Acquia™

Drupal and the Assembled Web

Why web assembly?

Drupal is a platform for assembling web applications. Drupal provides enterprise architects, web designers, and application developers with an open framework and a social content infrastructure for rapidly developing content applications. With Drupal, it is easy to produce engaging, content rich websites, incorporate both structured and unstructured data sources, and blend published information with user-generated content.

Drupal delivers a web application development environment for the enterprise, oriented around content and community. Rather than simply relying on a framework, a database, and application-level programming tools, designers and developers begin with a rich set of content and community-enabled modules. They proceed to quickly assemble an interactive application from these building blocks. When required, they can extend existing modules to deliver additional functions, or develop new ones to accomplish new tasks. As a result, organizations can rapidly and cost-effectively deploy dynamic web experiences that foster engagement and interaction, and thus enhance their online brand.

Let's consider how the assembled web transforms website design, development, and deployment by focusing on the underlying social content infrastructure.

Exploiting a social content infrastructure

Great expectations

As enterprise customers doing business on the web, we expect great experiences that solve business problems. The information we access must make sense to us in the context of doing our work.

MSKU#: 0100-110831

For example, if we are considering attending a professional conference, we quickly want to know the agenda, the speakers, the cost, and the location, and then be able to register with just a few clicks. When we have a problem with a product and need support, we want to easily find the answers to our questions and also engage with communities facing comparable issues. As we track the trends related to a critical financial trend, we want timely insights from multiple news sources, a quick overview of performance metrics, and rapid reporting of the significant facts that help to influence our decisions.

While our tasks may change from one situation to another, we rely on the web to easily access, organize, and consume business-related content. Things change quickly in the digital age, and our online resources need to keep up with the speed of business. We expect our web-based applications to rapidly adapt to our changing business environment.

Building dynamic experiences

But web experiences are the results of design decisions. When building a website, designers organize and present information. First they identify the content that we (the target audience) should receive. Then they define where this information is going to come from, how it is going to be presented, where communities will engage, and what we should do to complete our tasks.

Site builders and application developers are no longer limited to displaying information within self-contained, static, web pages. Rather, to produce an engaging experience, a contemporary website encompasses various content chunks, organized and managed in a dynamic fashion. There is an order among these chunks. They can be grouped into different content types, have descriptive fields (or metadata) associated with them. Fields, in turn, are useful for tagging the meaning of content, such as when optimizing results for search engine crawlers.

When building a website, it's essential to define the underlying content types. Different types have different predefined sets of fields. Site builders create connections among content types based on field-level values. They can assemble multiple content chunks into a web experience and render the results as a series of content mashups. If site builders need additional links among content types, they can add fields to create new types.

Application developers, in turn, can focus their efforts on developing the underlying modules for websites and enriching the site builders' activities. Modules invoke actions and deliver predefined functions. Developers can easily create a new module by simply adding functions to an existing one.

In short, site builders and application developers need to be able to work at the speed of business, to rapidly assemble and transform websites that meet changing business requirements. It is important that they start at the right level – the content level rather than at the programming tools level. It is critical to begin with a modular, flexible, and extensible social content infrastructure for deploying engaging web experiences.

Assembling a web experience with Drupal

This is where Drupal makes a difference. Drupal is a content-aware framework for assembling web experiences. It includes the underlying repository and application development tools for managing web content and for supporting online communities. In addition, Drupal provides site builders and application developers with a range of content-aware capabilities to jump-start their own website development efforts.

Let's focus on four key capabilities: how Drupal maintains content, secures content, connects content-centric modules, and renders content. The end result is a flexible and extensible social content infrastructure. Enterprises can readily use this infrastructure to easily assemble and rapidly deploy compelling websites and communities.

Maintaining content

To begin with, Drupal defines content in terms of content types, nodes, references, and views, as shown in Illustration 1. When assembling a web experience, it is important to first define the relevant content types and then identify the activities for generating the individual instances of particular types. Here's how Drupal structures a web experience.

