TECH DOCUMENTATION

Feb 2018

M

RCoA
Royal College of Anaesthetists

INTRODUCTION

You asked us to make clear recommendations for a number of parts of a new digital platform and in this part of the report we'll focus on doing so from a technical perspective.

As part of the production of this report we spoke to a number of key stakeholders in your organisation, digested a range of internal documentation and previous strategic advice. From this research we determined the following high level needs:

- A platform capable of delivering personalised content to both the general public and to members of the RCoA
- A platform that's easy to use and flexible, one that streamlines and supports the creation of content for web, mobile & digital documents
- A platform that centralises user login removing the need for multiple usernames and passwords
- A platform that looks and feels consistent across the range of services that it offers

In addition we believe that as an organisation you should be looking to describe a platform that is designed and built to accommodate change. What we mean by this is that your platform should be designed in a way to support the addition and removal of components over time as a first order capability. This capability seeks to ensure that although you might look to add and remove individual products and services from your digital platform over time, the underlying architecture that describes how these products integrate with each other remains stable.

In order to make clear recommendations to you about how best to meet these needs, we undertook the following activities

- We reviewed your current digital landscape and provided a high level critique of the as is state
- We described a future state architecture that provides an aspiration for future programmes of work
- We performed a Web Content Management System product selection
- We performed a Single Sign On framework product selection

KEY RECOMMENDATIONS

Supporting the delivery of personalised content

There are a number of capabilities required to deliver a good personalised experience for your users. They can be described as:

- Single user view
- Content and data tagging
- Integration between your systems and the single user view
- Deliver tailored content based on users behaviour in context

Single user view

Perhaps the most important of these capabilities is a coherent, centralised view of your users. A single view of your members, the systems they have access to and their preferences.
Whilst SSO primarily provides the ability to consolidate user logins - so that a member only needs to remember one set of credentials - it also presents the opportunity to centralise the data that is gathered and stored against members accounts regardless of where the data is to be ultimately stored. As part this report we've described a solution that provides centralised user profile management suitable to underpin the capability to provide personalised content for your users and members.

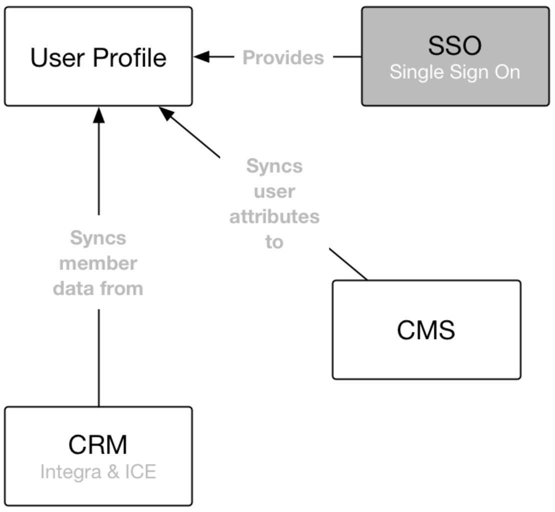
Content and data tagging

In order to be able to serve relevant content to our users we need first to have categorised and tagged that content in a way that ties it to a particular group or segment of users. For content, this takes place in the CMS and highlights a couple of important capabilities we need to look for when selecting a CMS. The first of these is metadata management - described as the ability to manage metadata such as controlled values, tags, etc. The second important capability is the ability to manage content using a modular content modelling paradigm. In simpler terms this means the ability to create and manage content at a lower level of granularity than a web page.

Integration between your systems and the single user view

Now with a view of the user, their preferences and the systems they have access to, and a repository of content that has been categorised and tagged, we can start to look at how to tie things together.

Our first recommendation is to describe an integration pattern where the single user view becomes the central integration point for any user information or data. What this means in practice is that systems such as the CMS and CRM don't integrate with each other directly, rather that they both talk to the Single User View instead. This integration pattern is described in more detail in the SSO Solution Design document.



Our second recommendation is to make data from your systems available over RESTful APIs. This process should start with the content stored in the CMS and with the data stored in Single User View and extend to systems such as the events and exams management systems. Providing access to this data over a RESTful API describes the way that we could, for example, render information about events users have signed up for in a dashboard view delivered from the CMS.

Deliver tailored content based on users behaviour in context

The personalisation we've described to this point is driven from the data that you've already captured and stored from your users. We typically refer to this type of personalisation as **explicit**. It is also possible, however, to make decisions about the content to deliver to your users even if you don't know anything explicitly about them. We typically call this **implicit** personalisation and it relies on the ability to monitor the behaviour of users on your site. For example, if a user repeatedly searches for and browses content about having an operation, you may be able to infer that the user is a patient and start to deliver content tailored to this understanding. Of the content management systems we have assessed only one, EPIServer provides, any native implicit personalisation functionality. All of the others rely on integrations with external products. For example Drupal on Acquia relies on the use of Acquia Lift to provide this capability.

Our recommendation here is that this kind of implicit personalisation should be on your roadmap, and that you should plan for its introduction once you have embedded the capability to deliver explicit personalised content.

