

WAN-IFRA Report | June 2022

Expert advice on choosing a CMS



World Association
of News Publishers

ABOUT THE REPORT

The path to choosing a new CMS is littered with roadblocks, conflicts and challenges.

That is why we have tried over the past few years to showcase roadmaps that help publishers and newsrooms to better navigate their CMS journeys.

And quite honestly, this “report” fell in our laps. Ariane Bernard was asked to speak at our CMS Day (as part of WAN-IFRA’s Newsroom Summit) in late April, essentially to give her recommendations to publishers which are perhaps in the procurement phase for a CMS. Her comprehensive, concise presentation not only checked all the boxes of what we are looking for in our reports, but also for newsrooms searching for a CMS. So it was a simple decision to publish this, more or less, as is from her presentation.

So in case you don’t know Ariane, she is a product leader with a specialty in content management and delivery tools – CMS, subscription and eCommerce, analytics, CRM. She was formerly the chief digital officer at Le Parisien.

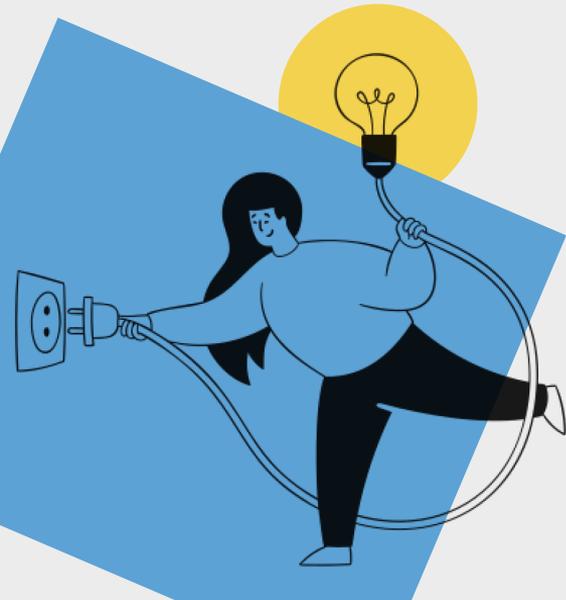
Prior to this, she was the product lead for editorial products at Taboola, and served in a variety of roles during her 14 years at The New York Times, including working on the CMS team.

Today she , as the Founder and CEO of Helio, is building this new publishing platform oriented at independent creators, journalists, video creators, micro-influencers, course creators, etc. And here is her disclaimer for those news media organisations which might think this is some sales pitch for her product:

“We’re in alpha and launching soon (and fundraising soon). Always happy to connect if you want to hear more (but Helio, if it needed to be clarified, is absolutely not the right tool for an established media company).”

Take home
version

So, you're shopping for a new CMS...



CMS Day @ WAN-IFRA - 2022

Ariane Bernard - bonjour@arianebernard.com

Hi, I'm Ariane Bernard 🇫🇷 🇺🇸

I am the founder of *helio*

A product leader, I've built some of the most advanced CMS and ad-tech products in the world



The New York Times

Taboola

Le Parisien

Outline

Part I: Ahead of the Journey

Part II: Getting the RFP Ready

Part III: My Framework

Part IV: Platform Vision

Part V: The Key Items & Scores

Part VI: Takeaways

Part I: Ahead of the journey ...

What's hard about the decisions of replatforming?

CMSs tend to have similar capabilities. What differentiates them is:

- Architecture (which will speak to modularity, evolution) → Your tech team owns this but your subject-matter expert stakeholders should care about it
- The tool layer. What you see as a non-technical stakeholder

And, broadly:

- Very few truly 'bad' CMS (but some really outdated frameworks)
- No 'best' CMS

CMSs are opinionated by their nature. Their toolsets are oriented at an industry or group - that's why only certain vendors are in this room. The more particular your particular goals or need, the smaller the list of specialist CMS. But you may be happy with a generalist CMS + some speciality tools added to your platform.

Replatforming received knowledge

The market leader in my industry is probably one of the best choices for me too

→ not necessarily at all. [Example of digital only product looking at platform whose strong suit was print integration. Broadcasters]

I already know all the 'usual suspects' – these 5 CMSs that equip 50 percent of my competitors

→ many new options in the “CMS in a box” category. This category used to be a bad idea, but there are some interesting, very modular options now that can be inspiring → Don't jump head just yet and keep an open mind

Buy vs. Build is clear cut for me

→ Maybe so. But the modularity of many of the platform components (ie, paywall, frontend etc) asks the question of whether a mix of ready made components tied by some proprietary framework may not be your best options. Don't discard this just yet.

90 percent of your
decision should be based
on what you may need
within 2 to 5 years.

