



MARIA KERN – NETZ98 GMBH

JS

MAGENTO 2 UI COMPONENTS

THE JS PART

Magento 2 UI Components – The JS part



Maria Kern

Senior Frontend Architect at netz98 GmbH

www.netz98.de

 @maja_kern

CommerceHero: maria-kern



MAGENTO 2 UI COMPONENTS – THE JS PART

UI components definition

UI COMPONENTS DEFINITION



The UI components are:

- the **base** for many **user interface (UI) elements** (buttons, fields, tables, minicart)
- designed for simple and flexible user interface rendering
- defined in the module `Magento_UI`

Magento uses **knockoutjs** as javascript library especially for binding data to html elements.



They are used in Magento **frontend** and **backend**

UI COMPONENTS DEFINITION



BACKEND

Attribute Set Default

Product Name *

[store view]

SKU *

[global]

Price *

[global]

[Advanced Pricing](#)

Tax Class Taxable Goods

[website]

Quantity

[global]

[Advanced Inventory](#)

Filters Default View Columns

Actions 6384 records found 20 per page 1 of 320

	ID	Thumbnail	Type	SKU	Price	Quantity	Visibility	Status	Action
☐	1		Downloadable Product	9783829781114	€11.99	10000.0000	Catalog, Search	Enabled	Edit
☐	2		Downloadable Product	9783829799737	€7.99	10000.0000	Catalog, Search	Enabled	Edit
☐	3		Downloadable Product	9783829782920	€2.99	10000.0000	Catalog, Search	Enabled	Edit

FRONTEND

Subtotal €23.98

Order Total €23.98

[Proceed to Checkout](#)

[Shopping Cart](#)

Search entire store here...

2

2 items Cart Subtotal: 23,98 €

[Go to Checkout](#)

[Example Product](#)

[See Details](#)

11,99 €

Qty:

UI COMPONENTS DEFINITION



BACKEND

Magento_Ui/js/form/components/fieldset

Attribute Set: Default

Product Name *

SKU *

Price * € [Advanced Pricing](#)

Tax Class: Taxable Goods

Quantity [Advanced Inventory](#)

FRONTEND

Subtotal €23.98

Order Total €23.98

[Proceed to Checkout](#)

Search entire store here...

2 items

Cart Subtotal: 23,98 €

[Go to Checkout](#)

Example Product

See Details

11,99 €

Qty: 2

Magento_Ui/js/grid/toolbar

Filters Default View Columns

Actions 6384 records found 20 per page 1 of 320

Actions	ID	Thumbnail	Type	SKU	Price	Quantity	Visibility	Status	Action
☐	1		Downloadable Product	9783829781114	€11.99	10000.0000	Catalog, Search	Enabled	Edit
☐	2		Downloadable Product	9783829799737	€7.99	10000.0000	Catalog, Search	Enabled	Edit
☐	3		Downloadable Product	9783829782920	€2.99	10000.0000	Catalog, Search	Enabled	Edit

Magento_Ui/js/grid/listing

UI COMPONENTS DEFINITION



BACKEND

Magento_Ui/js/form/components/fieldset

Attribute Set: Default

Product Name * [store view]

SKU * [global]

Price * € [global] **Magento_Ui/js/form/element/abstract**

Advanced Pricing

Tax Class [website]: Taxable Goods

Quantity [global] **Magento_Ui/js/form/components/fieldset**

Advanced Inventory

Magento_Ui/js/grid/toolbar

Filters | Default View | Columns

Actions | 6384 records found | 20 per page | 1 of 320

Actions	ID	Thumbnail	Type	SKU	Price	Quantity	Visibility	Status	Action
☐	1		Downloadable Product	9783829781114	€11.99	10000.0000	Catalog, Search	Enabled	Edit
☐	2		Downloadable Product	9783829799737	€7.99	10000.0000	Catalog, Search	Enabled	Edit
☐	3		Downloadable Product	9783829782920	€2.99	10000.0000	Catalog, Search	Enabled	Edit

Magento_Ui/js/grid/listing

FRONTEND

Magento_Checkout/js/view/cart/totals

Subtotal €23.98

Order Total €23.98

Proceed to Checkout

Shopping Cart

Search entire store here...

