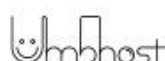


UMBRACOSPARK #SPARKQA

The new Backoffice of Umbraco

Jacob Overgaard and Lasse Fredslund





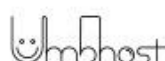
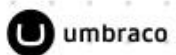
Jacob Overgaard
Frontend Team Lead & Architect
Umbraco HQ

 @iOvergaard



Lasse Fredslund
CMS Product Owner
Umbraco HQ

 @lassefredslund



#SPARKQA



Why a new Backoffice?

- AngularJS is EOL
- Limited extension points →
 - Usage of Angular Interceptors
 - Impossible to introduce none breakable updates

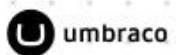


Web
Components

New
Backoffice

Extension
API

Management
API



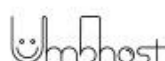
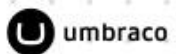
#SPARKQA



What we use

- Typescript
- Lit

Web
Components



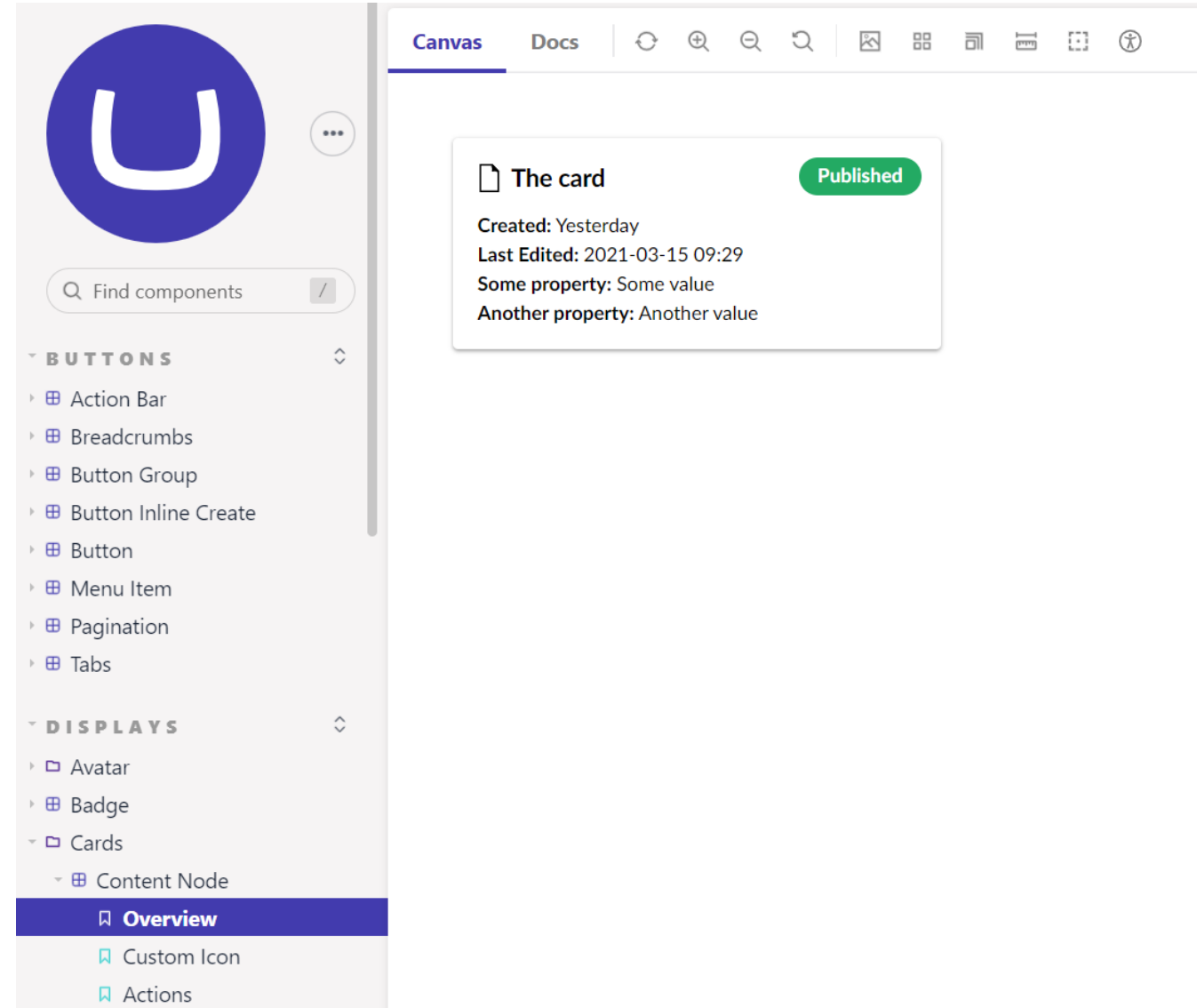
#SPARKQA



What we use

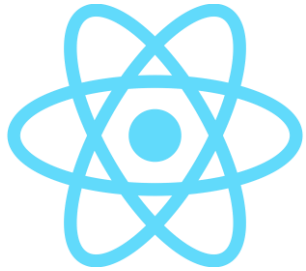
- Umbraco UI Library

Web
Components

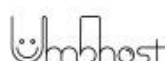
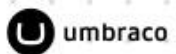




You can use the framework you like



Web
Components



#SPARKQA

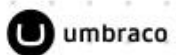


Web
Components

New
Backoffice

Extension
API

Management
API



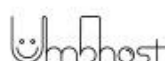
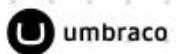
#SPARKQA



Management API

More responsibility to the frontend

Management
API



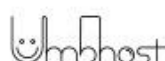
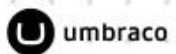
#SPARKQA



Management API

Part of our Headless offering

Management
API



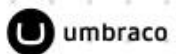
#SPARKQA



Management API

- OpenAPI Spec
 - Clients can be autogenerated
 - Authentication – OpenID Connect with usage of OpenIddict
- Rest API
- Doc available in Swagger

Management
API



#SPARKQA



Umbraco Backoffice API 1.0 OAS3

This shows all APIs available in this version of Umbraco - including all the legacy apis that are available for backward compatibility

Authorize



Audit Log



GET	/umbraco/management/api/v1/audit-log	▼	🔒