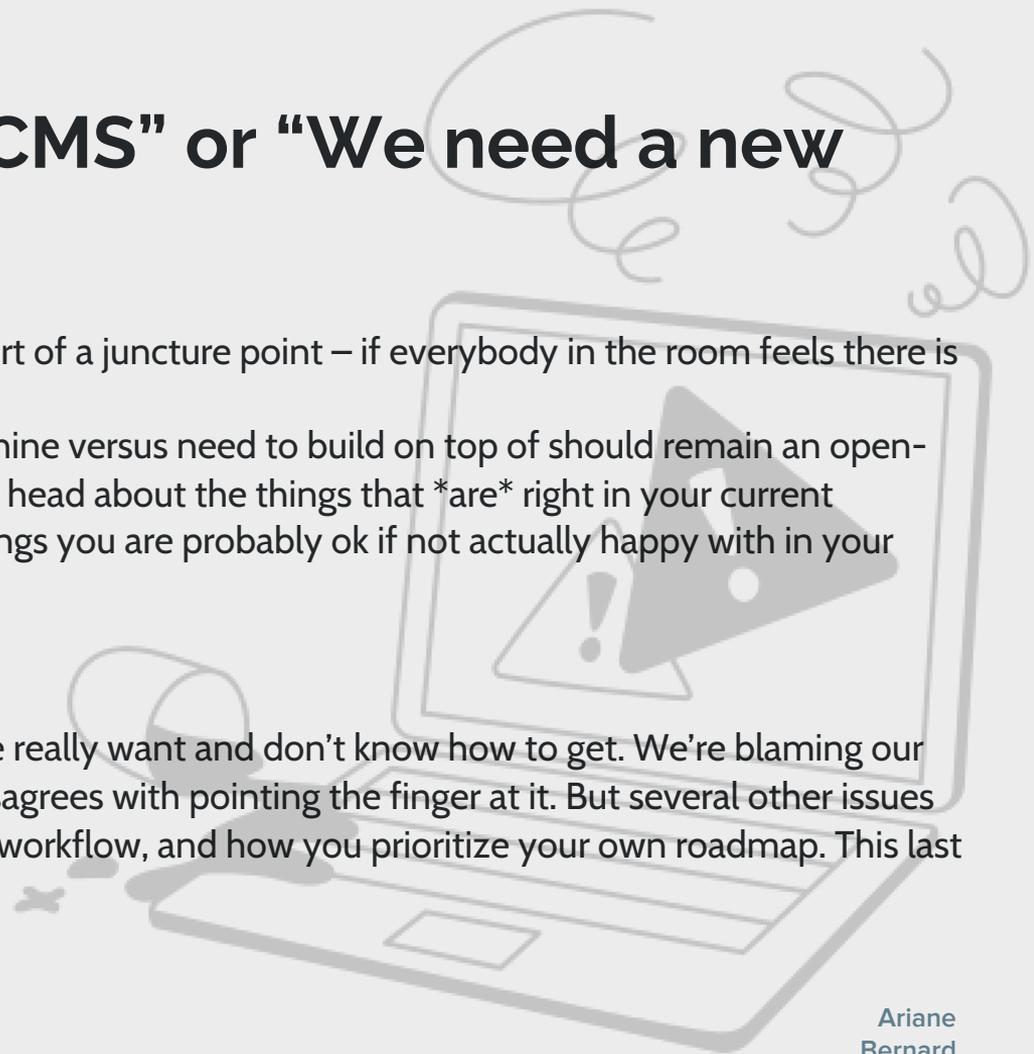
#1

“Everybody hates our CMS” or “We need a new CMS” (What I hear...)

- Probably right that you’ve come to some sort of a juncture point – if everybody in the room feels there is a problem, there probably is.
- Whether you need to burn the current machine versus need to build on top of should remain an open-ended question. It will help you keep a clear head about the things that **are** right in your current platform. It may surprise you how many things you are probably ok if not actually happy with in your current system.

But i also hear

- ... shorthand for: There’s a few big things we really want and don’t know how to get. We’re blaming our CMS because it’s not a person so no one disagrees with pointing the finger at it. But several other issues are often playing a part – in particular, your workflow, and how you prioritize your own roadmap. This last bit is the painful, painful bit of it all...



The problems you are here to solve are strategic, not functional

And this is the other big reason you need to work very hard on the problem before the solution:

- Your #1 problem is the business you want in 2 to 5 years
- Your #2 problem is the way you want the company to run

I've literally had a publisher ask me 'if we get a better CMS, we will have a better workflow'

→ You won't. You can get a good workflow out of an old CMS. You can be equipped with the latest and greatest and run a really messy ship. There is no shortcut for strategizing #1 and #2 and do this well ahead of writing your first Request for Proposal (RFP)

If you are spending 90 percent of your time discussing your problems of today, you're discussing **functionality, not strategy**



Part II: Getting the RFP Ready

You write your perfect RFP:
you'll drop most vendors in
one round.