2 items | Cart Subtotal: 23,98 €

Go to Checkout

Example Product

See Details

11,99 €

Qty: 2

Magento_Checkout/js/view/minicart



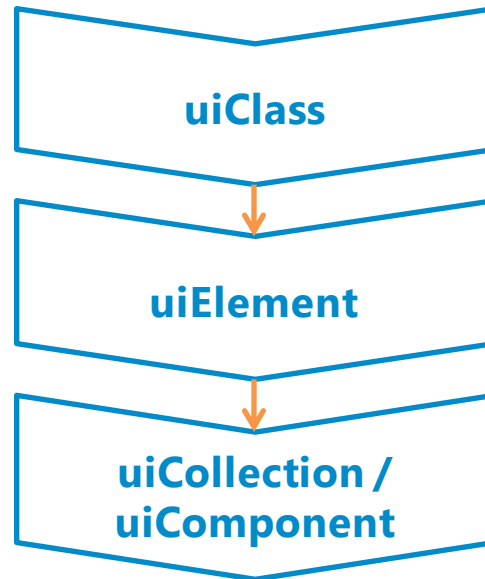
MAGENTO 2 UI COMPONENTS – THE JS PART

Basic structure of UI components

BASIC STRUCTURE OF UI COMPONENTS



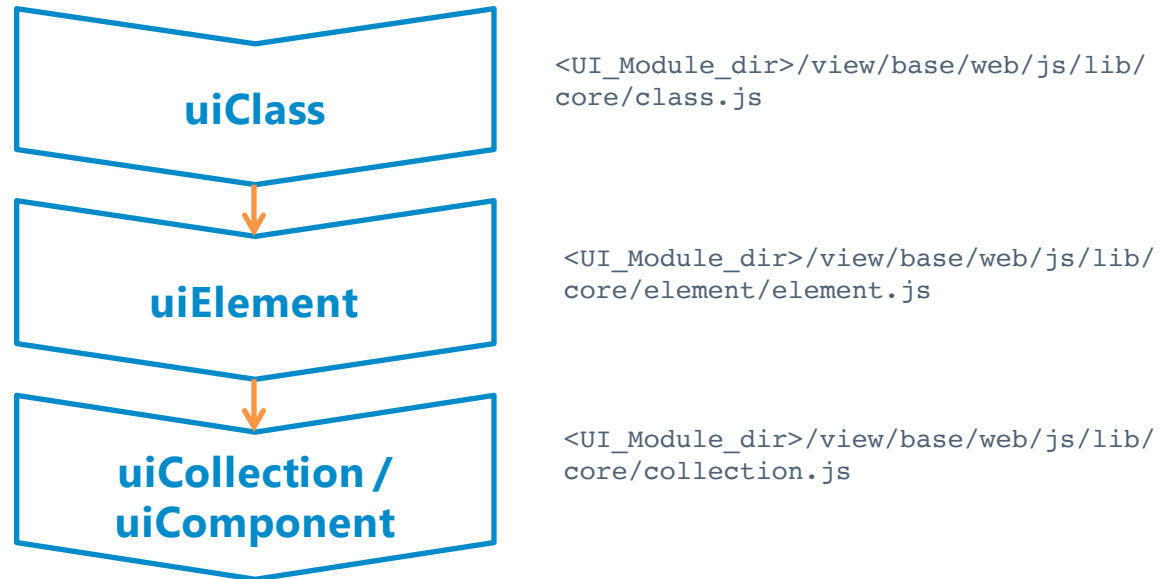
JS DEPENDENCY AND INHERITANCE



BASIC STRUCTURE OF UI COMPONENTS



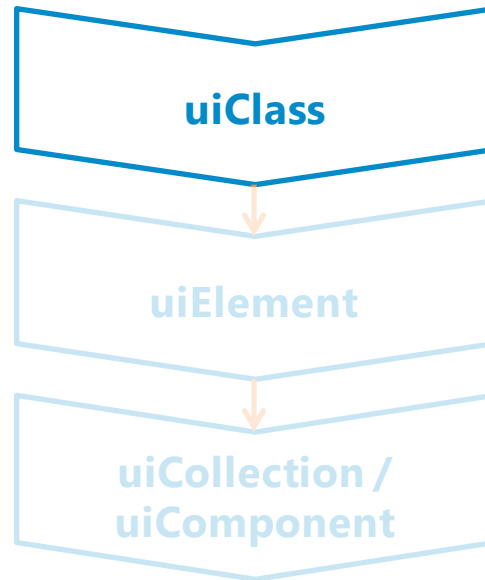
JS DEPENDENCY AND INHERITANCE



BASIC STRUCTURE OF UI COMPONENTS



JS DEPENDENCY AND INHERITANCE

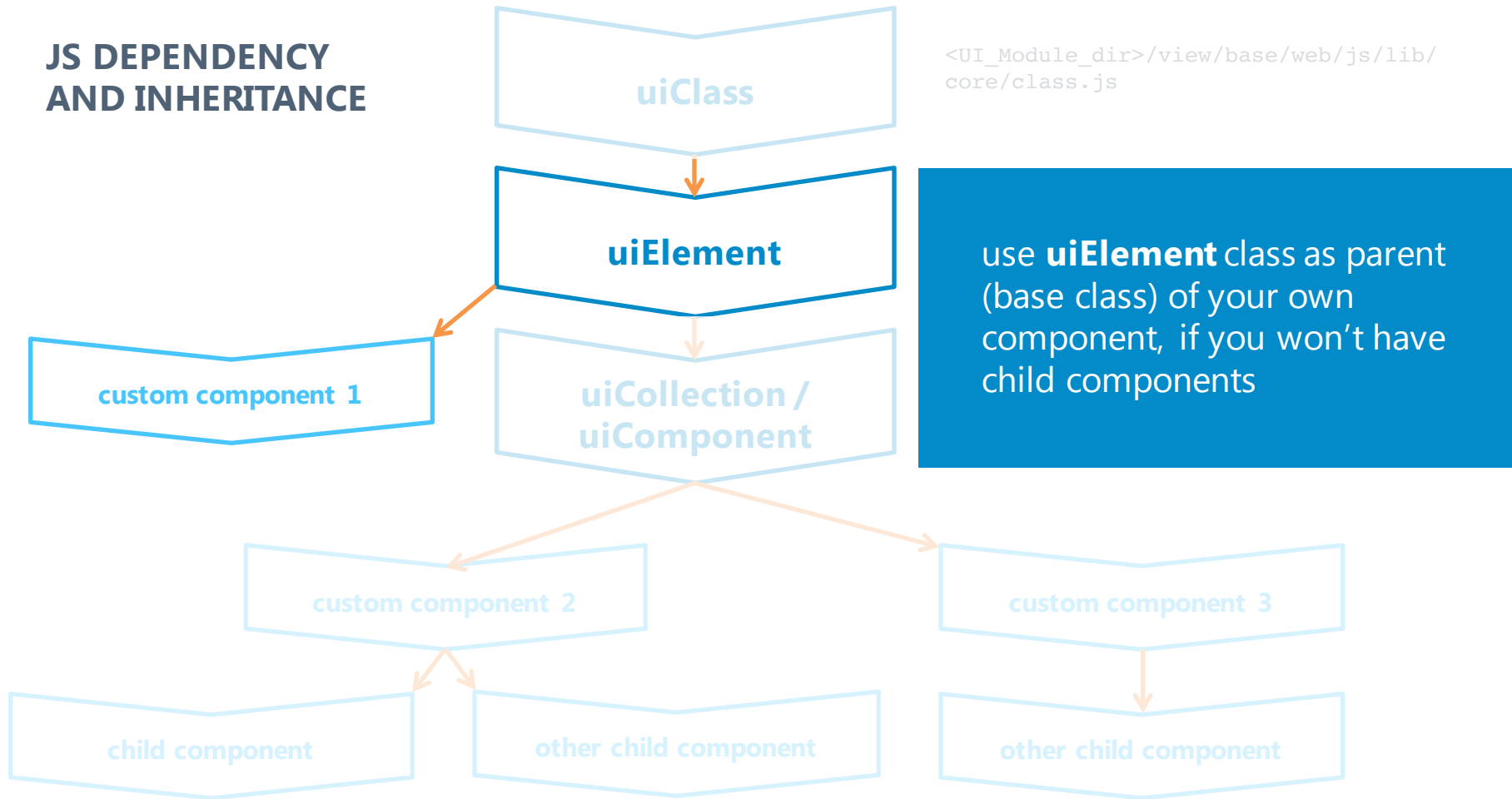


the **uiClass** is an abstract class where all components are extended from

`<UI_Module_dir>/view/base/web/js/lib/core/collection.js`

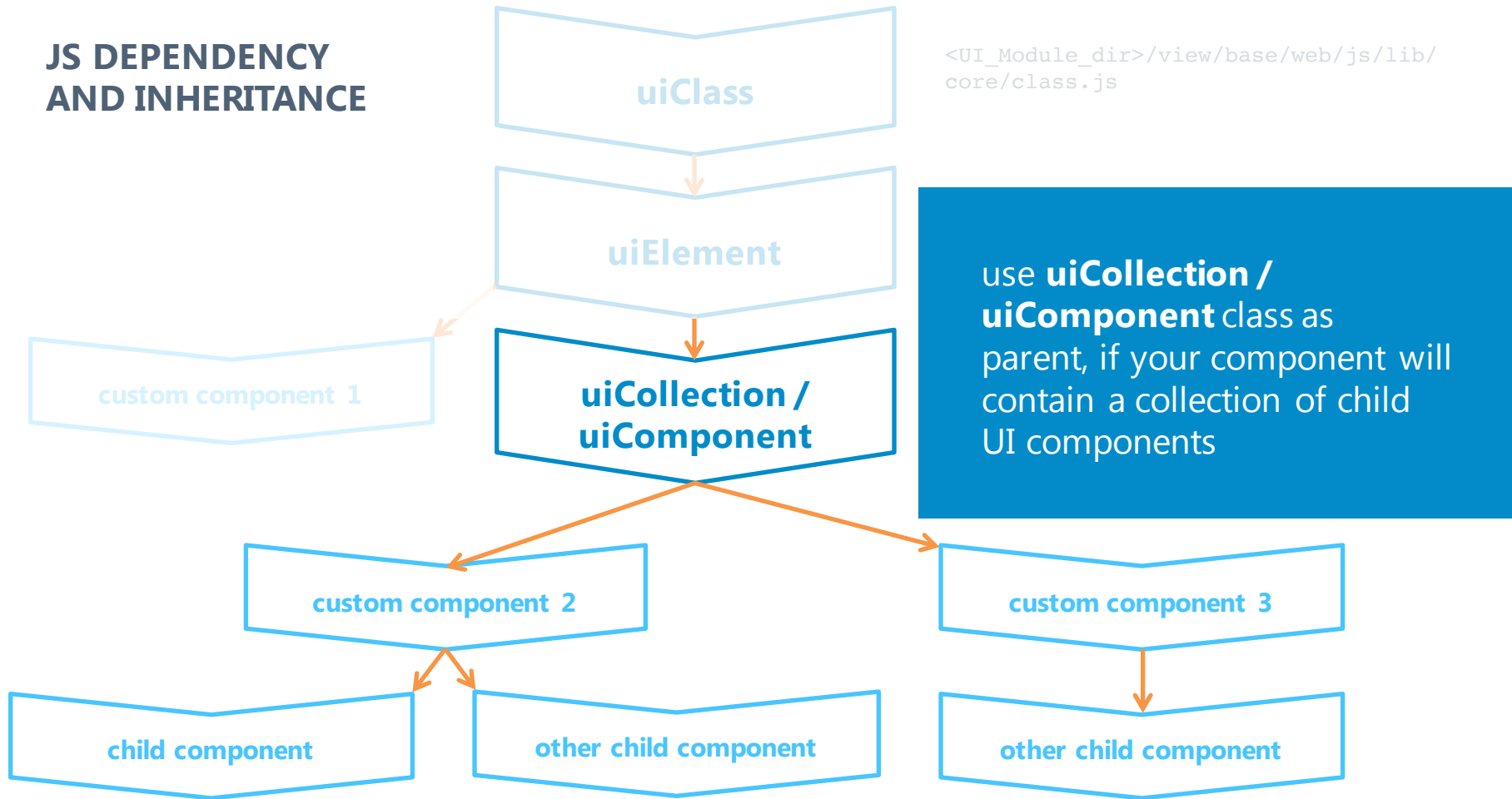
BASIC STRUCTURE OF UI COMPONENTS

JS DEPENDENCY AND INHERITANCE



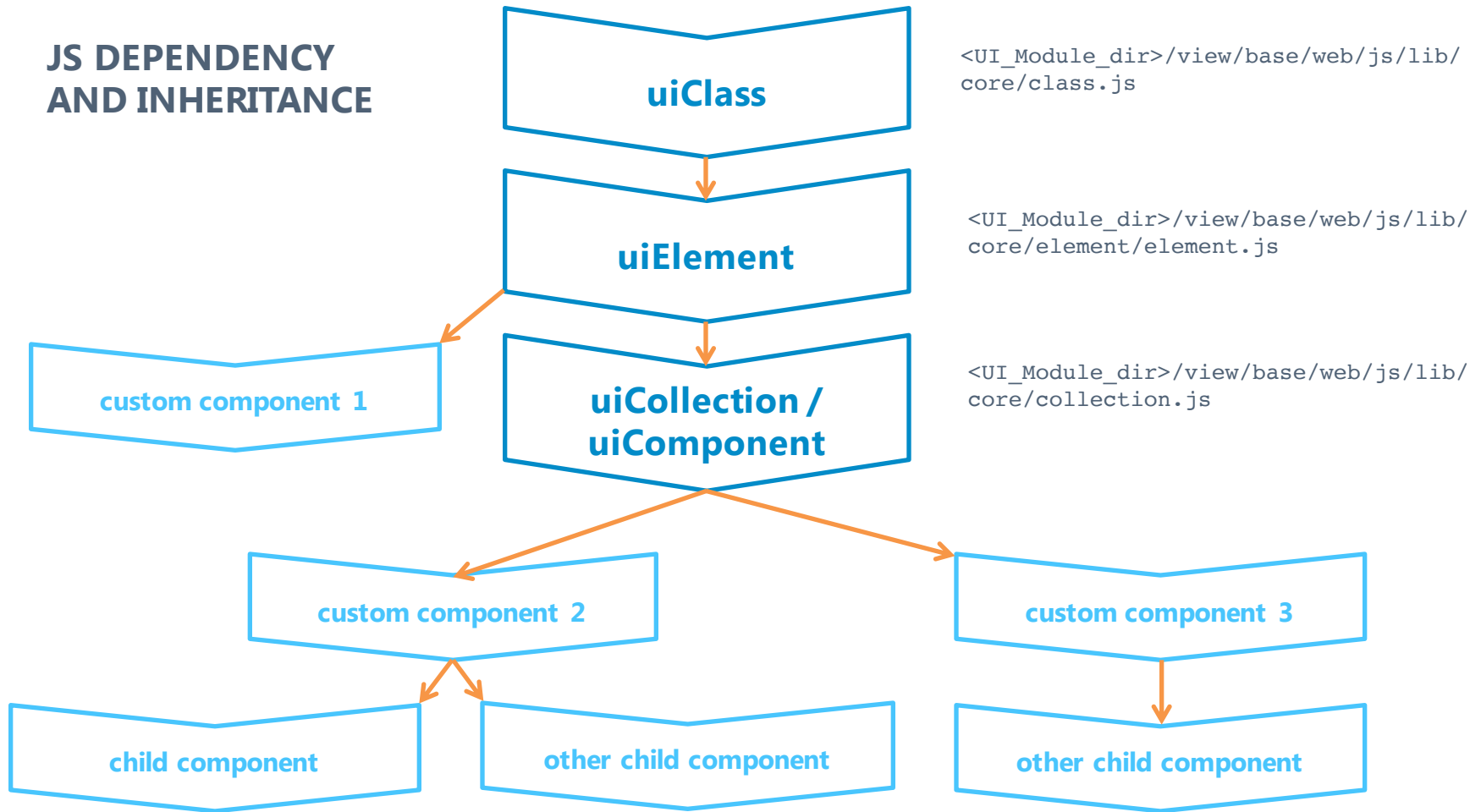
BASIC STRUCTURE OF UI COMPONENTS

JS DEPENDENCY AND INHERITANCE



BASIC STRUCTURE OF UI COMPONENTS

JS DEPENDENCY AND INHERITANCE





MAGENTO 2 UI COMPONENTS – THE JS PART

Useful functions and properties

USEFUL FUNCTIONS AND PROPERTIES

EXTEND AND INITIALIZE

- `extend()` implements inheritance of UI components
- `initialize()` is called during instantiation

```
define([
    'uiComponent'
], function (Component) {
