

Magento 2 Overview



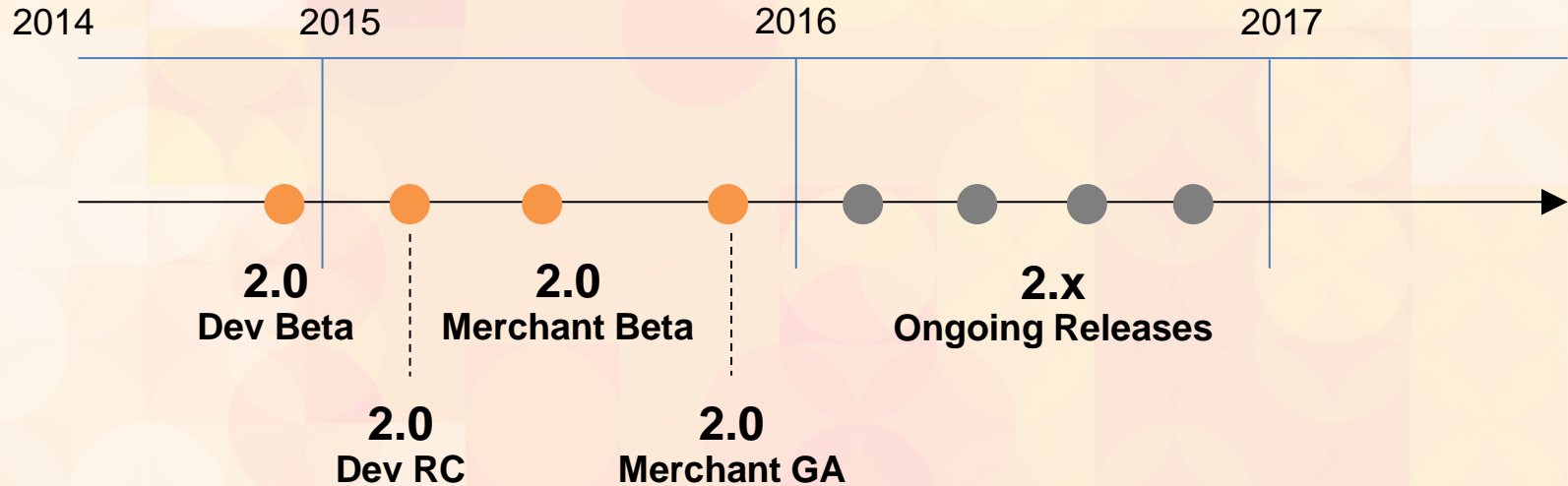
Paul Boisvert

Director Product Management,
Magento

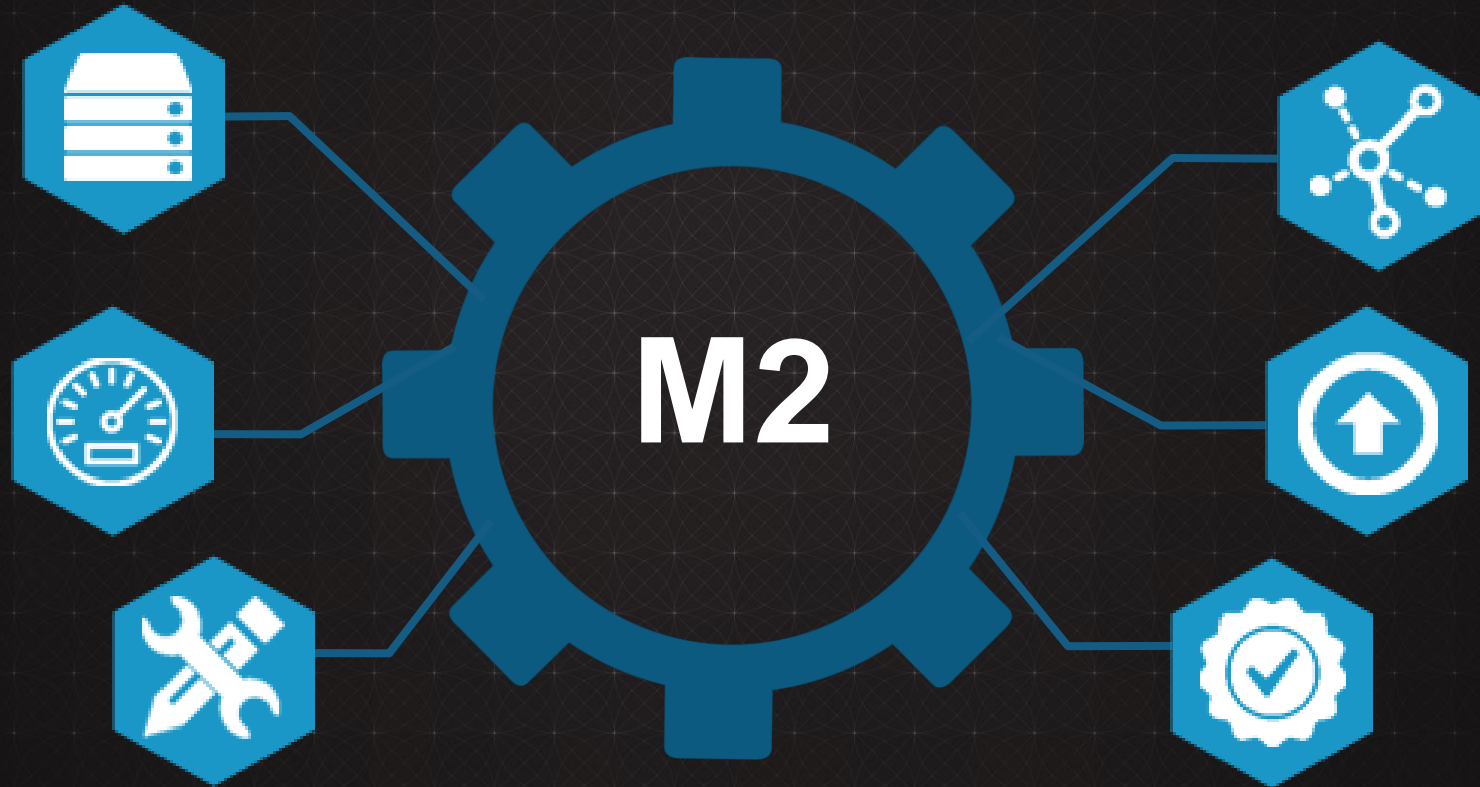


| Platform Goals

Release Approach



MAGENTO 2: PLATFORM GOALS





Modern tech stack



Improved performance & scalability



Streamline customizations



Simplify external integrations



Easier installation & upgrades



High code quality & testing



Nikolay Babiy

Development Manager, Magento

Magento 2 Overview



magento/magento2