**Your shortlist will be shorter
(2 or 3), and all good options**

#3

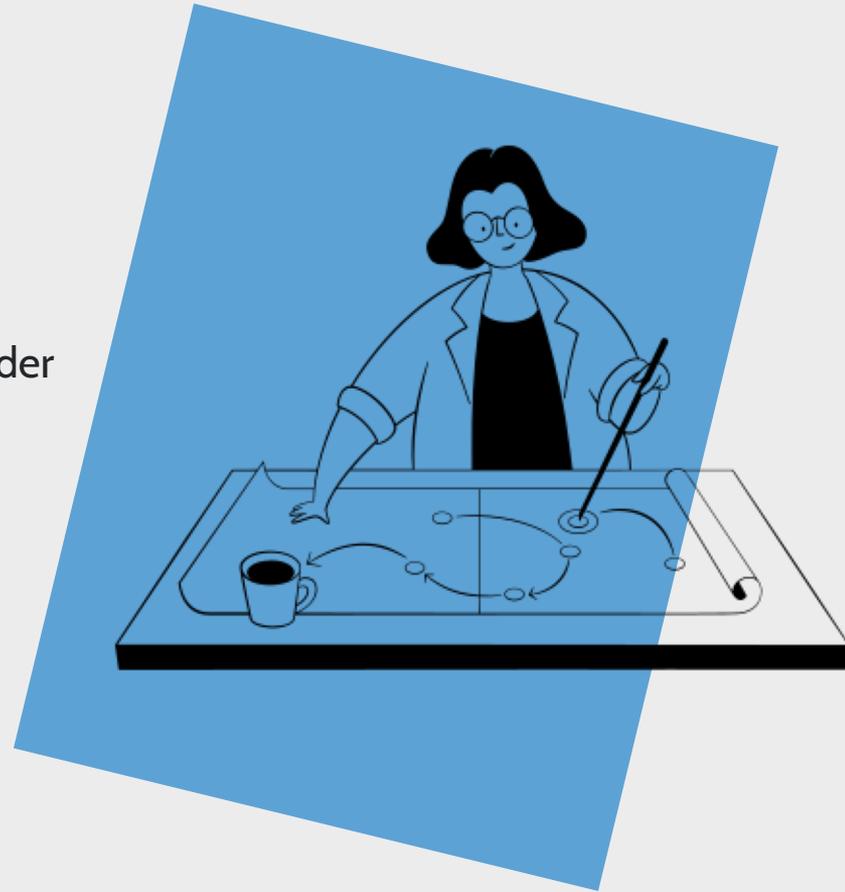
Step 1 - The vision for your publishing

This is actually a vision for:

- Your business, at 5 years
- How you work as a company: Who does what
 - This may be quite a broad rethink: I've learned that this was actually where you had to think of things like your unions and any existing labor situations that you may have to take into account for how you may want to change roles and responsibilities
 - Are you trying to shift most of your company to working on digital and minimally on your legacy product?
- If you think you may acquire or divest yourselves of certain assets in your company (does audio matter? We'll sell our radio division in 2 yrs)
- Are you expanding overseas with new bureaus and therefore new workflows?
- Will you be localizing in new languages?

Sidebar | Why 5 years?

- Alas, no platform is forever - if you try to draw a wider vision, you'll over engineer (and you'll still replatform)
- The industry, your workforce, technology, and our ecosystem will change beyond our imaginings.



What does the replatforming task force look like

My recommendation is that it has two groups:

The **Core project team** works on the project end to end, from Vision to deployment.

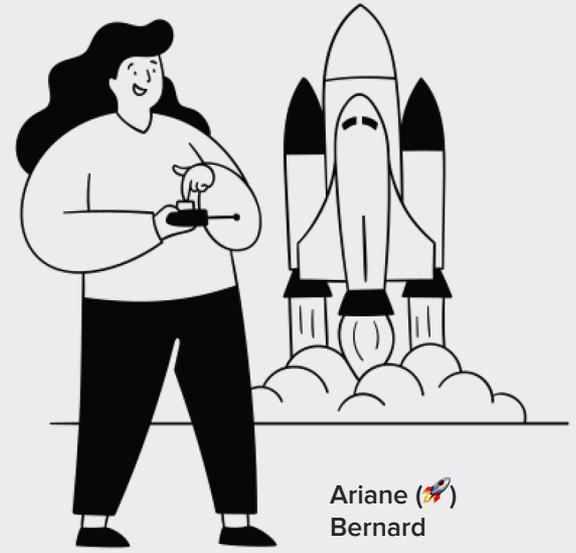
The **Stakeholding team** contains owners and representatives of the various goals. It is made of little Stakeholding pods which will each own one of the goals.

The Core project team is therefore working ahead of the Stakeholding team:

- Creating vision, which we will talk about
- Interviewing across the business. This may include interviews with folks who will end up taking part in the Stakeholding team!
- May need to get C-Suite level information on long-term goals for company if some aren't public.

Who is the boss of the Core Project team?

Tempting to put a key stakeholder. But this may create conflict and, usually, the best person to lead the core project team is your best project manager. Prepare to give her a long vacation, a bonus and a promotion when this is over.



