UMBRACO SPARK #SPARKQA

Not your grandmother's headless

BY MIKKEL KELLER STUBKJÆR

$ whoami

Long-time passionate Umbraco developer and seasoned software architect.

- Head of Development at Novicell UK
- 10+ years of experience with Umbraco (2007)

Mikkel Keller Stubkjær
Head of Development
Novicell UK

mks@novicell.co.uk
+44 (0) 7756 634 808

https://www.linkedin.com/in/mikkelkeller/

Novicell
BUSINESS ONLINE

About 3 years ago

we embarked on an exciting adventure

Novicell
BUSINESS ONLINE

Decomposing the monolith

of large scale e-commerce

Novicell
BUSINESS ONLINE

Approach

using microservice architecture, headless api's
and assembling best of breed platforms and services

Novicell
BUSINESS ONLINE

We wanted freedom of choice

CMS system, PIM system, commerce platform, recommendation engine, personalisation engine etc.

Novicell
BUSINESS ONLINE

Choosing best of breed for our client

Contextual to the marked, the costumers, the competitors as well as the internal capabilities of their organisation

Novicell
BUSINESS ONLINE

We are
CMS Platform Agnostic

Novicell
BUSINESS ONLINE

Best of breed CMS

Part of that journey

To apply headless at scale

Novicell
BUSINESS ONLINE

Strategic requirements

- Performance and scalability as a first principle
  - Minimising request-time workload
  - Caching as a last resort
- Freedom to choose different .NET versions
- Freedom to choose the frontend tech stack
- Reusability, flexibility and extendability
- Headless APIs

Unfortunately

No mature headless offering was available at the time

Novicell
BUSINESS ONLINE

Without headless

- No ability to isolate the workload of the website from the workload of the CMS
  - No control of ressource consumption within the website runtime (eg. database requests from within Umbraco etc.)
  - The performance and scalability of the website is affected by Umbraco
- Dependencies on the .NET version of Umbraco
- Dependencies on the frontend framework of Umbraco (ASP.NET mvc)

Luckily

We are builders

Novicell
BUSINESS ONLINE

Actually

Even with a headless api, there will still be a direct performance dependency on the CMS

Novicell
BUSINESS ONLINE

Why #1

Direct performance dependency on the workload

Novicell
BUSINESS ONLINE

Request-time transformations

When a controller generates a view model by transforming node data. Traversing the content tree, aggregating data from multiple nodes, doing dictionary look-ups etc.

Novicell
BUSINESS ONLINE

What if

we could prepare data in the desired form

and update it whenever changes occur

Novicell

BUSINESS ONLINE

Publish-time transformations

When a publish action generates a view model by transforming node data. Traversing the content tree, aggregating data from multiple nodes, doing dictionary look-ups etc.

Novicell
BUSINESS ONLINE

Publish-time transformations

- Triggered on publish action in the Umbraco backoffice
  - Listening to Umbraco events
- Generating view models
  - Tailored for a specific page or purpose
- Storing the generated view model in fast cache-level storage
  - Without the need for TTLs and cache invalidation (hardest thing ever)
  - Whenever content changes, a new publish event triggers the existing model to be overridden

Why #2

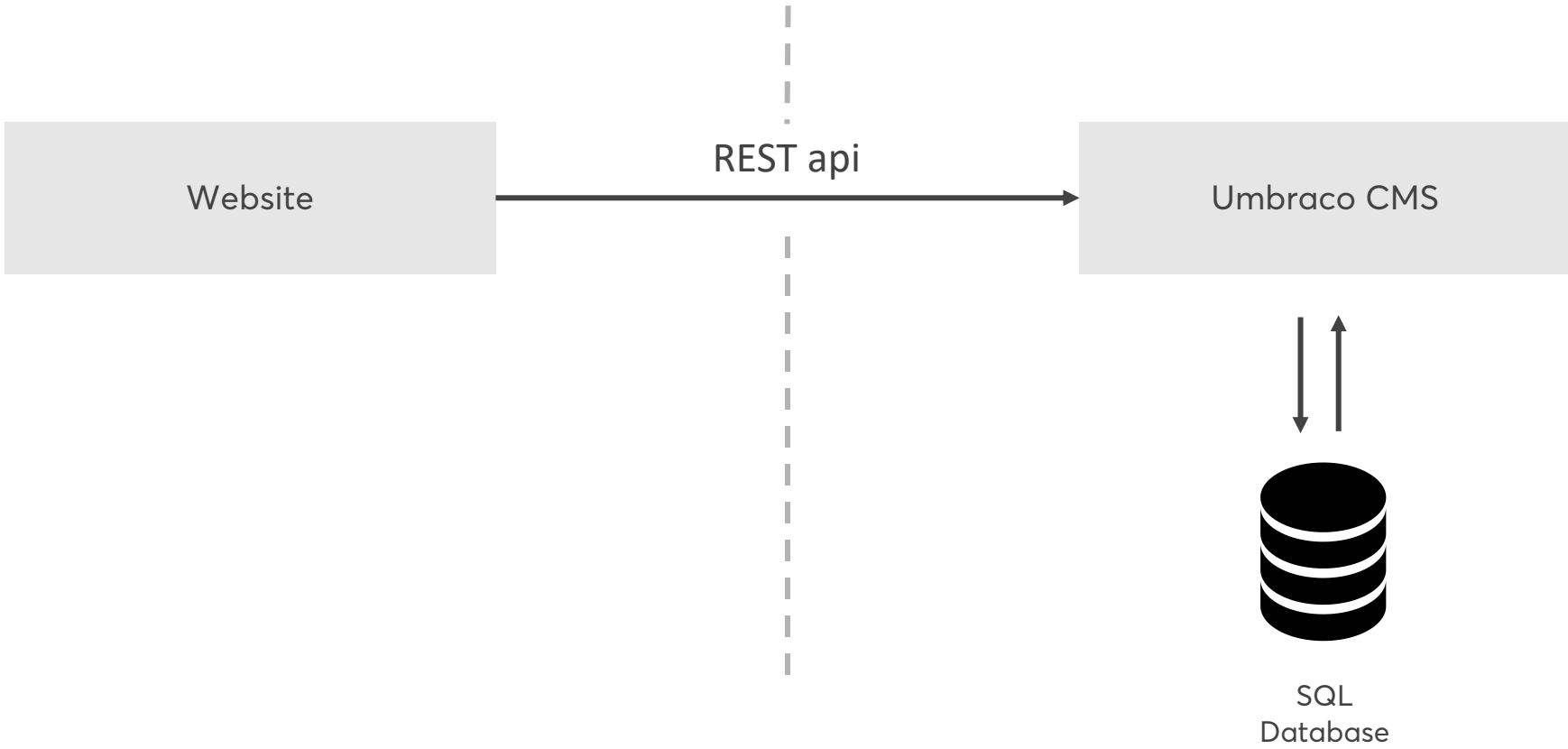
Direct performance dependency on the CMS

