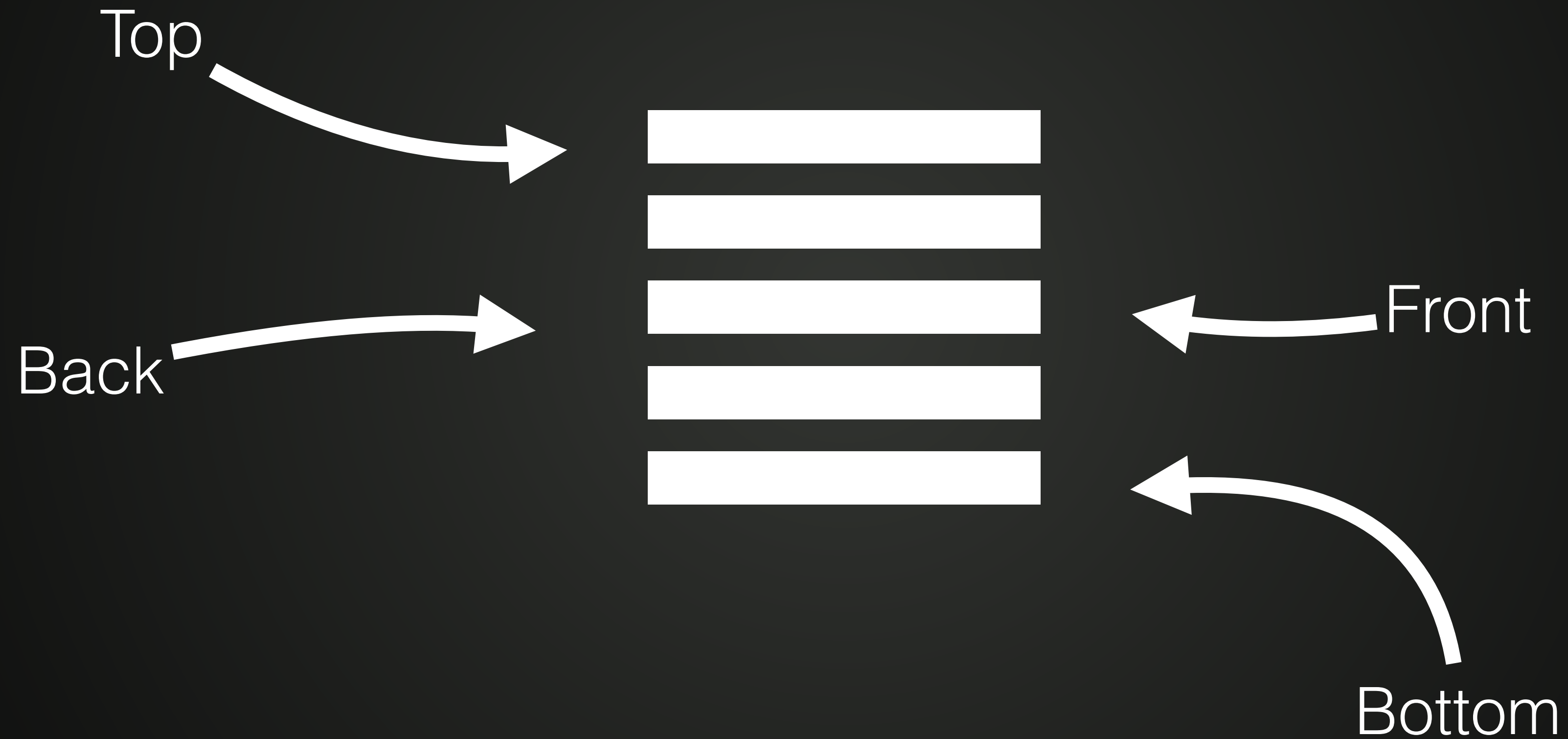


Mastering Message Queues

Tobias Nyholm

@tobiasnyholm

The data structure



Why?

Tobias Nyholm

- Full stack unicorn on Happyr.com
- Certified Symfony developer
- Symfony core member
- PHP-Stockholm
- Open source

Open source

Assert
Neo4j
FriendsOfApi/boilerplate
Backup-manager/symfony Guzzle Buzz nyholm/effective-interest-rate
Swap Puli LinkedIn API client
NSA
php-http/discovery
PSR7
PHP-cache PHP-Translation
CacheBundle
league/geotools Mailgun
KNP Github API happyr/normal-distribution-bundle
HTTPlug
Stampie SymfonyBundleTest
MailgunBundle php-http/httpplug-bundle php-http/multipart-stream
SimpleBus integrations
PHP-Geocoder
PSR HTTP clients BazingaGeocoderBundle