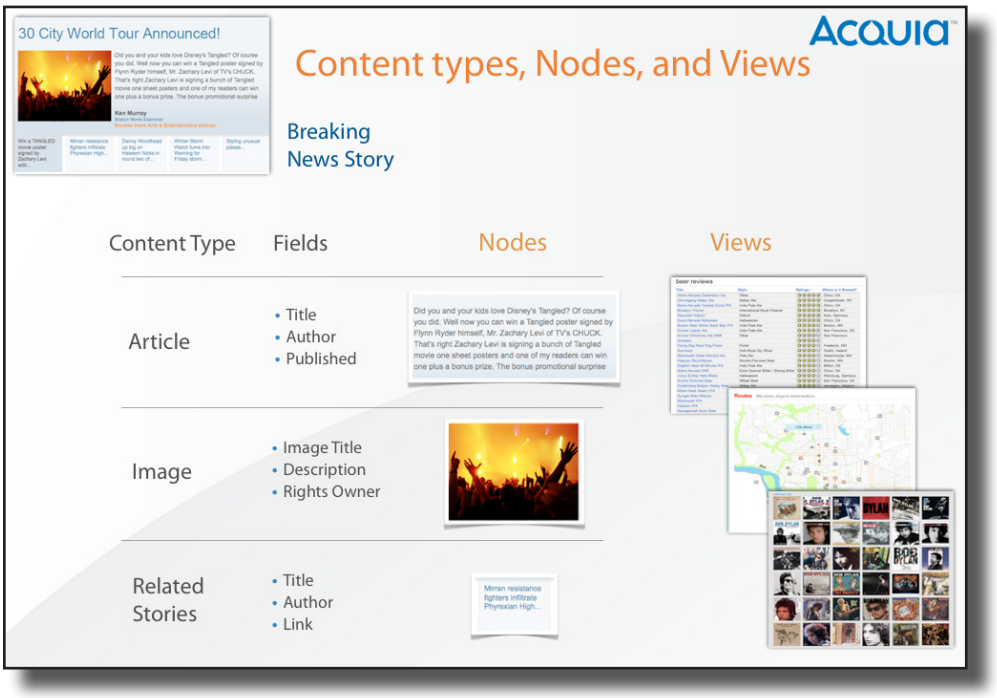

Illustration 1. Drupal defines content in terms of content types. Each content type has a predefined set of fields. Nodes are instances of content types and include the field-level values. Views relies on the fields to render content into an on-screen display.

- **CONTENT TYPES**. To begin with, Drupal defines content in terms of content types – chunks of digital information tagged with relevant fields that describe this information. Examples of content types include a blog post, a wiki entry, a news article, a research summary, an instructional video, and a financial report. Thus a news article is a text file and has such fields as "title," "body," "image," and "subject keyword." An album track is an audio file and has such fields as "artist," "title," and "genre."

- **NODES**. Nodes are instances of content types. Drupal proceeds to collect, store, and manage web-related content as discrete nodes. These self-contained content chunks can include such items as text for a report, the data elements for a table, a customer-service poll, a video snippet, or a blog entry, plus the sets of fields that describe them.

  Each node has a predefined set of fields, defined by its content type. As a result, nodes provide a very flexible foundation for content delivery. Site builders can rely on fields to easily combine nodes in many different ways to manage and publish the content for a website.

- **REFERENCES**. Content in Drupal can be "referenced." Thus, the terms provided by content types can be used to create and manage references among nodes. For instance, a financial analyst can post a blog about a company's earning report that is automatically indexed by the firm's name and associated with a catalog of the firm's goods and services. References add meaning to links among content types.

- **VIEWS.** Finally, Drupal includes Views, a powerful content mashup capability to query the content encapsulated by the nodes and to present the content to business users in meaningful ways. Views can be used to do everything from presenting simple, reverse-chronological lists of financial analysts comments, to creating a rich display of a company's performance against key performance indicators in multiple locations, to publishing syndicated research reports, and much more. With Views, site builders and application developers have extensive flexibility to present content for maximum impact.