    'use strict';

    return Component.extend({

        initialize: function () {
            /* call the parent UI component method: */
            this._super();

            /* here you can add custom functionality (executed only once) */
            this.myLocalProperty = 'abc'
        },
    });
});
```


USEFUL FUNCTIONS AND PROPERTIES

EXTEND AND INITIALIZE

- `extend()` implements inheritance of UI components
- `initialize()` is called during instantiation

```
define([  
    'uiComponent'  
], function (Component) {  
    'use strict';
```

DEPENDENCY

```
    return Component.extend({
```

```
        initialize: function () {
```

```
            /* call the parent UI component method: */
```

```
            this._super();
```

```
            /* here you can add custom functionality (executed only once) */
```

```
            this.myLocalProperty = 'abc'
```

```
        },
```

USEFUL FUNCTIONS AND PROPERTIES

EXTEND AND INITIALIZE

- `extend()` implements inheritance of UI components
- `initialize()` is called during instantiation

```
define([  
    'uiComponent'  
], function (Component) {  
    'use strict';
```

```
    return Component.extend({
```

INHERITANCE

```
        initialize: function () {  
            /* call the parent UI component method: */  
            this._super();  
  
            /* here you can add custom functionality (executed only once) */  
            this.myLocalProperty = 'abc'  
        },
```

USEFUL FUNCTIONS AND PROPERTIES

EXTEND AND INITIALIZE

- `extend()` implements inheritance of UI components
- `initialize()` is called during instantiation

```
define([  
    'uiComponent'  
], function (Component) {  
    'use strict';  
  
    return Component.extend({
```

```
        initialize: function () {  
            /* call the parent UI component method: */  
            this._super();  
  
            /* here you can add custom functionality (executed only once) */  
            this.myLocalProperty = 'abc'  
        },
```

OVERWRITE / EXTEND A
METHOD

USEFUL FUNCTIONS AND PROPERTIES

CONFIGURATION AND FIRST LEVEL PROPERTIES

```
return Component.extend({  
  
  defaults: {  
    // custom properties:  
    isVisible: true,  
    headline: '',  
    pageSize: 20,  
    selectors: {  
      mapItem: '.map_item'  
    },  
    // linking properties:  
    listens: {  
      'isVisible': 'onIsVisibleChange'  
    },  
    exports: {},  
    imports: {},  
    links: {},  
    modules: {}  
  },  
},
```

Configuration properties will
be available as “**first level