2013

- ▼  src
 - ▼  Carlin
 - ▼  Advert
 - ▶  PatternBundle
 - ▶  PersonalQuestionBundle
 - ▶  PositionBundle
 - ▶  RecruitmentBundle
 - ▶  RequirementBundle
 -  readme.txt
 - ▼  Application
 - ▶  ApplicationBundle
 - ▶  ResumeBundle
 - ▶  SpontaneousBundle
 - ▶  SubscriptionBundle
 - ▶  UserPotentialBundle
 -  readme.txt
 - ▼  Communication
 - ▶  BookingBundle
 - ▶  BufferedMailBundle
 - ▶  CompanyMailBundle
 - ▶  MailerBundle
 - ▶  QnaBundle
 -  readme.txt
 - ▶  CoreBundle
 - ▶  Integration
 - ▶  Location
 - ▶  Statistics
 - ▶  User
 -  readme.txt
 -  .htaccess

- ▼ CompanyBundle
 - ▶ Controller
 - ▼ DependencyInjection
 - Ⓢ Configuration.php
 - Ⓢ WebfishCompanyExtension.php
 - ▶ Entity
 - ▶ Event
 - ▶ Form
 - ▶ Manager
 - ▶ Model
 - ▼ Resources
 - ▼ config
 - 📄 services.yml
 - ▼ public
 - ▶ css
 - ▶ js
 - ▼ translations
 - 📄 messages.sv.yml
 - ▶ views
 - ▶ Service
 - ▶ Tests
 - Ⓢ CompanyEvents.php
 - Ⓢ WebfishCompanyBundle.php

```
<?php

namespace Webfish\CompanyBundle\Controller;

use Money\Money;
use Rawls\Bundle\Controller\BaseController;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;
use Symfony\Component\HttpFoundation\RedirectResponse;
use Symfony\Component\HttpFoundation\Request;
use Webfish\CompanyBundle\Entity\Company;
use Webfish\CompanyBundle\Entity\SaleImport;
use Webfish\CompanyBundle\Form\SalesImportType;
use Webfish\CompanyBundle\Model\SalesImportModel;

class ImportController extends BaseController
{
    /**
     * @param Request $request
     *
     * @Route("/import/sales", name="sales_import")
     * @Template()
     *
     * @return array|RedirectResponse
     */
    public function importAction(Request $request)
    {

```


“Testing your controllers.”

–*Twitter*

[DX] Added CurrentUserProvider service #14407

Edit

Closed

 Nyholm wants to merge 1 commit into symfony:2.8 from Nyholm:CurrentUserProvider

Conversation 11

Commits 1

Checks 0

Files changed 5

+140 -65



Nyholm commented on Apr 19, 2015

Member



Q	A
Bug fix?	no
New feature?	yes
BC breaks?	no
Deprecations?	no
Tests pass?	yes
Fixed tickets	none
License	MIT
Doc PR	symfony/symfony-docs#5192

I believe it would be nice to make it little easier to retrieve the current user. This provider will hide the complexity with the token and check if the `$token->getUser()` is an object or a string.

Reviewers

No reviews

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Unsubscribe

You're receiving notifications because you modified the open/close state.