GET	/umbraco/management/api/v1/audit-log/{key}	▼	🔒
GET	/umbraco/management/api/v1/audit-log/type/{logType}	▼	🔒

Culture



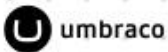
GET	/umbraco/management/api/v1/culture	▼	🔒
-----	------------------------------------	---	---

Data Type



POST	/umbraco/management/api/v1/data-type	▼	🔒
GET	/umbraco/management/api/v1/data-type/{key}	▼	🔒
DELETE	/umbraco/management/api/v1/data-type/{key}	▼	🔒
PUT	/umbraco/management/api/v1/data-type/{key}	▼	🔒
POST	/umbraco/management/api/v1/data-type/{key}/copy	▼	🔒
POST	/umbraco/management/api/v1/data-type/{key}/move	▼	🔒
GET	/umbraco/management/api/v1/data-type/{key}/references	▼	🔒
POST	/umbraco/management/api/v1/data-type/folder	▼	🔒
GET	/umbraco/management/api/v1/data-type/folder/{key}	▼	🔒

Management
API



#SPARKQA

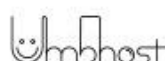
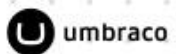


Updated controllers

- Versioned (to avoid breaking changes)
- Smaller controllers
- Bigger services
- All APIs use keys (instead of integer IDs)



Management
API



#SPARKQA

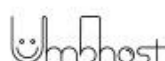
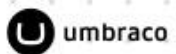


Attempt Pattern



```
Attempt<ILanguage, LanguageOperationStatus> result =  
    await _languageService.CreateAsync(created, CurrentUserId(_backOfficeSecurityAccessor));
```

Management
API



#SPARKQA



Attempt Pattern

```
public enum LanguageOperationStatus
{
    Success,
    CancelledByNotification,
    InvalidFallback,
    NotFound,
    MissingDefault,
    DuplicateIsoCode,
    InvalidIsoCode,
    InvalidFallbackIsoCode,
    InvalidId
}
```

Management
API

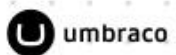


Web
Components

New
Backoffice

Extension
API

Management
API




#SPARKQA



“Extension-First Architecture”

Extension
API

ContentMediaMembersSettingsPackagesTranslationUsers

English (United States)ContentInfo

Document 1

Headline"Extension First Architecture"

Text string property

DescriptionEvery UI Component in The New Backoffice is implemented as an extension!