Securing content

To secure the content delivered over the web, Drupal provides an elegant user management framework that adapts to business needs. To begin with, Drupal supports different sets of users, as shown in Illustration 2. Some might be anonymous, while others are authenticated and organized into groups based on roles. In addition, Drupal maintains access controls on individual nodes or on groups of nodes, defined by content types and/or stored within predefined collections (comparable to folders within a file system). Permissions determine access to Views.

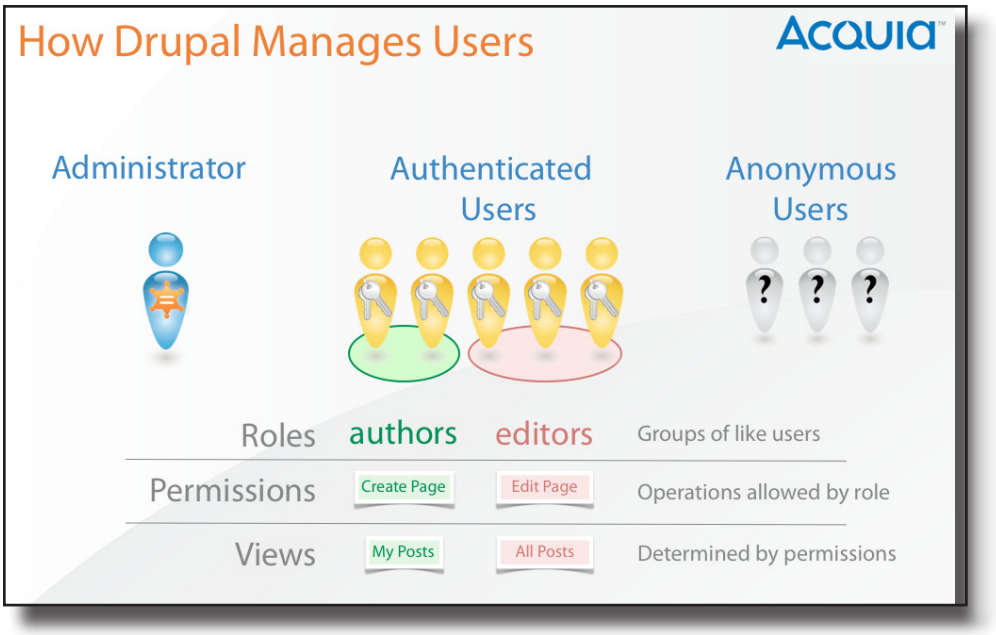

Illustration 2. Drupal manages users by their roles and permissions. Access to Views is determined by permissions.

For example, authenticated users within a market research firm can include analysts, editors, and supervisors.

- Analysts can create and modify their own content (including reports, blog-posts, and videos), and review research produced by other team members.
- Editors can also modify content produced by all analysts on a team.
- Supervisors can not only modify all of a team's content, but also decide when to publish it to external audiences and thus make it accessible to anonymous users.

Moreover, authenticated and anonymous users can access the same content and yet have entirely different experiences based on separate perspectives. Drupal can easily change how content is rendered to various devices, based on roles. Thus, as power users, analysts and editors might see company names and addresses in a list; by comparison, anonymous users might find the same information plotted on a map.

With Drupal, security and access rights can be associated with actions. This provides an additional level of interaction and automation. For instance, blog posts by analysts in a market research firm might be subject to a four-hour review period for regulatory compliance. Drupal can delay publishing a blog post by this period of time, automatically alert a regulatory editor to review it, and log what happens.

How modules create experiences

Drupal provides a range of capabilities for directing the flow of content to meet predefined business requirements. Drupal is based on a modern, modular development environment where both application developers and site builders are able to extend and enhance the environment.