Delivering a platform that supports change

In order to deliver an underlying platform that provides stability over a longer period, we believe you should invest in describing an architecture that supports change as a first order capability. What this means in practice is a platform that is capable of withstanding the change of the introduction of a new product or the withdrawal of a legacy one, without significant disruption. In order to deliver this capability we recommend that you describe a platform architecture that includes the following patterns.

Introduce an API layer and make service data available over APIs

Introducing a centralised API management tier that provides a facade over the APIs provided by your existing products allows you to do a number of things.

- Firstly it makes data available for delivery channels other than the Web - mobile is a good example here
- Secondly it makes data from your systems available so that they can be integrated together, this provides opportunities for you to provide real value for your users
- Thirdly by introducing a facade over existing APIs you provide a way to add and remove services from your platform without affecting the platforms underlying capability

Introduce a User Experience framework

Describing a framework for how your users interact with all of your services, not just the pages that the content managed website provides a point of separation between the user experience that you provide to your users across your digital engagement platform and it's various implementations. In practice this means the creative design, user experience and front end build (HTML, CSS and Javascript) takes place as a separated project or phase from the main CMS build. By designing and building this part of the work separately we recognise the ability to progress and change our user experience and front end build with impacting the CMS implementation to the same extent. As part of this report we've made two recommendations with respect to constituent parts of a user experience framework.

The first is to utilise a design framework called Atomic Design - http://bradfrost.com/blog/post/atomic-web-design/

Atomic design

Atomic design is methodology for creating design systems. There are five distinct levels in atomic design:

- Atoms
- Molecules
- Organisms
- Templates
- Pages

Atoms

Atoms are the basic building blocks of matter. Applied to web interfaces, atoms are our HTML tags, such as a form label, an input or a button.

Molecules

Molecules are groups of atoms bonded together and are the smallest fundamental units of a compound. These molecules take on their own properties and serve as the backbone of our design systems. An example of a molecule might be a site search control including a label, input and search button.

Organisms

Molecules give us some building blocks to work with, and we can now combine them together to form organisms. Organisms are groups of molecules joined together to form a relatively complex, distinct section of an interface. An example of an organism might be the Header bar of a website, including things such as site search molecule, a navigation bar and a site logo.

Templates

At the template stage, we break our chemistry analogy to get into language that makes more sense to our clients and our final output. Templates consist mostly of groups of organisms stitched together to form a genericised view of a page type. An example here might be a landing page template.

Pages

Pages are specific instances of templates. Here, placeholder content is replaced with real representative content to give an accurate depiction of what a user will ultimately see.

Front end frameworks

There are a number of frontend technologies and frameworks that effectively enable this separation although they can all be described broadly as offering a component based view of front end development. Currently the three most popular choices according to a range of measures are:

- React
- AngularJS
- Vue.js

Each of these three technologies would provide a good choice to use as a frontend framework however we would currently recommend the use of React. It provides a greater level of flexibility than AngularJS and a more mature ecosystem than Vue.js.

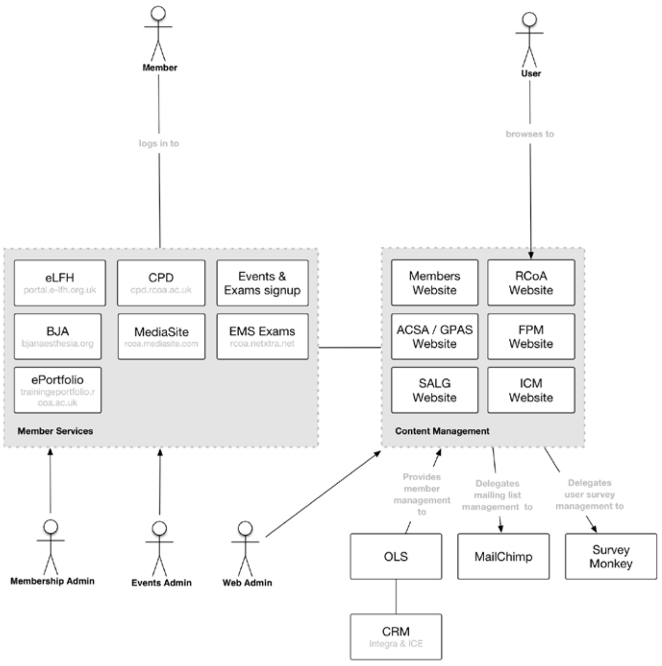
ARCHITECTURAL OVERVIEW

To start with we looked at your current digital platform to understand the current digital landscape. This understanding is based primarily on the information contained in the supplied TSP Architecture document.

Current State System Context

The current architectural landscape is described by a number of separate applications each with their own user management and authentication subsystems. In the diagram below these applications are described in the grouping Member Services. In addition you have a number of applications delivered using the Drupal 7 Content Management System. In the diagram below these application are described in the grouping Content Management. Finally you have a number of applications that provide support for the applications delivered by the CMS.