Modern tech stack

PHP



jQuery



HTML5 CSS3 Preprocessor



RequireJS



Coding Standards

- **PSR-0** (Autoloading standard)
- **PSR-1** (Basic Coding standard)
- **PSR-2** (Coding Style Guide)

www.php-fig.org

PHP Framework Interop Group

Tech Stack





Improved performance & scalability

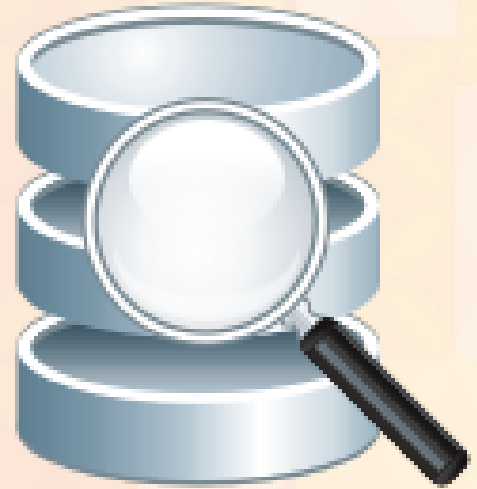
Page Caching

- Full Page Caching out of box in CE
- Cache Applications:
 - Built-in Application: Filesystem/Redis
 - Varnish
- Private/public content
- Invalidation logic



