

Consumer Mesh on Website Factory

Table of Contents

Consumer Mesh on Website Factory.....	1
1. Introduction.....	1
2. From the field.....	1
3. Diving into the details.....	2
4. An approach to the solution.....	2
5. Design and Design Systems.....	3
6. Design To Code and Content Sources.....	3
7. Consumer Mesh approach.....	3
7.1. Design.....	4
7.2. Content Model.....	4
7.3. Front-end code.....	4
8. Roadmap.....	5
9. Contact Us.....	5

1. Introduction

Multi-brand CPG organizations face a continual challenge with managing a multitude of websites across countries and brands. There are several key forces at play to balance, the key being standardization globally versus customization locally.

Combine this with the fact that:

- Browser technologies are evolving
- Core Web Vitals impact rankings
- Martech stack keeps innovating for Digital Experiences
- UI/UX focus and innovation keeps changing
- there is an higher need to delivering content to myriad marketing channels including Social
- Headless architectures seem to be making headway.
- Brands need increasing flexibility to create engaging experiences

This becomes a complex challenge.

2. From the field

From L'Oreal to GlaxoSmithKline Consumer Healthcare (now Haleon), we have seen different solutions to this problem, each creating their own homegrown solution.

- Standardizing Web templates at GSK CH developed in the 2010s, were constraining the brands and agencies who effectively ignored the templates, but still had to contend with the loading the

template JS and CSS, creating poor performing sites. And moving 1200 sites to a new platform became a multi-million dollar challenge over several years.

- L'Oreal created extensive infrastructure and empowered brands to create exceptional engaging experiences, but at a high operational cost.
- Ocean Spray standardized around Sitecore and its Components for the US brands, and were dealing with high cost of hosting the solution on Azure.
- Nature's Way moved from several Kentico CMS versions, to Sitecore, a project which took over a year for a handful of sites.

There doesn't seem to be a universal solution to this challenge, and each approach seems to have its own challenges.

3. Diving into the details

Brands want performant web presences, that execute their marketing and communication strategies to engage with their target market, whether it is to gather first party data, modify consumer buying behavior, or launch new products, or other tactical and strategic goals.

To keep costs low and adoption high, there is a need to standardize the approach to creating digital experiences.

The factory approach which began with building front-end code with templating languages, now extends to managing the entire digital stack that spans Cloud, CMS, Front-end and Design tools.

There are several benefits to the Factory approach:

- reduce cost of operations
- reduce time to market
- pre-built integration with sources of content
- focus on business self-service removing IT being a bottleneck

There are several disadvantages, depending on the original design and vision of the Factory.

- framework updates are time consuming
- deviation from standards are expensive
- smaller brands might find it too expensive
- platform has to be refreshed from 4-6 years given the innovations in digital
- backlog bottleneck for changes
- agencies have to be trained to adopt on the specifics of the factory

4. An approach to the solution

Keeping the consumer as the core focus, and re-orienting the solution from their perspective yields an ideal solution, or at least a North Star to the ideal solution.

User Experience, then becomes the single most important aspect of the entire digital supply chain.

- Consumers are digital savvy, and their expectations keep evolving to new standards pushed by Amazon, Google, and Social Networks.
- Giving what the customer wants becomes paramount, while still enabling brands to modify consumer behavior (educate, engage, etc.) and achieve strategic business goals.

The key principle to enabling good or even great User Experiences is simple:

Good design at the outset, coupled with delivering changes quickly as we learn more about the consumer needs and when business goals change.

How can we orient a technology platform to deliver on this seemingly simple concept?

5. Design and Design Systems

Branding has always focused on key elements of communication, a unique design that communicates their brand, and distinguishes from competition.

Standardizing on design systems and a visual language for each Brand is important, especially given the marketing channels for content, including Social Media that can flood the consumer and make engagements challenging.

Each brand or umbrella of related brands can create a standardized Design System and establish a Visual language.

The Design principles, including Design Systems, with a focus on enabling exceptional User Experience is built into the mature design process.

The focus on Design and User Experience needs to be implemented in the final website.

6. Design To Code and Content Sources

If good Design, and delivering constant changes are key, then exploring a solution that translates Design to Code is a worthwhile endeavor.