Novicell
BUSINESS ONLINE

Direct Headless

Separation of website and CMS

But direct dependency on the CMS



Indirect Headless

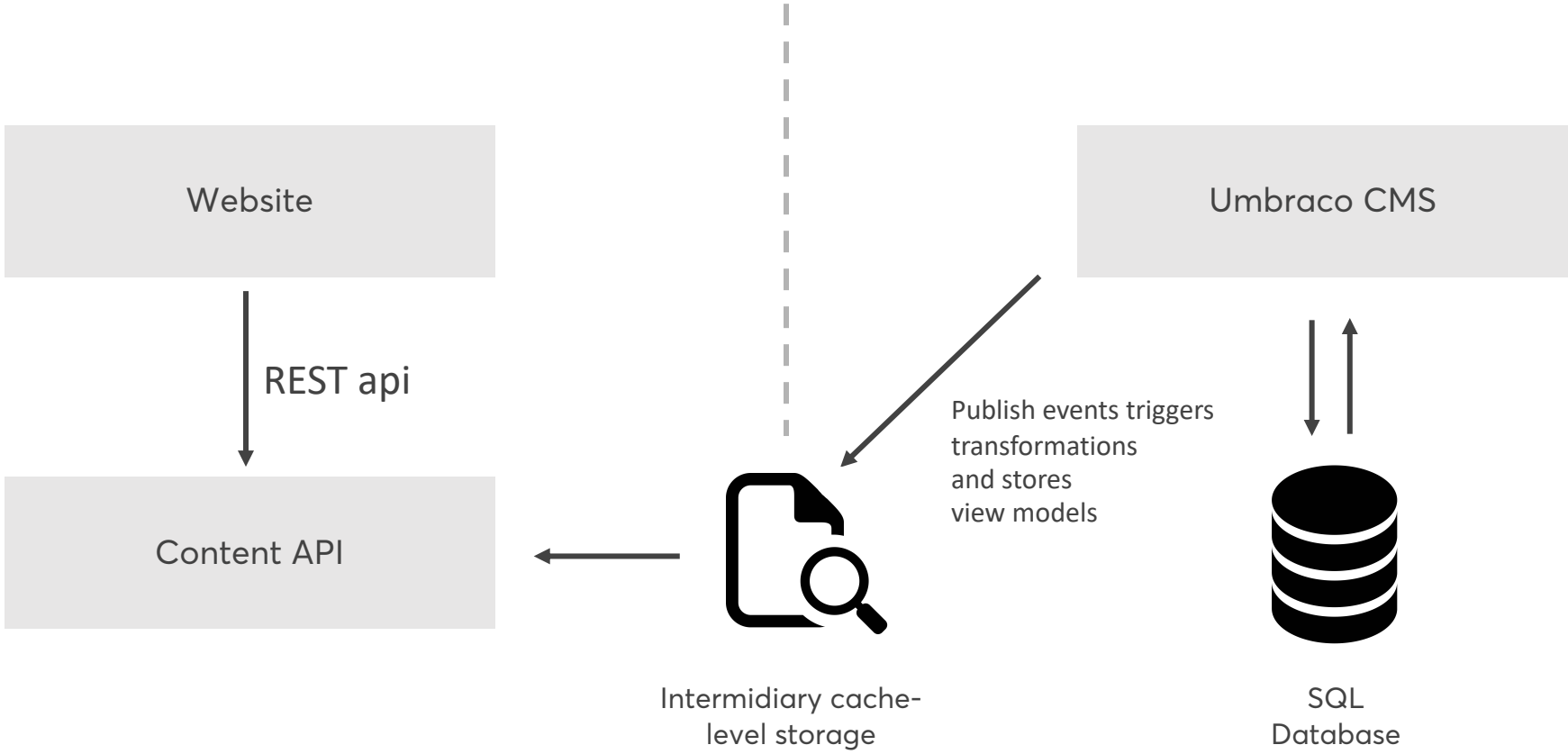
Serving up the view models directly from intermediary storage

Novicell
BUSINESS ONLINE

Indirect Headless

Complete separation of website and CMS

No direct performance dependency on the CMS



Novicell
BUSINESS ONLINE

Effectively

Minimising workload at request-time by displaces the workload from request-time to publish-time

