

## Chapter 2 Points of Comparison

- Target Site Type
  - Systems Versus Implementations
  - Open Source Versus Commercial
  - Technology Stack
  - Management Versus Delivery
  - Coupled Versus Decoupled
  - Installed Versus Software-as-a-Service (SaaS)
  - Code Versus Content Versus Configuration
  - Uni- Versus Bidirectional Publishing
  - Practicality Versus Elegance, and the Problem of Technical Debt
- نوع الموقع المستهدف-
  - الأنظمة مقابل التطبيقات
  - المصدر المفتوح مقابل التجاري
  - تكنولوجيا المكس
  - الإدارة مقابل التسليم
  - تفرق مقابل موصول
  - المثبتة مقابل البرامج كخدمة (SaaS)
  - الرمز مقابل المحتوى مقابل التكوين
  - النشر أحادي الاتجاه ثنائي الاتجاه
  - التطبيق العملي مقابل الأناقة ، ومشكلة الدين الفني

19

## Why do we compare

من خلال فهم مجموعة الخيارات المتاحة ، يمكننا البدء في فهم النطاق الكامل للخيارات

- By understanding the range of **available options**, we can begin to understand the full breadth of options.
- To demonstrate this, it might help to examine the aspects of content management as a series of **comparisons** or dichotomies



20

نوع الموقع المستهدف

## Target Site Type

تستهدف أنظمة إدارة المحتوى المختلفة أنواعًا مختلفة من المواقع

- Different CMSs are **targeted** at different types of sites: -\_- نحس مش مهمات بنفوتهم

1. A small, static marketing site for a dental office
2. A multinational newspaper publishing hundreds of articles a day
3. A single-author blog for a technology writer
4. An intranet for a medium-sized accounting firm
5. An extranet for the dealers of a farm implement manufacturer
6. The product documentation library for a software developer
7. The course materials and syllabi for a small university
8. An online community and social network for owners of Porsches



1. موقع تسويق صغير وثابت لمكتب طب الأسنان.
2. صحيفة متعددة الجنسيات تنشر مئات المقالات يوميًا.
3. مدونة مؤلف واحد لكاتب تقني.
4. إنترانت لشركة محاسبية متوسطة الحجم.
5. إكسترانت لتجار الشركة المصنعة لأدوات المزرعة.
6. مكتبة وثائق المنتج لمطور البرامج.
7. المواد الدراسية والمتاهج الدراسية لجامعة صغيرة.
8. مجتمع على الإنترنت وشبكة اجتماعية لأصحاب سيارات بورش.

21

## Systems Versus Implementations

- It's important to **separate** a content management **system** from a CMS **implementation**.
- An implementation is the process by which a CMS is **installed, configured, templated, and extended** to deliver the website you want.
- Unless you build your CMS from **scratch**, you are not the only one using it. Other organizations are using the same software to solve different problems and deliver different types of websites, so it's not going to be preconfigured to do any one thing particularly well.
- This means a necessary step is the **initial effort** of adapting the CMS to do exactly what your organization and circumstances **require from it**.

التنفيذ هو العملية التي يتم من خلالها تثبيت نظام إدارة المحتوى وتهيئته ووضع نموذج له وتوسيع نطاقه لتقديم موقع الويب الذي تريده.



هذا يعني أن الخطوة الضرورية هي الجهد الأولي لـ تكيف نظام إدارة المحتوى للقيام بالضبط بما تتطلبه مؤسستك وظروفك

22

important

## Platform Versus Product

لا توجد وظائف  
CMS

على الإطلاق ، مجرد منصة برمجة خام

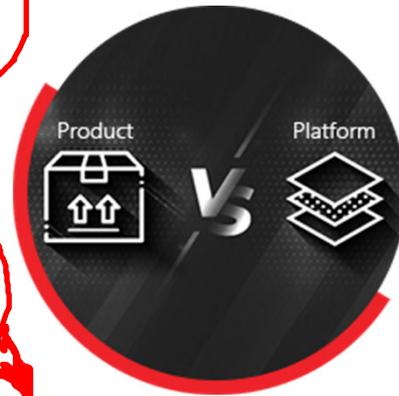
- If we consider a CMS a range of functional “completeness,” the extremes might look like this:

