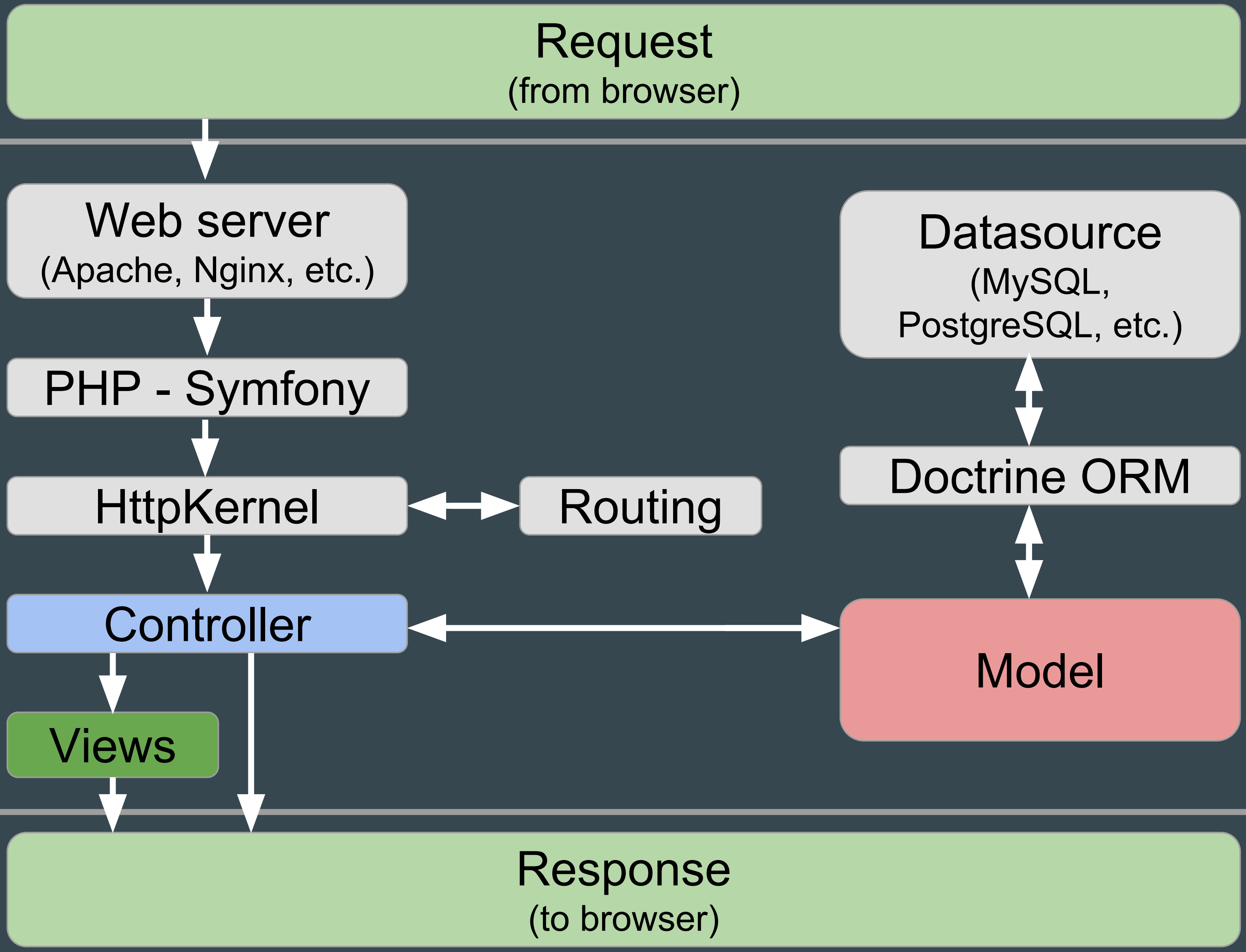
High Level Project Architecture

Request-Response Flow

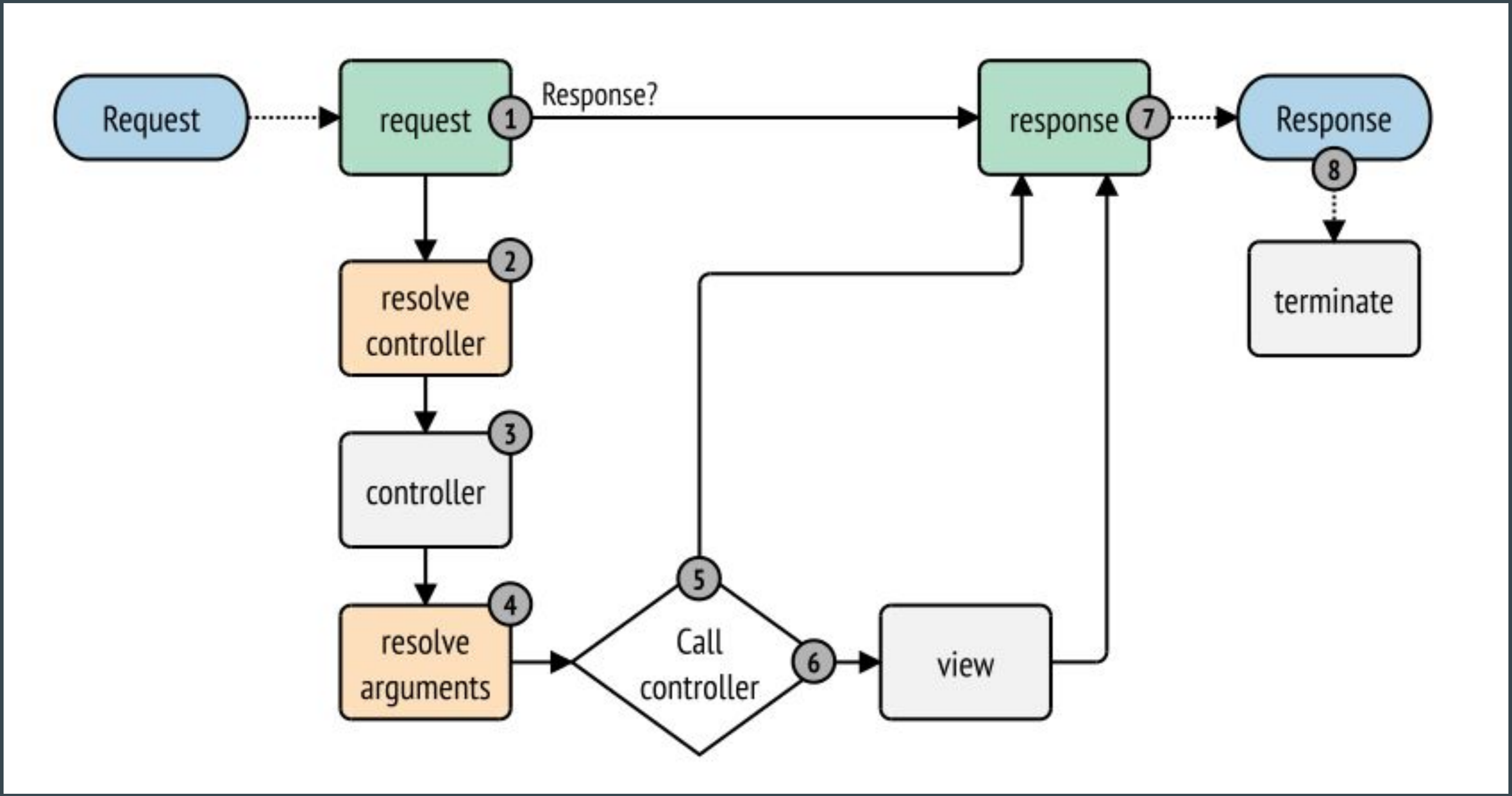

Request
(from browser)
Web server
(Apache, Nginx, etc.)
PHP - Symfony
HttpKernel
Routing
Controller
Views
Datasource
(MySQL,
PostgreSQL, etc.)
Doctrine ORM
Model
Response
(to browser)

Kernel Events

- kernel.request
- kernel.controller
- kernel.controller_arguments
- kernel.view
- kernel.response
- kernel.terminate
- kernel.exception
- kernel.finish_request

Dispatched by HttpKernel

Kernel Events



Bundle

Has two main responsibilities:

- Store structured set of files related to a single feature
- load extensions and configs for the Service Container (example)

Also it allows to hook to bundle boot and shutdown

- BundleInterface

Bundles Structure

- Resources/...
- DependencyInjection/...
- All other stuff ...