properties**”

`this.isVisible`



CONFIGURATION AND FIRST LEVEL PROPERTIES

A lot of **default properties** and **methods** are already set by the base classes, e.g.:

```
this.index // name of component
```

```
this.name // full name of component including parents
```

```
this.template
```

```
this.component
```

```
this.getTemplate()
```

```
this.hasTemplate()
```



TRACKS AND INIT OBSERVABLES

For Knockout JS binding:

```
this.tracks
```

```
initObservable()
```

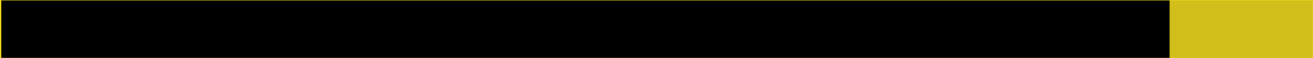
```
<div class="map-widget" data-block="map-widget" data-bind="visible: isVisible">  
  <div data-bind="html: headline"></div>  
  <div data-bind="html: text"></div>  
  <div class="map_wrapper">  
    <div class="map" data-bind="bindHtml: map"></div>  
    <!-- ko foreach: getRegion('layer') -->  
      <!-- ko template: getTemplate() --><!-- /ko -->  
    <!--/ko-->  
  </div>  
</div>
```



INIT ELEMENT AND HANDLING WITH CHILDREN

The **uiCollection class** (mainly) provides a lot of methods for handling with child components, e.g.:

```
initElement(child) {}  
this.elms()  
this.containers  
this.getChild('nameOfChild') // UI component  
this.hasChild('nameOfChild') // boolean  
this.insertChild('nameOfComponent')
```



Initialize...