Novicell
BUSINESS ONLINE

Effectively

Moving the workload from our website visitor to the editor

Novicell
BUSINESS ONLINE

Effectively

Removing the direct performance dependency on the CMS

Novicell
BUSINESS ONLINE

Remember

We wanted freedom of choice

Novicell
BUSINESS ONLINE

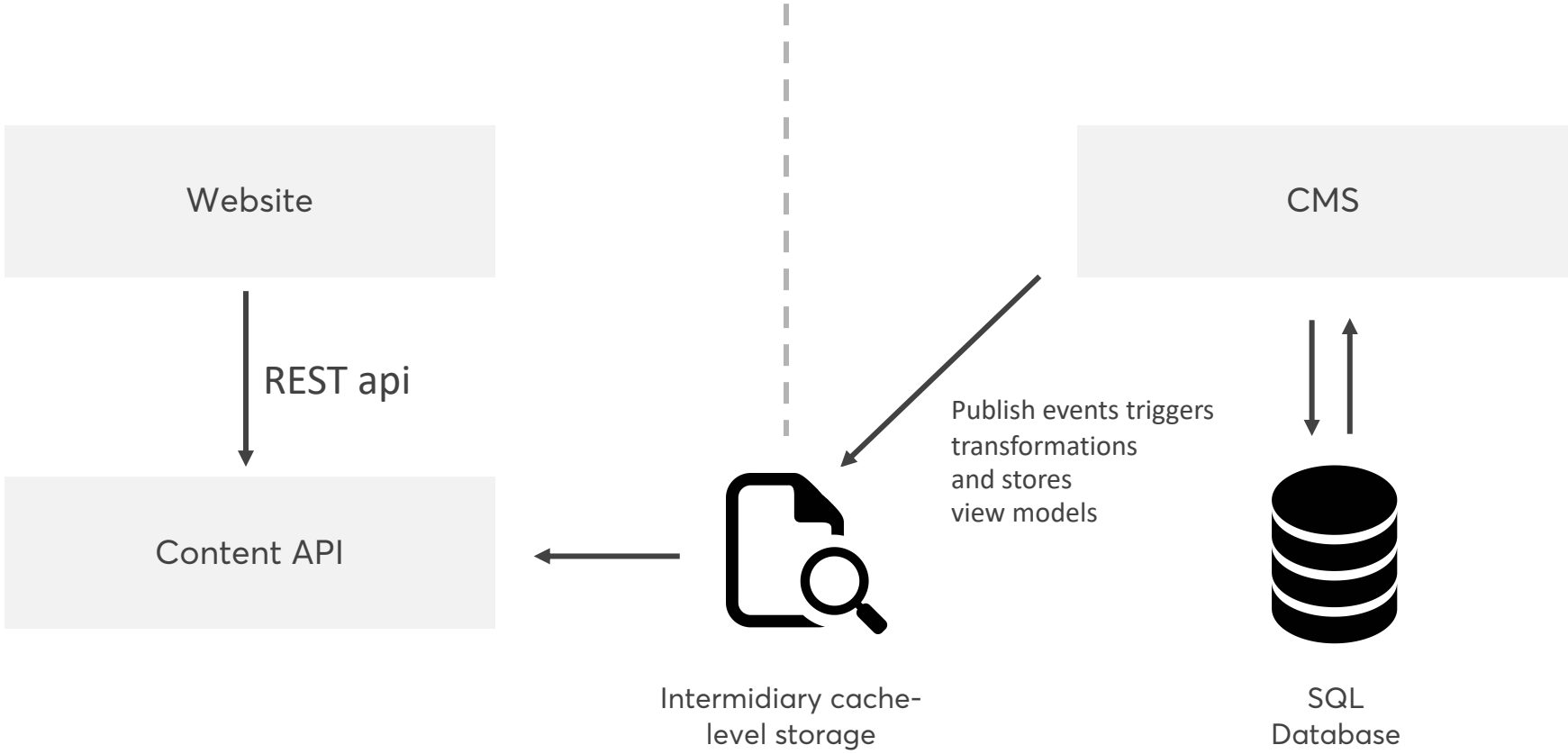
Building

A CMS agnostic
headless transformation engine

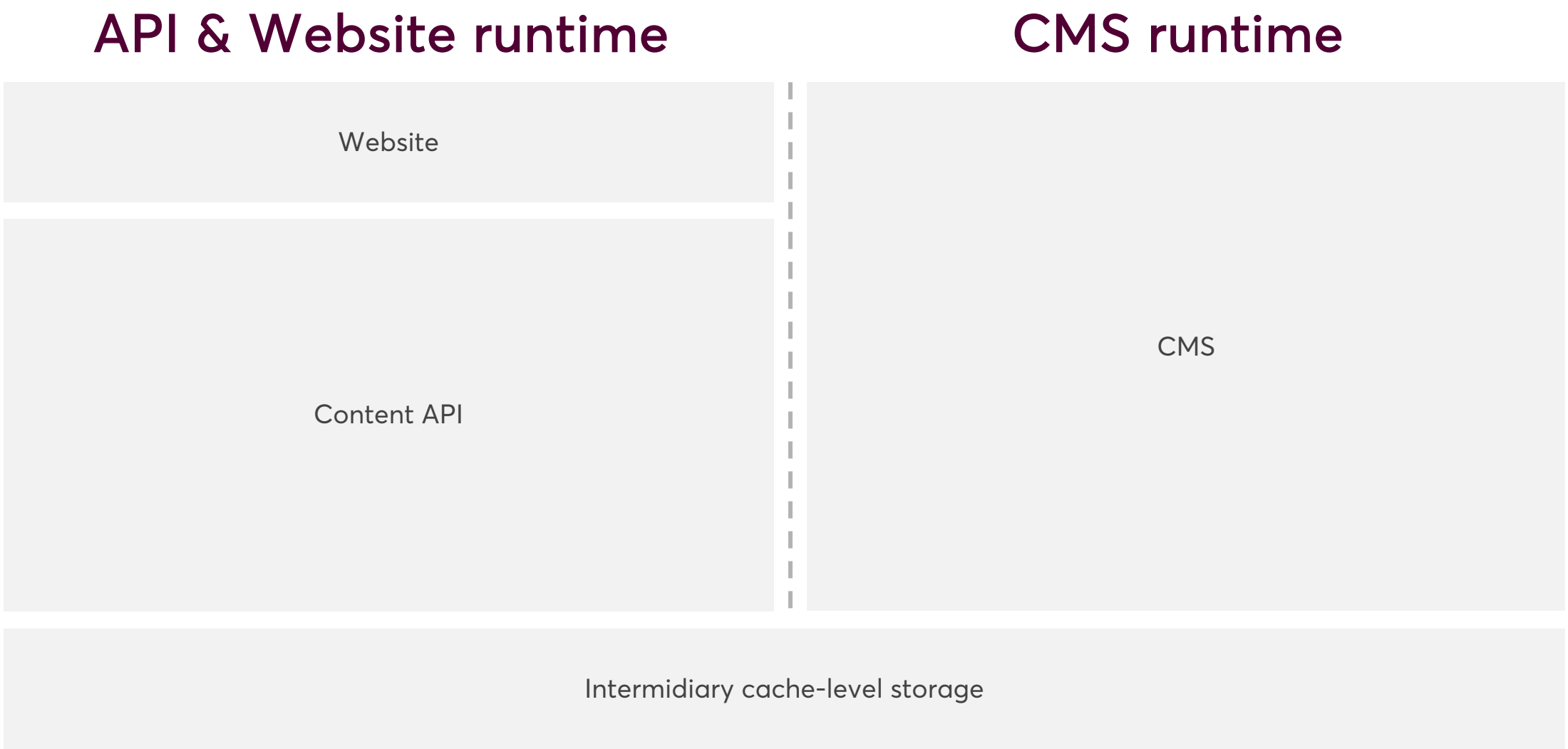
Novicell
BUSINESS ONLINE

API & Website runtime

CMS runtime



Novicell
BUSINESS ONLINE


API & Website runtime
CMS runtime
Website
Content API
CMS
Intermidiary cache-level storage