- **No CMS functionality** at all, just a raw programming platform

- **A fully functional CMS** ready to go out of the box, complete with prebuilt features to solve all your content problems with no changes

- In between these two extremes, we can insert a third option:

- A programming framework providing flexible **API** access to common content management features and functions that can be used to develop your own **solution**, along with a default **user interface** and configuration to support common needs



بين هذين النقيضين ، يمكننا إدراج خيار ثالث: • إطار عمل برمجة يوفر وصولاً مرئياً إلى واجهة برمجة التطبيقات إلى ميزات إدارة المحتوى الشائعة والوظائف التي يمكن استخدامها لتطوير وظائفك الخاصة الحل ، جنباً إلى جنب مع واجهة المستخدم الافتراضية والتكوين لدعم الاحتياجات المشتركة

23

## Open Source Versus Commercial



24

## Technology Stack

تعمل جميع البرامج على "مكدس" من برامج الدعم وقواعد البيانات واللغات، يتم دائماً تنفيذ بلغة وتخزين معينين CMS (والتي قد تكون "قابلة للتبديل" أو لا).

- All software runs on a **"stack"** of supporting software, databases, and languages. A CMS is always implemented in a specific language and storage framework (which may or may not be **"swappable"**).
- This language and storage framework strongly influence what **hosting environment** the CMS needs to run.
- The stack includes the following:
  - The CMS itself
  - A programming framework
  - A programming language
  - A database server
  - A web server
  - An operating system



يؤثر إطار عمل اللغة والتخزين هذا بشدة على بيئة الاستضافة التي يحتاجها CMS لتشغيلها

- تشمل المكدس ما يلي
- نظام إدارة المحتوى نفسه
- إطار برمجة
- لغة برمجة
- خادم قاعدة البيانات
- خادم الويب
- نظام تشغيل

25

## Management Versus Delivery

كل ما يحدث للمحتوى من لحظة إنشائه حتى موته هو "إدارة"

- Everything that happens to content from the moment it's created until the moment it dies is **"management."**
- The subset of everything that happens to the published version of that content from the moment it's published is **"delivery."**
- The two disciplines are quite different.

النظامان مختلفان تماماً



المجموعة الفرعية لكل ما يحدث للنسخة المنشورة من ذلك المحتوى منذ لحظة نشرها هي التسليم.

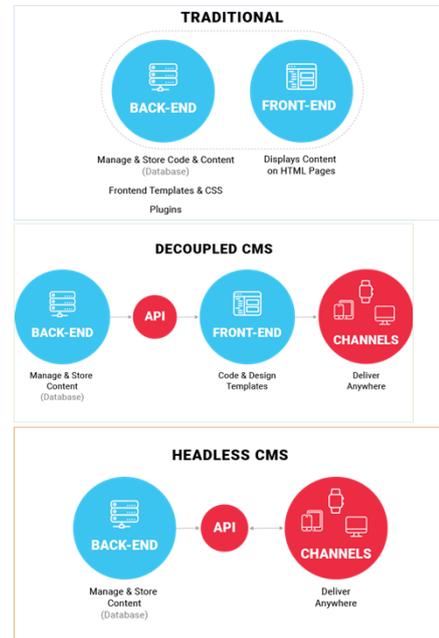
26

## Coupled Versus Decoupled

في النظام المقترن ، تحدث الإدارة والتسليم على نفس الخادم. يدير المحررون المحتوى على نفس النظام حيث يستهلكه الزوار.

- In a **coupled system**, management and delivery occur on the **same server**. Editors manage content on the same system where visitors consume it.
- In a **decoupled system**, management and delivery are **separated**. Content is managed in one environment and then published to a separate environment (another server).