Indexing

- Refactored: Catalog, Price, EAV, Stock, URL Rewrite, Target Rule, Search Indexers
- Partial indexing
- Work in background (do not block processes)
- No manual re-index required
- Materialized View (Mview)





Streamline customizations

DI

- Object Manager
(Dependency Injection Container)
- Configurable di.xml
- Compiler



Mage Class

God Class “Mage::” is eliminated












```
Mage::getModel('Magento_Customer_Model_Address')
```

Interceptors & Plugins

- Before method
- After method
- Around method
- Sorting

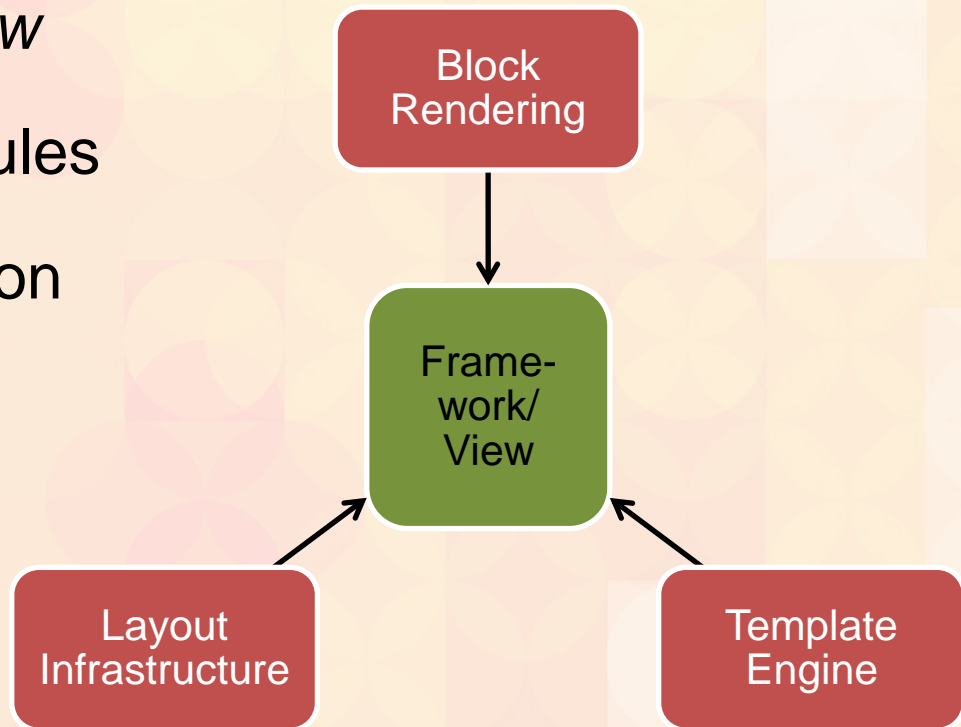


Magento Framework

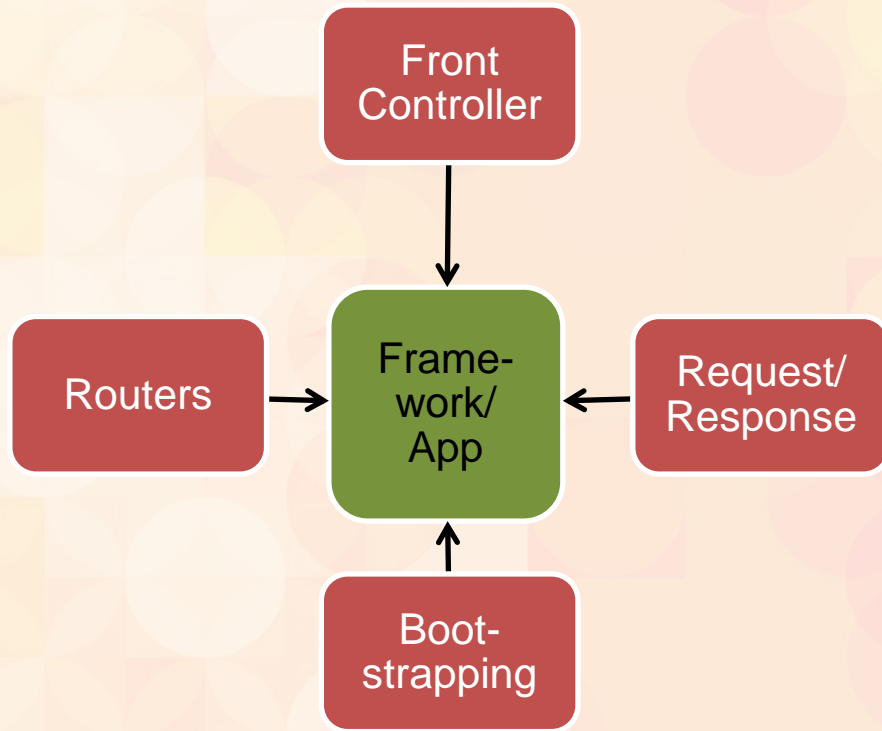
- ▼  Magento
 - ▼  Framework
 - ▶  Acl
 - ▶  App
 - ▶  Archive
 - ▶  Authorization
 - ▶  Autoload
 - ▶  Backup
 - ▶  Cache
 - ▶  Code
 - ▶  Config

Modularity: View

- Library Component *View*
- Independent from Modules
- Resolved dependency on Store
- Abstract blocks & widgets in Library