Current System Context: v0.1



Future State System Context

The future state architecture described below differs from the current state in the following ways:

Instead of describing the applications that form part of RCoA's digital architecture in two main groupings (Member Services and Content Management) we've detailed a three tier architecture.

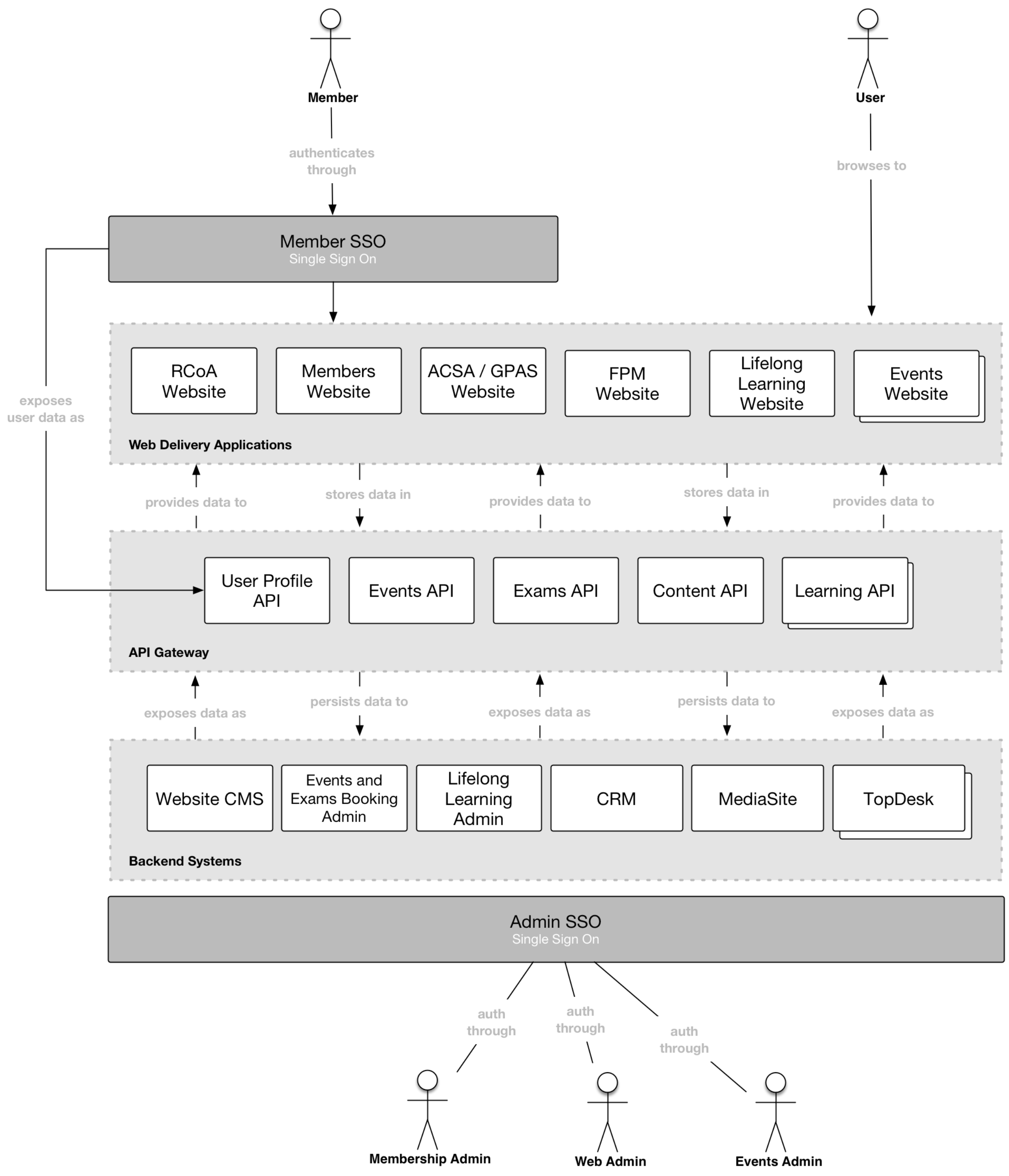
This architectural view makes a distinction between the applications that deliver content and information to the end user (Web Delivery Applications), the applications that provides administrative access to the underlying systems (Backend Systems) and introduces a tier in between that provides an API centric view of the data stored in the backend systems (API Gateway). Additionality access and authorisation to both the Backend Systems (for college administrators) and the Web Delivery Applications (for members and other logged in users) is provided by the introduction of an SSO framework.

As discussed before, the API gateway provides two main capabilities in this model.

The first is that it provides access to the data in the underlying systems in a uniform manner making it possible to consume and present data from separate systems to users in the same page/context.

Secondly by describing a uniform API that provides, for example, a view of events we are introducing a facade that enables the ability to change the supplier of events data without impacting any integrations created against the API.