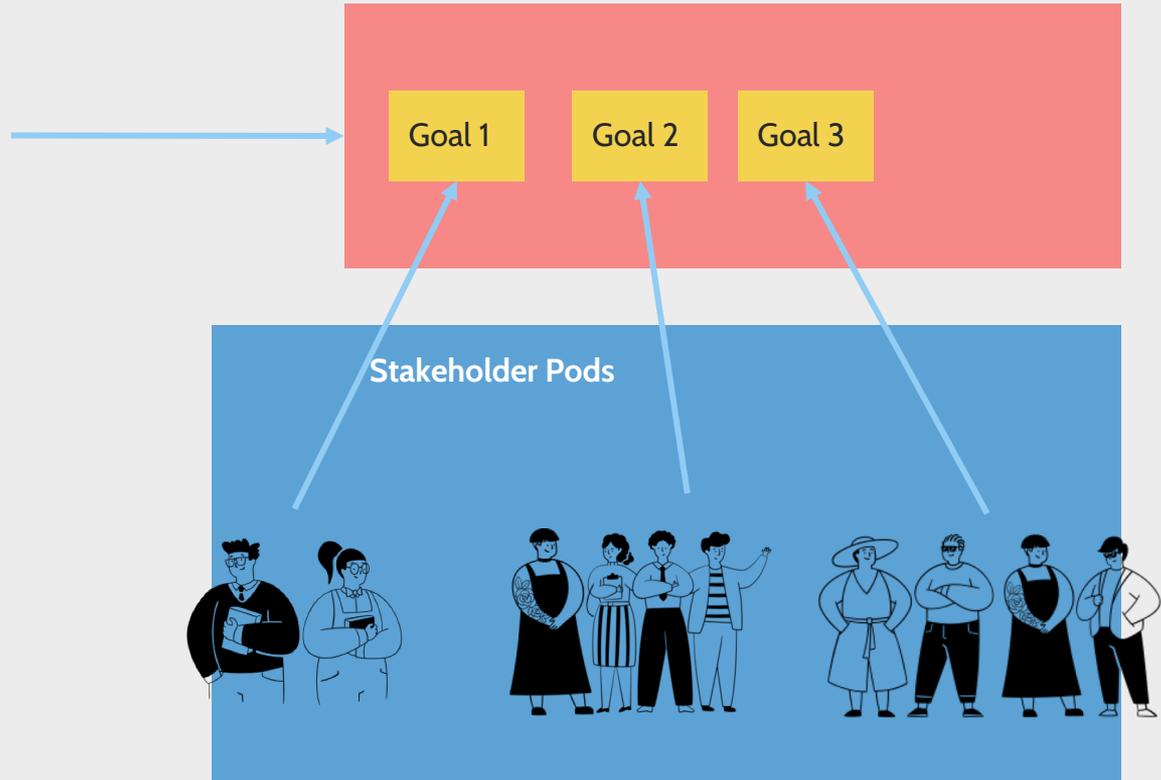
Ariane (🚀)
Bernard

Replatforming Task Force at Step 1



Core team

- Each goal has a stakeholder pod
- Some team members may be in several stakeholder pods (most likely the case, in fact)



The Core Team's first job is to inventory the goals

You can spend months describing all you don't like about your current platform. I suggest you don't spend time there but instead **work only on goals and vision**.

Narrowly, the goals trickle from the 5 year vision.

Imagine your workflows before you worry about how your workflow tools will support it. If there was ever a moment to paint a blue sky, it is now.

→ Imagine the end to end lifecycle of an article (or video). Don't put any names or departments behind the jobs to be done. Imagine that current technology faultlines don't exist (not, "but this team doesn't work in this tool"). That becomes your workflow goal.

→ By far one of the biggest reason for messy discussions in the RFP stage is because this step was treated as just a fun thought exercise.

→ Whatever time you spend here to have these discussions and break down what they may entail will make for a significantly clearer and complete RFP.



The time you take aligning on the problem is time you will save aligning on the solution: the solution has more unknowns and marketing spin.

Align on the problem

#4

Stakeholder Pods flesh out how to serve the goals

There are various frameworks for working through roles and responsibilities - your company probably uses its own approach (problem owner, business owner, key stakeholders, informed party etc).

For your project, this will translate to something like this:

The Core team will probably work through assembling that stakeholding team. Some stakeholder pods are obvious (Video Product person, video producer, and video ad person will probably be stakeholder pod members of the Video goals)

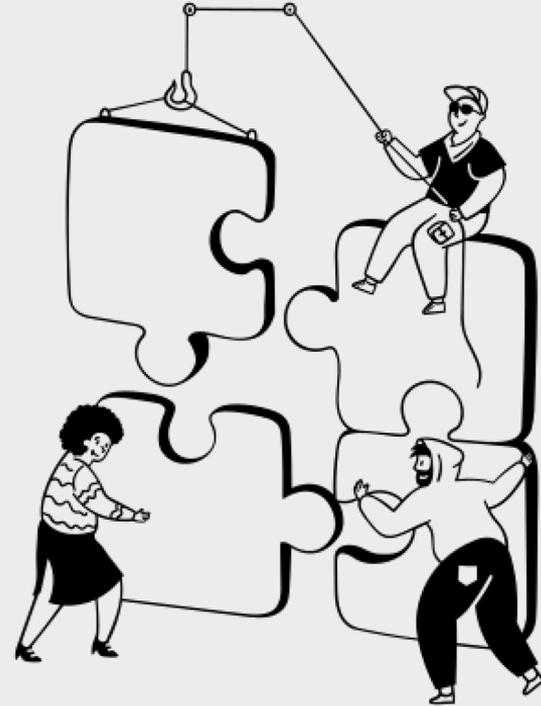
But some are harder:

- who owns your goal for how Support will work?
- Who owns your goal for overall infrastructure? This dips across goals and depending on your company, the Core team may have a harder time coming up with the right Stakeholder pod on the goal. It may be a pure engineering Pod. It may have product in it.

Part III: My framework

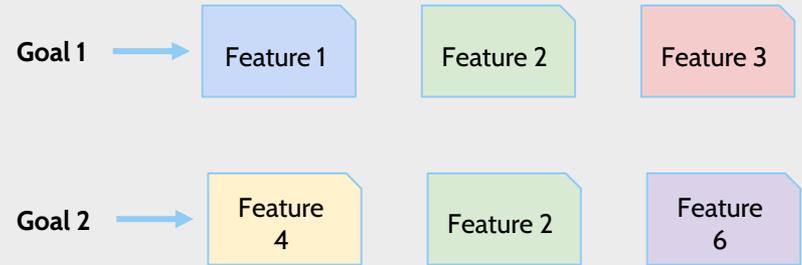
Step 2: Stakeholder pods assembled. Goals -> Features

The upside of having spent enough time working on your goals is that you will have more clarity of vision for your features.