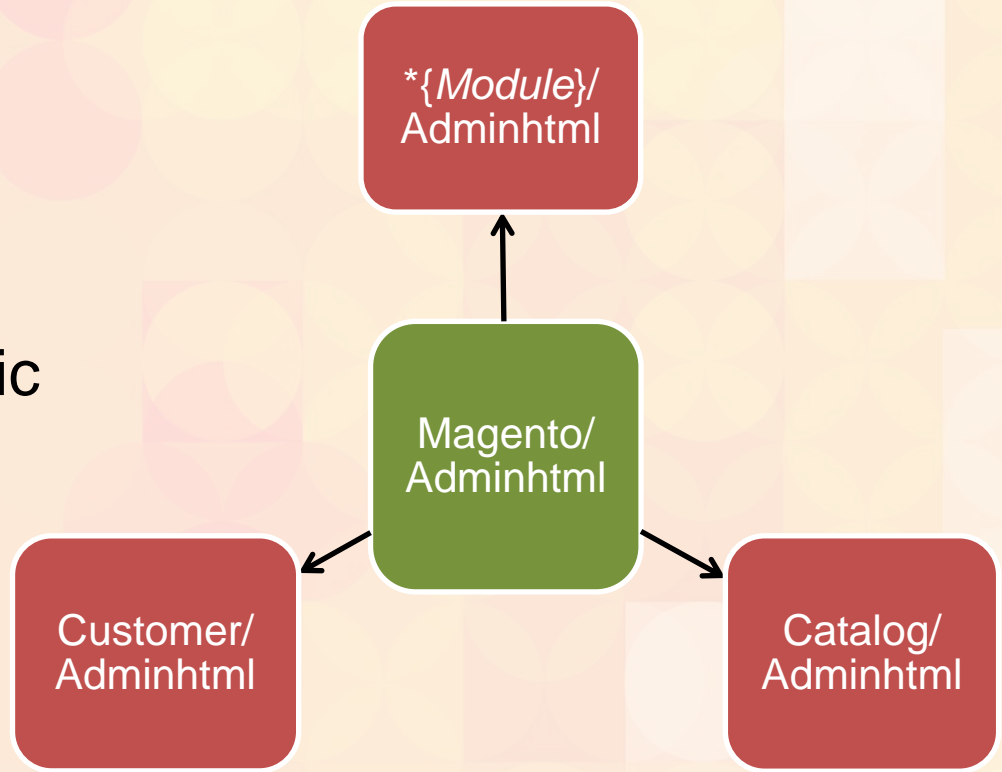
Modularity: App



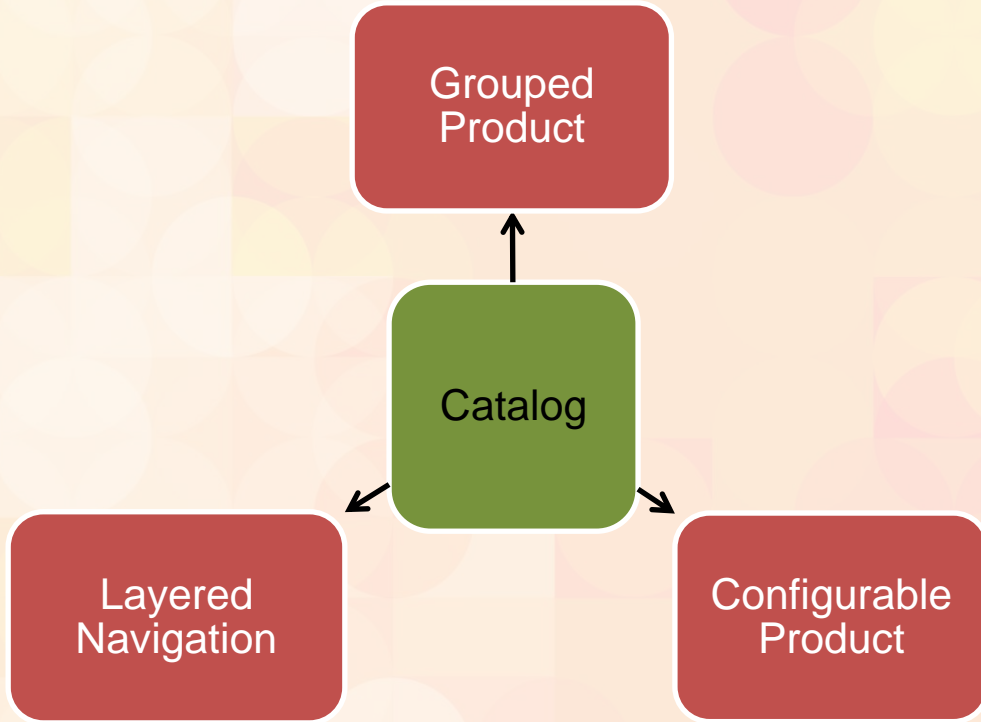
- Library Component *App*
- Resolved global dependency from Core modules

Modularity: Admin

- Global *Adminhtml* module is eliminated
- Modules are monatomic