API & Website runtime

Website

Content API

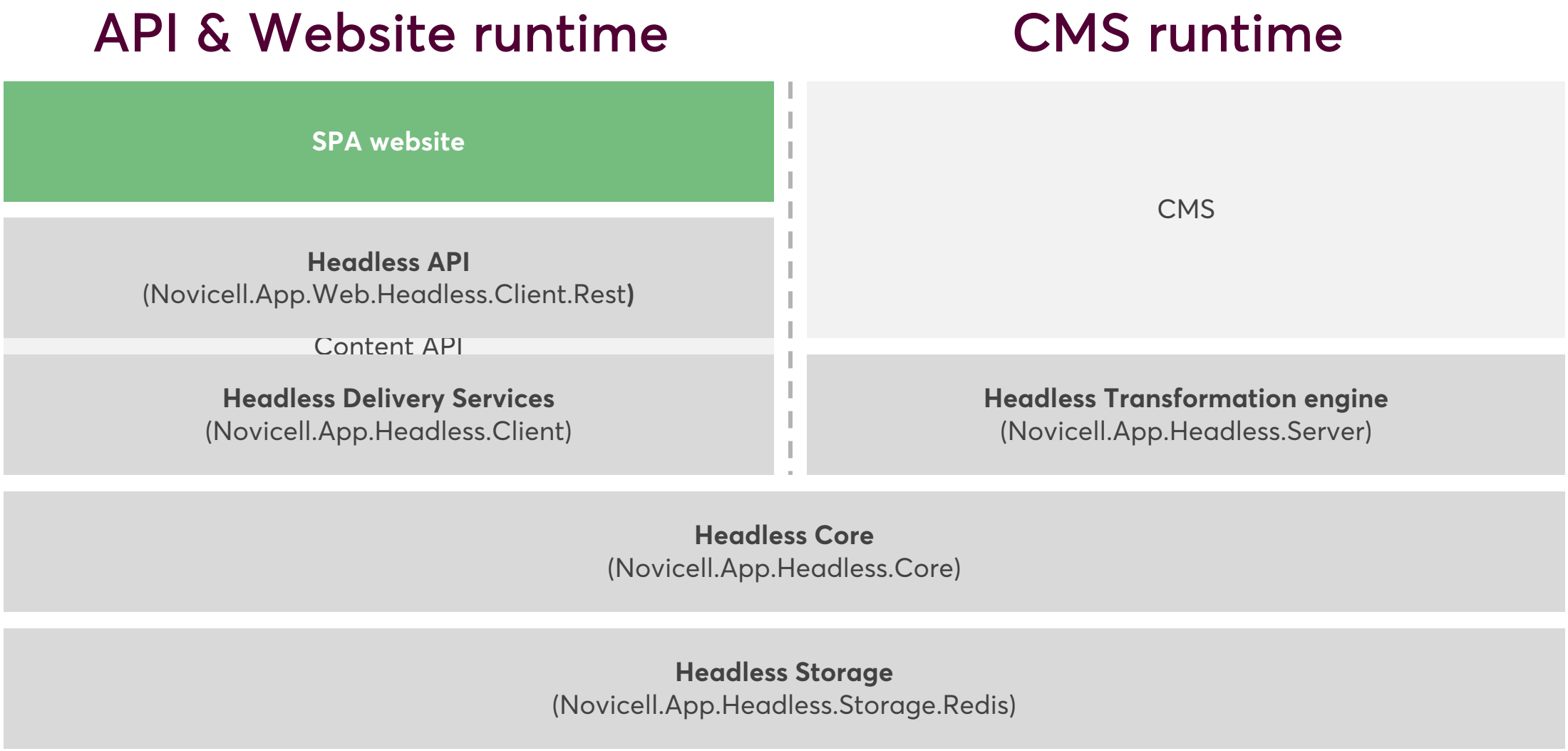
CMS runtime

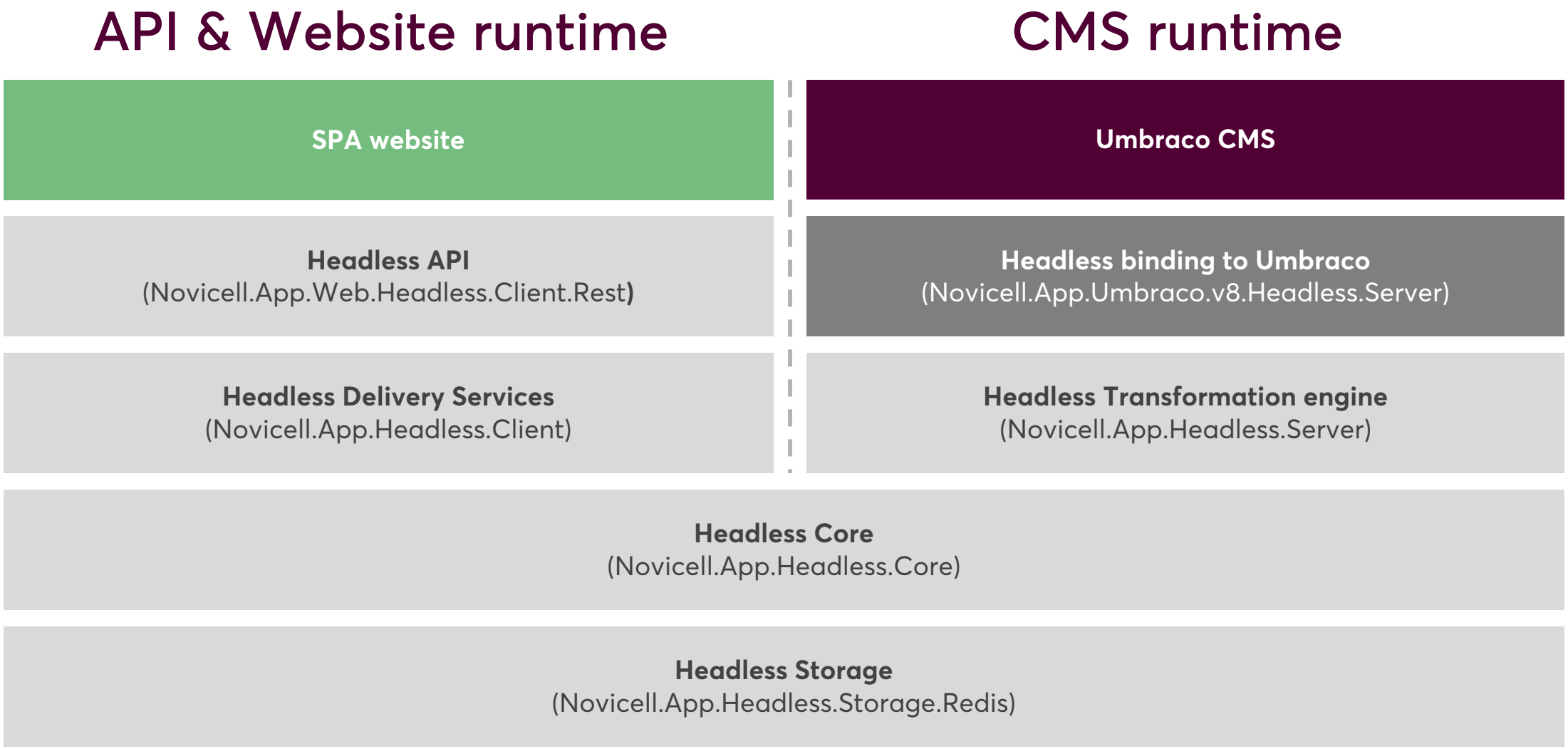
CMS

Headless Transformation engine
(Novicell.App.Headless.Server)

Headless Core
(Novicell.App.Headless.Core)

Headless Storage
(Novicell.App.Headless.Storage.Redis)


API & Website runtime
CMS runtime
SPA website
CMS
Headless API
(Novicell.App.Web.Headless.Client.Rest)
Content API
Headless Delivery Services
(Novicell.App.Headless.Client)
Headless Transformation engine
(Novicell.App.Headless.Server)
Headless Core
(Novicell.App.Headless.Core)
Headless Storage
(Novicell.App.Headless.Storage.Redis)


API & Website runtime
CMS runtime
SPA website
Umbraco CMS
Headless API
(Novicell.App.Web.Headless.Client.Rest)
Headless binding to Umbraco
(Novicell.App.Umbraco.v8.Headless.Server)
Headless Delivery Services
(Novicell.App.Headless.Client)
Headless Transformation engine
(Novicell.App.Headless.Server)
Headless Core
(Novicell.App.Headless.Core)
Headless Storage
(Novicell.App.Headless.Storage.Redis)