My framework: From goals to features

- You know who is a stakeholder in each feature (sometimes multiple stakeholder pods)
- You'll actually know what are features that are organically meaningful to you.
- Because we will be reviewing some stuff regardless of whether it came up for you or not ...
- Giving goals a weight in your strategy will allow you to weight the importance of the features ... **in turn allowing you to score your options**



You could have come up with a list of features abstractly, but by deriving them from goals



My framework: From features to categories

Goal 1



Feature 1

Feature 2

Feature 3

Goal 2



Feature 4

Feature 2

Feature 6

Categories

Audience & Optimization

Feature 1

Business environment & Support

Content Creation

Feature 4

Content Management

Feature 6

Feature 2

Delivery & Distribution

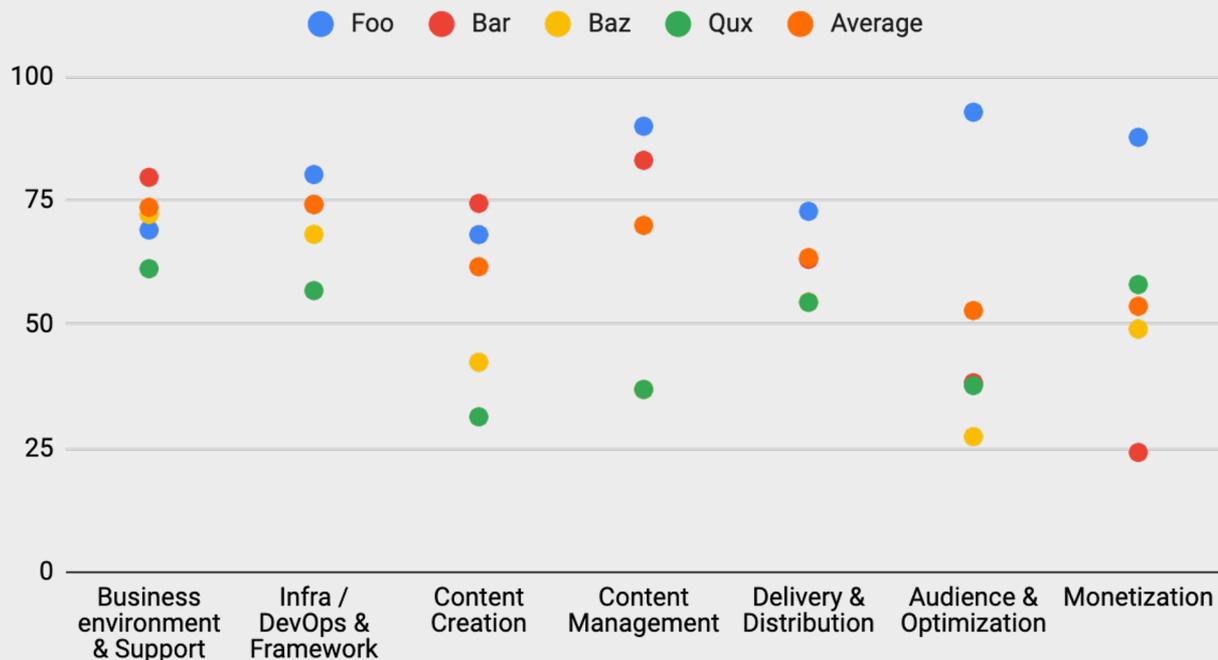
Infra / DevOps & Framework

Monetization

Feature 3

Analysis broken down by area - with *weighted* scores

By Category, Scatter Plot



**My
framework:
7 categories,
56 standard
feature items
(for now)**

Sidebar | About the Grande Dame: Print

(this is also applies for whatever is your legacy media if you have one: broadcasting, radio etc)

There are a few first filters that I'd qualify as hard: They will be so structuring that they will rule out entire approaches or possible solutions. Print is one such hard filter (multi-site or multi-language are other examples).

I can think of several 'good' publishing platforms that are so not oriented at use cases of integrating with a print CMS that I'd always rule them out for a legacy publisher.

Integrating with **Print** is a **specialty use case that your vendor is either ready for** because this vendor has already built the pipelines for *your* print CMS... **or walk away.** You don't want to be the first client here.

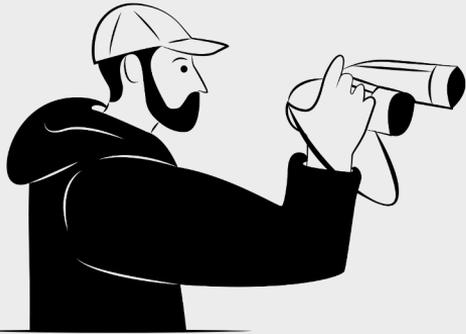


Part IV: Platform Vision

Your needs are 95% the same as other publishers your size but how you want to solve for the 5% will orient the platform you want

#5

Your platform vision



Flexible to 'code your own'?

Owned infrastructure / high devOps ownership?

Complexity of design requirements? On the-fly-layouts?

One-size-fits-all tools or speciality tools?

Mix and match components highest problem-solution fit? Buy the kit from one vendor?