Textarea property

Breadcrumbs

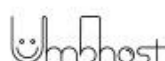
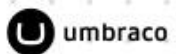
Save and previewSaveSave and publish



Comment from Nathan

Extension
API

"The extension registry pattern is lovely to work with - I put together a new manifest type for RTE plugins, and everything just works. It's a really pleasant DX, extending is smooth and painless."



#SPARKQA



Example: Custom Dashboard

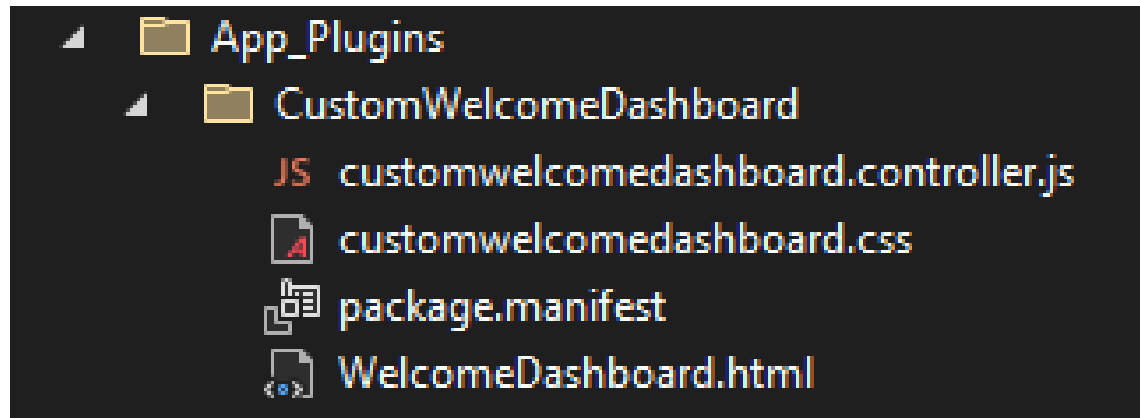
Extension
API

The screenshot shows a web browser window with the Umbraco Spark V13 backoffice. The browser tab is labeled 'Umbraco' and the address bar shows the URL '127.0.0.1:5173/section/content/dashboard/customwelcomedashboard'. The navigation menu includes 'Content', 'Media', 'Members', 'Settings', 'Packages', 'Translation', and 'Users'. The 'Content' section is active, showing a list of items: 'All property editors' and 'Blog post A'. The main content area displays a custom welcome message: 'Welcome to Umbraco Spark!' followed by 'This is the V13 Backoffice' and '© Umbraco Spark 2023'.



Custom Dashboard – prior to V13

Extension
API





Custom Dashboard – in V13

Extension
API

```
import { UITextStyles } from '@umbraco-ui/uui-css/lib';
import { css, html, LitElement } from 'lit';
import { customElement } from 'lit/decorators.js';

@customElement('custom-welcome-spark')

export class CustomWelcomeElement extends LitElement {
  static styles = [UITextStyles, css`];

  render() {
    return html`
      <h1>Welcome to Umbraco Spark!</h1>
      <p>This is the V13 Backoffice</p>
      <p>© Umbraco Spark 2023</p>
    `;
  }
}

export default CustomWelcomeElement;

declare global {
  interface HTMLTagNameMap {
    'custom-welcome-spark': CustomWelcomeElement;
  }
}
```



imports



Register custom element {

```
import { UITextStyles } from '@umbraco-ui/uui-css/lib';
import { css, html, LitElement } from 'lit';
import { customElement } from 'lit/decorators.js';
```

```
@customElement('custom-welcome-spark')
```

```
export class CustomWelcomeElement extends LitElement {
  static styles = [UITextStyles, css``];
```

```
  render() {
    return html`
      <h1>Welcome to Umbraco Spark!</h1>
      <p>This is the V13 Backoffice</p>
      <p>© Umbraco Spark 2023</p>
    `;
  }
}
```

```
export default CustomWelcomeElement;
```

```
declare global {
  interface HTMLElementTagNameMap {
    'custom-welcome-spark': CustomWelcomeElement;
  }
}
```

Extension
API

} Create class with Lit

} Lit html

Present as extension {

For intellisense {

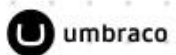


Web
Components

New
Backoffice

Extension
API

Management
API

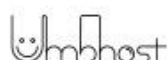
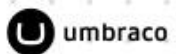


#SPARKQA



When we started the journey

New
Backoffice

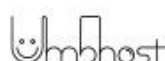
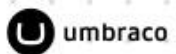