RCoA: Future System Context: v0.1



PRODUCT SELECTION - WEB CONTENT MANAGEMENT SYSTEM

Goal - to provide a framework for choosing a Web Content Management System that best supports the needs of the RCoA

Overview and recommendations

This section of the document is separated into the following

- Selection recommendation
- Comparison highlights
- Methodology
- Market research
- Selection criteria
- Scoring scale

Selection recommendation - Drupal 8 on Acquia

Of the five products we looked at, we feel that three would provide good platforms to implement the next phase of the RCoA digital platform with and that one stands out particularly.

In our opinion Drupal 8 provides the best all round set of capabilities to deliver the ambition you have for your digital platform. In particular its support for modular content management paradigms and the easy with which access can be provided to the underlying content over RESTful APIs.

Comparison highlights

Drupal 8 - Acquia

+ Excellent modular content management and delivery capabilities
+ Good content workflow capabilities
+ Easy to expose content over RESTful APIs
+ Good value for money
- no personalisation functionality out of the box

EPIServer 11 hosted on Digital Experience Cloud Services

+ Good personalisation functionality out of the box
+ Good form building and handling
- basic content workflow capability
- the most expensive of the options described

WordPress hosting on WPEngine

+ Excellent easy to use content editing user interface
+ Easy to expose content over RESTful APIs
+ Good value for money
- no personalisation functionality out of the box
- very basic content workflow capability

Methodology

- Determine a long list of options
- Determine a shortlist of 5 prospective products
- Identify based on user needs a range of grouped assessment criteria
- Apply a weighting to describe our priorities
- Score each product against the criteria described

Market research

In order to determine a list of prospective content management systems to pick from, we performed some initial market research.

The Long List

The following list comes from the this report - https://www.realstorygroup.com/Reports/CMS/ and details 34 separate Web CMS products and platforms.

Acquia - Drupal
Adobe - AEM Sites
Atex - Polopoly
BloomReach - Digital Experience Platform
CCI Europe - Escenic
Contentful - Contentful
CoreMedia - Digital Experience Platform
Crafter Software - Crafter CMS
CrownPeak Technology - CrownPeak CMS
DotNetNuke - DotNetNuke
e-Spirit - FirstSpirit
Episerver - Episerver
eZ Systems - eZ Enterprise
GX Software - XperienCentral
IBM - Web Content Manager
Ingeniux - Ingeniux Content Management System
Joomla Project - Joomla!
Kentico - Kentico CMS
Magnolia - Magnolia CMS
Microsoft - SharePoint 2016 WCM
MODX - Revolution
OmniUpdate - OU Campus
OpenText - TeamSite
OpenText - OpenText Web Experience Management
Oracle - WebCenter Sites
Perfect Sense Digital - Brightspot
Plone - Plone CMS
Progress Software - Sitefinity CMS
SDL - SDL Web
Sitecore - Experience Platform
TERMINALFOUR - Site Manager
TYPO3 - TYPO3
Umbraco - Umbraco CMS
WordPress - WordPress

From this long list we filtered down to a list of 5 options based on the following criteria:

- Excluding legacy platforms
- Excluding products that would be too expensive to implement and run
- Excluding products that were a poor fit for our scenario, this included products that:
    - focused on specific use cases such as multilingual
    - provided too little pertinent functionality

Legacy platforms

IBM: Web Content Manager

Microsoft: SharePoint Server 2016

OpenText: TeamSite

OpenText: Web Experience Management

Oracle: WebCenter Sites

Expensive platforms

Sitecore

Adobe AEM

CoreMedia: CMS

BloomReach - Digital Experience Platform

Poor fit

SDL - SDL Web

CCI: Escenic

CrownPeak: CrownPeak CMS

e-Spirit: FirstSpirit

eZ Systems: eZ Enterprise

Ingeniux: Content Management Systems

Magnolia: Magnolia

Progress Software: Sitefinity

Atex: Polopoly

GX Software: XperienCentral

Perfect Sense: Brightspot

Plone: Plone

TYPO3: TYPO3

Crafter Software: Crafter CMS

DotNetNuke: DotNetNuke

Joomla!: Joomla!

Kentico: Kentico CMS

MODX: Revolution

OmniUpdate: OU Campus

TERMINALFOUR: Site Manager

This left us with the following 5 options to compare

Comparison shortlist

Episerver - Episerver

Umbraco - Umbraco CMS

Acquia Drupal 8

WordPress - WordPress

Contentful - Contentful Web Content Management Selection Criteria

Web Content Management Selection Criteria

With your input we identified the following capability groupings

- Core Web Content Management
- Content Delivery
- Integration & APIs
- Digital Asset Management
- Cost of Ownership
- Platform Management Complexity
- Personalisation
- Security

What follows are descriptions of each of the capabilities by grouping.

Core Web Content Management

Intuitive content editing paradigm

Ability for non-technical authors to edit and maintain content with an easy-to-learn, intuitive interface which empowers those who actually know, own, and manage content to perform content management activities without intermediaries.

WYSIWYG authoring

Authoring environment is user-friendly, modern, and enables authors to see content presentation as it will appear when rendered in a web browser or mobile device, etc.

