

Meet Magento  

MM21IT

June 10/11 2021 - Virtual Conference

Principles & Advantages of Hexagonal Architecture on Magento



Alessandro Ronchi

COO @ Bitbull

Testing is hard

it is even harder on Magento

There must be a way

to simplify testing

Hexagonal architecture

to the rescue

Advantages

- ✓ **Decoupling**
- ✓ **Focus on the domain**
- ✓ **Focus on testability**

About me

- ✓ **Team mentor at Bitbull**
- ✓ **Passionate about Software Design**
- ✓ **Magento Master & Maintainer**



@aleron75

Hexagonal architecture

a different way of layering components

Decoupling

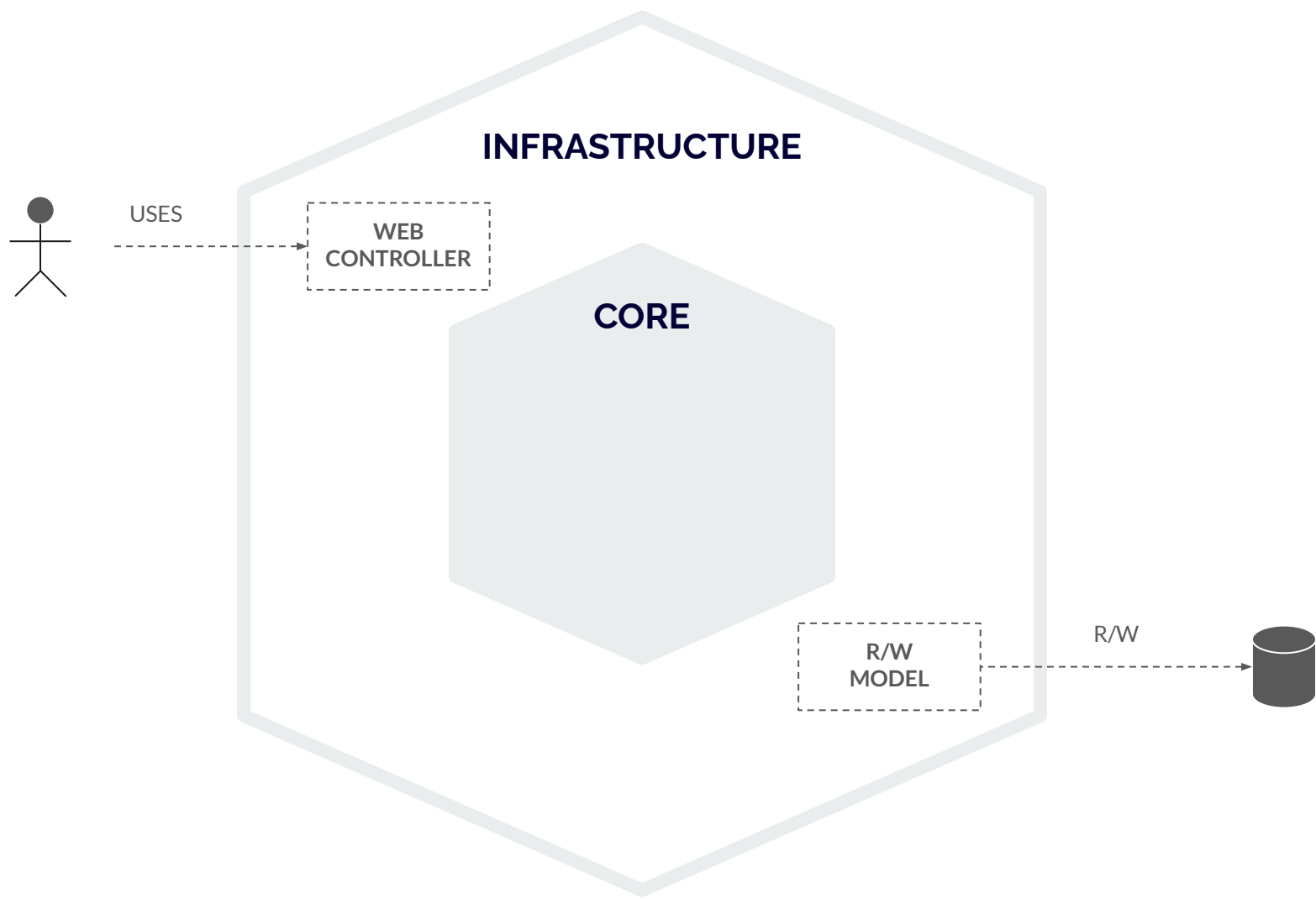
Core vs Infrastructure

CORE CODE

DOES NOT DEPEND ON
EXTERNAL SYSTEMS

INFRASTRUCTURE CODE

DEPENDS ON
EXTERNAL SYSTEMS



CORE CODE

DEFINES & DEPENDS ON
ABSTRACTIONS

INFRASTRUCTURE CODE

IMPLEMENTS
ABSTRACTIONS

CORE CODE

DEFINES & DEPENDS ON
ABSTRACTIONS

<REPOSITORY>

INFRASTRUCTURE CODE

IMPLEMENTS
ABSTRACTIONS

MYSQL REPOSITORY

IMPLEMENTED BY

CORE CODE

DOES NOT DEPEND ON
CONTEXT



USED BY

INFRASTRUCTURE CODE

DEPENDS ON
CONTEXT



CORE CODE

DOES NOT DEPEND ON
CONTEXT

SERVICE

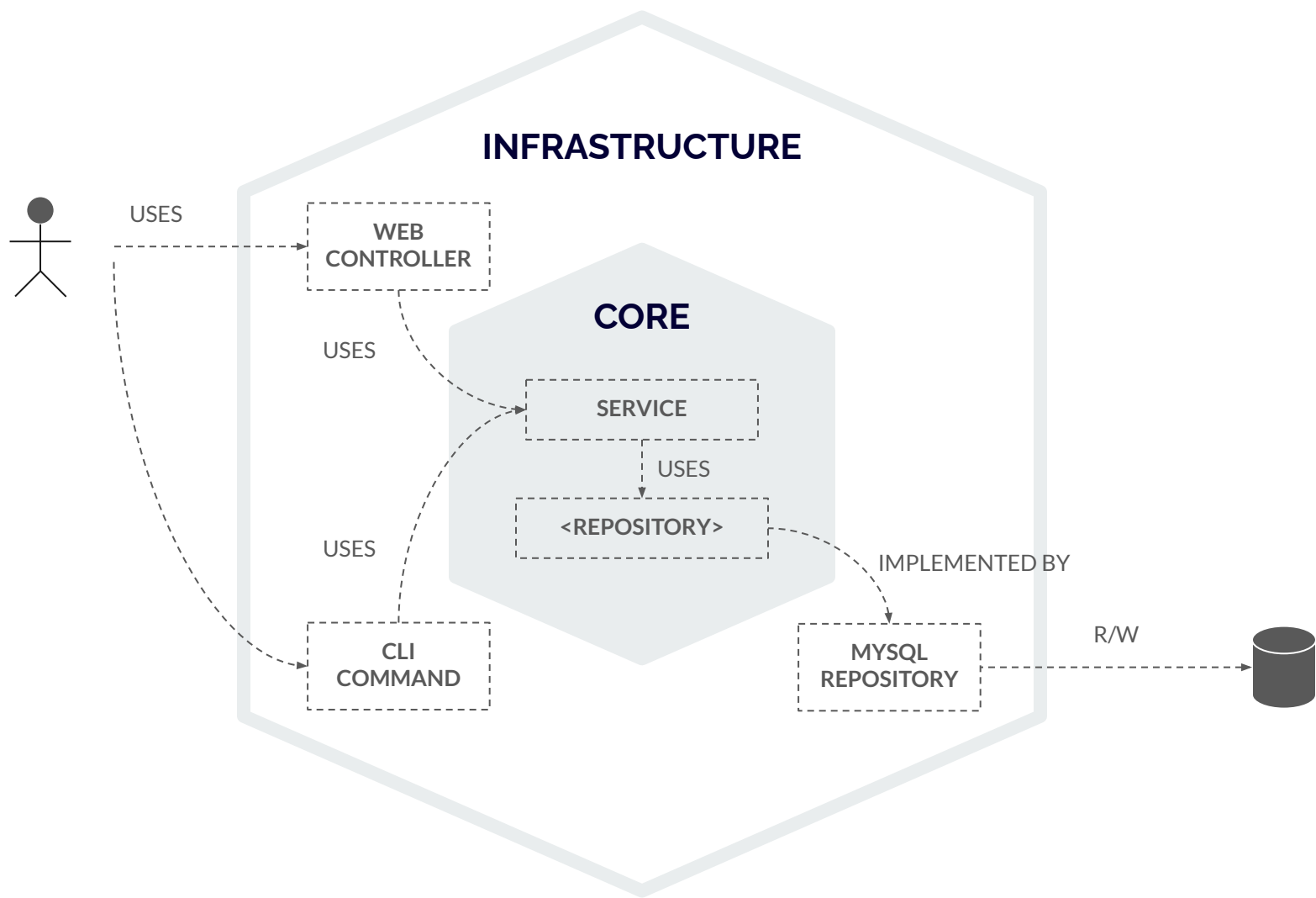
INFRASTRUCTURE CODE

DEPENDS ON
CONTEXT

CLI COMMAND

USED BY



```
graph LR; SERVICE[SERVICE] -.->|USED BY| CLI_COMMAND[CLI COMMAND]
```





Search entire store here... 



[What's New](#) [Women](#)  [Men](#)  [Gear](#)  [Training](#)  [Sale](#)

[My Account](#)
[My Orders](#)
[My Downloadable Products](#)
[My Wish List](#)

[Address Book](#)
[Account Information](#)
[Stored Payment Methods](#)

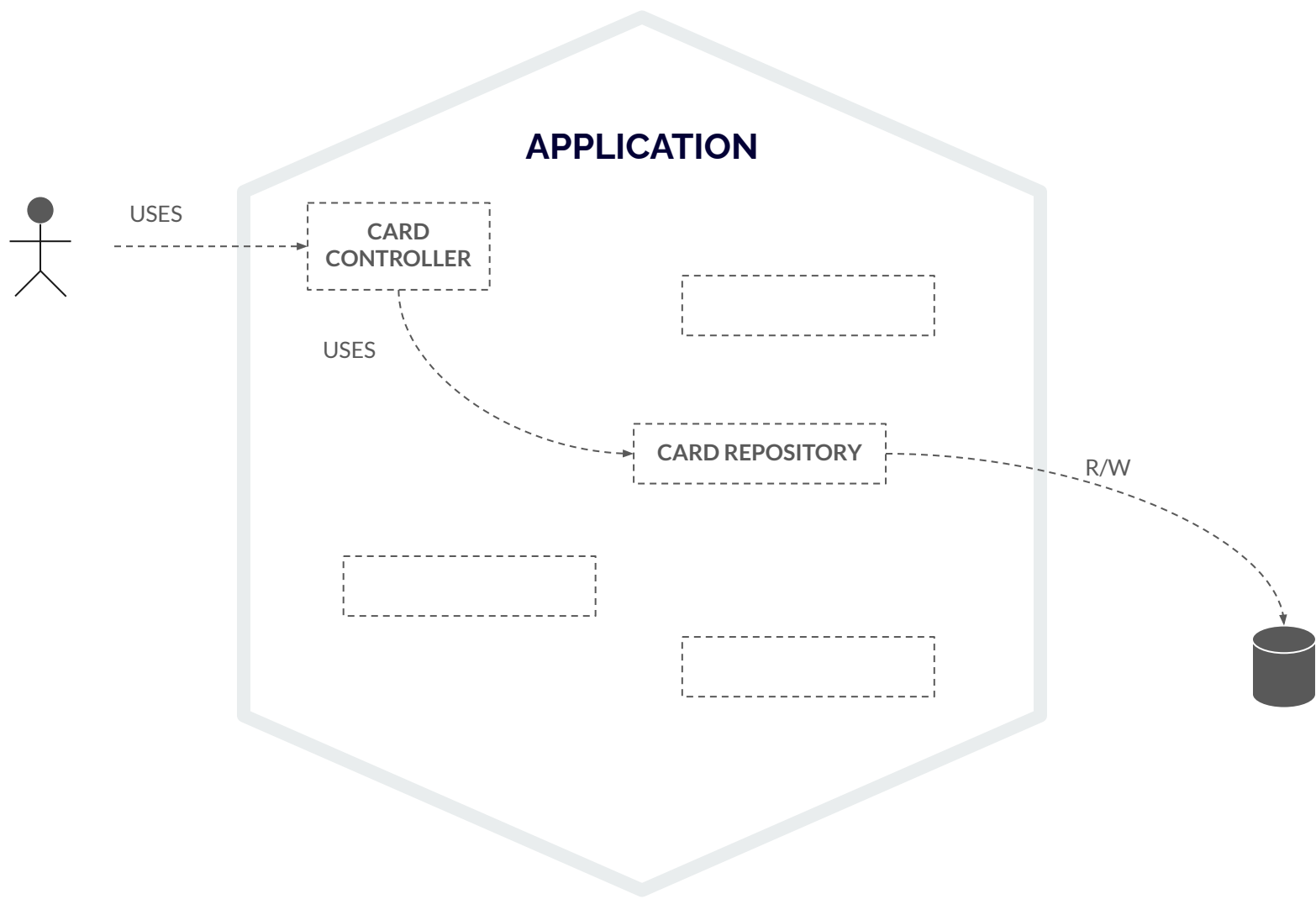
