id="contacts">
  {{#each contacts}}
    {{> contact}}
  {{else}}
    <i>(empty)<i>
  {{/each}}
</div>
```

Partials are globally accessible

HTML Escaping

Using the "triple-stash"

```
title: "All about <p> Tags"
body: "<p>This is a post about &lt;p&gt; tags</p>"
```

```
<div class="entry">
  <h1>All About &lt;p&gt; Tags</h1>
  <div class="body">
    <p>
      This is a post about&lt;p&gt; tags
    </p>
  </div>
</div>
```

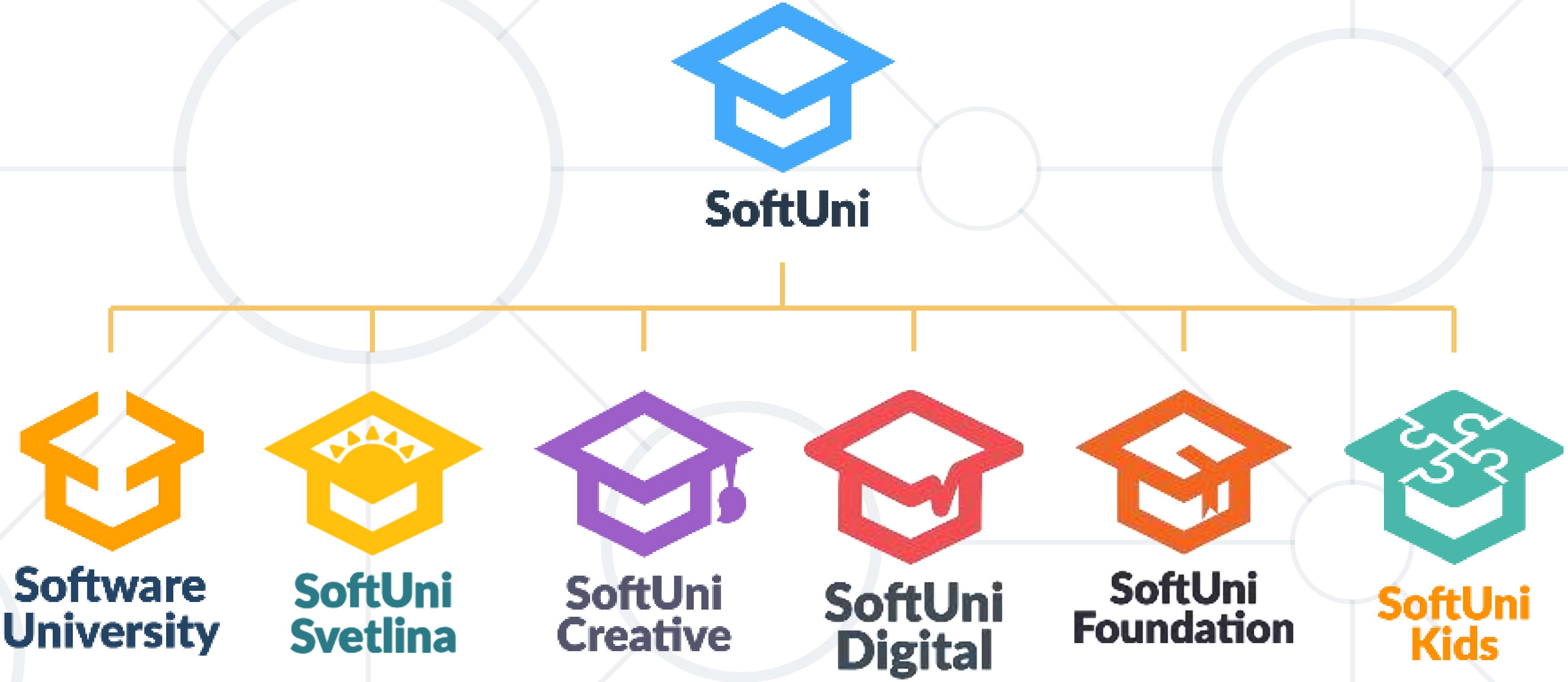
Live Exercise

Summary

- **Templates** speed up and simplify the development process
- **Handlebars** offers effective **templates** and simple helper functions