Time for some code

Novicell
BUSINESS ONLINE

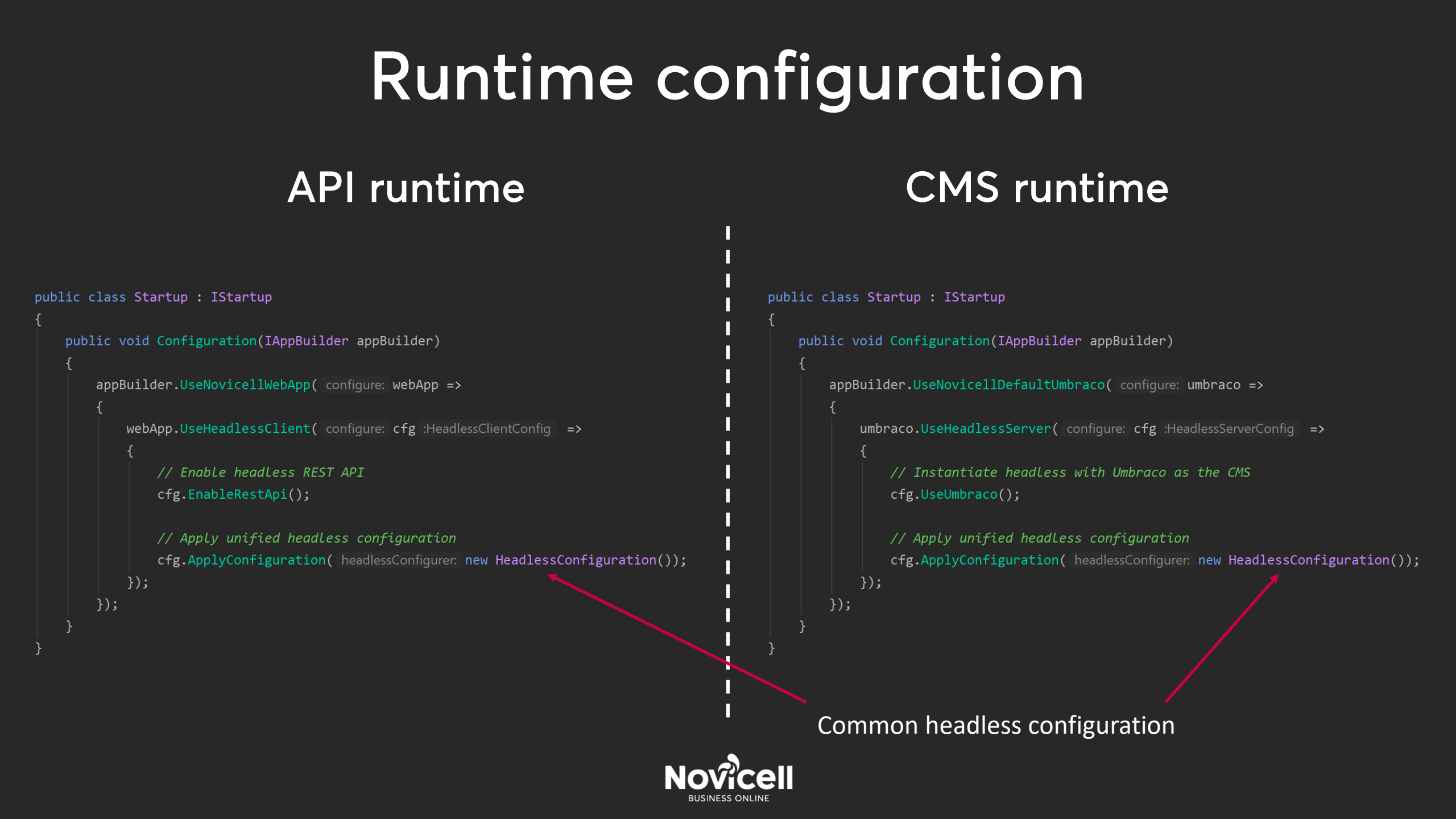
Runtime configuration

API runtime

```
public class Startup : IStartup
{
    public void Configuration(IAppBuilder appBuilder)
    {
        appBuilder.UseNovicellWebApp( configure: webApp =>
        {
            webApp.UseHeadlessClient( configure: cfg :HeadlessClientConfig  =>
            {
                // Enable headless REST API
                cfg.EnableRestApi();

                // Apply unified headless configuration
                cfg.ApplyConfiguration( headlessConfigurer: new HeadlessConfiguration());
            });
        });
    }
}
```

CMS runtime

```
public class Startup : IStartup
{
    public void Configuration(IAppBuilder appBuilder)
    {
        appBuilder.UseNovicellDefaultUmbraco( configure: umbraco =>
        {
            umbraco.UseHeadlessServer( configure: cfg :HeadlessServerConfig  =>
            {
                // Instantiate headless with Umbraco as the CMS
                cfg.UseUmbraco();

                // Apply unified headless configuration
                cfg.ApplyConfiguration( headlessConfigurer: new HeadlessConfiguration());
            });
        });
    }
}
```



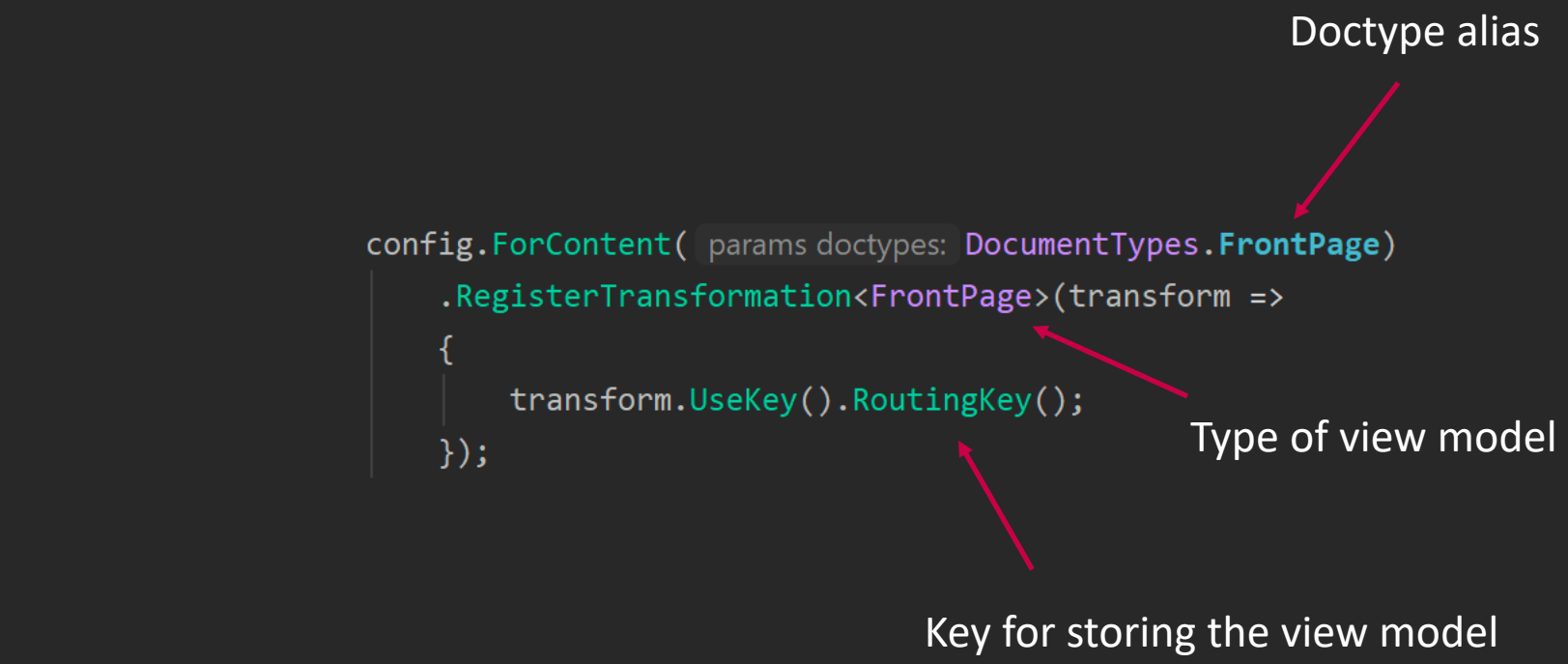
Novicell
BUSINESS ONLINE

Common headless configuration

```
public class HeadlessConfiguration : IHeadlessConfigurer
{
    public void Configurer(HeadlessConfigurator config)
    {
        StorageConfiguration.Register(config);

        // Global elements
        ConfigureGlobalElements(config);

        // Pages
        ConfigurePages(config);

        // Compositions
        ConfigureCompositions(config);

        // Items
        ConfigureItems(config);
    }
}
```

Novicell
BUSINESS ONLINE

Configuring a transformation



Novicell
BUSINESS ONLINE

The view model

```
public class FrontPage : ViewModel
{
    public string Headline { get; set; }
    public IGridViewModel GridContent { get; set; }
}
```

Novicell
BUSINESS ONLINE

The transformation

View model type

```
public class FrontPageTransformation : IContentTransformation<FrontPage>
{
    public FrontPage Transform(IContentModel content, ITransformationHelper helper, IUrlProvider urlProvider)
    {
        return new FrontPage
        {
            Headline = content.GetPropertyValue<string>( alias: "headline"),
            GridContent = content.TransformPropertyValue<IGridViewModel>( alias: "grid", helper, urlProvider)
        };
    }
}
```