MAGENTO 2 UI COMPONENTS – THE JS PART

Initialization process

All the configurations are translated into json and added to the body

```
<script type="text/x-magento-init">{"*": {"Magento_Ui/js/core/app":{<JSON_configuration>}}}</script>
```

<!-- Example: -->

```
<script type="text/x-magento-init">
{
  "[data-role=compare-products-link]": {
    "Magento_Ui/js/core/app": {
      "components": {
        "compareProducts": {
          "component": "Magento_Catalog/js/view/compare-products",
          "customProperty1": "abc",
          "customProperty2": true,
          "children": {
            "childNameOfCompareProducts": {
              "component": "Magento_Catalog/js/view/subfolder/nameOfFile",
              "customProperty1": "abc",
              "displayArea": "nameOfDisplayArea"
            }
          }
        }
      }
    }
  }
}
</script>
```

INITIALIZATION PROCESS



- The app calls `Magento_Ui/js/core/renderer/layout` and passes the UI component's configuration into the layout

The screenshot shows a code editor with the `app.js` file open. The code defines a function that takes `types` and `layout` as arguments and returns another function that takes `data` and `merge` as arguments. The returned function calls `types.set(data.types)` and then `layout(data.components, undefined, true, merge)`. The `layout` parameter is highlighted in blue on line 14.

On the right side, the 'Paused on breakpoint' panel shows the execution context. The 'Local' scope contains the following variables:

- `data`: Object
 - `components`: Object
 - `compareProducts`: Object
 - `component`: `"Magento_Catalog/js/view/compare-products"`
- `__proto__`: Object
- `__proto__`: Object
- `__proto__`: Object
- `merge`: `li.item.link.compare`

INITIALIZATION PROCESS



- layout.js creates instances of UI components and its child components

The screenshot shows a code editor with the `layout.js` file selected. The code defines a function `initComponent` that creates a new component instance and registers it. The debugger is paused at line 68, where `registry.set` is called. The local scope shows the `component` object, which is an instance of `UiClass` with various properties and methods.

```

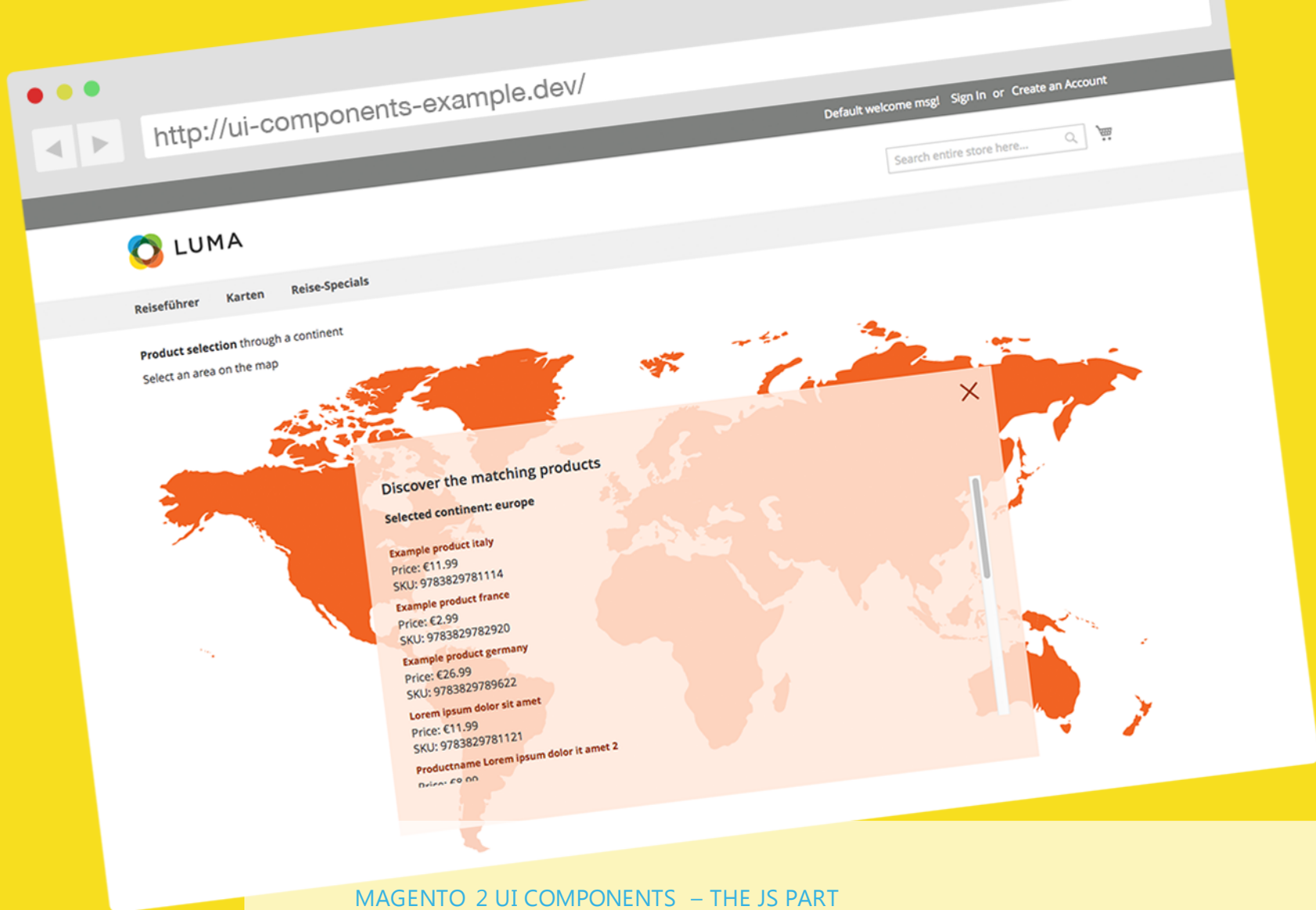
57
58     require([source], function (constr) {
59         loaded.resolve(node, constr);
60     });
61
62     return loaded.promise();
63 }
64
65 function initComponent(node, Constr) {
66     var component = new Constr(_.omit(node, 'children'));
67
68     registry.set(node.name, component);
69 }
70
71 function run(nodes, parent, cached, merge) {
72     if (_.isBoolean(merge) && merge) {
73         layout.merge(nodes);
74
75         return false;
76     }
77
78     if (cached) {
79         cachedConfig[_.keys(nodes)[0]] = JSON.parse(JSON.string
80     }
81