Best Practices

ProductBundle

Bundles Loading priorities

- Usually doesn’t matter
  - Because you control how they loaded
- Important for ORO
  - Because of autoloading
- /var/cache/prod/bundles.php
- Symfony toolbar tab

Oro Foundation

- Symfony Framework
  - No significant issues with adding symfony-specific components to ORO applications
- Doctrine ORM
  - Pay attention on the performance
- Some best practices aren’t used
  - Eg. bundles.yml / routing.yml

https://symfony.com/projects/orocrm
https://symfony.com/projects/orocommerce

Oro applications

- OroCRM (Community and Enterprise)
- OroCommerce (Community and Enterprise)
- Akeneo PIM (external)
- Diamante Desk (external)
- Marello (external)
- All of them can* be used together, eg. crm+commerce
- Features and differences between CE and EE
  - Data volume, scalability, performance, support and more

Application environments

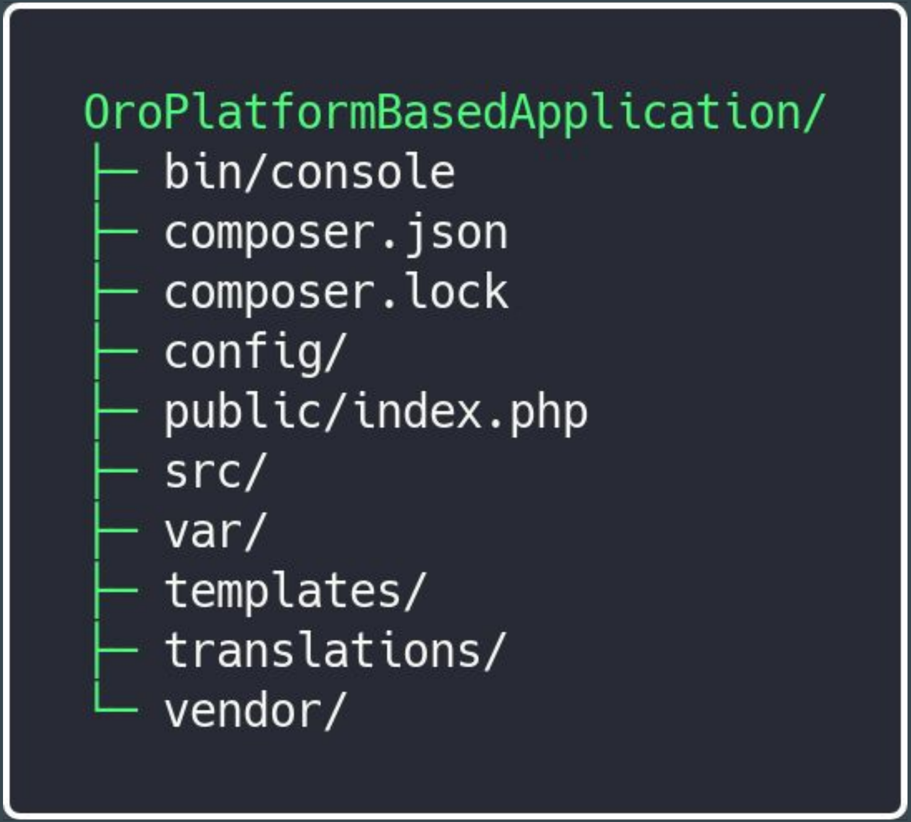
- dev
  - Symfony toolbar + ORO extensions
  - The slowest one
  - Good for debugging, mail catching etc.
  - System configuration
- prod
  - Performance optimized
  - Recommended for demo
- test
  - No demo data!

Demonstration >>>

Application environments

- Best practices from ORO team
  - Dev env for development
  - Use prod for demo for the clients
  - Check features in prod env before merging
  - Sometimes is worth to do clean install
  - In prod JS/assets are minified

Application structure



```
OroPlatformBasedApplication/
├─ bin/console
├─ composer.json
├─ composer.lock
├─ config/
├─ public/index.php
├─ src/
├─ var/
├─ templates/
├─ translations/
└─ vendor/
```

platform-application

Dependencies of OroCRM application

- https://github.com/oroinc/crm-application/blob/3.0.0/composer.json#L21-L28
- https://github.com/oroinc/crm/blob/3.0.0/composer.json#L17-L22
- https://github.com/oroinc/platform/blob/3.0.0/composer.json#L13-L117

Application configuration: Files

- Symfony
- config/config.yml ; parameters.yml
- config/security.yml
- config/routing.yml

- ORO
- <bundle>/Resources/config/oro/app.yml
- <bundle>/Resources/config/oro/routing.yml
- and others...