5 participants



Nyholm referenced this pull request in symfony/symfony-docs on Apr 19, 2015

[DX] Added info about security.current_user_provider #5192

Closed

[DX] Added CurrentUserProvider service

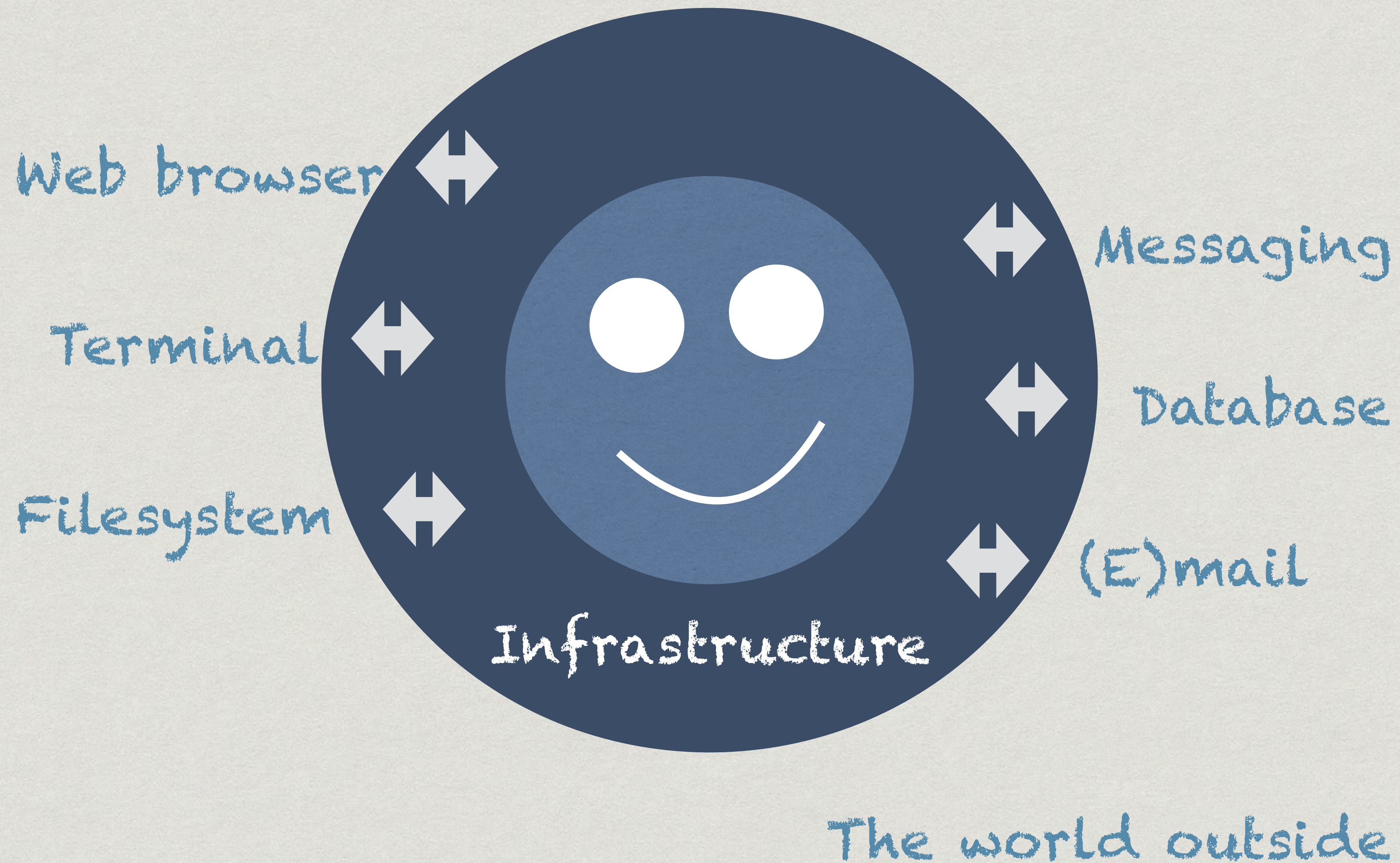
0532d49