```

Debugger paused

Local

- ▶ `Constr: function ()`
- ▼ `component: UiClass`
 - ▶ `compareProducts: function observable()`
 - ▶ `component: "Magento_Catalog/js/view/compare-products"`
 - ▶ `containers: Array(0)`
 - ▶ `dataScope: ""`
 - ▶ `elems: function observable()`
 - ▶ `exports: Object`
 - ▶ `extendProvider: true`
 - ▶ `hasUnique: undefined`
 - ▶ `ignoreTmpls: Object`
 - ▶ `imports: Object`
 - ▶ `index: "compareProducts"`

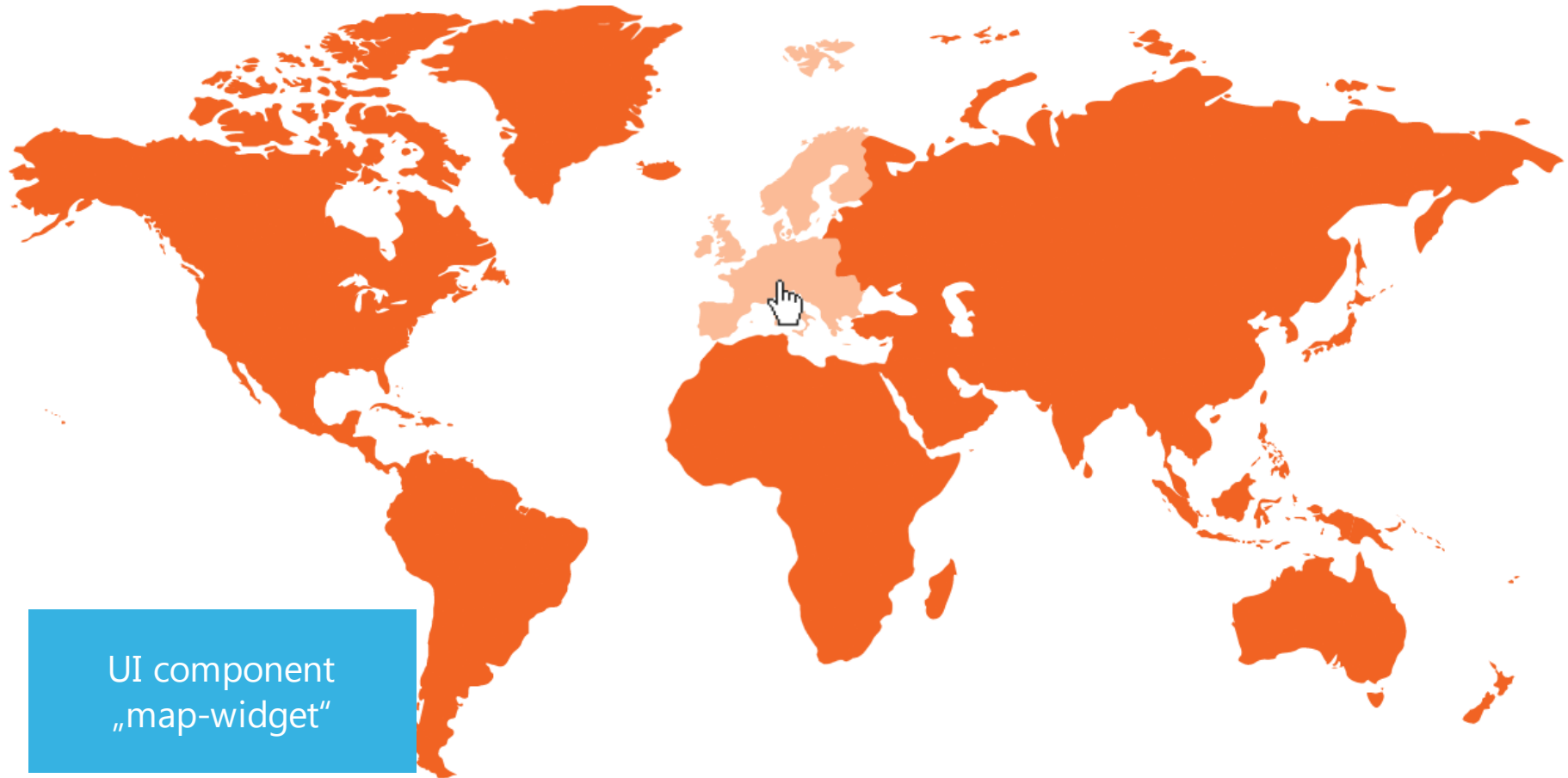


MAGENTO 2 UI COMPONENTS – THE JS PART

Real life example

Product selection through a continent

Select an area on the map



UI component
„map-widget“

Product selection through a continent

Select an area on the map



UI component
„map-widget“

Product selection through a continent

Select an area on the map



child UI component
„productlist“

productlist could be used in other components as well



MAGENTO 2 UI COMPONENTS – THE JS PART

Options for debugging

OPTIONS FOR DEBUGGING



Chrome Knockoutjs Context Debugger

The screenshot shows the Chrome DevTools interface with the KnockoutJS Context Debugger. The left pane displays the DOM tree, and the right pane shows the Knockout context for the selected element.

DOM Tree (Left Pane):

```

<!-- ko foreach: getRegion('layer') -->
<!-- ko template: getTemplate() -->
<div class="productlist" data-block="productlist" data-bind="visible: isVisible" data-mspdevtools-ui="b4019872c3e85a1138fc7145f78dd123">
  <div class="content">
    <div data-bind="html: headline">
      <h3>Discover the matching products</h3> == $0
    </div>
    <div class="selected_value">...</div>
    <div class="products">...</div>
    <button type="button" class="action close" data-bind="click: function() { isVisible(false); }, attr: { title: $('Close') }" title="Close">...</button>
  </div>
</div>