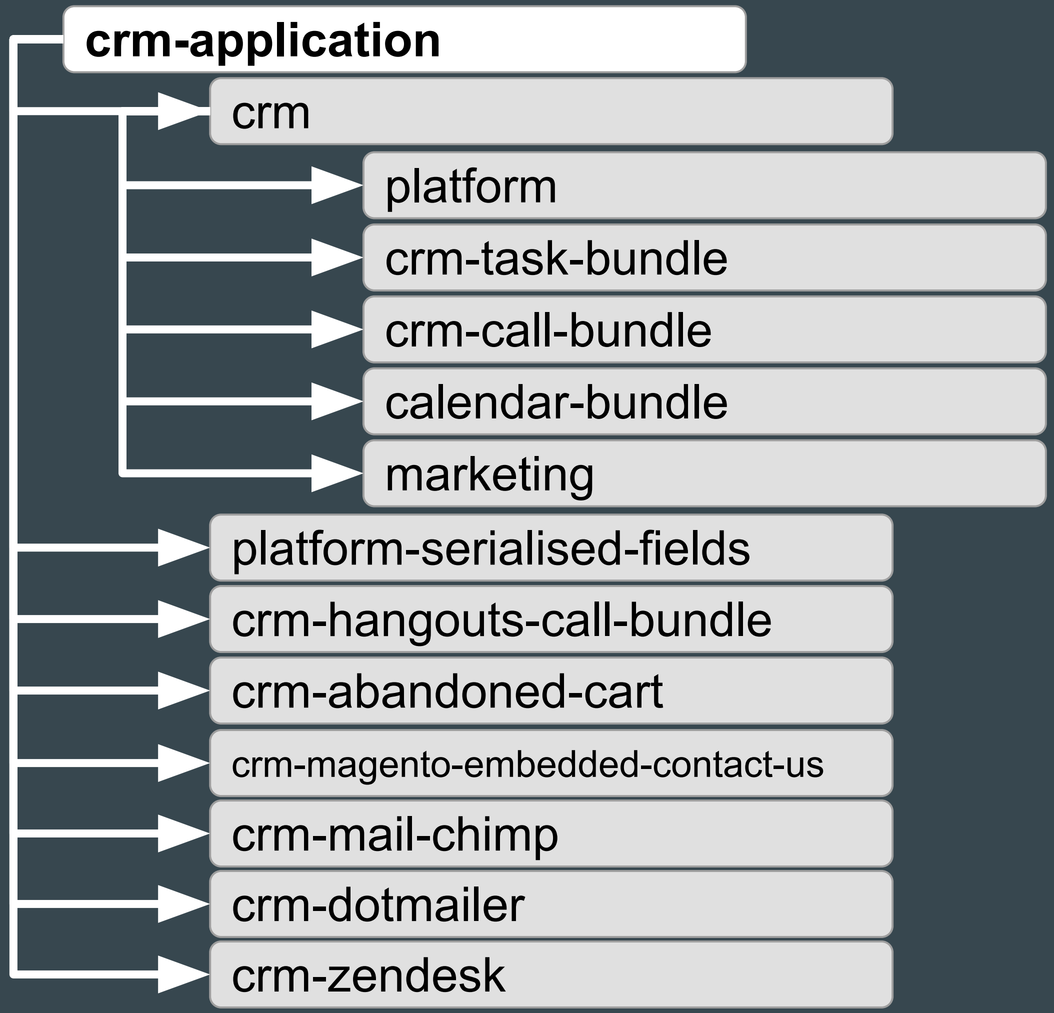
https://github.com/oroinc/platform/tree/master/src/Oro/Bundle/PlatformBundle
https://github.com/oroinc/platform/tree/master/src/Oro/Bundle/DistributionBundle