```
<div class="entry">
  <h1>{{ title }}</h1>
  <div class="body">
    {{ body }}
  </div>
</div>
```

Questions?



SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

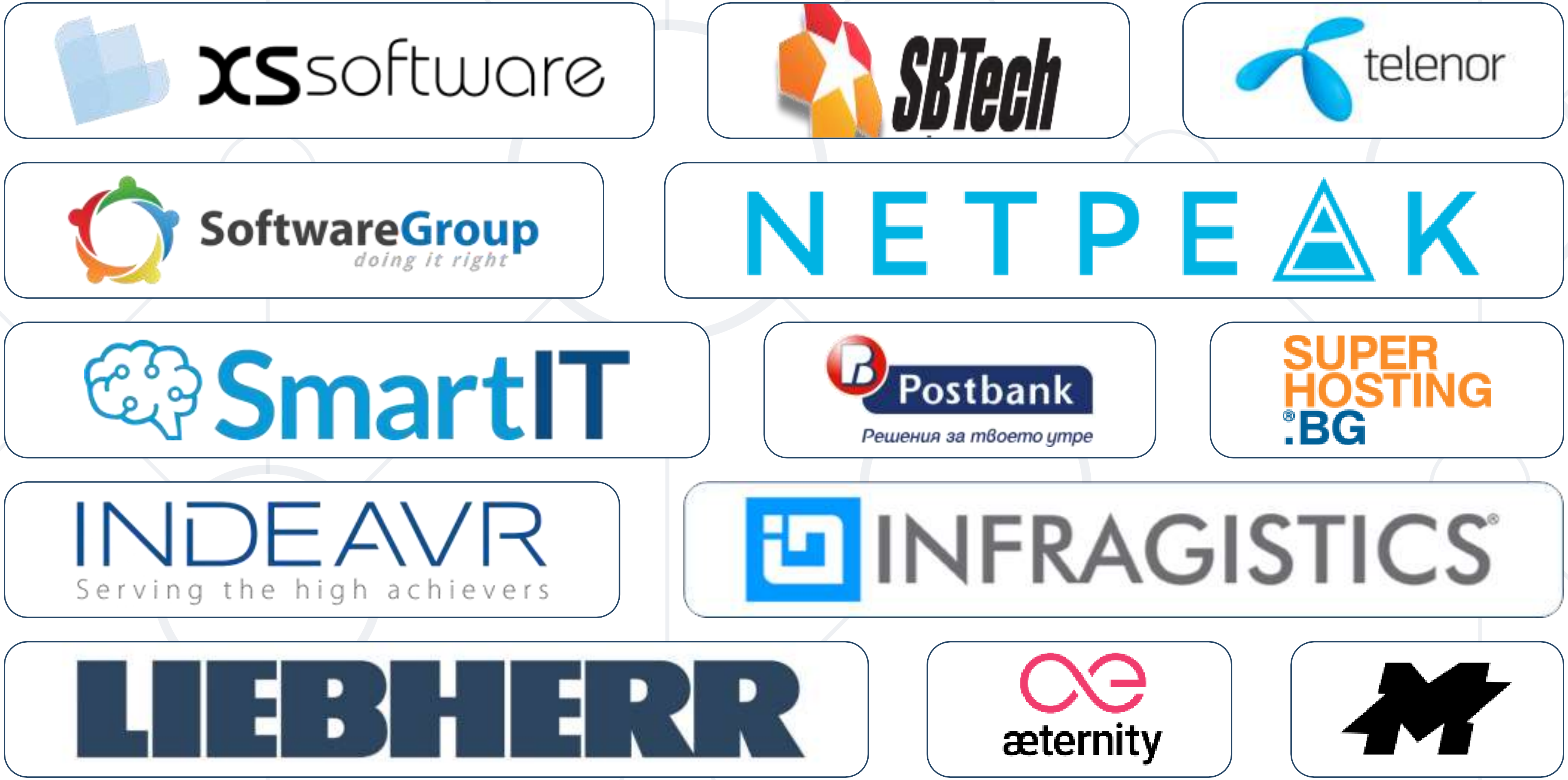
SoftUni Digital

SoftUni Foundation

SoftUni Kids

https://softuni.bg/courses/js-apps

SoftUni Diamond Partners



SoftUni Organizational Partners

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license