Modularity: Catalog



- New Modules
- Enable/Disable
- Delete from code

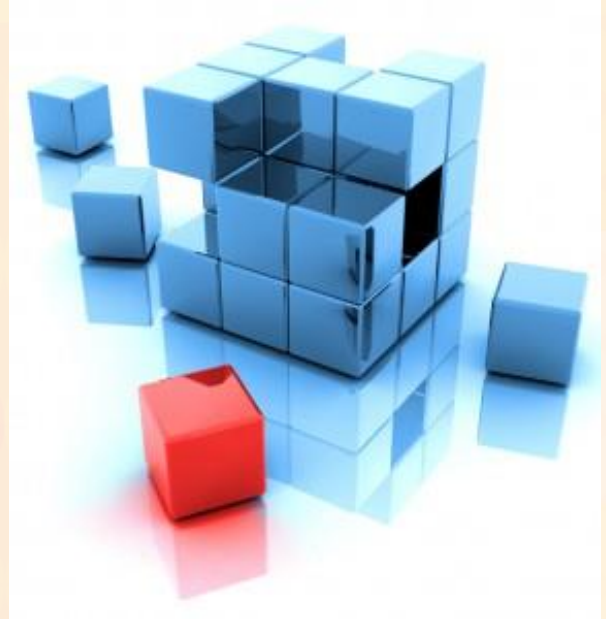
Modularity: Search



- Enable/Disable
- Delete from code
- No search logic in other modules
- Ability to change search engine to any other 3rd party