Transfer values to view model

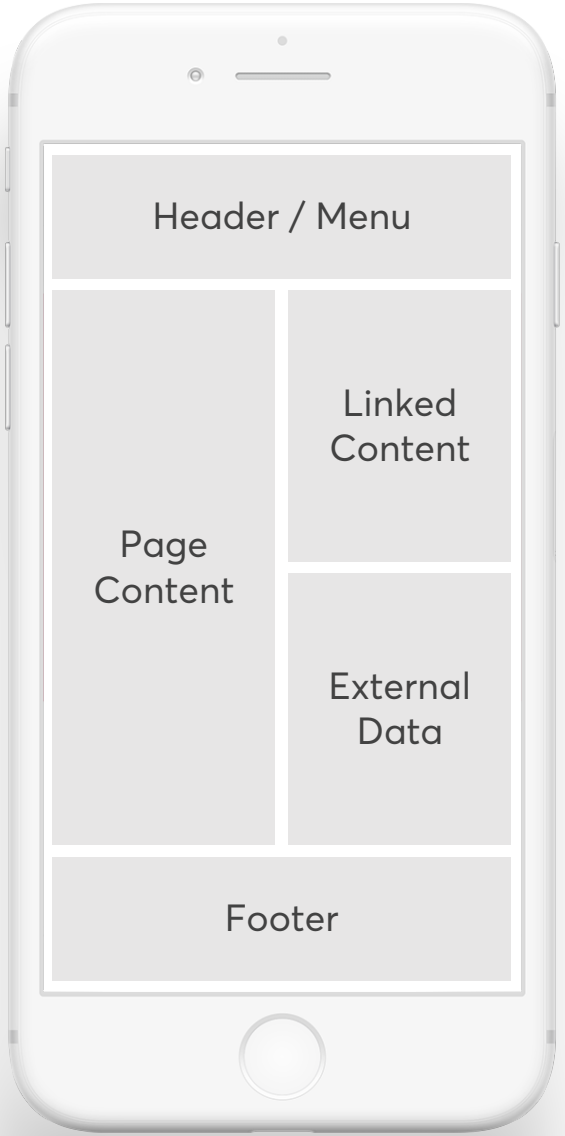
Novicell
BUSINESS ONLINE

Re-visiting transformations

Novicell
BUSINESS ONLINE

Anatomy of a webpage

- Global elements
  - Header, Menu, Footer etc.
- Routable content
  - Articles, Pages etc.
- Linked Content (aggregated)
  - News, Faqs, emplyees etc.
- External Data
  - Product information, prices, stock count etc.



Novicell
BUSINESS ONLINE

Anatomy of a webpage

- Global elements
  - Header, Menu, Footer etc.
- Routable content
  - Articles, Pages etc.
- Linked Content (aggregated)
  - News, Faqs, employees etc.
- External Data
  - Product information, prices, stock count etc.

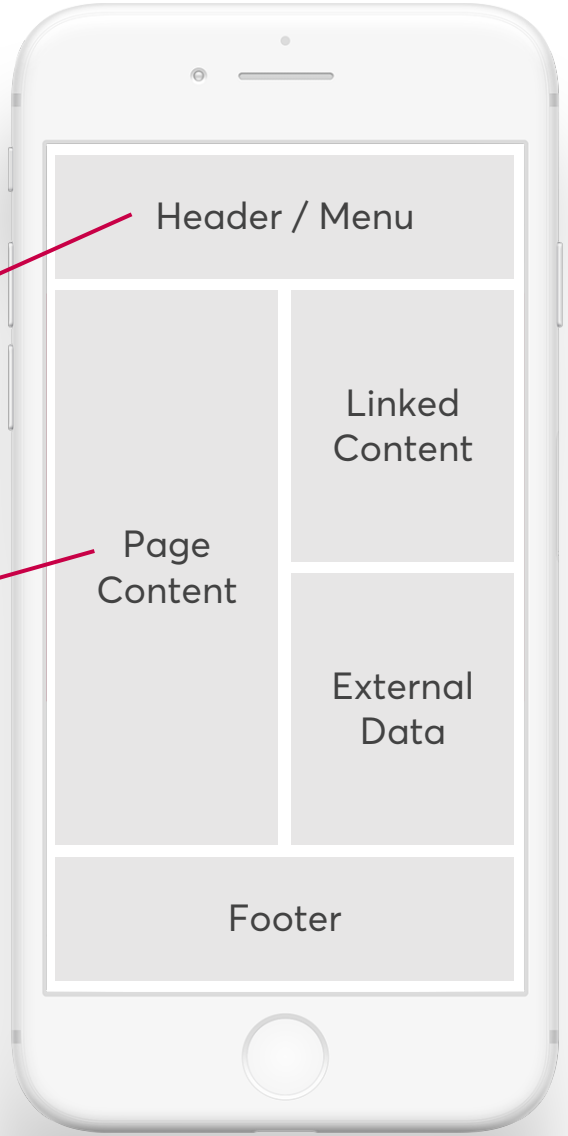
Translates to

View Models

Multiple transformations

2 transformations are triggered when a FrontPage node is published:

- Menu view model
- FrontPage view model



Novicell
BUSINESS ONLINE

Configuring global transformation

Register transformation on multiple doctypes

```
config.ForContent(
        params doctypes: DocumentTypes.FrontPage,
        DocumentTypes.TextPage)
    .RegisterTransformation<Navigation>(transform =>
    {
        transform.UseKey().DynamicContentKey(DynamicKeyConstants.NavigationKey);
    });
```

Navigation model key

```
public class DynamicKeyConstants
{
    public static Func<IContentModel, string> NavigationKey => $"navigation:{x.GetSiteNodeId().ToString()}";
}
```

Novicell
BUSINESS ONLINE

A single request for aggregated content

Linking global elements to routable elements

Novicell
BUSINESS ONLINE

Linking global view models

Going back to the FrontPage doctype

```
config.ForContent( params doctypes: DocumentTypes.FrontPage)
    .RegisterTransformation<FrontPage>(transform =>
    {
        transform.UseKey().RoutingKey()
            .WithDynamicPartialKeys(DynamicKeyConstants.NavigationKey);
    });
```

Linking a global element key to the FrontPage view Model

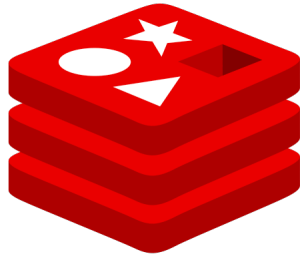
Novicell
BUSINESS ONLINE

Intermediary storage

And internal data structures

Novicell
BUSINESS ONLINE

Underlaying storage



redis

- Key / value storage
  - Efficient for retrieval
- Json in NoSQL storage
  - Efficient for serialization and deserialization
- Utilising fast cache-level storage
  - Originally build using Redis for performance
  - Now also supporting CosmosDB for replicated datasets
  - And Examine, Lucene, SQL server for scaled down instances

Routing and content keys

Routing and Content is divided into two distinct key/value storages

- Routing storage:
  - keys are urls (or paths) eg. "domain.co.uk/", "/" or "/en/"
  - Only routable content has a key/value pair in the routing storage

- Content storage:
  - keys are static eg. "menu" or dynamic eg: $"navigation:`{x.SiteId()}`"
  - All view models have a key/value pair in the content storage

Novicell
BUSINESS ONLINE

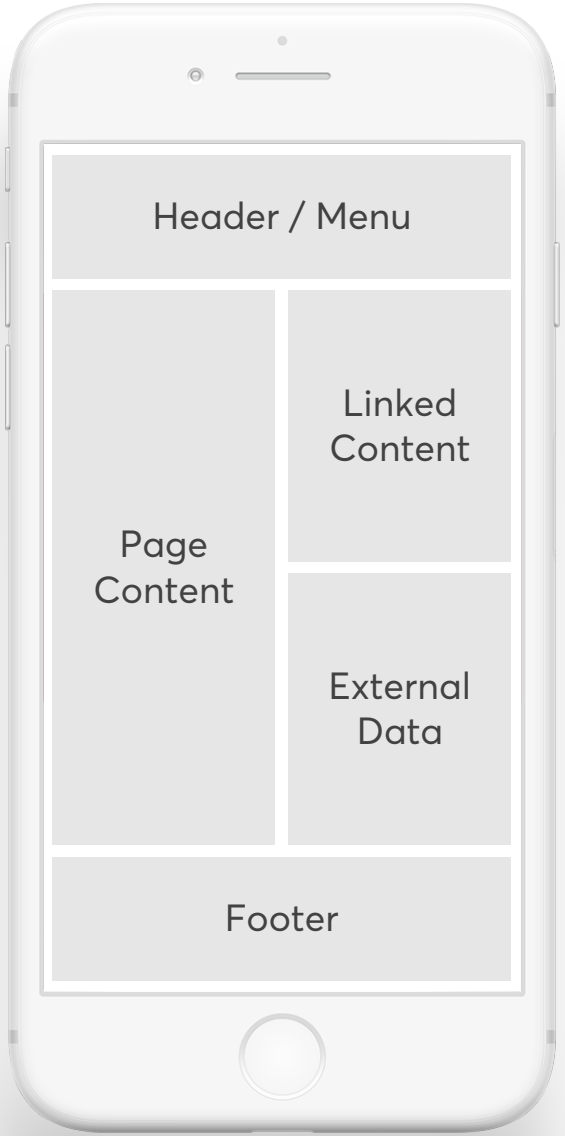
Routing and content keys

Routing storage

- Key: "/"