Figma as a tool for creating Design and components for Design Systems is quite powerful. It is also a tool that is familiar to designers and agencies worldwide. It gives Brands the Visual Language and Design Standards for their brand. You can rapidly prototype ideas from changes to color, to user journeys, advanced micro-interactions and animations.

For content on the final website, everything that appears in Design needs to come from a content source. Either directly from a CMS, or from another system like PIM, ECommerce, etc.

7. Consumer Mesh approach

With Consumer Mesh, Designs are converted to Front-end code, and content model for authoring. The current tech stack outputs Content Model for Drupal 10, and front-end code with NextJs. The front-end

code builds as a Static Site. As part of the build, it pulls the latest content from Drupal, and pushes to a CDN.

7.1. Design

Design to Code is a complex problem, with several tools that limit the flexibility of design.

Consumer Mesh supports most features of Figma, to enable a fully functioning responsive website.

- Frames, simple Fills, Vectors, Components, Component Sets with Variants, Interactive Components, Basic Animation are all supported.
- more features are added throughout the evolution of Consumer Mesh

On the flip side, not everything that is possible in HTML, CSS, and JS is supported by Figma (or would be time consuming to design out completely).

We allow additional configuration in the plugin to support more than what is possible in Figma, while preserving the designers' intent. This includes integrating with third-party React components.

For example, you can design a Radial Chart using Ellipses, and angles in Figma, and we translate it to recharts, a Javascript library to render what the designer intends. This particular example of visualization is a sore spot for Designers, as they are limited to what the developers choose for the visualization library.

The goal is to give designer's full control of the User Experience, without constraining their creativity.

Forms are an important aspect of User Experience, that is automatically generated based on design. Designers can create complex user interfaces for a Form field, which is passed to the Form Submit.

7.2. Content Model

From the same design, Consumer Mesh also generates a Content Model for Drupal.

This content model includes features for:

- Repeating Content like Cards that are separate content types
- Common content that is used across pages
- Taxonomy terms that are added to simplify relationships and support tagging
- Responsive images feature of Drupal is used to render images for the static site
- Drupal Webforms are used to manage form submissions from the Front-end code. Notifications and workflows around Drupal Webforms can be implemented easily.
- Open Graph and other meta tags are managed within Drupal by the content authors.

7.3. Front-end code

The front-end code of Consumer Mesh is a NextJs web application. NextJs' focus on Core Web Vitals, and support for Typescript lends quite well to Consumer Mesh's approach to generating code. The key aspects of front-end code:



- Static Site Generation, with client-side data refresh. This allows content to be available on the server, which is great for SEO and Search crawling. And client-side data refresh allows for the newest content to be updated. This is useful for long lists of data with pagination. This approach brings the best of static server content rendering, while keeping large content pulls on the client-side.
- This also means, that all content that needs to be shown on a website does not need to be residing inside the CMS. So, data integration does not need to move content from other systems into the CMS for display on a website. Content is pulled at build time.
- Static Sites are hosted without an actual server that needs to be managed. A simple CDN works. Reduces complexity and cost of the infrastructure. Note: images are still served through Drupal or other Digital Asset Management. This leads to possibly the best Core Web Vital scores for performance globally.
- Styles accurately reflect in generated CSS from design.
- On Azure Static WebApp, Consumer Mesh generates API proxies to avoid CORS issues. These APIs run as Azure Functions inside the Static WebApp. This is useful for Contact form submissions, preserving the security of API tokens, etc.

8. Roadmap

Consumer Mesh aims to improve its product continuously, to be on the latest platform and technologies. Our features are built based on real customer needs. With that in mind, this is the general road map:

- Simplified content management within Figma, to remove Drupal dependency for simple websites. This reduces the cost of operations to just a static website for simple web presences.
- Support disparate content sources based on REST API. This includes PIM (Product Information Management) systems and other internal systems for pulling content directly to generate websites. Mapping the visual element to the content source is done through the plugin.
- Authenticated experiences. Integrate with Identity providers to create a secure authenticated experience.
- Integration with headless systems, like eCommerce, Payment Providers, and other internal systems.
- Portal experiences with dashboard widgets, and Form capture to custom API destinations.

9. Connect with us

 Senthil Prabakaran, co-founder, CEO. senthilp@consumermesh.com

 Dena Borsand, co-founder, CPO, dena@consumermesh.com

Website: <https://consumermesh.com>