wouterj reviewed on Apr 19, 2015

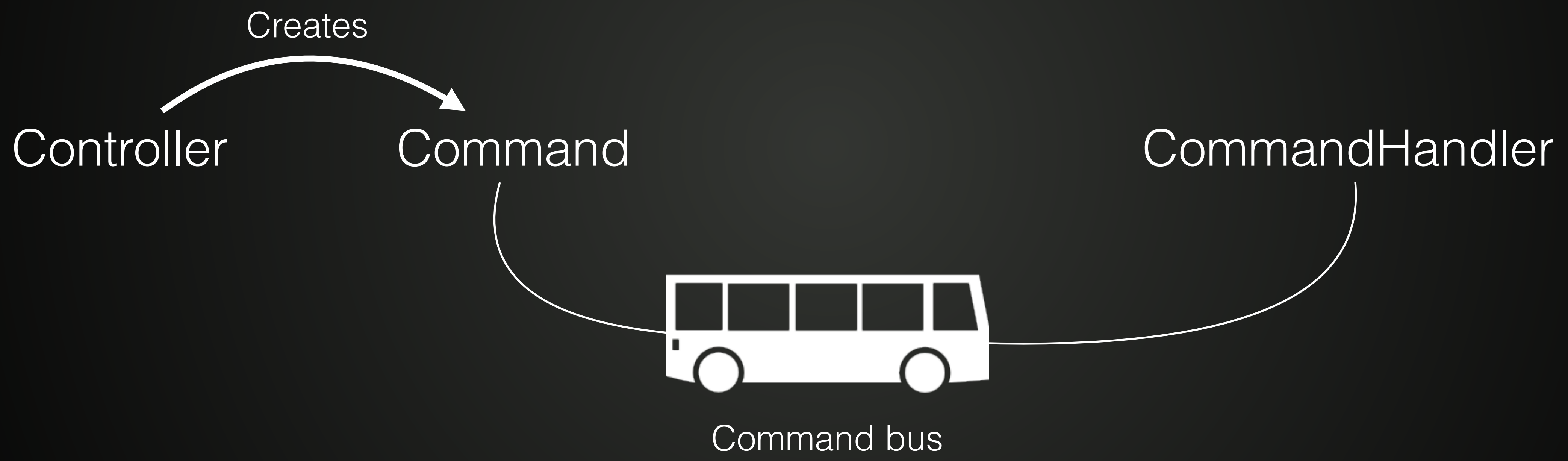
View changes

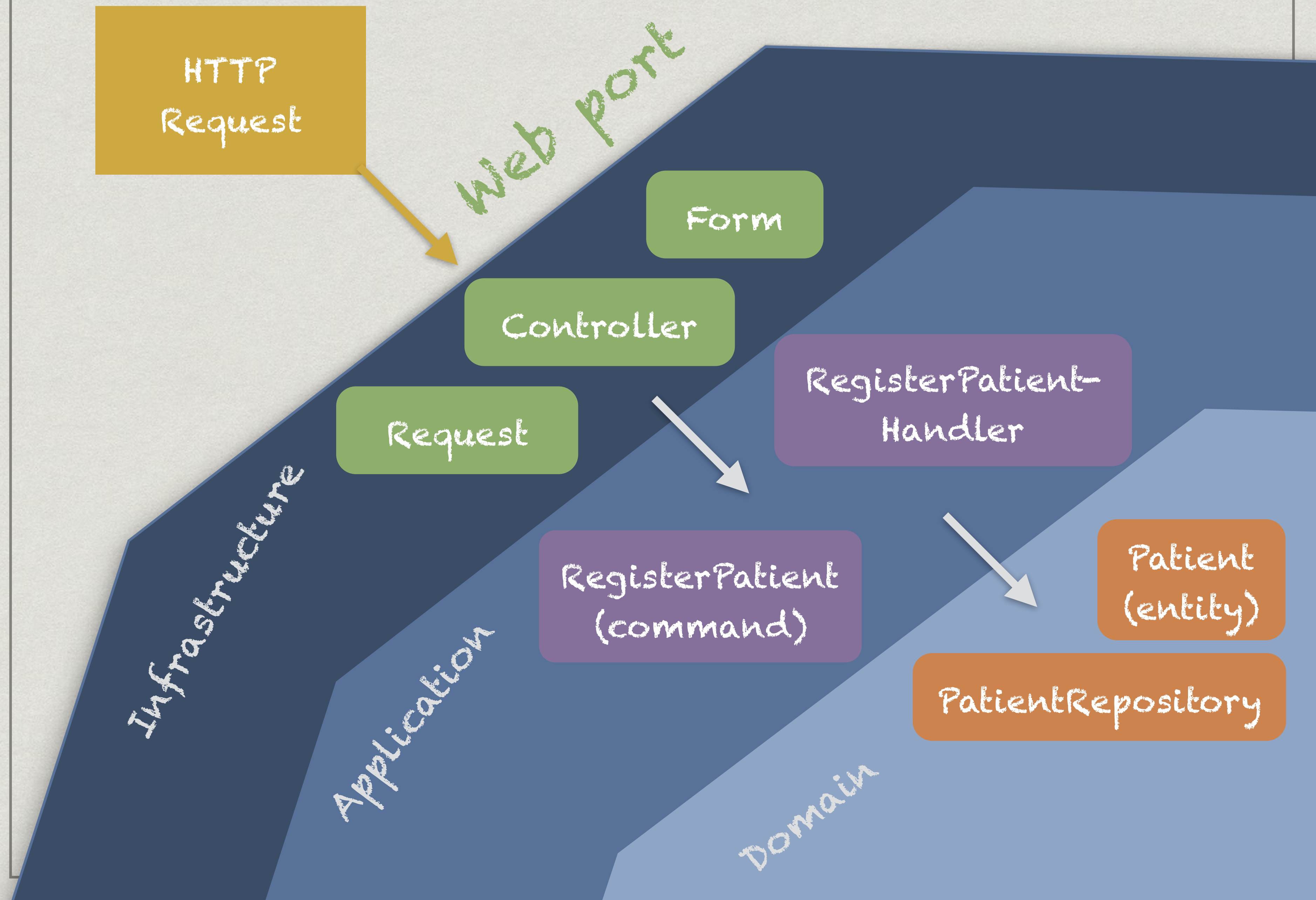
Matthias Noback

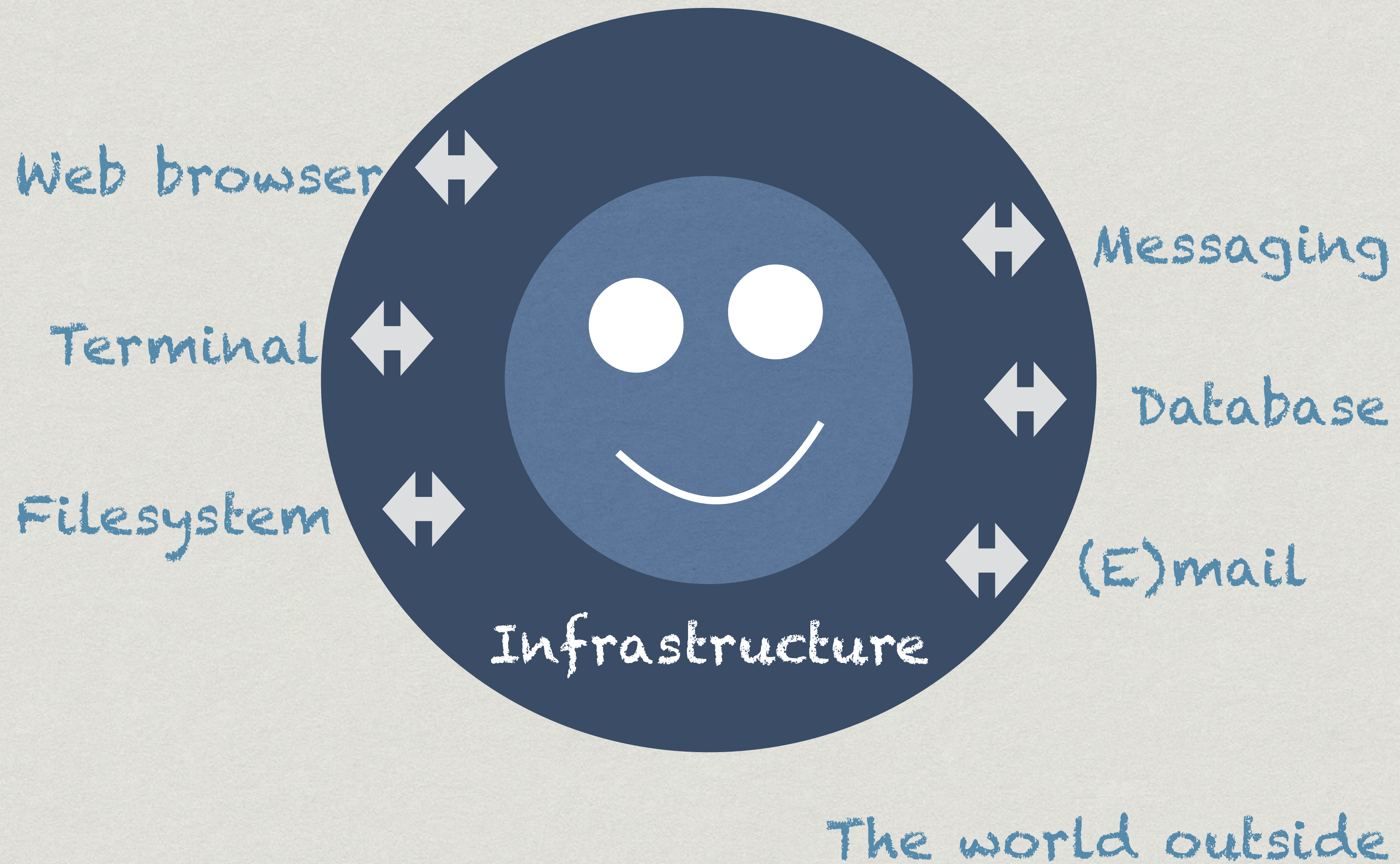




- ▼ src
 - ▼ App
 - ▼ Message
 - ▼ Command
 - ▶ Advert
 - ▼ Company
 - CreateCompany.php
 - ▶ Competence
 - ▶ Industry
 - ▶ Interest
 - ▶ Location
 - ▶ User
 - ▼ CommandHandler
 - ▶ Advert
 - ▼ Company
 - CreateCompanyHandler.php
 - ▶ Competence
 - ▶ Industry
 - ▶ Interest
 - ▶ Location
 - ▶ User
 - BaseCommandHandler.php