Content storage

- Key: "1145"
- Key: "menu"
- Key: "footer"



Novicell
BUSINESS ONLINE

Routing

- Primary key
  - Key: "/"
- Routable content key
  - ContentKey: "1143" (eg. Frontpage)
- All content keys (including globals)
  - ContentKeys: ["1142", "menu", "footer"]

Novicell
BUSINESS ONLINE

Meta model


```
{
  "$type": "Novicell.App.Headless.Core.Models.Route,
  Novicell.App.Headless.Core",
  "RoutingKey": "/",
  "Url": "http://novicell.dk/",
  "Controller": "frontpageController",
  "Restricted": false,
  "ContentKeys": {
    "$type": "System.String[], mscorlib",
    "$values": [
      "1143",
      "menu",
      "footer"
    ]
  },
  "LinkedContentKeys": {
    "$type": "System.String[], mscorlib",
    "$values": []
  },
  "PublishAt":   null,
  "UnpublishAt":   null,
  "Culture": "da",
  "RedirectKey":   null,
  "ContentKey": "1143",
  "Path": "-1,1133,1143"
}
```

Content

- Primary key
  - Key: "1143"
- Meta information
  - Culture,
  - Scheduled publishing
  - Etc.
- View Model

Novicell
BUSINESS ONLINE

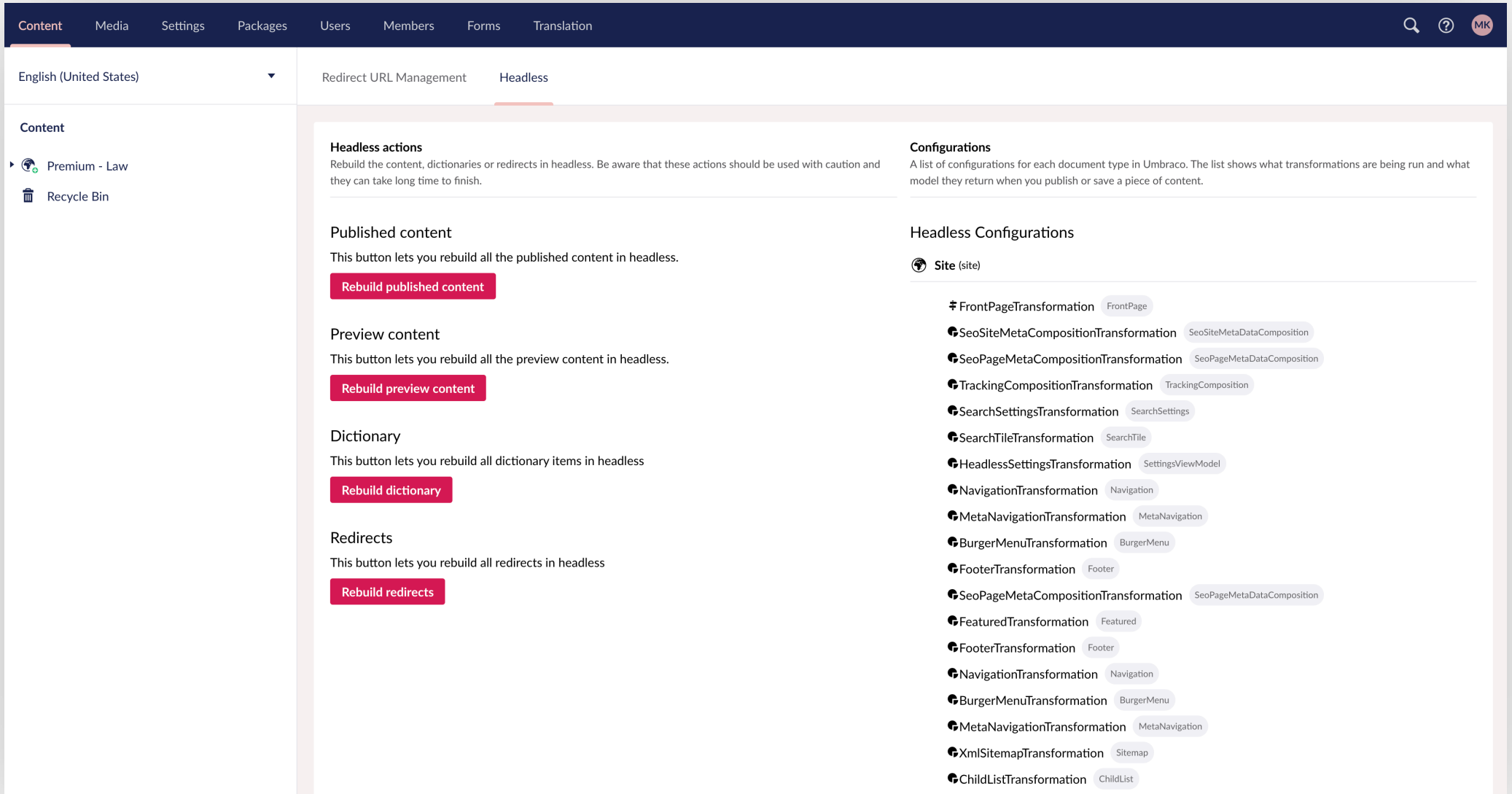
Meta model


```
{
  "$type":
  "Novicell.App.Headless.Core.Models.ContentMetaModel`1[
  [Novicell.App.Headless.Core.Models.IViewModel,
  Novicell.App.Headless.Core.Models]],
  Novicell.App.Headless.Core",
  "Key": "1143",
  "ViewModel": {
    "$type":
    "Web.v8.Mvc.Website.Core.Models.FrontpageModel,
    Web.v8.Mvc.Website.Core",
    "ActiveFrom": "0001-01-01T00:00:00",
    "ActiveTo": "9999-12-31T23:59:59.9999999",
    "Headline": "A typical headline",
    "body": "<b>body</b>"
  },
  "Culture": "da",
  "PublishAt": null,
  "UnpublishAt": null,
  "LastUpdatedAt": "2019-09-12T21:29:33.0904614+02:00",
  "ContainsTimedVariations": false,
  "LinkedContentKeys": {
    "$type": "System.String[], mscorlib",
    "$values": []
  }
}
```