Modular authoring paradigm

Template and component based authoring experience whereby authors can assemble and edit content using a modular approach. Where template based authoring describes the ability for authors to use pre-defined content-type focused templates to ensure consistent user experience around content presentation and component based authoring describes the ability for authors to use components to assemble content on the template.

Multi-site management

Ability to support multiple distinct sites via a common platform while supporting isolation of data and code as needed

Metadata management

Ability to manage metadata such as controlled values, tags, etc.

Global presentation management

Ability for authors to manage globally presented content like headers, footers, navigational elements, country selectors, etc.

User generated content moderation

Authors have a management dashboard for reviewing UGC submissions and moderating what is posted live

Entitled content

Ability to manage entitled content via Information Architecture and permissions-based standards to grant access to the groups with the appropriate entitlement, from a content consumption perspective

Bulk content editing

Ability to edit content in bulk; for example, to make changes to a particular metadata value across an array of assets

Web forms authoring

Ability for authors to create web forms for data capture

Versioning

Ability for authors to manage multiple versions of content to allow for content to be rolled back to a previous state

Audit history

Ability to view a log of lifecycle events associated with each page; a robust capability in this area would track content edits

Publishing

Ability to publish content live and pull content down (immediately or on a schedule)

Publishing workflow

Ability to launch new content via publishing workflow with review and approvals, etc.

Scheduling

Ability to configure and manage scheduling of tasks such as publishing content live and pulling content down on a scheduled basis

EOL/Archive

Ability for authors to deactivate content and update links from other content; this may include an impact assessment. Ability to remove and archive EOL content

User provisioning (author)

Ability to provision users to access the authoring interface; this includes internal and external (agency, subcontractor, partner) users

Roles and permissions (author)

Ability to assign roles to users to allow each to perform their required duties while enforcing protection against rogue or accidental modifications to other content

Visual/Creative design

Ability to design page layouts i.e. the arrangement of elements (content) on a page, such as image placement, text layout and style

Content Delivery

Scalability

Ability of content delivery layer to scale to meet increasing demand, either automatically or admin-assisted

Caching

Ability of content delivery layer to support caching to improve performance and reduce load on publishing hardware application servers

Device detection and routing

Ability of platform to detect user device and route to appropriate delivery channel

Content syndication

Ability to distribute content via alternative channels

Social media

Ability of platform to publish content to social media sites like Facebook or Twitter

Integration & APIs

Content import and export

Ability to import content to the platform; for example, that received from a 3rd party authoring partner. And the ability to select and download content from the platform in order to transfer content between environments or for archiving purposes

Integration support

Ability to expose external data and services to the platform via an integration layer

Content repository APIs

Access to the repository that supports the collection, managing, and publishing of digital information using well described RESTful APIs

Application logic

Ability to introduce custom application layer logic (i.e. JAVA, .NET, PERL, PHP, etc.) to implement specific business features

Personalisation

Personalisation

Ability to accommodate the differences between individuals. For instance, social network websites use personal data to provide relevant advertisements for their users

Personalised content

Process of providing content geared towards accommodating the differences between individuals provide relevant

Persona simulation

Ability for authors to intuitively set the attributes on which personalization is based to simulate the experience of each persona as a means of testing personalized content at design-time

Digital Asset Management

Rich media asset management

Ability to manage rich media assets including configuration files and other associated static assets

Ingestion

Ability to add assets to the system, individually and in bulk

Metadata and tagging

Ability to tag and assign metadata to assets for internal findability as well as external presentation

Cost of ownership

Looking beyond the basic licensing costs of the product the TCO needs to be calculated incorporating line items such as annual support/maintenance costs, patching costs (any customizations may require additional work every time patching occurs), cost to source staff for customizations and other relevant ancillary costs.

First year cost

Is there a licensing cost for the software?

Ongoing annual support / maintenance costs

What are the ongoing support and maintenance costs?

Upgrade complexity

How complicated are upgrades for the software, do they typically require development resource?

Development costs

How complicated is the product to implement, does it require development resource with specific product knowledge or skills? Is there a strong market of developers with the skills required to implement this product?

Additional module costs

Are there additional modules required to perform basic functionality that incur further cost?

Platform Management Complexity

How hard is it to maintain the various platforms required to maintain the ecosystem. This needs to consider items such as product patching, server maintenance and deployment stories.

Deployment pathway

How easy is it to deploy updates to the application? Is there a well describe automated deployment pathway or is the process primarily manual?

Maintenance effort

How much input from the customer is required to keep the product running smoothly?

Security

Infrastructure

Does the product lend itself to a deployment architecture that allow best practice network/infrastructure security to be followed.

Compliance and governance

Does the product itself allow a fine grain definition of users roles to be defined and managed in order to secure features, content and assets. It is particularly important that integration stories for complementary products allow propagation of these security rules, or at worst permit similar definitions of their own to match the relevant requirements.