Hexagonal architecture

<https://www.youtube.com/watch?v=fgQWngInGeU>

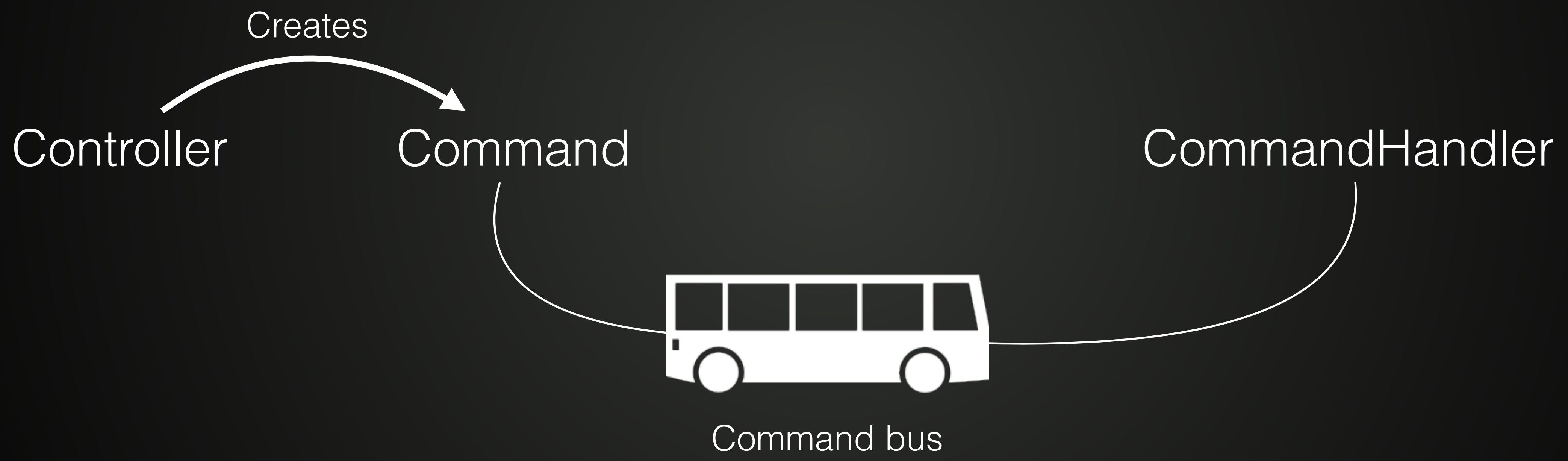
<https://matthiasnoback.nl/2015/01/a-wave-of-command-buses/>

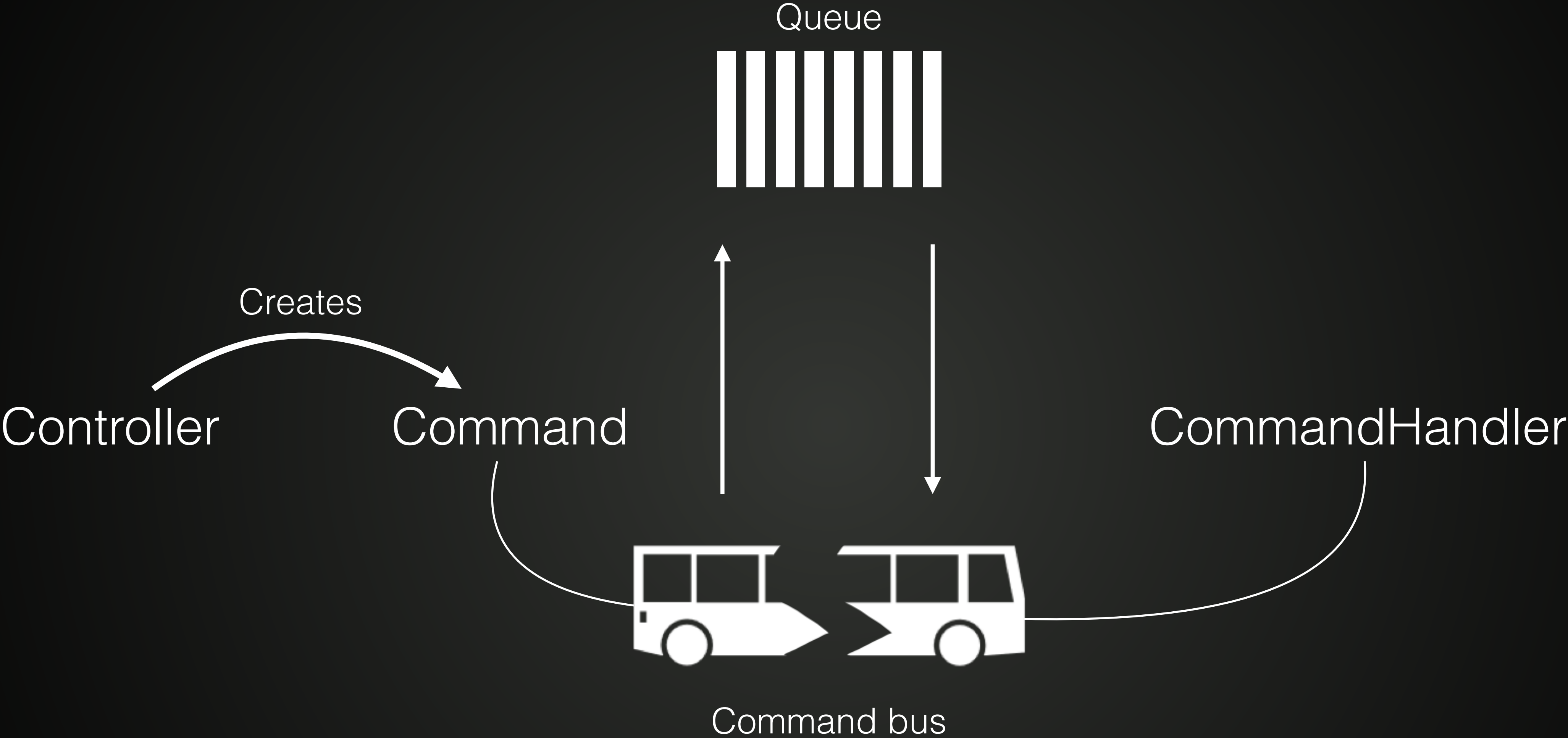
<https://martinfowler.com/bliki/CQRS.html>