Loyalty Card

[My Product Reviews](#)
[Newsletter Subscriptions](#)

Loyalty Card

Your card number

Save



```
class Save extends Action implements HttpPostActionInterface
{
    // ...
    public function execute()
    {
        $redirectResult =
            $this->resultFactory->create(ResultFactory::TYPE_REDIRECT);

        $cardNumber = $this->getRequest()->getParam('card_number');
        $customerId = (int)$this->currentCustomer->getCustomerId();
        // validation and error handling...
        $card = $this->cardFactory->create();
        $card->setData([
            'card_number' => $cardNumber,
            'customer_id' => $customerId,
        ]);
        $this->cardRepository->save($card);

        return $redirectResult->setPath('loyalty/card/index');
    }
}
```

```
class Save extends Action implements HttpPostActionInterface
{
    // ...
    public function execute()
    {
        $redirectResult =
            $this->resultFactory->create(ResultFactory::TYPE_REDIRECT);

        $cardNumber = $this->getRequest()->getParam('card_number');
        $customerId = (int)$this->currentCustomer->getCustomerId();
        // validation and error handling...
        $command = new RegisterCardCommand($customerId, $cardNumber);
        $this->application->registerCard($command);

        return $redirectResult->setPath('loyalty/card/index');
    }
}
```

```
class Save extends Action implements HttpPostActionInterface
{
    // ...
    public function execute() {
        // ...
        $this->cardRepository->save($card); // framework-coupled
        // ...
    }
}
```

