

# **Drupal as the Head in Headless Commerce**

# Kyle Einecker

ctrlAdel

**Drupal Lead at**



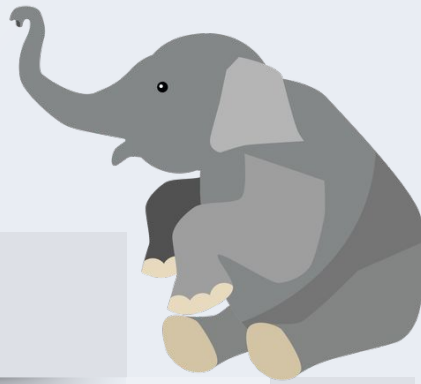
**AVIONOS**





**AVIONOS**

# Drupal as the Head



# This guy says...



and this guy says...





 Centarro

 salesforce



commercetools

 Magento®  
An Adobe Company

 *shopify*

 elasticpath

SAP Hybris 

 OROCommerce™

 BIGCOMMERCE

 WOO COMMERCE



# Practical Reasons

## Integrations

Payment gateways

Shipping providers

ERP

PIM

Taxes

Marketing automation

## Backend Functionality

Customer Service

Returns

Invoicing

User management

Account management

Promotions

# Assumption

**The commerce  
platform is not Drupal  
Commerce**

# Assumption

**The commerce API  
supports headless use**

# Assumption

**CMS capabilities of the  
commerce platform  
are insufficient**

# Why Drupal?

Drupal is a full featured CMS

Built in commerce framework with a flexible data model,Drupal Commerce

# Assumption

**Drupal is the CMS**



# **Architecture Considerations**

# Authentication

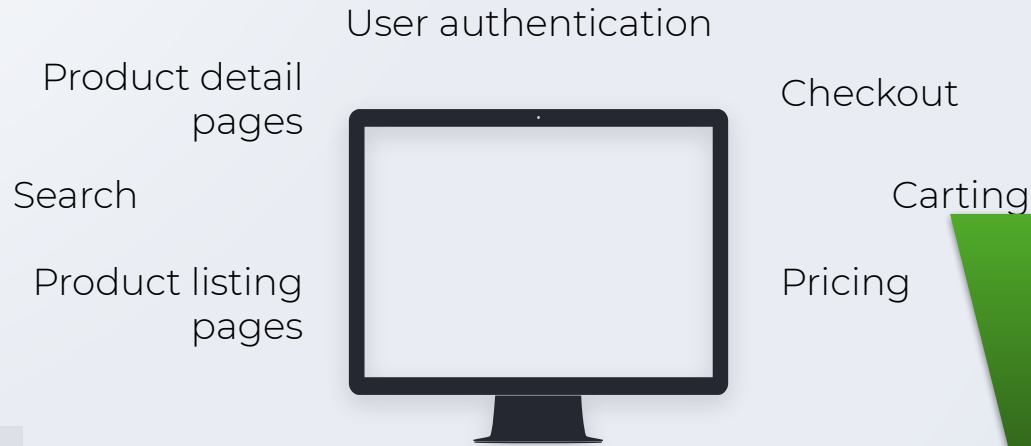
Integration user

VS

Customer authentication



# Disconnected Behavior



## Implementation Tip #1

Write a health checker and notification.

# Commerce API Usage & Flexibility

Is it fully featured?

What are the api usage limits?

Does it support querying?

Will the business logic fit within existing endpoints?

Does it support custom endpoints?

Is there a php sdk or library available?

Webhooks?

# Data Ownership

What system owns what data?

- Users
- Orders
- Carts
- Products
- Product Attributes
- Inventory
- Accounts
- Content
- Catalog Structure