Drupal includes a core set of approximately 30 modules that define its basic functions, together with over 8,000 community modules, contributed by members of the Drupal community.

COMBINING MODULES. In many situations, site builders can combine existing modules in new ways to target content delivery, extend interactive experiences, and address particular business requirements. They can add new capabilities to their web experience without relying on application developers.

For example, there are separate Drupal modules for rating content, organizing views, and performing bulk operations. Using these three modules, a site builder working in a financial services firm can easily produce a Drupal-powered experience that tracks customers' ratings of research reports and provides feedback to analysts who produced them.

- Customers rate content using the content rating module
- Ratings are then organized into a view for analysts using the views module
- The rating process runs at a periodic interval that makes business sense (for instance daily), using the bulk operations module

To assemble the business experience, the site builder defines the common fields and content types used by all three modules. These design insights are based on business requirements. With the needed functions encapsulated within the modules, there is no need for application developers to get involved.

DEVELOPING MODULES. Of course, there are times where coding assistance from application developers is required. As an Open Source project, application developers can access the Drupal source code, and add new code to extend and enhance particular functions. To address unique business situations, application developers always have the option of modifying an existing module or developing a completely new one.

For instance, suppose a market research firm needs to support mobile workers while also ensuring compliance with regulatory requirements. The firm can deploy a Drupal site that supports varying levels of content security, depending on the network connection.

- When working within the firewall (defined as the company intranet), financial analysts can simply post their comments to a blog running on a Drupal site.
- When working remotely, the firm can require its staff to further authenticate themselves before posting to a financial blog and add a time period for internal review before the post is published.

An application developer can easily modify the security and publishing modules for Drupal, and add just a few lines of code to satisfy the firm's requirement for mobile workers.

Rendering content with themes

To render content, Drupal includes a series of themes, comprising one or more cascading style sheets (CSS) and one or more PHP-based templates. A theme defines the presentation of a node or a collection of nodes displayed within a device (such as a laptop web browser).

Themes separate the presentation of content from the storage of content. Theming provides site builders with the flexibility for unique designs, so that sites can be coordinated with corporate branding guidelines and design standards. Themes can be created to render content based on the requesting client type – enabling a laptop browser experience to include Adobe Flash animations while an iPhone or iPad experience to produce animations with other technologies (such as Java script or HTML5). Themes can be used in conjunction with content security to ensure that the right content and functionality is presented to the right users, based on their roles and permissions.

Calculating the Drupal difference

In short, Drupal highlights the power of a content and community infrastructure for the assembled web. Not only can application developers and site builders rapidly produce the interactive experiences that meet current business requirements. With Drupal, they can also rapidly adapt and easily extend existing websites to keep up with the speed of digitally driven businesses. They can deliver socially aware environments that support business communities.

Key to Drupal's success is its relentless attention to all things content-related. Drupal defines the essential capabilities for organizing, managing, and delivering all kinds of web-related content and for building communities. Drupal provides the flexibility and the extensibility to insure that developers and site builders can leverage an underlying platform, build out the unique capabilities for web experiences and social interactions, and rapidly assemble the dynamic environments that organizations require.

As a result, organizations can optimize their investments, and realize solutions to their business problems, at an affordable cost. With Drupal, organizations can build out their unique web experiences by leveraging the content-centric foundations of the assembled web infrastructure.

About Acquia

Acquia empowers enterprises with the open-source content-management system Drupal. Co-founded by Drupal's creator in 2007, Acquia helps customers manage their growth and scale their online properties with confidence. Acquia's software, consultation, cloud infrastructure, and services enable companies to realize the full power of Drupal while minimizing risk, as it's done for Examiner.com, Al Jazeera, and over 700 others.
See who's using Drupal at **http://showcase.acquia.com** and learn more at **http://acquia.com**.

Acquia, Inc.
25 Corporate Drive, 4th Floor
Burlington, MA 01803
USA

www.acquia.com
sales@acquia.com

+1.781.238.8600