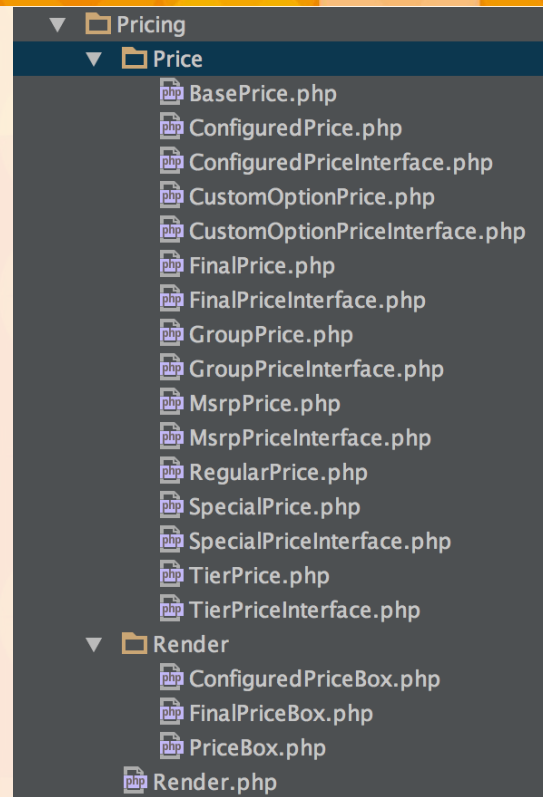
Modularity

- Sales: extracted Recurring Payment
- Shipping: Carrier, decoupled shipping methods
- Checkout – extracted Terms & Conditions and Multishipping
- Payments – OfflinePayment, decoupling payment methods



Price Logic

- Price logic decomposition per price type
- Price Interfaces
- Price decoupled from Tax and Weee
- Price “adjustment”



i18n

- Global function __()
- Phrase
- RendererInterface
- Composite, Inline, Placeholder, Translate
- [Module]/i18n

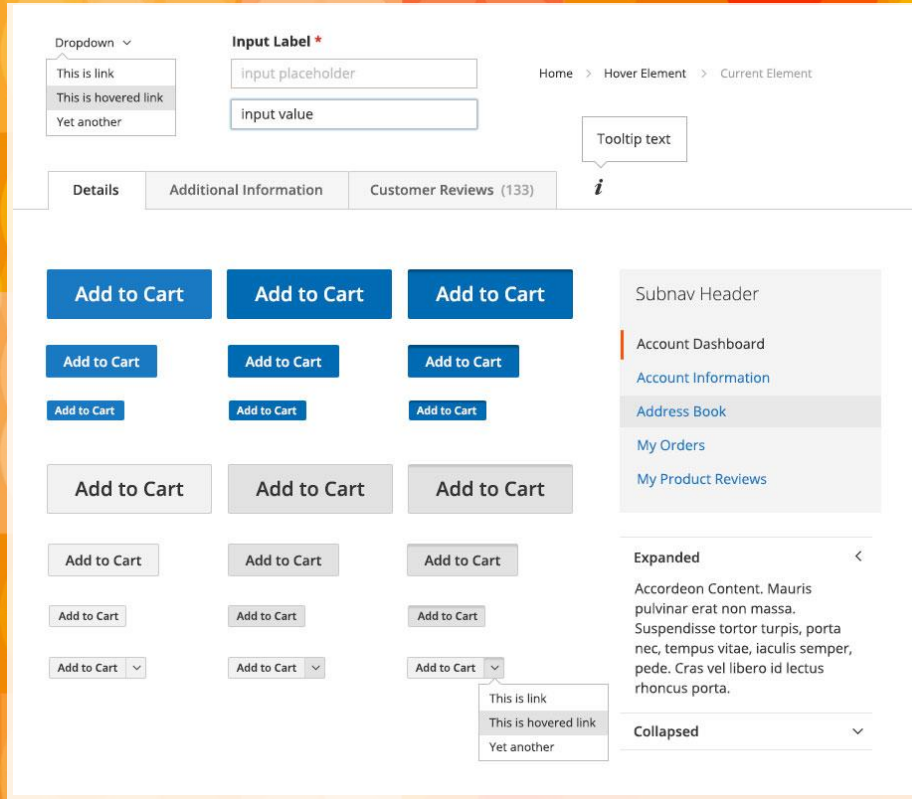


Layouts

- Layouts, templates (view) in Modules
- Layout file responsible only for single page/block
- Validation Layout XML (XML Schema)
- Extend Layout
- Merging Layouts
- Override Layout
- Page Types