<https://hexagonal-architecture.eu/>

What did we win?

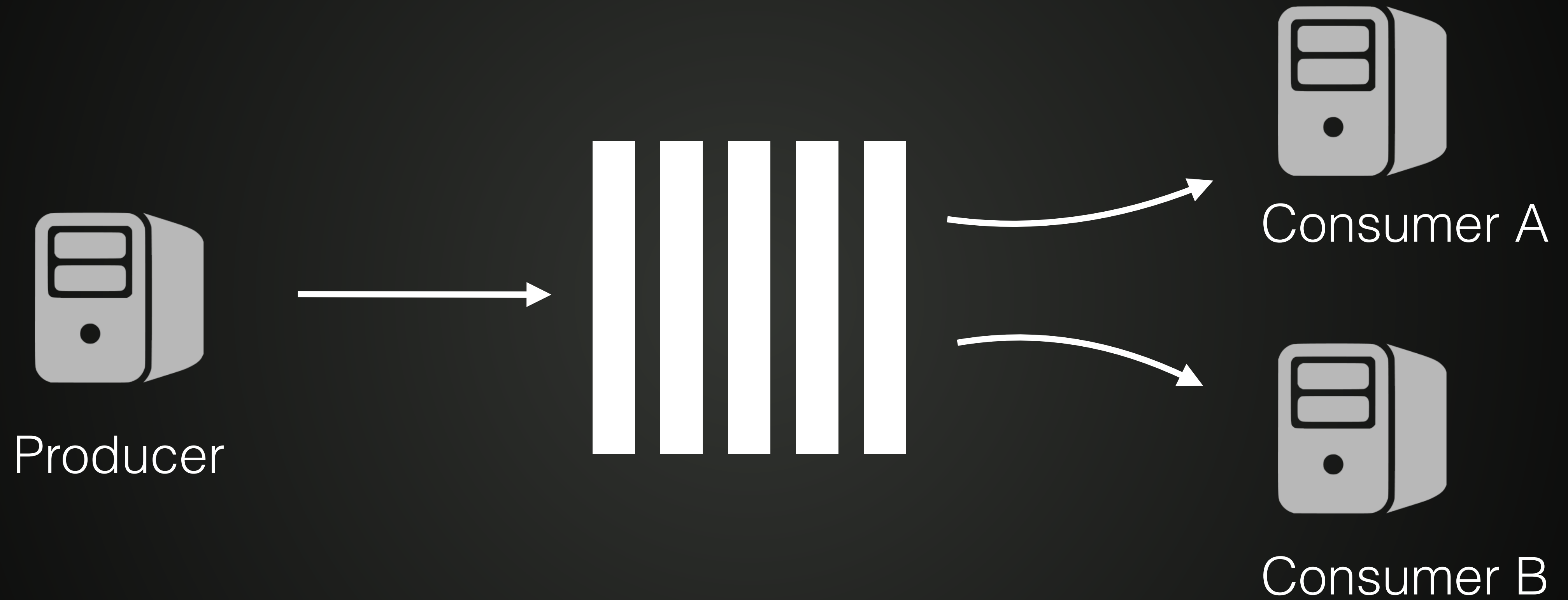
Middlewares!