Scoring

Scoring scale

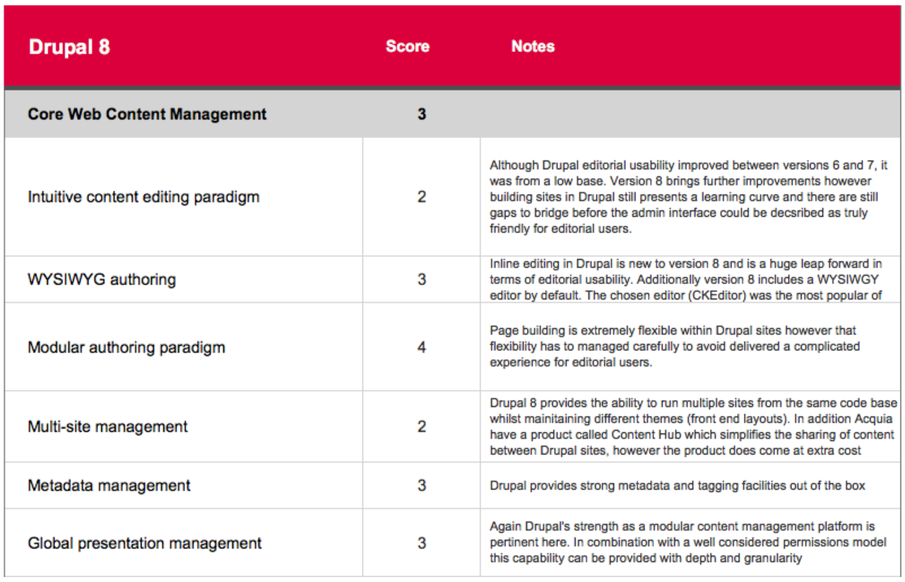
The following describes the scale used to score the capabilities described for each product

Rating	Description
0	No capability - completely custom solution.
2	Minimal custom development required to support the capability.
1	Extensive custom development required to support the capability.
4	Available out of the box, but some configuration required.
3	Available out of the box, but extensive configuration required.

Web CMS Product Selection Scores

Capability	Weighting	Drupal	EPIServer	Umbraco	Contentful	WordPress
Core Web Content Management	20%	2.789	2.421	2.316	2.105	2.421
Content Delivery	10%	1.800	2.200	2.200	2.600	2.200
Integration APIs and Extension	15%	4.000	3.667	3.667	4.000	4.000
Personalisation	10%	2.000	3.000	2.000	1.333	1.333
Digital Asset Management	10%	2.667	2.000	2.000	2.333	3.000
Cost of Ownership	10%	2.800	2.000	3.000	2.200	3.400
Platform Management Complexity	5%	2.500	2.500	2.500	3.000	3.000
Security	15%	2.500	2.000	2.000	2.500	2.000
Totals		**2.685**	**2.529**	**2.458**	**2.459**	**2.594**

Further scoring detail and narrative can be found in the associated file - rcoa-cms-product-evaluation-scoresheets an example of which is displayed below.



Drupal 8	Score	Notes
Core Web Content Management	**3**	
Intuitive content editing paradigm	2	Although Drupal editorial usability improved between versions 6 and 7, it was from a low base. Version 8 brings further improvements however building sites in Drupal still presents a learning curve and there are still gaps to bridge before the admin interface could be decsribed as truly friendly for editorial users.
WYSIWYG authoring	3	Inline editing in Drupal is new to version 8 and is a huge leap forward in terms of editorial usability. Additionally version 8 includes a WYSIWGY editor by default. The chosen editor (CKEditor) was the most popular of
Modular authoring paradigm	4	Page building is extremely flexible within Drupal sites however that flexibility has to managed carefully to avoid delivered a complicated experience for editorial users.
Multi-site management	2	Drupal 8 provides the ability to run multiple sites from the same code base whilst maintaining different themes (front end layouts). In addition Acquia have a product called Content Hub which simplifies the sharing of content between Drupal sites, however the product does come at extra cost
Metadata management	3	Drupal provides strong metadata and tagging facilities out of the box
Global presentation management	3	Again Drupal's strength as a modular content management platform is pertinent here. In combination with a well considered permissions model this capability can be provided with depth and granularity

PRODUCT SELECTION - SSO PLATFORM

Goal - to provide a framework for choosing a SSO platform that best supports the needs of the RCoA

Overview and recommendations

Of the four products we looked at, we feel that two would provide good platforms to implement as part of the delivery of the next phase of the RCoA digital platform.

Both Okta and Microsoft Azure Active Directory are excellent SSO products and the choice between these two is difficult to call. In fact our recommendation is contrary to the scoring outcome, however in spite of this, we would recommend Microsoft Azure Active Directory.