#SPARKQA



<umb-backoffice />

New
Backoffice



#SPARKQA



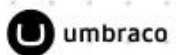
New
Backoffice

```
<!DOCTYPE html>
<html lang="en">
<head>
  <base href="/umbraco/" />
  <title>Umbraco</title>
  <link rel="stylesheet" href="backoffice/style.css" />
  <script type="module" src="backoffice/main.js"></script>
</head>

<body>

  <!-- THIS IS IT -->
  <umb-backoffice></umb-backoffice>

</body>
</html>
```

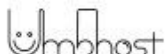
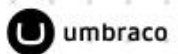


#SPARKQA

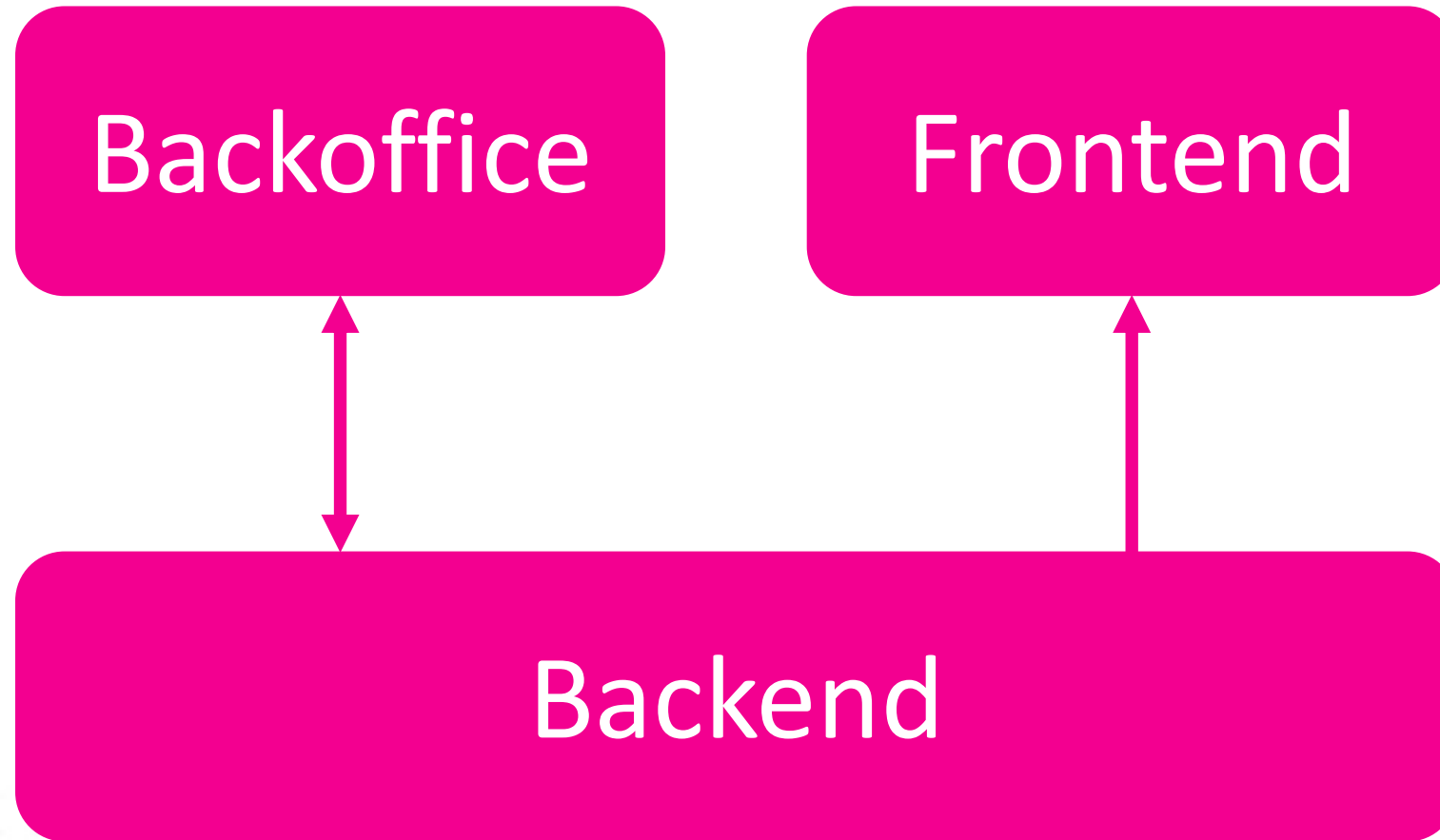


```
<umb-backoffice api-url="https://example.com"></umb-backoffice>
```