في نظام منفصل ، يتم فصل الإدارة والتسليم. تتم إدارة المحتوى في بيئة (واحدة ثم نشره في بيئة منفصلة (خادم آخر



27

## Installed Versus Software-as-a-Service (SaaS)

يتم نقل المزيد والمزيد من البنية التحتية لتكنولوجيا المعلومات إلى "السحابة" ولا تختلف أنظمة إدارة المحتوى

- More and more IT infrastructure is moving to **"the cloud,"** and CMSs are no different.
- While the norm used to be installation and configuration on your server infrastructure, vendors are now offering hosted or SaaS solutions more often.
- The benefit purports to be a CMS that is **hosted and supported** by the vendor that developed it.\*



تدعي الميزة أنها نظام إدارة محتوى يستضيفه ويدعمه البائع الذي طوره

\*<https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/>

28

## Code Versus Content Versus Configuration

سيشمل تنفيذ نظام إدارة المحتوى دائماً نوعين من كود البرمجة على مستوى ما. سيكون للنظام

- The implementation of a CMS will almost always involve two types of programming code at some level. The system will have
  - (1) customizations that are developed in the native code of the system (PHP, Java, or C#, for example), and
  - (2) custom templating and the associated HTML, CSS, and JavaScript.
- Content, on the other hand, is developed by **editors** and lives in the CMS. In coupled systems, it's often developed in the production CMS and just kept **unpublished** until it's ready to launch.\*



النخصيصات التي تم تطويرها في الكود

من ناحية أخرى ، يتم تطوير المحتوى من قبل المحررين ويعيش في نظام إدارة المحتوى. في الأنظمة المقترنة ، يتم تطويره غالباً في نظام إدارة المحتوى الخاص بالإنتاج ويتم الاحتفاظ به غير منشور حتى يصبح جاهزاً للإطلاق  
<https://themeisle.com/blog/how-to-install-wordpress/>

29

## Uni- Versus Bidirectional Publishing

قبل أن تكون وسائل التواصل الاجتماعي شيئاً ، وقبل توقع مشاركة المستخدم في مواقع الويب ، كانت معظم أنظمة إدارة المحتوى أحادية الاتجاه

- Before **social media** was a thing, and before user participation in websites was expected, most CMSs were very **one-way**.
- Editors published content that visitors consumed, blindly. Visitors didn't have accounts on websites. They couldn't create "**profiles**" or have "**discussions**."



قام المحررون بنشر محتوى يستهلكه الزوار بشكل أعمى. لم يكن لدى الزوار حسابات على مواقع الويب. لا يمكنهم إنشاء "ملفات شخصية" أو إجراء "مناقشات"

30

## Uni- Versus Bidirectional Publishing-cont.

يعني النشر أحادي الاتجاه (اتجاه واحد) أن مؤسستك تدفع دائماً المحتوى "للخارج" إلى المستهلك ، غالبًا بشكل أعمى (مما يعني أن المستخدم مجهول الهوية ، ولم يتم تسجيل دخوله أو تحديده بطريقة أخرى).

- **Unidirectional (one-way) publishing** means your organization is always pushing content "**outward**" to the consumer, often blindly (meaning the user is anonymous, and not logged in or otherwise identified).
- **Bidirectional (two-way) publishing** means the consumer can sometimes push content "**backward**" to the organization (by posting a comment, for instance).



يعني النشر ثنائي الاتجاه (ثنائي الاتجاه) أنه يمكن للمستهلك أحيانًا دفع المحتوى "للخلف" إلى المؤسسة (عن طريق نشر تعليق

\*Content coming back from the user is known as **user-generated content (UGC)**

(UGC) يُعرف المحتوى الذي يتم إرجاعه من المستخدم بالمحتوى الذي تم إنشاؤه بواسطة المستخدم

31

## Practicality Versus Elegance, and the Problem of Technical Debt

عند التفكير في وظائف جديدة ، هناك معركة مستمرة بين

- When considering new functionality, there's a constant battle between
  - (1) "**just getting it done**" and
  - (2) taking a **step back** and rationally considering the best way to fit this functionality into the bigger picture.



إنجازها للتو" و (2) التراجع والتفكير بعقلانية في " أفضل طريقة لملاءمة هذه الوظيفة في الصورة الأكبر.

32