Understanding a vendor's platform vision

**SaaS versus self-managed:
who drives the deployment of
your software and infra?**

→ SaaS is less effort (usually), but may be more expensive. Upgrades are usually not your problem // self-managed means infra is on you. More independence to apply upgrades but can be big overhead!

**Single-tenant vs. Multi-tenant:
Your own install or shared
code**

→ Single-tenant: + flexibility to make custom developments in the vendor's own code; Multi-tenant means customization happens only in modules you own, or via vendor-planned configuration management

**Plugins vs. microservice versus
Big App**

→ Plugins are molecules, Microservices are atoms, and a Big App is a whole organism.

**“You can integrate any third-
party”**

→ Do they mean it? Dig extra hard if the vendor actually makes a component where you're considering an alternate instead.

Mix-n-match or buy all from one vendor?

This is a question that often paralyzes: Vendor A does Goal 1 very well, Vendor B does Goal 2 very well. But it would just be easier to get everything from (just one of them).

Remember the scatter plot? My advice is to **divide and conquer** if one or two groups are **significantly weak** with one vendor but all other goals are high. Splitting things between vendors does introduce more overhead – but if there is a way to get fit with a vendor to a high degree on a **significant** portion of your platform, split the difference and investigate other vendors for specialty jobs. (if no vendor scores well across several categories, you haven't found a finalist)

Mix-n-Match for a particular function can be a slippery slope however: a specialized vendor has a strong bias to **serve their product vision with their 'full kit'**, and once you start to take only one component, you may never get the full benefits of whatever were the touted benefits.



Part V: The Key Items & Scores

Key items are the smaller subset of features that truly underpin your business. In terms of vendor selection, these are your king makers

#5

For Almost All Publishers, These Are Key Items

Audience &
Optimization

Business
environment &
Support

Content
Creation

Content
Management

Delivery &
Distribution

Infra / DevOps &
Framework

Monetization

Discovery:
Content model,
Rec engine, Tags

Article App

Workflow /
Newsroom
planning tool

Platform /
ecosystem
architecture

Revenue Service
/ Paywall

A/B Testing, site

Homepage and
section front
management

Key items?

Their weight in how I score a platform is kept at a fixed percentage to make sure that even if a platform 'does' more (more points), we don't lose track of the most important things

Scoring the features: what is 'state of the art'?

Do you need to worry about figuring out State of the Art for everything

- No, because not everything matters equally
- No, because many things can evolve in the future anyway (read: “don’t overthink it”)
- But I still recommend you grade against ‘state of the art’ – then you can just give less *weight* to something you ultimately are more flexible on. You therefore know which vendor is better than which vendor at a given feature... (“Jane is better than Joe at X - Jane gets an A and Joe gets a B; But we ultimately don’t care much about this thing, so it’s got a very light weight in our average. The difference between A and B is therefore negligible in the final grade.”)

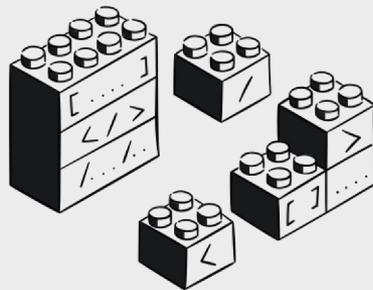


Under the hood: Platform architecture

Platform /
ecosystem
architecture

Micro-service architecture is both more desirable from a product perspective (**extensibility of framework to 3rd party tools where services are inherently all first-class citizens**) and more desirable from an engineering perspective since it allows **platform evolutions in less waterfall-y ways than large core apps**.

Reflecting the high multiplier I give this particular angle, I think it is the most structural factor to the choice of platform: it is not a feature the end user client (business or editorial) chooses on, but it does end up being what will paint the overall platform in a corner.



Plugin architectures are extensible by their nature but **are a two-class systems** and questions of compatibility between the core and plugins can quickly overwhelm. They are good if your needs are very fixed and you know that you can rely on a few industry-standard plugins with strong maintainers, whether large open source communities or private companies who are turning a good profit from their plugin users

In 2022, as a client with some budget flexibility, I wouldn't consider **any other architecture than micro-service**. If budget is far and away the main concern, plugin-based overall becomes appealing but note the caveat of that architecture.

The Article App: 85 percent of your traffic

One of the core features of a CMS for an article-led media publisher. I give this **at least 10 percent** of a final grade as one of the 'key items'.

The north star Article App has **great branching**, a **strong workflow-led editing experience** (articles are clearly expressing where they are in terms of readiness), **collaboration** (which means the editor allows co-creation - think Google Doc ideally - and the rest of the article app has **field-by-field locking**). Adding components to the article is at hand (any multimedia), adding social media is easy and doesn't lead users to behave badly (**no pasting of raw code**). A key test is **how abstracted your content is (API)** so future redesigns keep the article whole.

Good features for quality control like spellcheck, custom dictionaries are a minimal requirement for most newsrooms.

In terms of layout abilities, the north star is the possibility for the creator to manage **a number of presentation options**. *However*, this may not align with your own style as a publisher, so this may mean a readjustment of the article grade if you do not care for this at all (my grade would assume this is desirable).



Trains that leave on time: Workflow app

Workflow /
Newsroom
planning tool

There are two broad requirements scenarios, depending on whether you have a print component or not (if magazine with early closes: you are somewhere in between).