This advice is based primarily on the following:

- We believe the total cost of ownership will be lower for your with Azure AD
- You have already invested in products that integrate particularly with Azure AD, such as Office 365

Microsoft Azure Active Directory

+ Competitively priced
+ Excellent integration with other Microsoft offerings (Office 365, MS Dynamics)
+ Best of breed identity threat detection and analytics
- Administration user interface

Okta

+ Large number of out of the box application integrations
+ Good automated provisioning
+ Strong, extensible user directory
- Depending on options purchased, can become expensive

Methodology

- Determine a long list of options
- Determine a shortlist of 4 prospective products
- Identify based on user needs a range of grouped assessment criteria
- Apply a weighting to describe our priorities
- Score each product against the criteria described

Market research

In order to determine a list of prospective SSO single sign on platforms to pick from, we performed some initial market research. The following long list is determined from the following analyst report - The Forrester WaveTM: Identity-As-A-Service, Q4 2017

Comparison long list

- Okta
- Centrify
- OneLogin
- Microsoft Azure Active Directory
- Gemalto
- Oracle
- Ping Identity

Poor fit

- Gemalto
- Oracle
- Ping Identity

Comparison shortlist

- Okta
- Centrify
- OneLogin
- Microsoft Azure Active Directory

SSO SELECTION CRITERIA

Core IDM Platform

User directory support

What level of support is there for a range of user directory providers? Examples of which include, Active Directory, GSuite, WorkDay etc. Is there support for attribute synchronisation

Access policy administration

How easy is it to administer access policies, does this task require scripting or can be performed from admin user interfaces?

Protocol support and step-up authentication

How well does the solution support a range of authentication protocols, examples of which include SAML, OpenID Connect & OAuth2

On-prem app SSO support

How well does the solution support the integration of applications specifically hosted on premise?

Integration

Breadth of SAML and non-SAML application

Does the solution provide a range of pre-described integrations with 3rd party applications, using both SAML and other authentication protocols

Breadth of provisioning connector support, setup, and policy administration

What level of support does the solution provide for provisioning connectors, how easy are they to setup, support and administrate?

Analytics

Identity analytics

What level of support is provided for assessing threats to users identities?

Self service

End user self-service from the portal and from the IDaaS solution's native mobile applications

How easy is it for users to administer their identities both from the solutions portal and from the solutions native mobile application?

Dashboard customization

How easy is it to configure and customise the end user self service dashboard?

Mobile device management

What support does the solution have for managing access and identity across mobile devices

Breadth of support for mobile operating systems

What level of support does the solution provide for support a full range of mobile operating systems?

Reporting

Reporting breadth

How extensive is the out of the box reporting

Report creation and customization

How easy is it to customise out of the box reports or to create completely new ones?

APIs

API security and solution APIs

Cost of ownership

License cost

How expensive is the solution to license

Implementation Complexity

How complicated is it to implement the solution

Additional module costs

Does the solution come with everything you need out of the box or does it rely on the procurement of additional modules?

Security & operation

Scalability

How easy is it to scale the solution? Are there extra cost considerations beyond certain numbers of users?

Security certifications

To what level has the solution been certified and validated by external security authorities?

Scoring

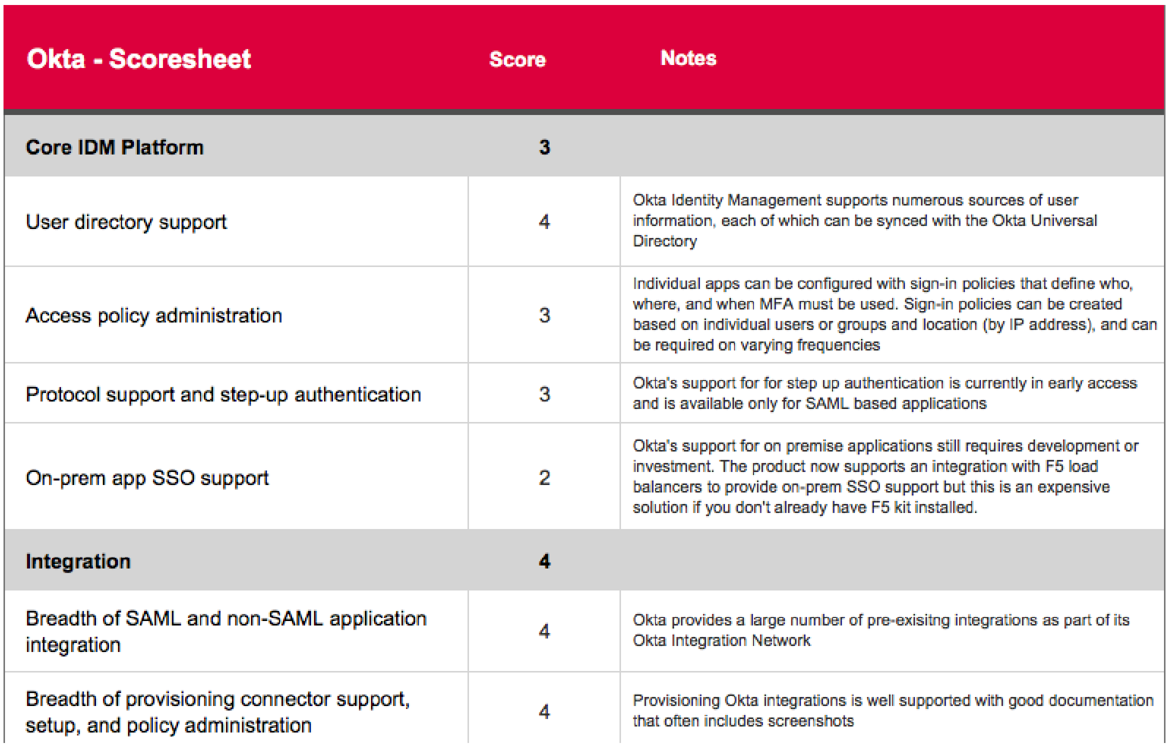
The following describes the scale used to score the capabilities described for each productAPI security and solution APIs

