DRUPALCON LONDON

Code & Coders

Developing with Drupal Commerce

Ryan Szrama, Richard Jones, Peter Philipp,
Jamie Wiseman, Jakob Torp

What can you *do* with it?

Ubercart offers a wide **feature set**, but operating outside of the core feature set is difficult at best.

"Application" mindset focusing on the modules' **performance out-of-the-box**.

Sacrificed API functionality and i18n for UI based customizations.

What can you *build* with it?

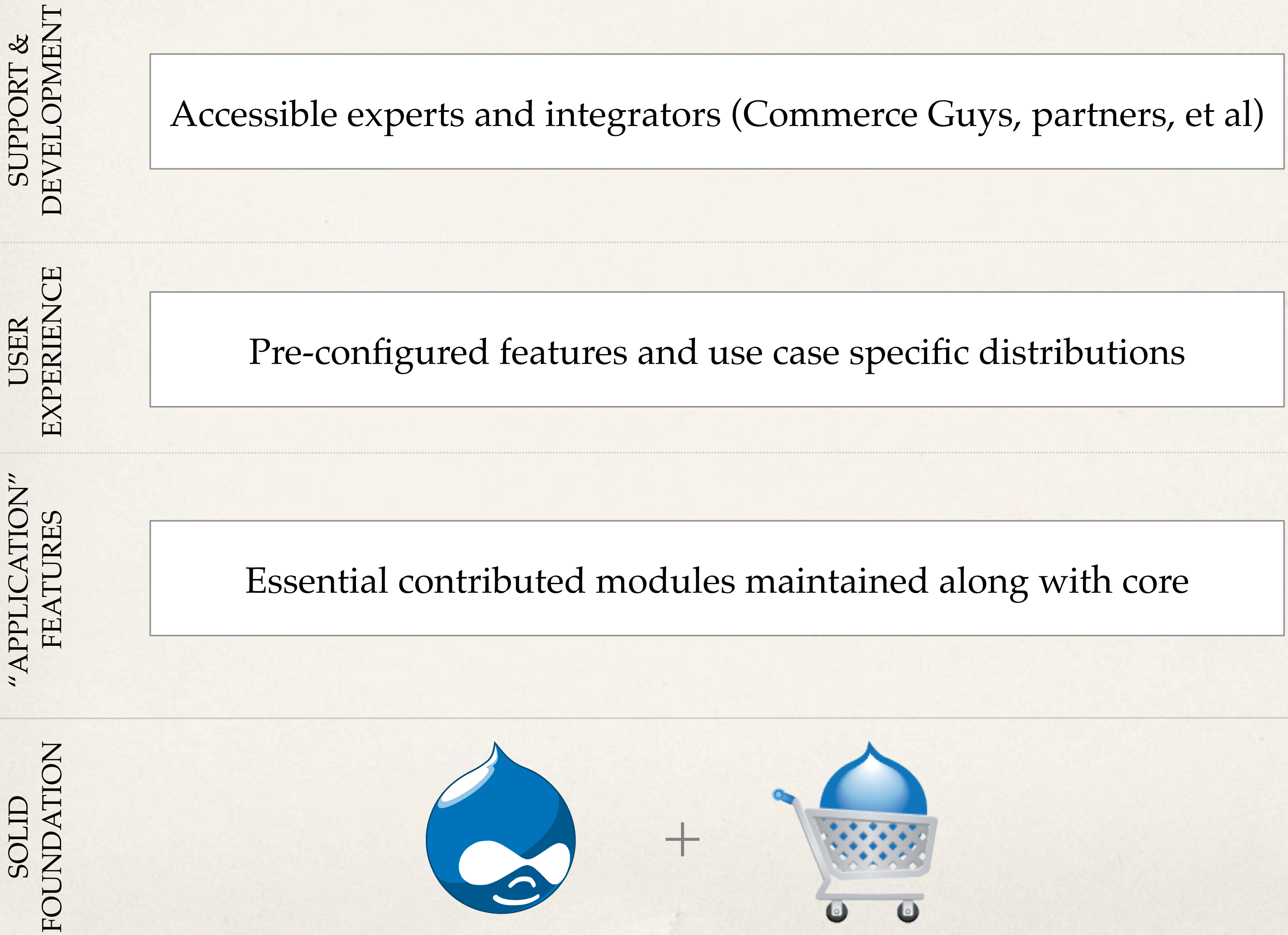
Commerce modules offer **core e-commerce systems and components**, providing basic functionality with amazing flexibility.

"Framework" mindset focusing on loosely coupled modules and **adaptability**.

Privileges developers and site builders at the core level, administrators and reusable feature development at the profile level.

“Our vision is for Drupal to be the number one e-commerce platform in the world.”

Formula for a winning Drupal Commerce ecosystem:



SUPPORT & DEVELOPMENT

Accessible experts and integrators (Commerce Guys, partners, et al)

USER EXPERIENCE

Pre-configured features and use case specific distributions

“APPLICATION” FEATURES

Essential contributed modules maintained along with core

SOLID FOUNDATION

Drupal 7 includes a “fieldable entity” system providing CCK functionality in core not just for nodes but for *any* type of data defined by your modules.

Commerce entities and fields:

1. **Product** ► **SKU, title, and price with custom attribute fields**

   Uses a *price field* for the base price; discounts, taxes, and currency conversion via Rules.
   Displayed on nodes with an Add to Cart form via the *product reference field.*

2. **Line item** ► **Any item on an order calculated into its total**

   Uses two *price fields* for the unit price and total price (unit price times quantity).

3. **Order** ► **Order ID / number, references, and order total**

   Uses a *line item and customer profile reference fields* to associate them with the order.
   Uses a *price field* to keep track of the order total based on referenced line items.

4. **Customer profile** ► **Profile / user ID, address, and custom fields**

   Uses an *address field* to collect country-specific address data.

5. **Payment transaction** ► **Created for every attempted transaction**

Drupal Commerce's core systems make as few hard-coded assumptions about business rules as possible, providing a small set of base systems meant to be built on in contrib.

Commerce systems:

1. **Product pricing** ► **Price lists, discounts, taxes, multi-currency**

   Allows for Rules based calculation of product sell prices on display and in the cart.
   Actions allow for manipulating prices, calculating taxes, and converting currency.

2. **Shopping cart** ► **Dynamic Add to Cart form, cart block / edit form**

   Add to Cart form handles one or many products on a form with dynamic page updates.
   Shopping carts are full-fledged orders from the first "Add to Cart" on.
   The cart block and edit form are built with Views for UI based customization.

3. **Checkout** ► **Drag-and-drop form builder**

   Checkout form builder provides customization for single and multi-page workflows.
   Functions as an order edit form tracking changes through status updates and order revisions.

4. **Payment** ► **Profile / user ID, address, and custom fields**

   Any number of payment methods with checkout availability determined via Rules.
   Order payment log allows tracks all payment attempts and allows for direct payment entry.

How does the core product pricing system work?

How can I customize the buying experience for non-physical product sales? Do I *need* a cart?

How should I best integrate with the payment and line item systems?

What should I keep in mind when quoting complex Drupal Commerce projects?

Where can I go for development support and how can I give back?

What did you think?

DRUPALCON
LONDON

Locate this session on the DrupalCon London website:

http://london2011.drupal.org/conference/schedule

Click the "Take the survey" link

THANK YOU!