Daily Print: You definitely do want a strong workflow tool, that allows you to:

- Budget your editions (Web, and any print editions), and your promotion schedule (what you plan for social channels etc)
- Have a sense of what your staff is allocated, and have everyone's 'marching orders' be visible. Anyone on the team can know who is working on Story X
- Have a high level view of the production workline: Where stories are at in terms of production, or if any are just waiting for attention

Digital-only:

- Your need are lesser for budgeting. You still may have needs when it comes to commitments to publish certain content on a certain day (think advertising contract) but otherwise, your budgeting needs are easier.

What a good Workflow tool should HAVE:

- The ability to view your planning by many different facet: By desk, by team, by type of assignment (news vs enterprise), by day, by issue, by platform...)
- The ability to get an glance understanding of bottlenecks in your daily production.
- The ability to 'pitch': to a team, a surface, a delivery channel
- The ability to define custom workflows, ideally without coding.
- The ability to enforce your rules for advancing the workflow based on permissions, and 'role on duty' (this is more rare)
- Assignment features: Tasks and stories are assigned to CMS users. These CMS users can find 'their' tasks on their own screens
- Notification features: Minimally, in-browser and email. In 2022, Slack or MS Team
- Workflow app proactivity: Notifications of the app when irregular activity is noticed (your custom rules). I.e, "Article is 'ready for copy' but untouched for 15 minutes.



The flagship: Homepage management

The requirements diverge significantly based on three factors:

A. You want to manage these screens actively with a group of editors who are focused on these tasks (larger publishers in general)

B. You want to be able to create new sections and/or change their presentation and behavior with a wysiwyg tool

C. You want most of the content ranking be automated with little to no human engagement

Most organizations will be A + B or B + C. Some organizations want A, B and C -- this means part of their site is highly curated and other parts are not.



For A: easy simple control of ranking and styling. You also need the ability to control and possibly **fork these rankings** and stylings for a mobile screen. Good queues for newly arrived items

For B: You'll want **WYSIWYG tool to essentially bring in previously created blocks** that you are rearranging for a particular display effect. The ability to **start pages from scratch** (and setting metadata for these pages) should be limited to higher-responsibility roles in the CMS.

For C. You will want to see powerful-yet-simple (ever the dream) querying for the module: Ideally, the ability to reuse common queries across the site, and the ability to create new ones. A combined mix of "by publishing state, by importance, by theme" is the general scope of queries.

Content Discovery Uplift



Discovery:
Content model,
Rec engine, Tags

These are actually very different things – but I am grouping them here because they touch the question of how your users discover your content.

Tags, Collections, Guided discovery: This is what is going to allow you to combine and recombine your catalogue of content from anything to Election guides to your coverage of a particular company or politician.

Pay attention to your ability to define or customize the **content model**, allow the content APIs to take inputs from content flows outside your CMS, and output this to other systems (We review this in its own category - Feed engines)

Recommendation engine / personalization: Generally, expect your solution to use an out-of-the-box rec engine (from Amazon, or if GCP, via TensorFlow).

You may want custom recommender ML, but if you are using a rec engine for the (more common) purpose of **simple click-thru uplift** (by opposition to more controlled use cases), out-of-the-box models will do a good job.

If this is a use case that is relevant to your organization, a recommendation engine should be able to use differentiated scenarios (like Lifetime value on the reader, likelihood to subscribe), **though the rec engine needs to be able to be provided training data to match.**

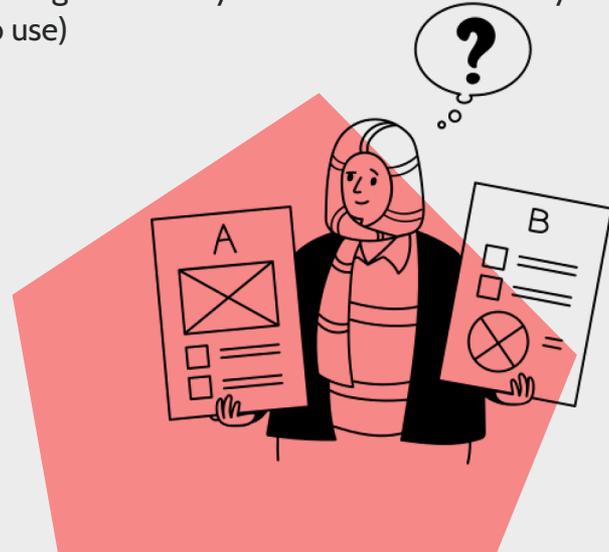
A/B Testing (Site)

Site A/B testing should be able to help you support things as broad as **entirely different functionality or templates to small details like copy changes**. For the most amplitude of tests, **both server-side and client-side** should be available.

This is an item that is **worth taking from the same platform as your CMS and Web frontend** (A/B testing for your native apps will come through dedicated systems) -- integration for A/B testing frameworks is delicate so taking a 'matching' component will be easier than bringing a third party. That said, **do not just settle for only client-side testing**.

Also, it is worth thinking of A/B testing capabilities early in your replatforming. The more deeply integrated, the most use you will be able to get for your tests.