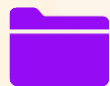
Magento UI Library



- Upgradability
- Consistency
- Extensive list of reusable components
- Styling and design best practices
- Good documentation
- Simplifies customization



Magento UI Documentation



pub/lib/css/docs

Primary button

The `.button-primary()` mixin is used to create a **primary button**. By default it uses Primary button variables from variables.less file.

Primary button

Button

Hover

Active

Disabled

```
<button class="button" type="button"><span>Button</span></button>
<button class="button :hover" type="button"><span>Hover</span></button>
<button class="button :active" type="button"><span>Active</span></button>
<button class="button disabled" type="button"><span>Disabled</span></button>
```

```
.button {
  .button-primary();
  &:active {
    box-shadow: inset 0 3px 1px rgba(0,0,0,.29);
  }
}
```

Blank Theme



Modern Technologies



Mobile First Responsive Design



Built with Magento UI library

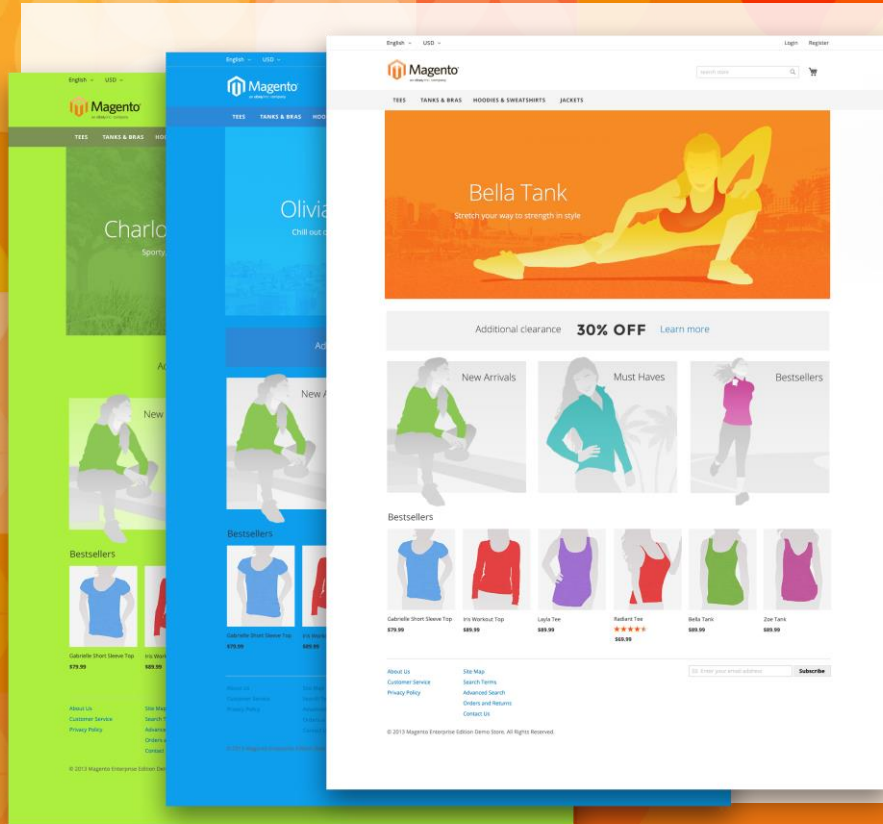


Compiled with Built-in PHP
LESS Compiler



WCAG 2.0 AA Compliant

Multilevel Theme Inheritance



- Simplifies customization
- Flexible store design
- Allows store to be special
- Maintainability
- Less code duplication