VS

```
class Save extends Action implements HttpPostActionInterface
{
    // ...
    public function execute() {
        // ...
        $this->application->registerCard($command); // framework-agnostic
        // ...
    }
}
```

CORE CODE

- FRAMEWORK: AGNOSTIC
- CHANGES: MANAGED

INFRASTRUCTURE CODE

- FRAMEWORK: BASED ON IT
- CHANGES: UNPREDICTABLE

Hexagonal architecture

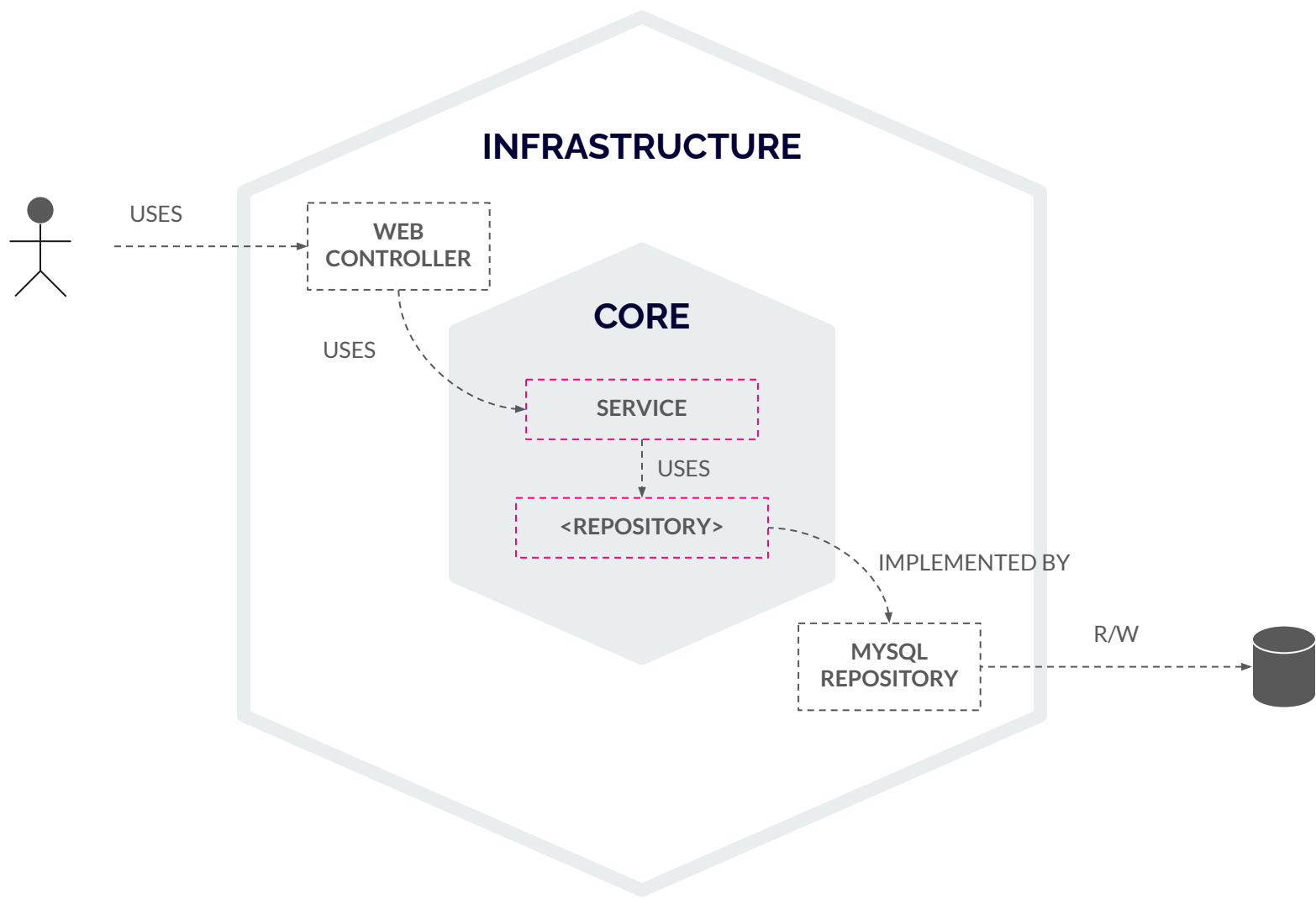
a better name: Ports & Adapters

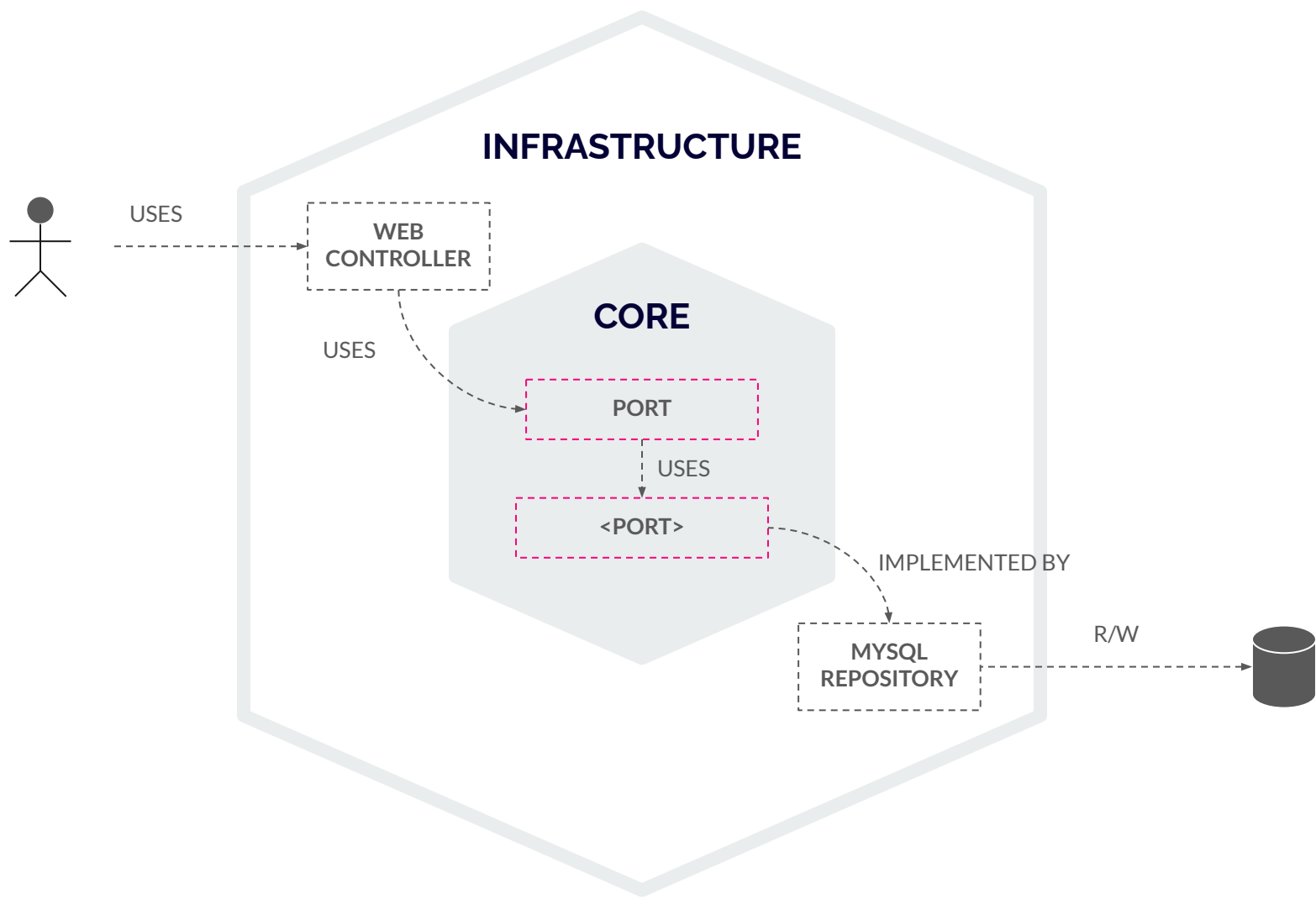
PORT

- WHAT OUR CORE CAN DO
- INTENT OF COMMUNICATION

ADAPTER

...



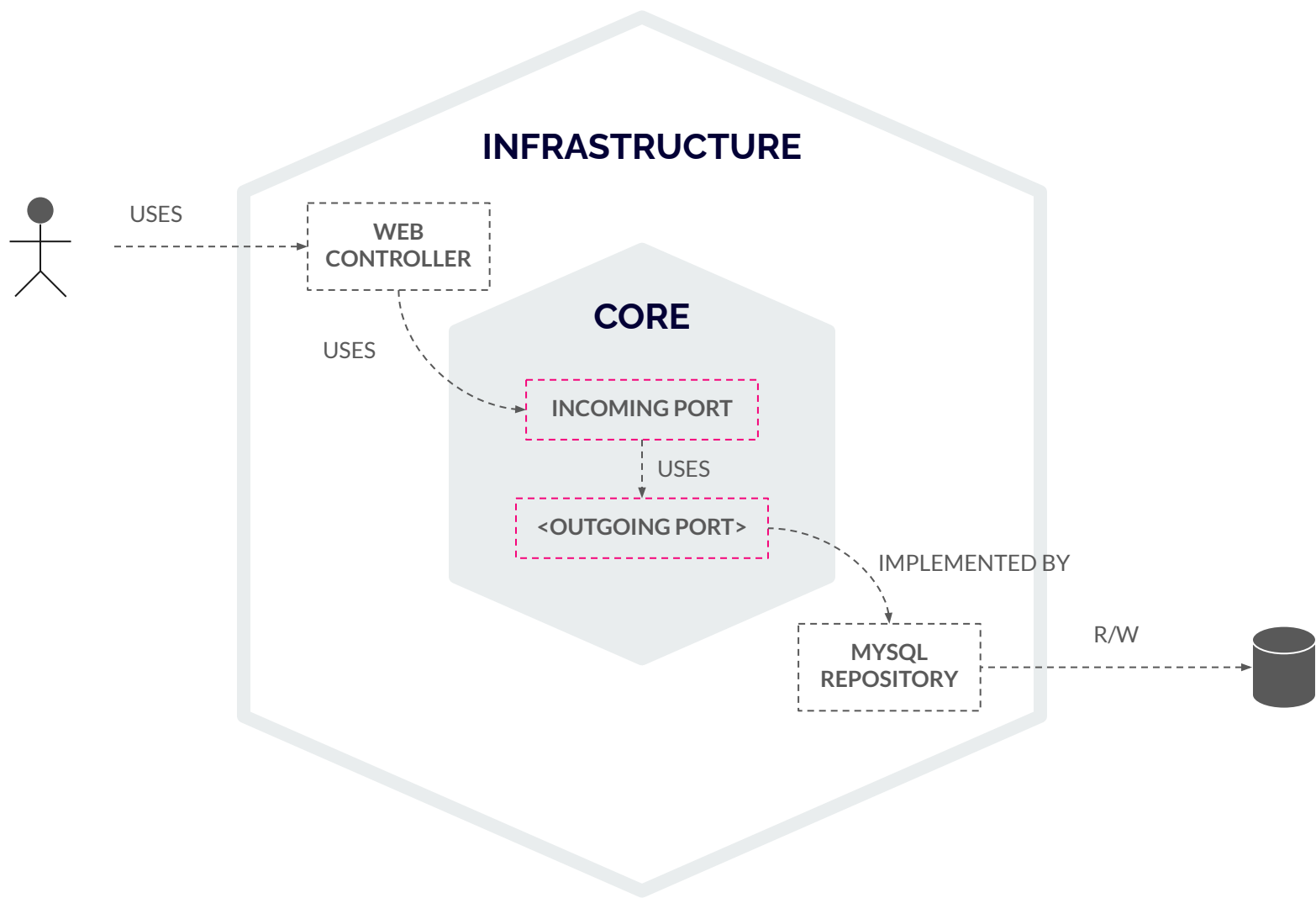


PORT

- INCOMING COMMUNICATION
- OUTGOING COMMUNICATION

ADAPTER

...