New
Backoffice

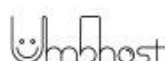
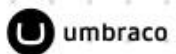


#SPARKQA





*We believe in the modern web and seek
only to sugar coat where it makes sense*



#SPARKQA



Tech Stack

Today

JavaScript

AngularJS

jQuery

Gulp

Future

Typescript

Web Components

Modern Web

Vite



Two New Concepts

Extension API

Auto register from server manifests

Manually register with JavaScript

Context API

Event-driven context based on DOM elements

Provider & Consumer



Extension
API

```
{
  "name": "My Awesome Package",
  "version": "1.0.0",
  "dashboards": [
    {
      "alias": "welcomeDashboard",
      "view": "/App_Plugins/MyPackage/WelcomeDashboard.html",
      "sections": [ "Content" ],
      "weight": -1
    }
  ]
}
```



```
{
  "name": "My Awesome Package",
  "version": "1.0.0",
  "extensions": [
    {
      "type": "dashboard",
      "alias": "welcomeDashboard",
      "name": "My Dashboard",
      "js": "/App_Plugins/MyPackage/dashboard.js",
      "weight": -1,
      "meta": {
        "label": "My Dashboard",
        "sections": [ "Umb.Section.Content" ],
        "pathname": "my-dashboard"
      }
    }
  ]
}
```

umbraco-package.json



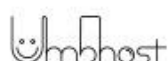
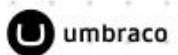
Extension API

```
const onInit = (host, extensionRegistry) => {  
  if ( condition ) {  
    extensionRegistry.register(  
      {  
        type: 'dashboard',  
        alias: 'my.conditional.dashboard',  
        name: 'MyDashboard',  
        js: '/App_Plugins/MyPackage/mydashboard.js',  
        meta: {  
          label: 'My Dashboard',  
          sections: ['Umb.Section.Content'],  
          pathname: 'my-dashboard',  
        },  
      }  
    )  
  }  
}  
  
export { onInit };
```



Extension
API

Umbraco CLI?



#SPARKQA



Two New Concepts

Extension API

Auto register from server manifests

Manually register with JavaScript

Context API

Event-driven context based on DOM elements

Provider & Consumer



Context API

```
export class UmbBackoffice extends UmbElementMixin(LitElement) {  
  constructor() {  
    this.provideContext('UmbNotificationContext', new UmbNotificationContext())  
  }  
}
```



Context API

```
export class MyElement extends UmbElementMixin(LitElement) {  
  constructor() {  
    this.consumeContext('UmbNotificationContext', (_instance) => {  
      _instance.peek('positive', { data: { headline: 'Hello, Spark!' } } )  
    })  
  }  
}
```



Context API

```
▼ <umb-backoffice> flex (slot) == $0
  ▼ #shadow-root (open)
    <!-->
    ▶ <umb-backoffice-header> ... </umb-backoffice-header>
    ▼ <umb-backoffice-main>
      ▼ #shadow-root (open)
        <!-->
        ▼ <umb-router-slot>
          ▼ #shadow-root (open)
            <!-->
            ▼ <router-slot>
              ▶ #shadow-root (open)
              ▼ <umb-section> flex (slot)
                ▼ #shadow-root (open)
                  <!-->
                  <!--?lit$396892467$-->
                  ▶ <umb-section-sidebar> ... </umb-section-sidebar> flex
                  ▼ <umb-section-main>
                    ▶ #shadow-root (open)
                    <!--?lit$396892467$-->
                    <!--?lit$396892467$-->
                    ▼ <umb-router-slot id="router-slot"> (slot)
                      ▼ #shadow-root (open)
                      <!-->
                      ▼ <router-slot>
                        ▶ #shadow-root (open)
                        ▼ <umb-section-dashboards> flex (slot)
                          ▼ #shadow-root (open)
                          <!-->
                          <!--?lit$396892467$-->
                          <!--?lit$396892467$-->
                          ▶ <ui-tab-group id="tabs" role="tablist"> ... </ui-tab-group> flex
                          ▼ <ui-scroll-container id="scroll-container" tabindex="0">
                            ▶ #shadow-root (open)
                            ▼ <umb-router-slot id="router-slot"> (slot)
                              ▼ #shadow-root (open)
                              <!-->
                              ▼ <router-slot>
                                ▶ #shadow-root (open)
                                ▼ <umb-dashboard-health-check> (slot)
```