<!-- /ko -->

```

Knockout context (Right Pane):

```

$uiClass
  component: "N98_MapWidget/js/view/productlist"
  containers: Array(1)
  dataScope: ""
  displayArea: "layer"
  elems: function observable()
  exports: Object
  extendProvider: true
  hasUnique: undefined
  headline: function observable()
  ignoreTmpls: Object
  imports: Object
  index: "productlist"
  isModal: true
  isVisible: function observable()
  links: Object
  listens: Object
  loader: function observable()
  maps: Object
  modules: Object
  name: "widgetmapwidget0.productlist"
  ns: "widgetmapwidget0"
  parentName: "widgetmapwidget0"

```

Annotations:

- A blue box highlights the `<div data-bind="html: headline">` element in the DOM tree.
- A blue arrow points to this element with the text: **click on DOM element inside UI component scope**.
- The right pane shows the Knockout context for the selected element, with the **Knockout context** tab selected.

OPTIONS FOR DEBUGGING



Magento DevTools by MageSpecialist, with Chrome extension

Magento DevTools by MageSpecialist

General Design Observers Blocks **UI** Profiler Plugins Queries

Search:

↓	Component	↑
	Magento_Checkout/minicart/content	
	Magento_Customer/authentication-popup	
	N98_MapWidget/map-widget	
▼ Object	<pre>component: "N98_MapWidget/map-widget" template: "N98_MapWidget/template/map-widget.html" type: "uiComponent"</pre>	
	ui/collection	

list of all initialized UI components on the actual page with link to inspect the DOM element



PERSONAL PROS & CONS

UI COMPONENTS – THE JS PART

Advantages

- easy teamwork – standardized programming
- common procedures and requirements have already been implemented
- configuring without inline scripting!

Disadvantages

- need time to get familiar with
- sometimes you feel restricted in frontend development