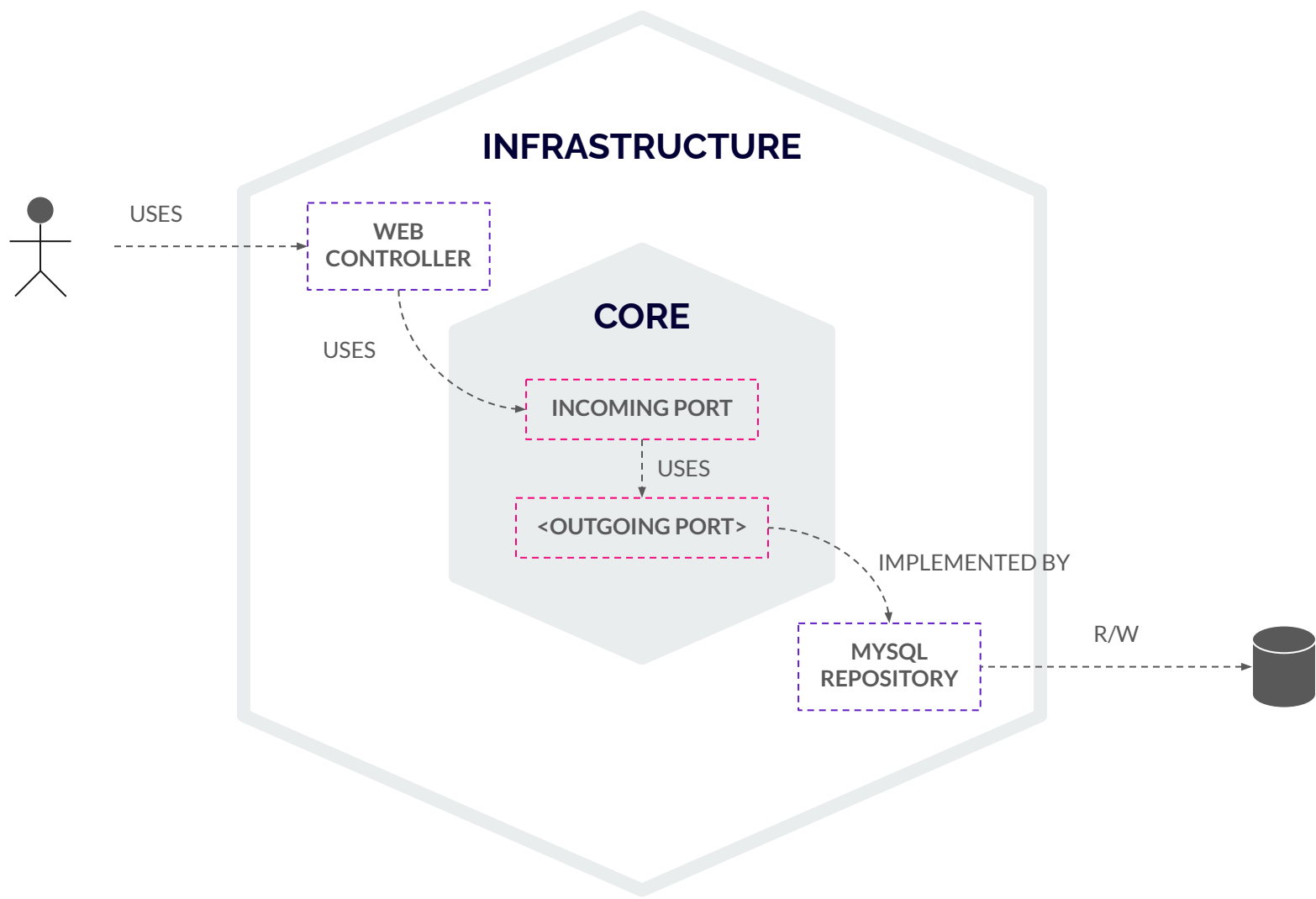


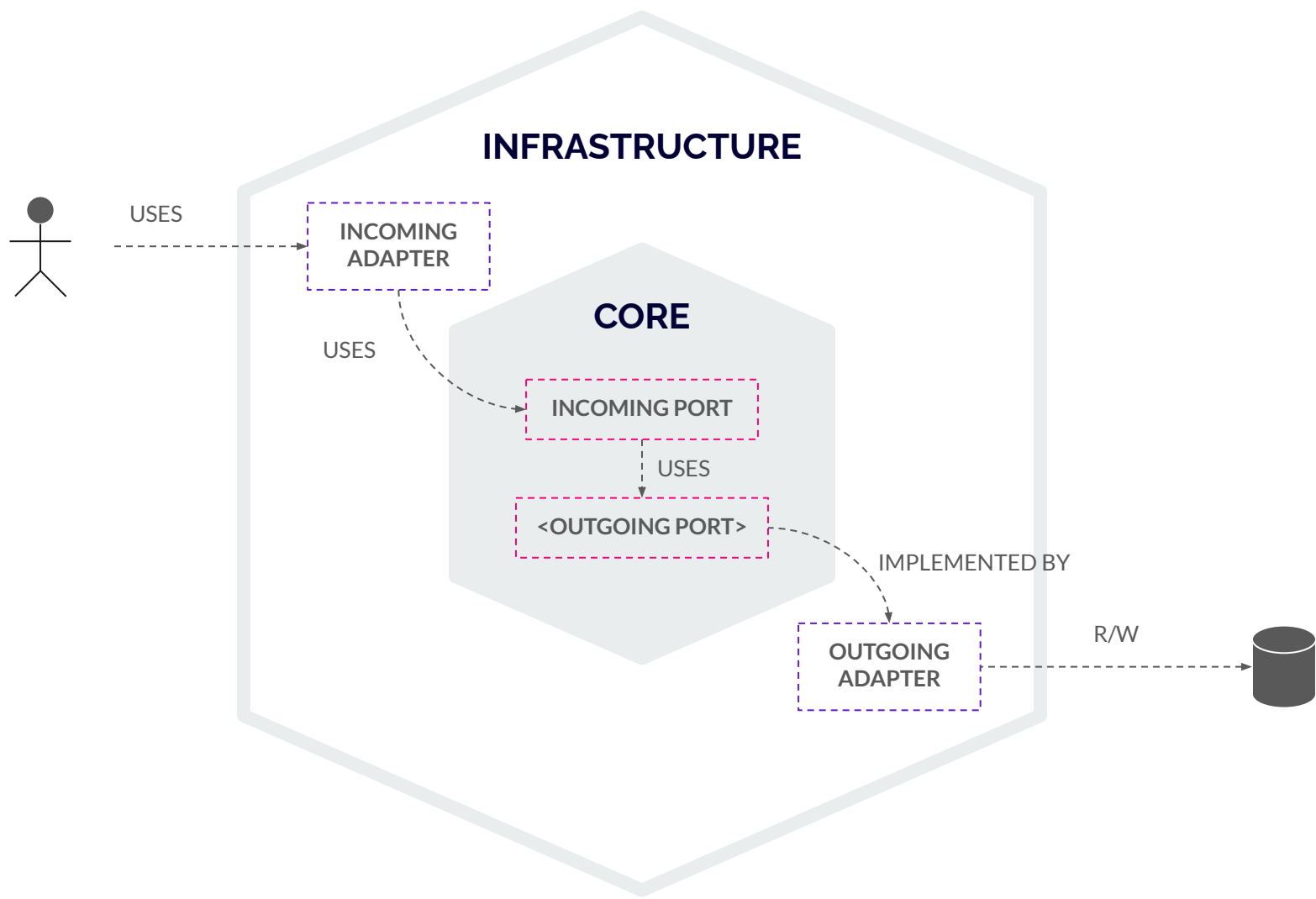
PORT

...