Provide: NotificationContext

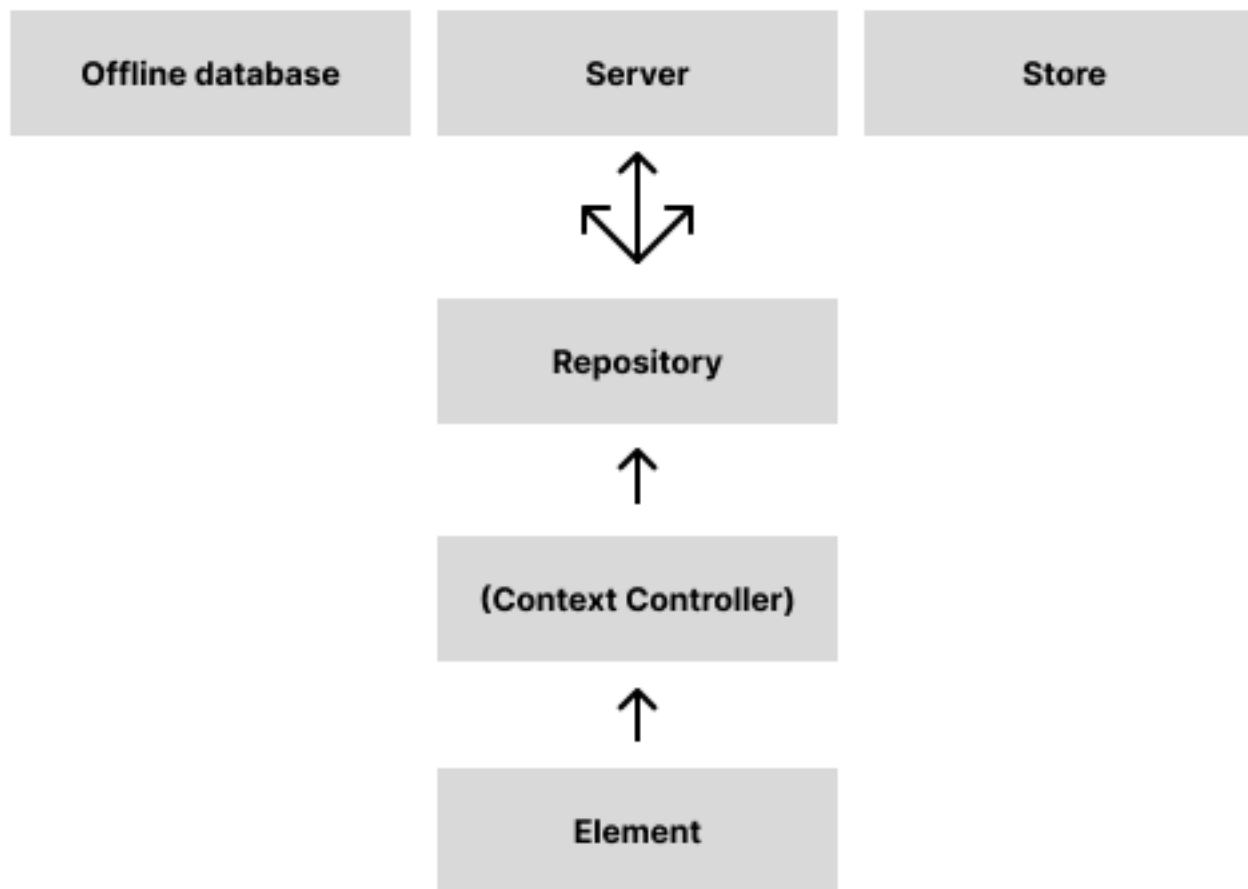
Override: NotificationContext

Consume: NotificationContext



Context
API

Expanding on the concept





Two New Concepts

Extension API

Auto register from server manifests

Manually register with JavaScript

Context API

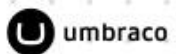
Event-driven context based on DOM elements