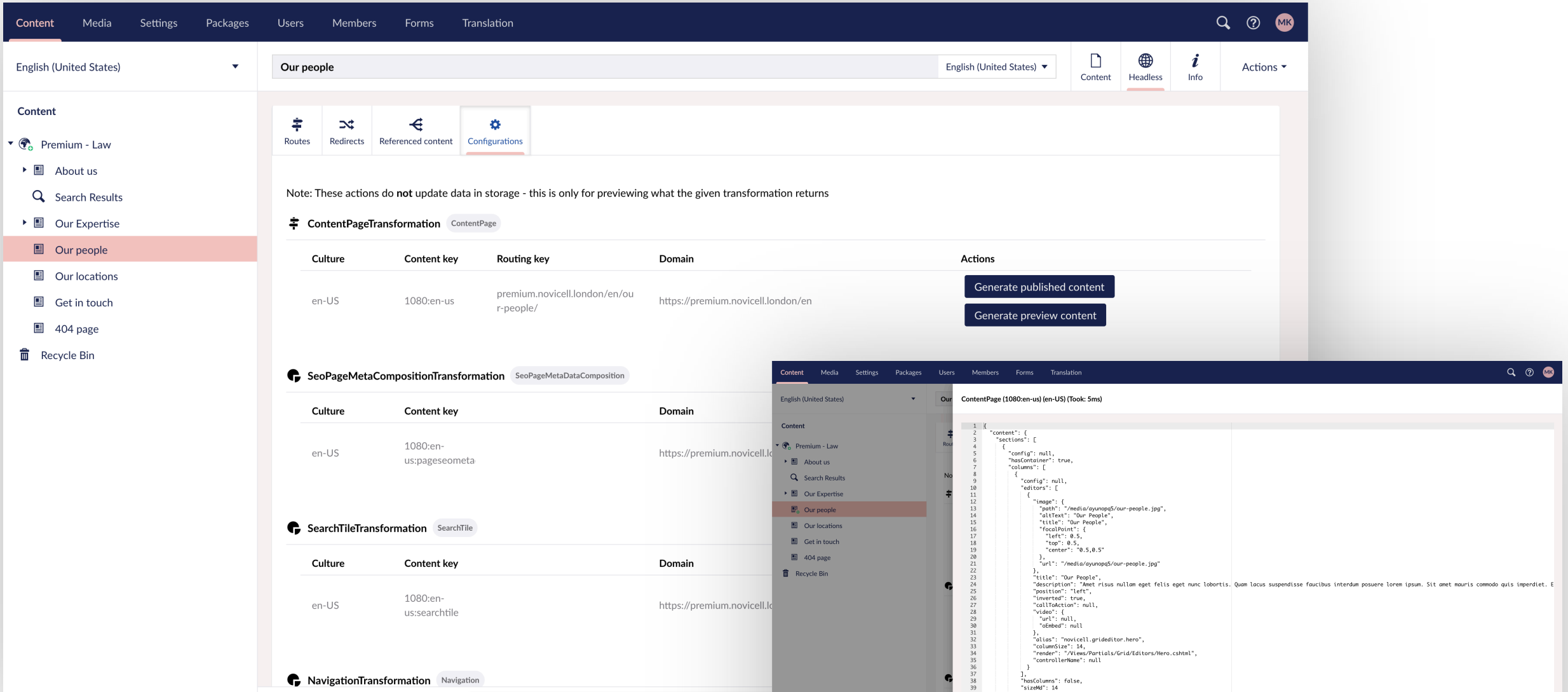
Headless backoffice extensions

Novicell
BUSINESS ONLINE

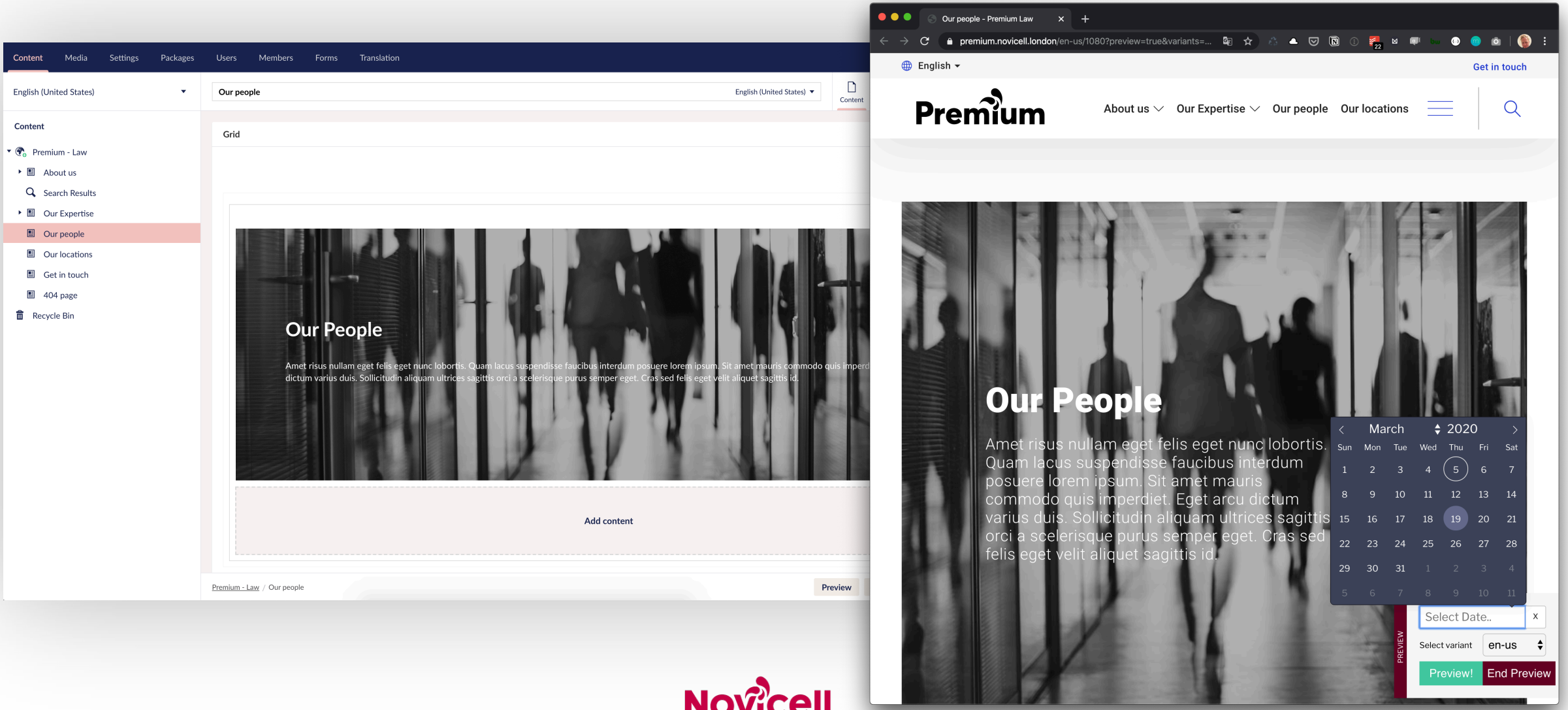
Headless Dashboard



Headless Content App



Headless Preview



Novicell
BUSINESS ONLINE

Headless Preview and storage

- Extra set of storages
  - Published storage
    - Route storage
    - Content storage
  - Preview storage
    - Route storage
    - Content storage

- Preview storage is routed by id (like native preview in Umbraco) and rewritten in IIS
- Transformations are triggered on save event

Novicell
BUSINESS ONLINE

Other supported features

- Dictionary service
- Routing service
- Strong typed grid transformations
- Search based on search models and view model annotations
- Taxonomy filtering, also based on view model annotations
- Linked Content
- External Linked Data
- Content listeners and View Model listeners

But what about Umbraco Heartcore?

Fantastic, use it, we didn't have the luxury

Novicell
BUSINESS ONLINE

However, we still need

- Full control of the Umbraco setup
- The ability to ship custom code with Umbraco
- Performance without the need for caching with publish-time transformations and fast cache-level intermediary storage
- Aggregating Umbraco content with external data
- Since we focus on websites and not Apps, IoT or chatbots, we can actually still allow the editor to preview the website

THANK YOU!

Mikkel Keller Stubkjær
Head of Development
Novicell UK
mks@novicell.co.uk
+44 (0) 7756 634 808

Novicell
BUSINESS ONLINE

+44 (0)20 8144 8142 hello@novicell.co.uk