



ALLAN THRAEN |

🕒 4 months ago |



PDF |



CMS Optimizely (EpiServer) Frontend Development Website Improvements Tech Talk

CHRISTMAS COUNTDOWN: #1 THE GRAND FINALE. GOING HEADLESS WITHOUT USING YOUR HEAD!



In 2014 the term 'headless cms' was coined - and presented as a cool new 'feature' in the Web CMS industry. And it quickly became a hot buzzword for a few years later. In a few days it's 2024 and we can celebrate that it's been a concept for 10 years. Strangely, I still encounter new implementations that want to go 'headless' based on an almost religious belief that it's the new cool thing.

The 2023 Christmas Countdown: 12 Common Pitfalls in Optimizely CMS - and how to avoid them

This blogpost is part of the 2023 Christmas Countdown series where I each day for the last 12 days before Christmas go through my Top 12 list of the most common and dangerous pitfalls I typically see in Optimizely (EpiServer) CMS 12 Implementations. If you want to learn more and perhaps have your own site evaluated feel free to reach out to us here at CodeArt.

Here are links to all the posts in this series as they are published:

12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

First, let's clear up a common misunderstanding: Almost all major content management systems had what would now be called 'headless' implementations many years before the term was coined. Essentially implementations where the content from the CMS was delivered through some other platform/technology and not straight from the CMS. Typically exposed through one of the API's most CMS's have. All the way back in 2004-5 I remember seeing Episerver implementations where content was accessed as xml and rendered through flash (yeah, some of you kids might not remember Flash - but it was a weird, but popular browser plugin). Or delivered through TextTV or used in some early mobile apps when they started to appear.

But Episerver and the other platforms didn't have to deliver headless. They could also deliver straight on the platform to a dynamic website.

So - the content management systems that started to market themselves as "pure headless" really had a stroke of marketing genius - they took a shortcoming: not being able to deliver/render content directly and turned it into buzzword feature. Sort of like a company launching a pure Engineless car - that can do everything a normal can do, but you have to invent some other contraption to get it to move :) But, once again I digress.

Reasons for Pure Headless Implementation:

Sure, there are valid reasons to go "pure headless" on Optimizely/Episerver. Perhaps your content is an exclusive VIP, only entering through a secret app entrance. Or maybe you're on the SaaS Core CMS, and being headless is the only ticket to the party. No website here, just a single-page app (SPA) soaking up the limelight.

And yes, I get it. Headless isn't the villain of the story. It's just that sometimes, we see it done for reasons that make us scratch our heads like confused reindeer.

Common Headless Missteps:

- "Our developers only want to code in [insert js framework here] and despise Marketing's love for Optimizely."

Ah, the age-old tale of love and hate, where business decisions born out of developer resentment seldom lead to the North Star.

- "Let's go headless so we can switch CMS like changing reindeer sleighs!"

Spoiler alert: It's not that easy. Santa's sleigh and CMS platforms have a lot in common - switching isn't as magical as it sounds.

- "Node gives the best performance, and we need it to serve our website."

The Node vs. .NET Core debate - a holiday feast where the roast has been debunked, reheated, and served again. Plus, throwing headless on top? Double trouble.

- "Headless is the only way to go cloud!"

Cloudy with a chance of misconceptions. While it may be true for pure multitenant SaaS, the cloud is vast, and scalable app services are still very much a part of the forecast.

- "Headless is the trend all the cool kids are using."

Are you sure you're not stuck in a time loop, reliving the trends of wannabe cool kids from a decade ago?

Consequences of Headless Shenanigans: ☒

- More code to juggle than a team of elves during peak season.
- Potential failure points that rival a precarious tower of gift boxes.
- Slower responses, making your website move at a speed slower than a snail in holiday traffic.
- Custom work galore to make 'built-in' features dance to your tune.

Again, let me clear. Headless architecture can be fine. I do very much like SaaS cms's as Contentful, and the new Optimizely SaaS offering. But most customers that choose Optimizely CMS today do it because they need a backend platform they can customize and integrate with many other internal systems - and in those cases where you can't go full SaaS then why not take advantage of the build-in head? All I'm saying is ***be pragmatic*** when choosing architecture. Use the one that best suits your requirements.

Closing: A Christmas Countdown Surprise

As we unwrap the #1 pitfall in our Optimizely CMS Christmas countdown, take a break! Christmas is a time to enjoy with family, just like a CMS should be with your development team. Reflect on your CMS implementations and how they can sparkle brighter in the future.

Merry Christmas and a Happy New Year! May your code be bug-free and your deployments smooth as fresh snow.

CMS

Optimizely (Epi)server

Frontend Development

Website Improvements

Tech Talk

RECENT POSTS