WHITEPAPER

Content vs. Data

A Key to Digital Disruption: Separate Digital Asset content from Business Data

pitss

For today's Market Share Wednesday, as part of the Modernization Readiness Boot Camp, we'll be discussing the differences between Digital Assets (or "content") and Business Data (or "data") in an architectural sense. Let's begin with some definitions.

Digital Asset Content

Information and presentation assets or graphics that are part of the UX. They tend to change rapidly with the needs of the marketing and sales departments, and also tend not to affect core business logic. Content is what users see, touch, and remember in an application's user experience. It's what users think of when they imagine your app. Content is close to users, and by extension, front-end applications. That makes it different from the deeper, more global parts of your architecture.

Business Data

Data is business information that drives how you fulfill customer requests and orders. Data is much closer to operations than to sales and tends to drive all core business logic. Shipping updates from your warehouses, billing history, and account information are all data, quite different in nature from what we call "content." Data is what makes your app work. You need data to process services for your users, and content to make users want to buy. Data is more global than content.

Monolithic applications tend to present all information directly from the database to the screen. Whether it's what we consider Digital Asset content or Business Data, monolithic apps store it all in the same database. This frustrates our efforts to make a legacy application truly helpful to marketing teams that want this application to play a customer-facing role. They want to re-use the logic, but they also need to control the aesthetic parts of the application.

Modern web-based applications make this separation of content and data possible. In the client layer, modern apps focus on the end user, the user experience, and the graphical experience of interacting with the enterprise. They leverage business logic and Business Data from the service layer. A well-made web application can re-use legacy logic and business data all while giving the marketing team the control they need to effectively communicate the organization's brand, style, and identity.

The user experience layer needs its own Digital Asset content to remain nimble and adaptable, different from the business logic layer that needs Business Data to be global and consistent.

When breaking down a monolith, we face an important question: how do we distinguish content from the business data that delivers customer value? UX is more than just styling and colors. It includes data that drives how the application functions at the UX level. Content may even be used to create business logic data to drive transactions or create sales, but the two are still different - as different as a one-time promotion offer is from a signed purchase order. This is the distinction between content and data.

Breaking Down the Monolith

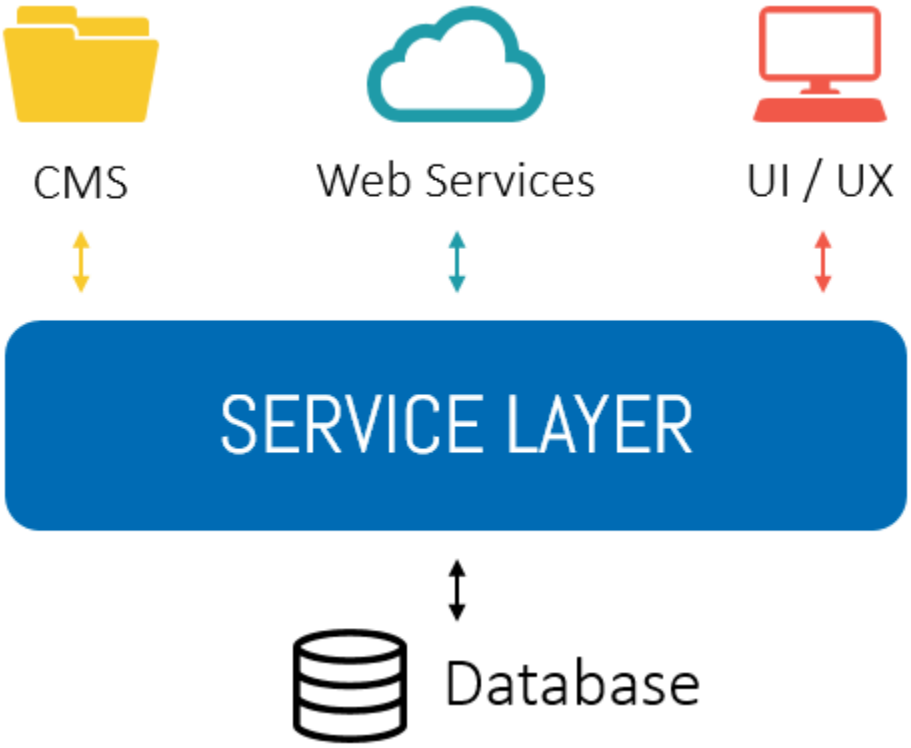
When breaking down the monolith into layers, we achieve a simpler overall architecture if we manage data through rigorous service layers, but keep content data managed closer to the UX directly.

For example: envision an eCommerce portal that collects product information from an API service layer that communicates with your legacy product catalog, but runs scheduled promos and discounts on the items out of its local database. If those promos are specific to that store, we might think of them as "content," just like the styling, because it does not affect the underlying data of the catalogue. It also separates the management of the catalog data from the promotional data, which makes it possible to easily align the software tools to the business units who use them (Product Management for the core catalog, Marketing for the eCommerce portal). The content here does not change the catalog or parameterize the process of order fulfillment. Instead, its goal is to consume the catalog, generate more inputs to the order process.

Setting the Standard

It takes a lot of rigor to update your service layer to speak to your systems just like your most important database. Content has a tendency to change its structure and volume very rapidly and it will not affect the underlying business logic. That's a key benefit to managing your content in a different way than your core (legacy) system - change is isolated to just the appropriate areas of your enterprise architecture, which enables the nimble processes that make digital transformation work.

This is an important distinction because we must treat data and how we manage and provide it with great rigor. If we give that same standard to content, we'd never get anything done. We'd lose our ability to represent the data and processes in flexible ways, losing our ability to gain market advantage.



The bottom line is that we need the right architecture to manage these concerns separately. For example a content management system can store all content and related information. Marketing can have control over this system so that they can take advantage of the content inventory to make rapid decisions and create custom campaigns. This CMS is managed separately from a suite of web services that provide access to the core business logic in the form of data.

This way, a web-based store or other consuming application can get access to both content and data to create a robust and flexible end-user experience, but the different areas are managed by the business units closest to them, creating more efficiency.

About the Author

Ross Smith is the Chief Architect at PITSS. He is responsible for designing and planning legacy application transformation projects at large enterprises in the United States and Europe. By specializing in rapidly transforming legacy applications into modular elements of modern enterprise architecture, Ross brings practical perspective on the implementation of microservices at enterprises that must account for their investments in older IT systems.

About PITSS

PITSS modernizes, customizes, migrates, and extends the life of valuable legacy systems. Using data, proprietary software, and deep-dive analysis, we reduce the cost and scope of your digital transformation effort by focusing on the highest ROI processes first. Then, we pave the complete road between you and your new horizons with confidence and expertise. Our full-stack team of experienced UI/UX designers, developers, engineers, technical leads and project managers can guide you through each step of a project. No one knows Oracle Forms like we do—we can take you from planning, to pilot, to the cloud and beyond.

Meet our team to see how we can transform the systems that matter most to your business.

Contact us today.

WWW.PITSS.COM **1.248.740.0935** **US.SALES@PITSS.COM**