ADAPTER

- USES INCOMING PORTS
- IMPLEMENTS OUTGOING PORTS





Focus on the domain

the CORE of our application



INFRASTRUCTURE

CORE





YOU NAME IT



Focus on testability

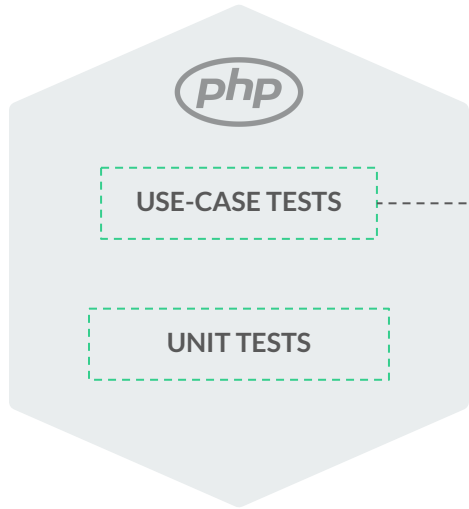
with framework-agnostic tests





USE-CASE TESTS

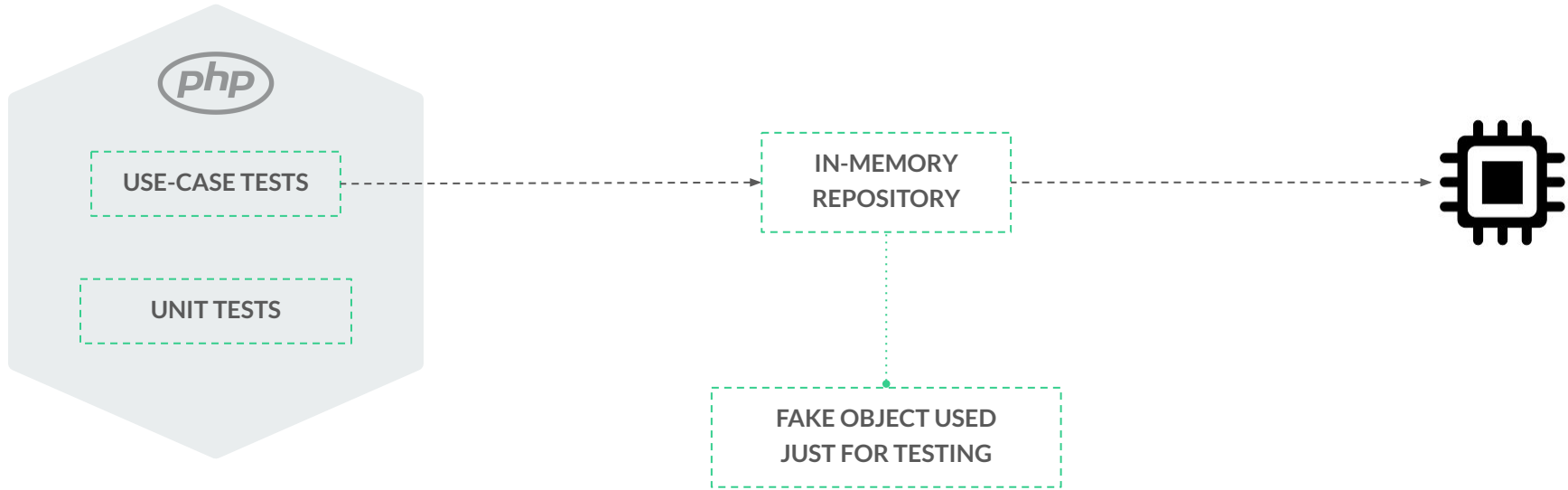
UNIT TESTS

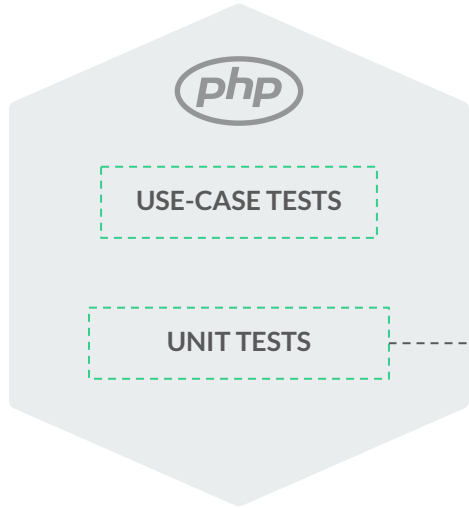


Registration

- ✓ The customer can register an unregistered card
- ✓ The customer can't register already registered card
- ✓ The customer can't register more than one card
- ✓ The customer can delete a registered card

OK (4 tests)

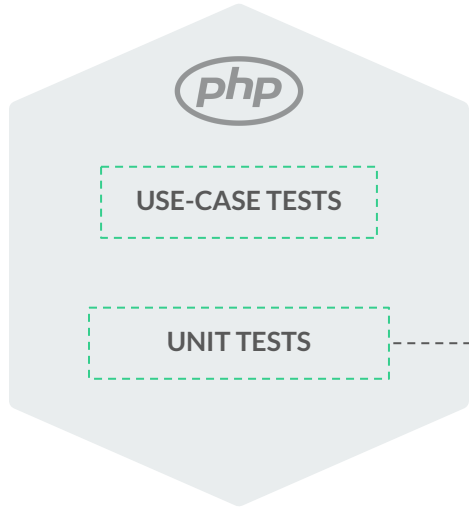




Card Number

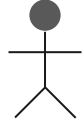
- ✓ It accepts ten digits
- ✓ It accepts all zeroes
- ✓ It doesn't accept an empty string
- ✓ It doesn't accept less than ten digits
- ✓ It doesn't accept more than ten digits
- ✓ It doesn't accept invalid chars

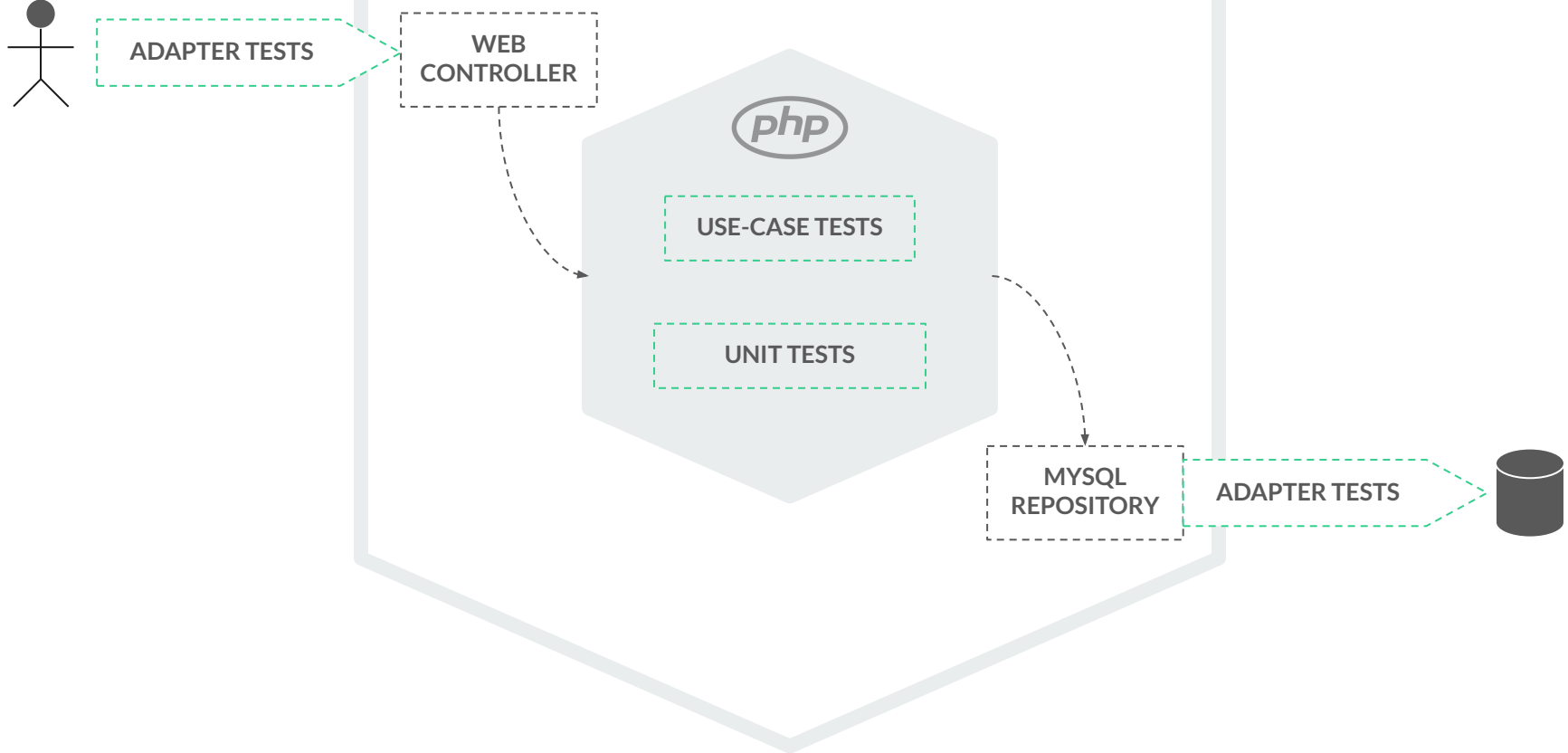
OK (6 tests)

