mattclarkecto.com // just technology

White Paper

Headless Content Management Architectures

Explore why Enterprise Digital Ecosystems are creating a new style of content management architecture

November 2016

mattclarkecto.com

cto bytes

thecto podcast

I recently finalised my usual technology trends for 2017, one subject covered was “The vendor innovation gap”. A condition I’m starting to see where our vendor partners are loosing ground between what customers are now starting to demand and the pace of their respective product roadmaps.

This may be, because, the roadmap demands are getting pulled in too many directions which is leaving vendors with challenges keeping up. One such area where I’m noticing this more acutely is amongst our CMS partners.

The CMS landscape is arguably a competitive vendor landscape which has evolved around key themes such as the marketing suite, customer engagement, customer conversion, mobile and responsive. This evolution, particular in the suite direction, has created an aurora of constraint, which is causing enterprise clients major issues as they start to join up and build enterprise grade digital ecosystems.

Why is this causing problems?

Because it is constraining our clients into a way of working around what the vendor sees as their roadmap and their best practice which often differs from the flexibility and enterprise nature that is required by our clients.

But surely going against a vendor recommended best practices could lead to problems down the line or simply not being able to enjoy some of the future benefits a vendor may introduce into the product at some later date?

This is a big consideration and not recommended for all clients particularly clients that are not looking to be too ambitious in their requirements. i.e. it’s just a website.

However at some point in the future all clients will need more from their digital estate, they will need to combine vendor capabilities such as CMS, Commerce, Optimisation, A/B Testing, Conversion, Product Information Management, IoT, AI, Mobile, Targeting, Single Customer View. Clients will need to avoid the constraining nature of a “one product meets all, suite approach”. No vendor can possibly deliver everything and this is one of the biggest concerns amongst enterprise clients who in their nature are often much bigger and have much more diverse needs than the vendors that are trying to serve them.

So what is the answer?

I find mid tier CMS vendors the most interesting, they are the vendors that are the most innovative and challenging to the guys at the top. However because of their nature they are marketing their products to users that like everything wrapped up into a suite of capability. This is where things can go wrong and prevent those vendors from enjoying some of the benefits of working with enterprise clients. A way to break through this barrier is through architecture of the product. Creating product architectures that enable customers to have a great degree of choice and flexibility in the implementation of the platform into their business. The best approach to do this is the decoupled approach or headless CMS model.

Web applications have become much more sophisticated, the front end interfaces have become applications in their own right using their own frameworks, integrations capability, service orientated architectures and rendering manipulation techniques. The back end application's capability have equally needed to keep up with the demand for cloud based hosting environments, vertical and horizontally scalable application capabilities and new lightweight approaches to application development using technologies like node.js. The combined sum of these two areas means a standalone application in its own right often gets developed for the average website build, regardless of the CMS choice. The hard bit then is retrospectively trying to make it all work within a content management platform. To be fair most vendors have recognised this and have adopted MVC approaches to their implementation architecture. However they still fail to create approaches for ensuring all of their functionality is made available to these environments without creating a mix-up of architectural techniques. The future needs to be based around a complete headless CMS strategy where the site CMS platforms focus on content storage, content management, editorial control and other engagement capability. All of which can be made available to the web application without interfering with the architecture approach of the application. This means that vendors need to drop the controlling nature of the template and start thinking services. Content services, engagement services, optimisation services and good solid content management at the backend. This then enables enterprise clients to focus on building their digital estates with a consistent and owned strategy of their own. The more vendors in the CMS and Commerce arenas open up their services the more we can build enterprise grade ecosystems, enjoying all the benefits of latest technology thinking in the areas of cloud, mobile, IoT and front end web innovations.

But does this create complications for clients who want to move quickly? i.e. do we have to build a complete application as well as implement the CMS. We feel that pace needs to be balanced with getting the architecture right. It is critical to think about the whole ecosystem and the connected way that an organisation's digital estate will need to evolve in the future. Vendors can help with pace by building accelerators in popular technology such as node.js, Microsoft.net or Swing to enable swift and flexible delivery.

What is critical in all of this thinking, is the good old service orientated approach to architecture. This includes content as a service, commerce as a service, customer as a service, optimisation and targeting as a service. Exposing these services to use by the web application will ultimately provide the hub needed to run your digital ecosystem. CMS vendors will then become true enterprise partners.

So what are the recommendations for implementing a CMS architecture today?

1) Decouple the web application layer and build an architecture that combines front and backend. Think independent application!
2) Look at the CMS and expose services needed. Think content as a service! Functionality as a service!
3) Think about upgrade strategy for the CMS.
4) Incorporate cloud based thinking into the architecture.
5) Consider the use of container technology for the web application. This will enable scaling and cloud deployments
6) Choose the web application technology wisely. Remember it needs to be independent from the CMS platform.
7) Build a consistent set of rules a guidelines for the whole stack. This will prevent the architecture getting messy.
8) If your CMS vendor supports MVC approaches engage your vendor in what you are trying to do. Try and understand your their roadmap.
9) Consider integration in the web application itself, remember it will be bigger than your CMS platform in the long term. Think about other platforms and tools you will need to include within the architecture.
10) Control your front end html and javascript application. Apply the same standards as you would to your backend, remember the front end discipline is still maturing and does not have the same level of architecture foundations as the backend ■