Provider & Consumer



DEMO

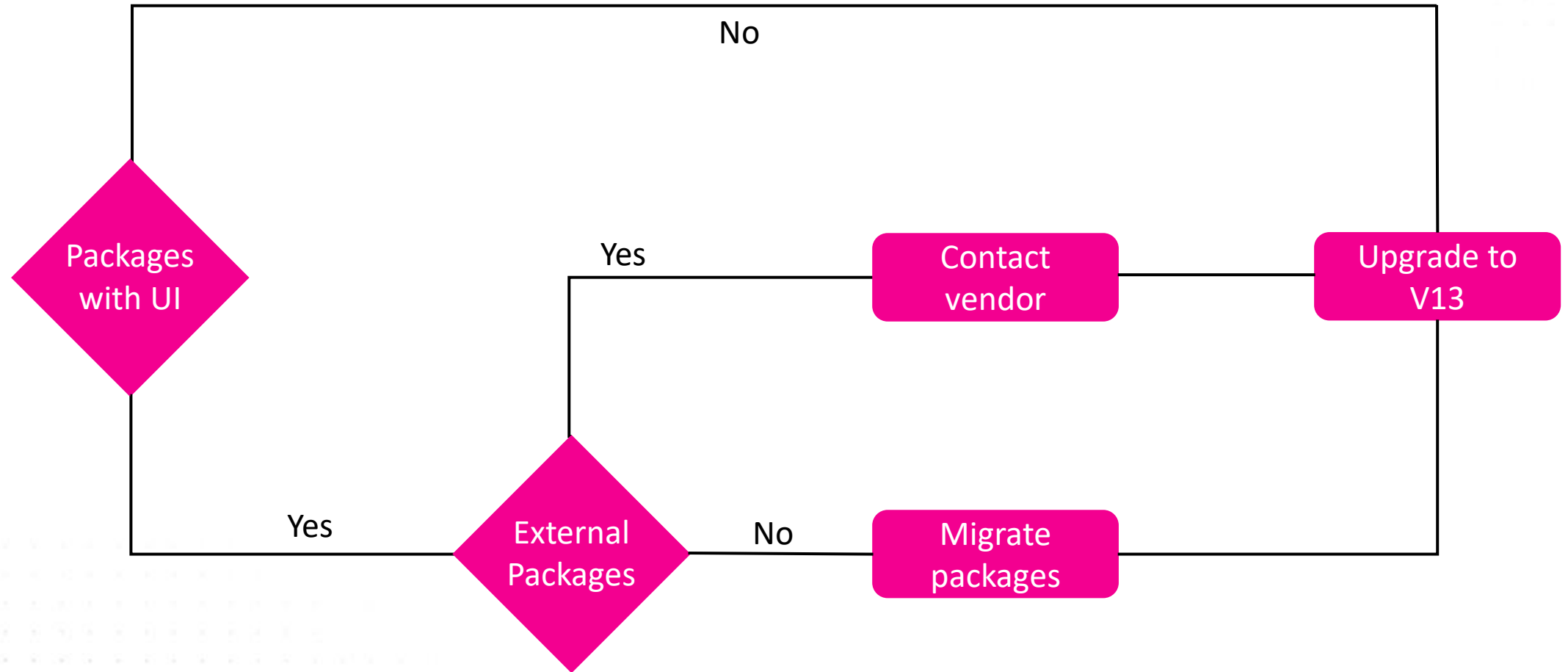
[Download demo files here](#)



#SPARKQA



High level update strategy

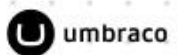




V13 Upgrade

Upgrade to
V13

System.Text.Json



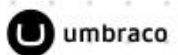
#SPARKQA



V13 Upgrade

- Media Picker (legacy) will be removed
- Grid Layout will be removed
- Nested Content will be removed

Upgrade to
V13



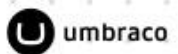
#SPARKQA



Milestones

- Beta → End June
- V13 RC → November 1st
- V13 Final → December 14th

Upgrade to
V13



#SPARKQA

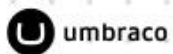


Not included in Beta*

- Documentation (except what's most needed for extensions)
- Search
- Preview
- Help
- ... and probably more

Upgrade to
V13

* This might change but is our best guess for now



#SPARKQA



Resources

Backoffice issue board:

<https://github.com/orgs/umbraco/projects/34>

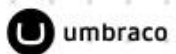
Backoffice repository:

<https://github.com/umbraco/Umbraco.CMS.Backoffice>

Umbraco UI Library:

<https://github.com/umbraco/Umbraco.UI>

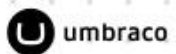
Try it out on the v13/dev branch today!



#SPARKQA



Thanks



#SPARKQA