Packages and bundles

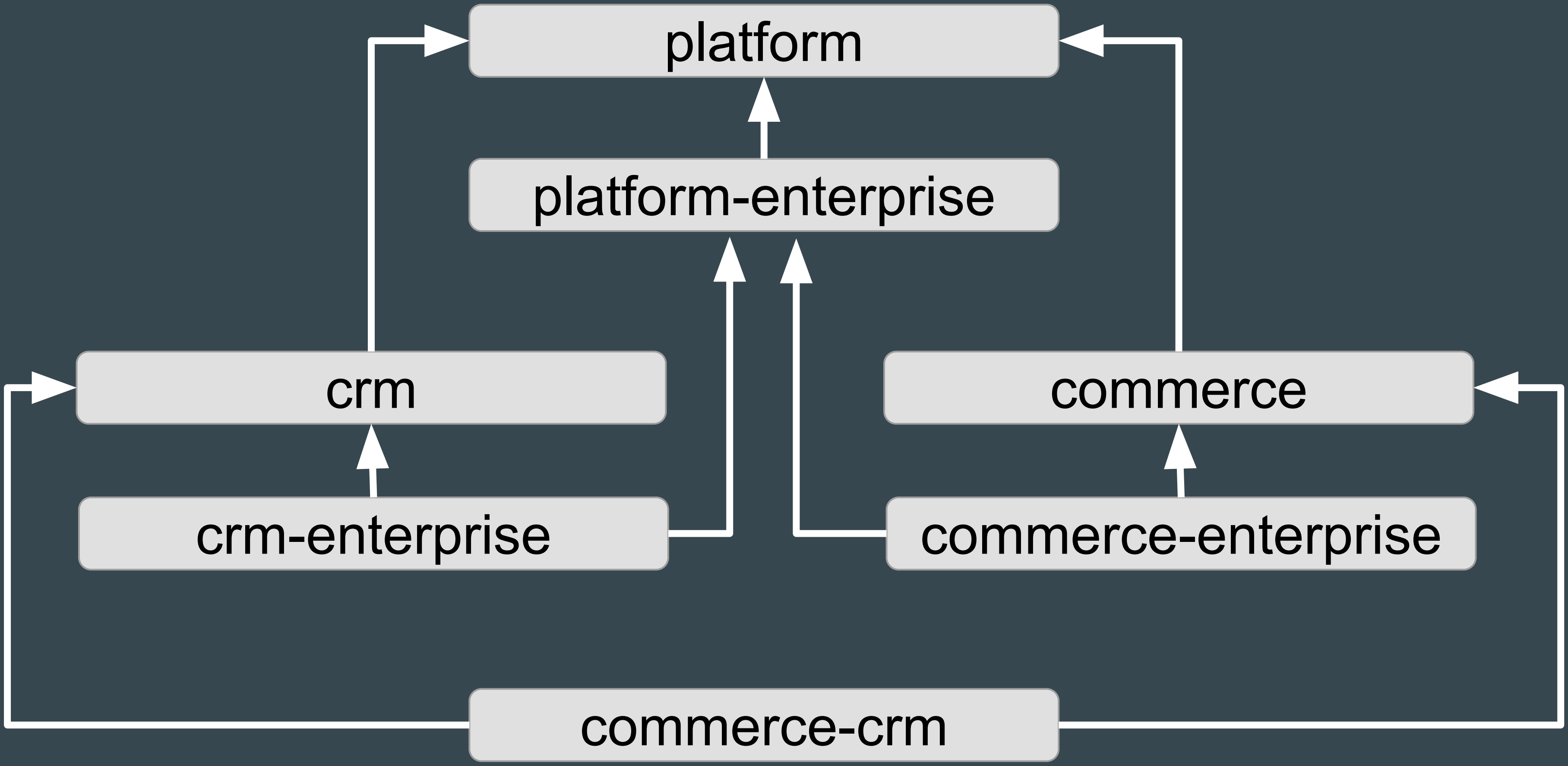
- Bundle contains finished code
- Package contains one or more bundles
  - eg. marketing
- No package manager for bundles

Demonstration >>>

Packages in OroCRM application



Main package dependencies



Additional packages

- Plugins
- Integrations
- Additional functionality
- Marketplace
  - https://marketplace.orocommerce.com/
  - How to Manage Extensions