Finally, look for a framework that can be extended to content testing (Sometimes, vendors will have dedicated tools for this, though fundamentally, content and site A/B testing use similar technologies -- a dedicated content A/B testing tool usually means there is a friendly UI for editors to use)



Make money: Paywall / Revenue



- Runs server-side for content delivery (this is the only fireproof paywall). Client-side only has inherent frailty.
- Handle hard, metered model, freemium, and mixed models. Donations and membership models
- Paywall runs on rules, or runs on learning models to optimize for conversions (higher end services)
- A WYSIWYG customer journey builder that a marketing team can directly manipulate and optimize as it learns what works and what doesn't
- Built-in A/B testing for the journeys and offers
- Offer management: Create, manage, end, convert offers (sunset and move to superior offer with old price), upsells, promo codes, time-limits, referral offers, bundles (may need to integrate with CRM)
- Entitlement management with third-party handling (ie, App store sales or Print entitlements)
- Analytics reporting on offers, and on funnel steps
- CRM integration (events of funnels kick back in CRM)
- Flagging for bad events: Failed purchased (has retries); Fraud attempts; and passive churning
- Paywall defense tools: Incognito browsers, AdBlockers, Multiple Devices login rules
- Identity management with SSO, SAML and the option to use other third parties
- Payment options to cover top credit card alliances and third-party payment handlers relevant in your key markets
- Payments in multiple currencies

Part VI: Takeaways

Front load **your vision for your business**, rather than an inventory of things you don't like about the current CMS

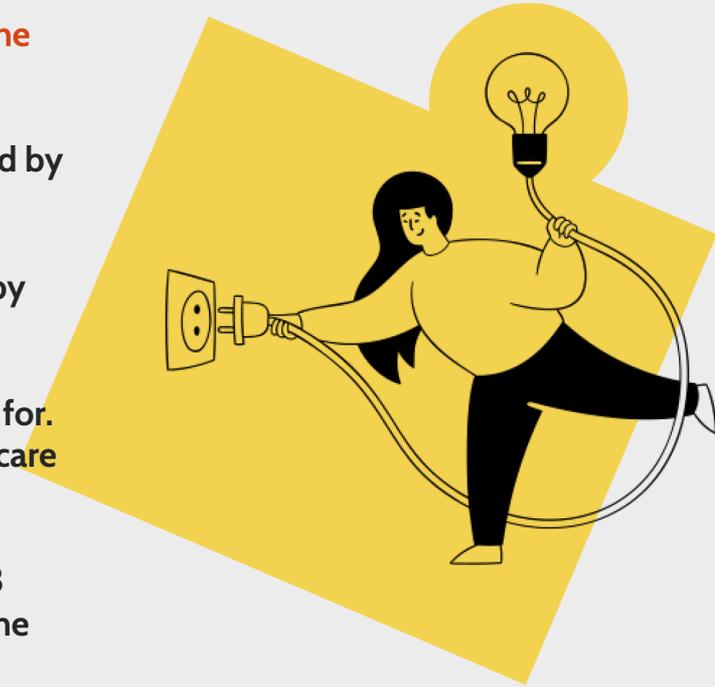
Approach building the team that will create the **RFP around the goals of the company**, rather than strictly the departments of the company

Get ready to **assess the RFPs in a structured manner** (so you're not swayed by who does the best sales deck).

Answer the **hard questions of your platform vision** so you're not swayed by great marketing. *"It's just one line of code" is. always. a. Lie.*

Have a sense of what is **state of the heart** for the features you are looking for. Grade against that north star, and lighten the weight if you don't actually care that much

Be ready to **split among vendors** if there a reasonable split between 2 or 3 max will give you **significantly more** than consolidating everything with one



Annex

- Often, we use **CMS** and **publishing platform** interchangeably: They aren't actually the same thing. CMS is narrow: it's the thing that allows you to create, manipulate your content internally. And technically, it wouldn't integrate your web frontend (headless CMSs are 'pure CMS', CMS that have the option of a frontend are so-called decoupled CMSs, and some come with a built-in web frontend)
Things like your paywall, your recommendation engine, your analytics or CRM are not part of your CMS or of the CMS. But they are often part of the items being considered in a **replatforming project**. Whether you'll end up actually taking 'the whole piece' as one thing or break it up may very well vary, in the end.
- I'll try to use the proper terms (publishing platform vs CMS narrowly) in this presentation
- **SME**: Subject matter experts
- **RFP - Request for Proposal**: the document(s) you will present to potential vendors that present your company, your problems, some of your high level goals and opportunities and sometimes but not always some of the solutions you want.

Annex: Lexicon and Concepts

Annex: What's *helio*

[Helio.cloud](#) is a SaaS publishing platform for independent creators (journalists, video creators, micro-influencers, course creators etc), specifically the more experienced who are branching across content types (like articles, newsletters, courses, ebooks, videos) and monetization models.

Helio is inspired by the kind of pro tools you'd find in a media company, but is boiled down for these micro-publishers. For example, it wraps in CRM for them and their monetization tools.

We are backed by [GMG Ventures](#) (The Guardian), and were incubated at [Founders Factory](#) in London. Helio is based in New York.

We're in alpha and launching soon (and fundraising soon). Always happy to connect if you want to hear more (but Helio, if it needed to be clarified, is absolutely not the right tool for an established media company)

WAN-IFRA Report | June 2022

Expert advice on choosing a CMS

Published by:

WAN-IFRA

Rotfeder-Ring 11
60327 Frankfurt
Germany
www.wan-ifra.org

CEO

Vincent Peyrègne

COO

Thomas Jacob

Director of Insights

Dean Roper

Author

Ariane Bernard, Founder, Hello

Credits

Illustrations: Milano via Streamline HQ (Figma)



World Association
of News Publishers

www.wan-ifra.org