Rating	Description
4	Available out of the box, but some configuration required.
3	Available out of the box, but extensive configuration required.
2	Minimal custom development required to support the capability.
1	Extensive custom development required to support the capability.
0	No capability - completely custom solution.

SSO Product Selection Scores

Capability	Weighting	Okta	Centrify	OneLogin	Azure AD
Core IDM Platform	25%	3.000	3.250	2.750	3.000
Integration	10%	4.000	2.500	4.000	2.500
Analytics	5%	2.000	4.000	4.000	4.000
Self service	20%	2.500	3.250	2.250	3.250
APIs	10%	4.000	4.000	2.000	2.000
Reporting	5%	3.000	4.000	3.500	2.000
Cost of ownership	10%	2.333	2.000	3.333	3.000
Security and operation	15%	4.000	1.500	3.000	4.000
Totals		**3.133**	**2.938**	**2.896**	**3.050**

Further scoring detail and narrative can be found in the associated file - rcoa-sso-product-evaluation-scoresheets an example of which is displayed below.



Okta - Scoresheet	Score	Notes
Core IDM Platform	**3**	
User directory support	4	Okta Identity Management supports numerous sources of user information, each of which can be synced with the Okta Universal Directory
Access policy administration	3	Individual apps can be configured with sign-in policies that define who, where, and when MFA must be used. Sign-in policies can be created based on individual users or groups and location (by IP address), and can be required on varying frequencies
Protocol support and step-up authentication	3	Okta's support for for step up authentication is currently in early access and is available only for SAML based applications
On-prem app SSO support	2	Okta's support for on premise applications still requires development or investment. The product now supports an integration with F5 load balancers to provide on-prem SSO support but this is an expensive solution if you don't already have F5 kit installed.
Integration	**4**	
Breadth of SAML and non-SAML application integration	4	Okta provides a large number of pre-exisitng integrations as part of its Okta Integration Network
Breadth of provisioning connector support, setup, and policy administration	4	Provisioning Okta integrations is well supported with good documentation that often includes screenshots

Appendix A: References

- CMS Product Evaluation | rcoa-cms-product-evaluation-scoresheets.xslx
- SSO Product Evaluation | rcoa-cms-sso-evaluation-scoresheets.xslx
- SSO Solution Design | rcoa-sso-solution-design.pdf
- The Web Content & Experience Management Evaluation Report | CMS-WebContentExperienceManagementVersi on231.pdf
- The Forrester WaveTM: Identity-As-A-Service, Q4 2017
- Manifesto Digital Engagement Future State Capability | manifesto-digital-engagement-future-state-capability.xlsx
- Royal College of Anaesthetists - A personalised digital approach | RCoA_Personalisation_Consultancy Outcomes_updated_09.05.17.pdf
- Technology Strategy Promgramme Architecture | TSP Architecture.pdf

Appendix B: Abbreviations, Acronyms and Definitions

- **Security Assertion Markup Language (SAML)** - An authentication and authorization standard commonly found in the enterprise, SAML differs from Open ID in that it does not dynamically discover and accept authentication from new identity providers. The IdPs that a service wants to trust must be specified and hard-coded into each login event. Typically used to give the users of a corporate network access to a specific 3rd party service—for instance, so you don't have to sign in again when you click a link to Salesforce on your company's intranet.
- **Single Sign On (SSO)** - A subset of federated identity management, a means through which authentication and interoperability can be achieved in a federated system.
- **Application programming interface (API)** - In computer programming, an application programming interface (API) is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program.
- **Content Management System (CMS)** - is a computer application that supports the creation and modification of digital content. It typically supports multiple users in a collaborative environment. CMS features vary widely. Most CMSs include Web-based publishing, format management, history editing and version control, indexing, search, and retrieval. By their nature, content management systems support the separation of content and presentation.
- **Customer relationship management (CRM)** - is a term that refers to practices, strategies and technologies that companies use to manage and analyse customer interactions
- **RESTful API** design (Representational State Transfer) is designed to take advantage of existing protocols. While REST can be used over nearly any protocol, it usually takes advantage of HTTP when used for Web APIs.
- **API gateway** is programming that sits in front of an application programming interface (API) and filters traffic. API gateways can be used to throttle traffic and enforce security policies. Besides combining a number of APIs with potentially different interfaces into a unified presentation interface, the separation of interface and API has a number of benefits.
- **Capability model** describes the complete set of capabilities an organisation requires to execute its business model or fulfil its mission.