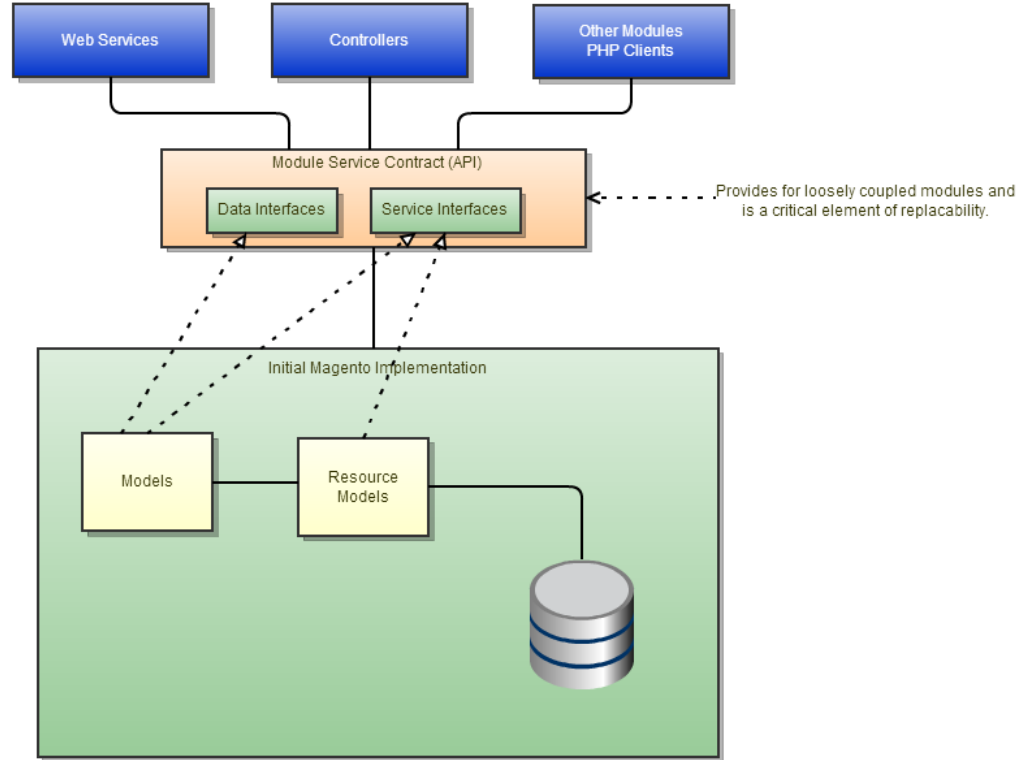
Simplify external integrations

MSC (Module Service Contracts)

Module contract that can be implemented by current Magento models – abstraction of business operations and entities for a module to produce a durable API for client to rely on

- Data Interfaces
- Service Interfaces
- Repository Interface

MSC (Module Service Contracts)





Easier installation & upgrades

Composer and Deployment Tool

- Deployment Tool
- **Composer**



`magento/magento-composer-installer`



High code quality & testing

- Full set of tests: Unit, Integration, Static, Legacy
- Functional Test – **Magento Testing Framework (MTF)**
- Continuous Integration & Continuous Delivery
- Performance Testing
- Regression Testing



magento/mtf





| **Magento 2 and Team**

Magento 2 Domains

Commerce
Services

Merchant
Experience

Platform
Services

Functional Areas

Payments

Shipping

OMS

Customer

Catalog

Promotions

CMS

Admin-Site

API

Lib

Performance
& Tools

Mobile

MTF

WebDev

Teams

WebAPI

QM

Tango

Ogres

WebDev

Firedrakes

Mustang

Trolls

Vanilla

South

Golnc

Dragons

Extensibility

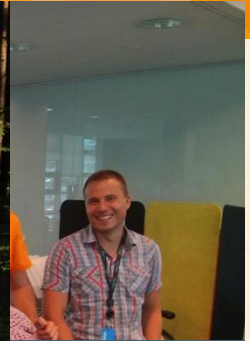
Nord

MPI

Fearless Kiwis

Folks

Teams





magento/magento2

Contacts

 nbabiy@ebay.com

 pboisvert@ebay.com



Thank you