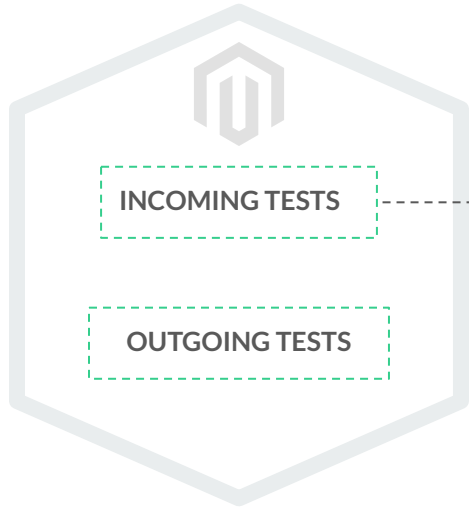


Test Infrastructure

with ~~integration~~ adapter tests



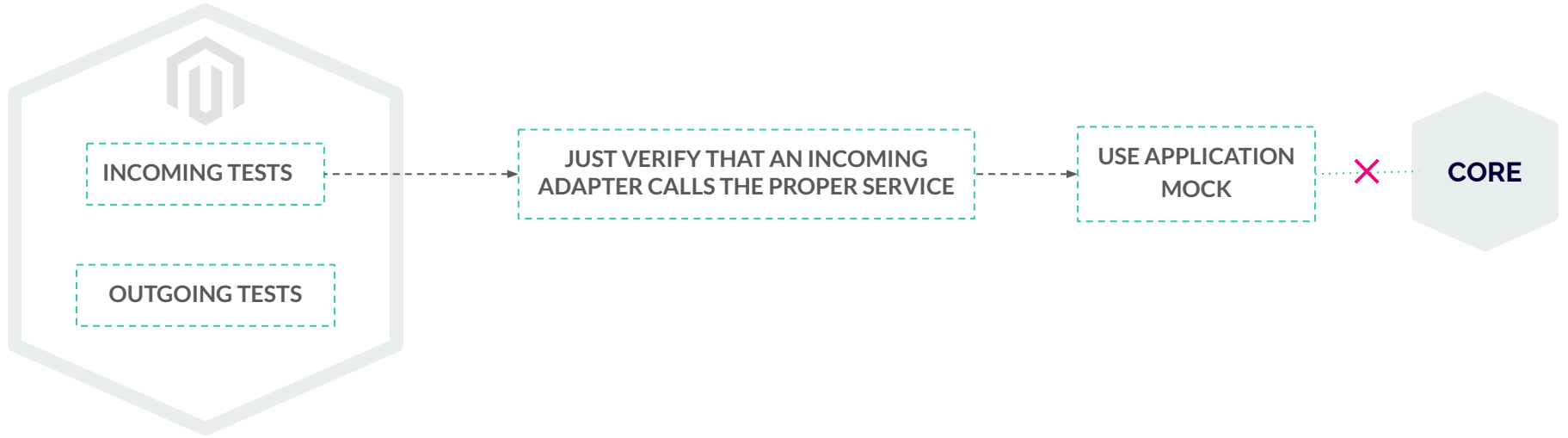




Card Controller

- ✓ It calls Application service `getCardByCustomer()`
- ✓ It calls Application service `registerCard()`
- ✓ It calls Application service `deleteCardByCustomer()`

OK (3 tests)



INCOMING TESTS

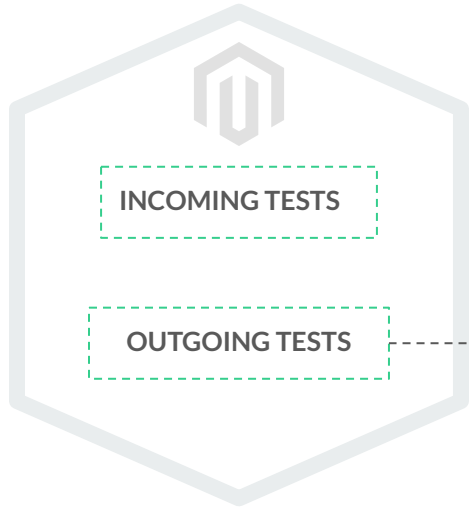
OUTGOING TESTS

JUST VERIFY THAT AN INCOMING
ADAPTER CALLS THE PROPER SERVICE

USE APPLICATION
MOCK



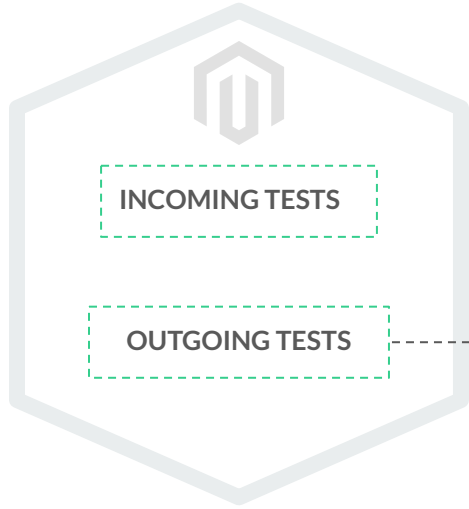
CORE



Card Repository

- ✓ It can save a card and load it by its number
- ✓ It can save a card and load it by its customer
- ✓ It can delete a card by its customer

OK (3 tests)