## Implementation Tip #2

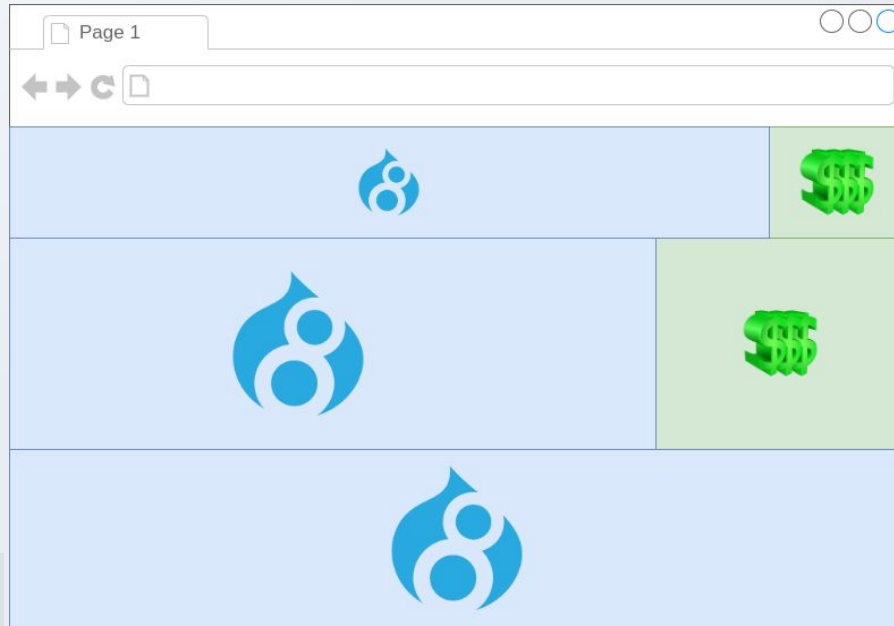
Respect data  
ownership.

## Implementation Tip #3

Make friends  
with all the  
teams.

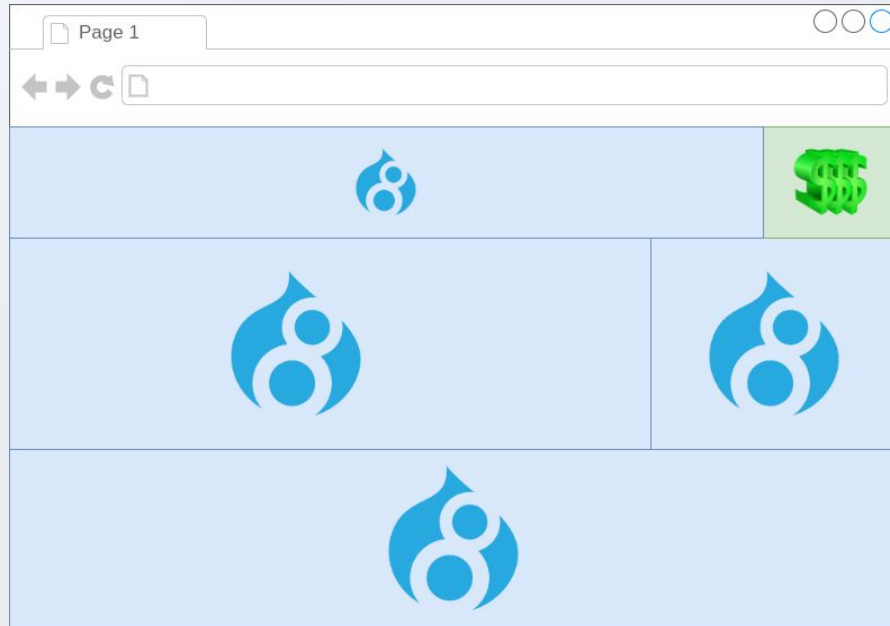
# Architecture #1

## Fully Headless



# Architecture #2

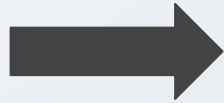
## Partially Headless



# Architecture #3

## Commerce Integration

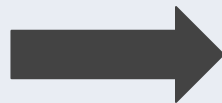
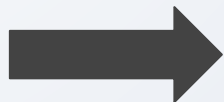
Commerce  
Platform



# Architecture #4

## Fully Headless Commerce Integration

Commerce  
Platform



# **Assumption**

**We've chosen  
Architecture #3  
the Commerce  
Integration**





**Integration**

**Considerations**

How often does data change?

Why do you need the data?

When do you need the data?

Who owns the data?

**Useful for:**

Products  
Accounts

**Approach:**

Push or Pull  
Message queue

# Import

(1 way sync)

**Implementation  
Tip #4**

Don't fail  
silently, log  
everything.



# Import & Export

(2 way sync)

**Useful for:**

Coupons  
Users

**Approach:**

Storage override  
Subscribers

# Lazy Load

**Implementation  
Tip #5**

Make sure you  
can debug on  
production.