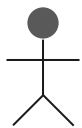


JUST VERIFY THAT AN OUTGOING ADAPTER PROPERLY HANDLES I/O



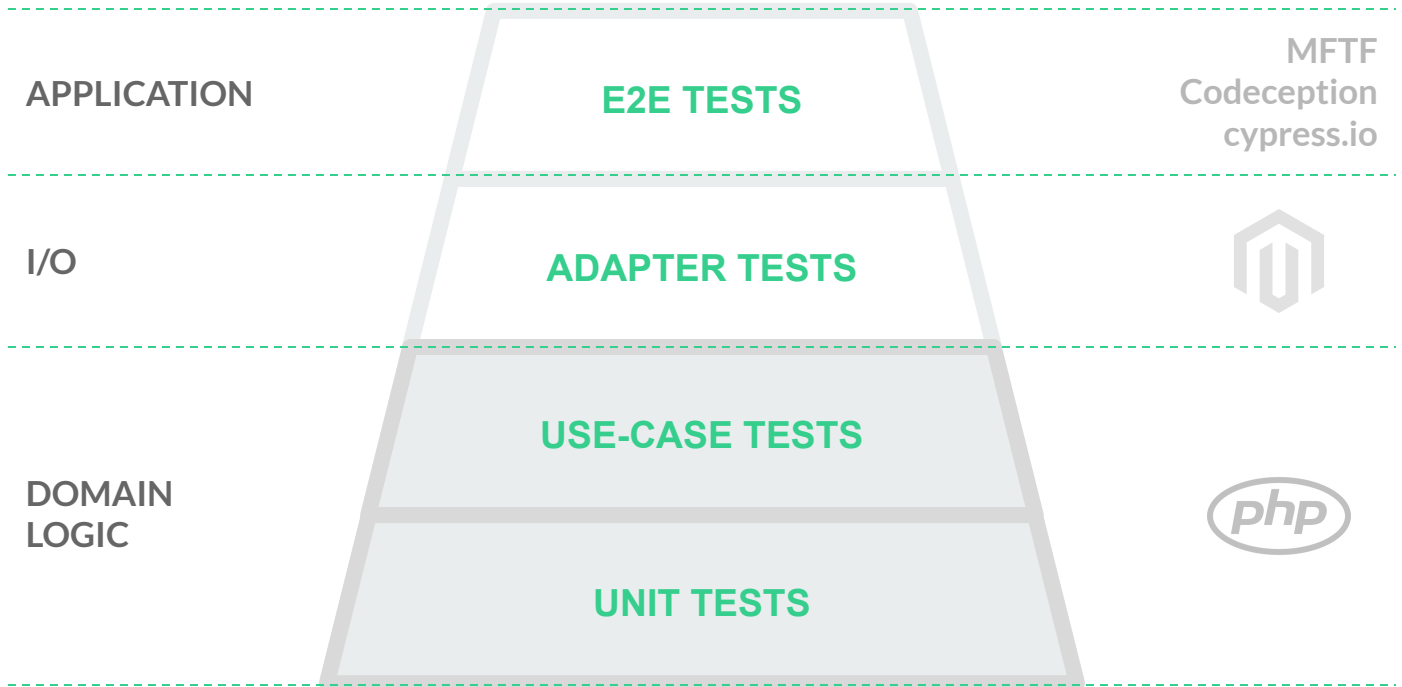
Increase confidence

with ~~functional~~ end-to-end tests



END-TO-END TESTS





APPLICATION

E2E TESTS

MFTF
Codeception
cypress.io

I/O

ADAPTER TESTS

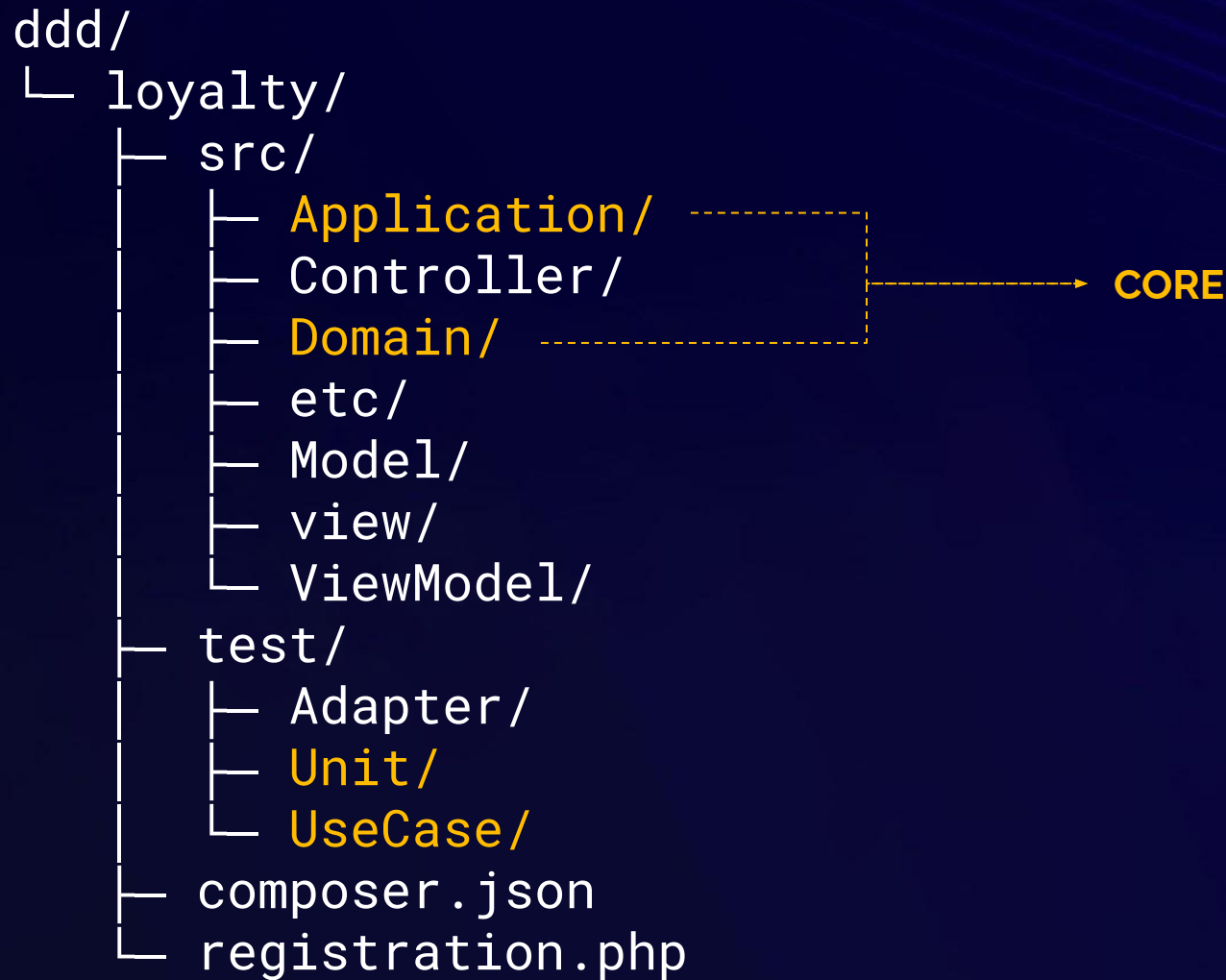


DOMAIN
LOGIC

USE-CASE TESTS



UNIT TESTS



When it is worth

- ✓ **Complex logic**
- ✓ **Long-lasting code**
- ✓ **Multi-platform code**

When it is **not** worth

- ✗ **Core hacks or fixes**
- ✗ **Short-term code**
- ✗ **Platform-specific code**

Takeaways

- ✓ **Split Core/Infrastructure**
- ✓ **Use own abstractions**
- ✓ **Apply a Testing Strategy**

The project

✓ <http://bit.ly/ddd-loyalty>



Inspired by



The book

- ✓ With a **special discount**
- ✓ <https://bit.ly/mn-awaa>



Thank you!

Questions: [@aleron75](#)

Meet Magento  

MM21IT

June 10/11 2021 - Virtual Conference