**Useful for:**

Order Confirmation  
Order History  
Shipping Methods  
Inventory

**Approach:**

Custom blocks  
Subscribers  
Plugins

# On Demand

**Implementation  
Tip #6**

If it's loaded  
Lazily or On  
Demand you  
should be able  
to delete it

**Useful for:**

User authentication  
Checkout  
Cart  
Taxes  
Pricing

**Approach:**

Custom blocks  
Subscribers  
Resolvers

# Transactional

**Useful for:**

Profile data  
Cart  
Address book  
Payment methods

**Approach:**

Subscribers  
Entity hooks

# Synchronization



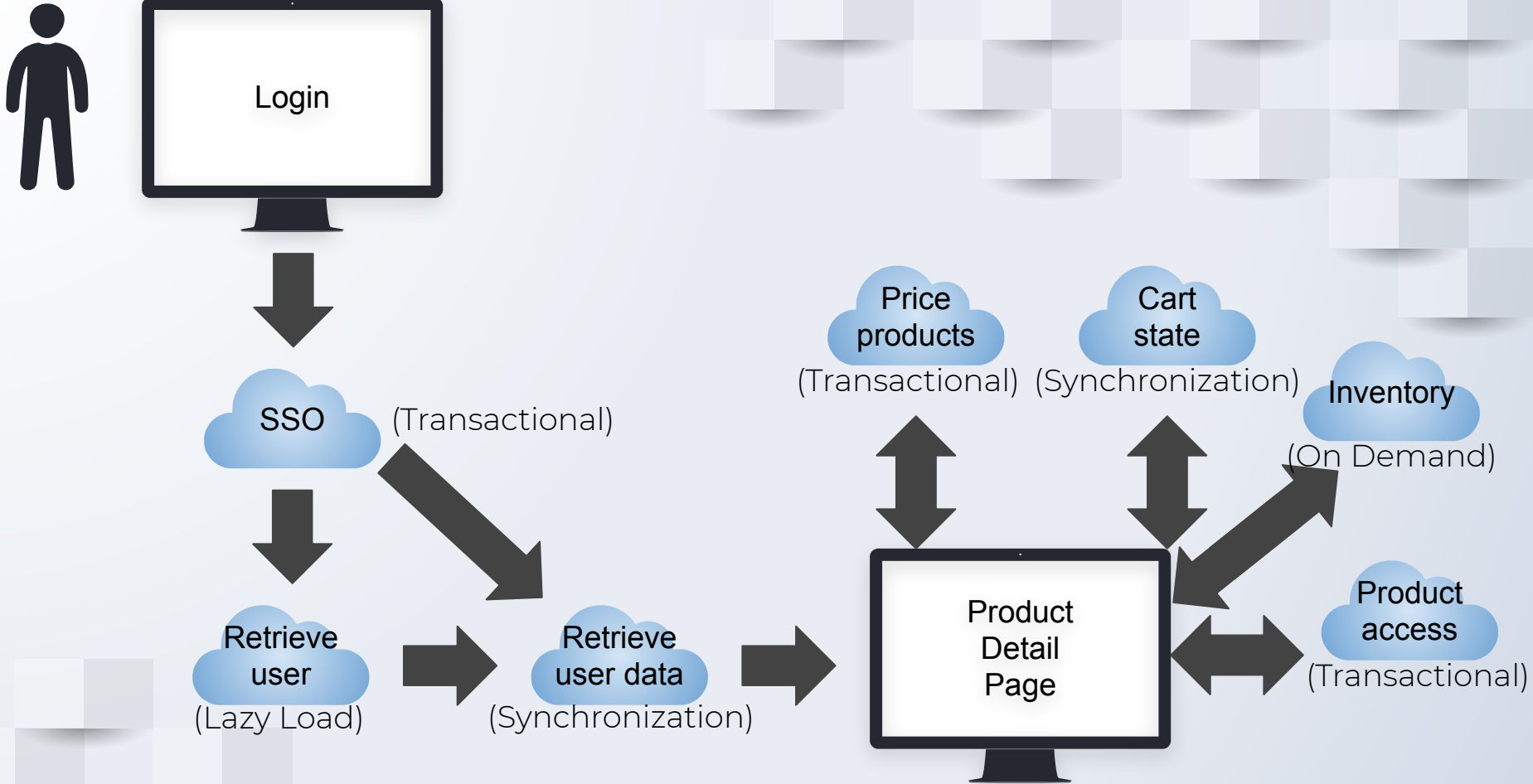
**Useful for:**

Address Book  
Payment Methods

**Approach:**

Storage override

# External Entities





# Questions?

# PLEASE PROVIDE YOUR FEEDBACK!

[mid.camp/6406](https://mid.camp/6406)



# CONTRIBUTION DAY

## Saturday 10am to 4pm

You don't have to know code to give back!

New Contributor training 10am to Noon  
with **AmyJune Hineline** of Kanopi Studios