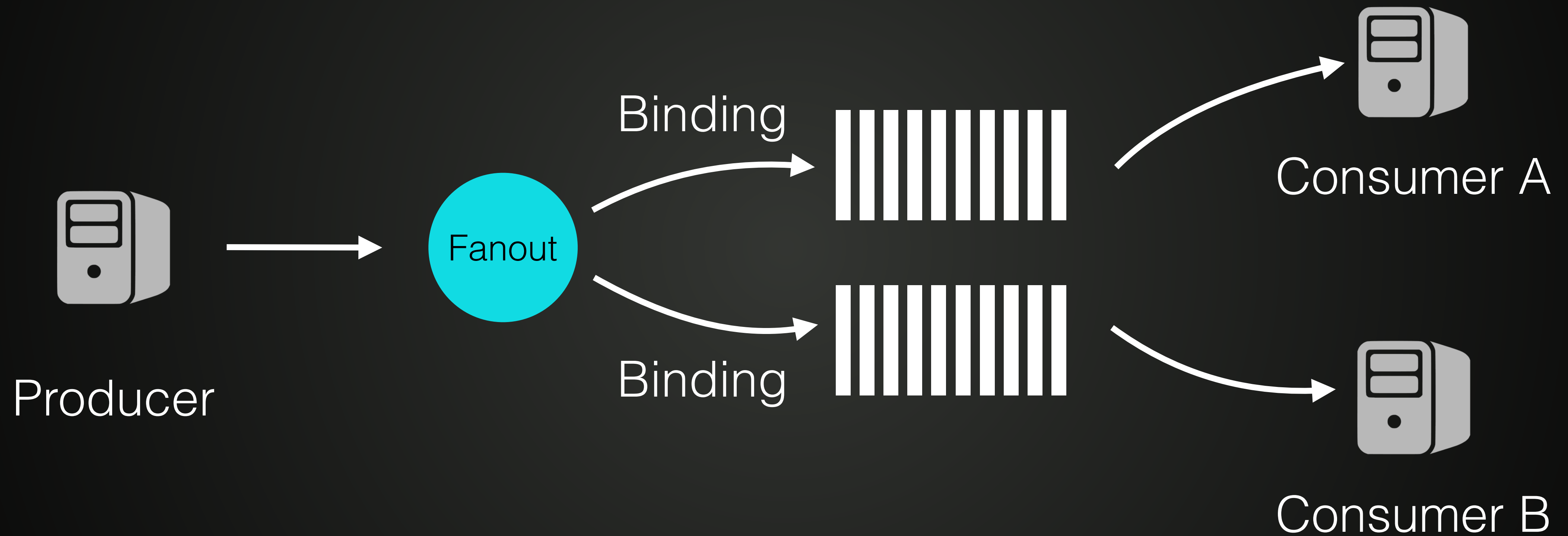


Let's talk about queues

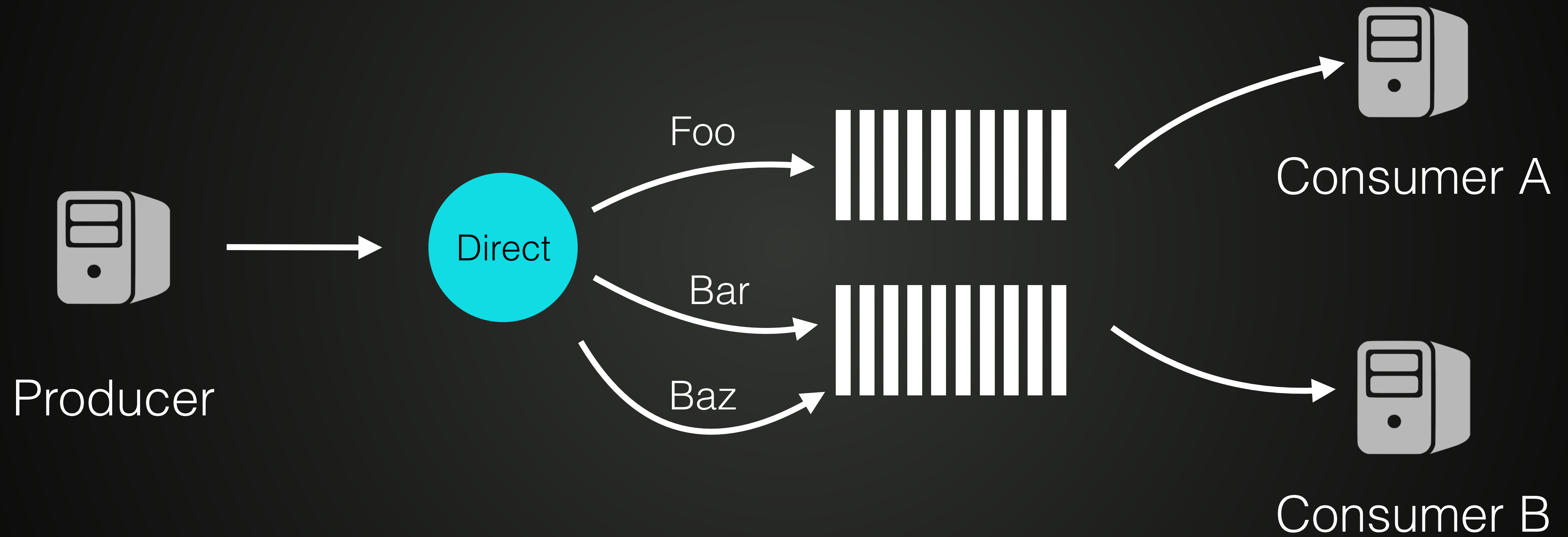
Work queue



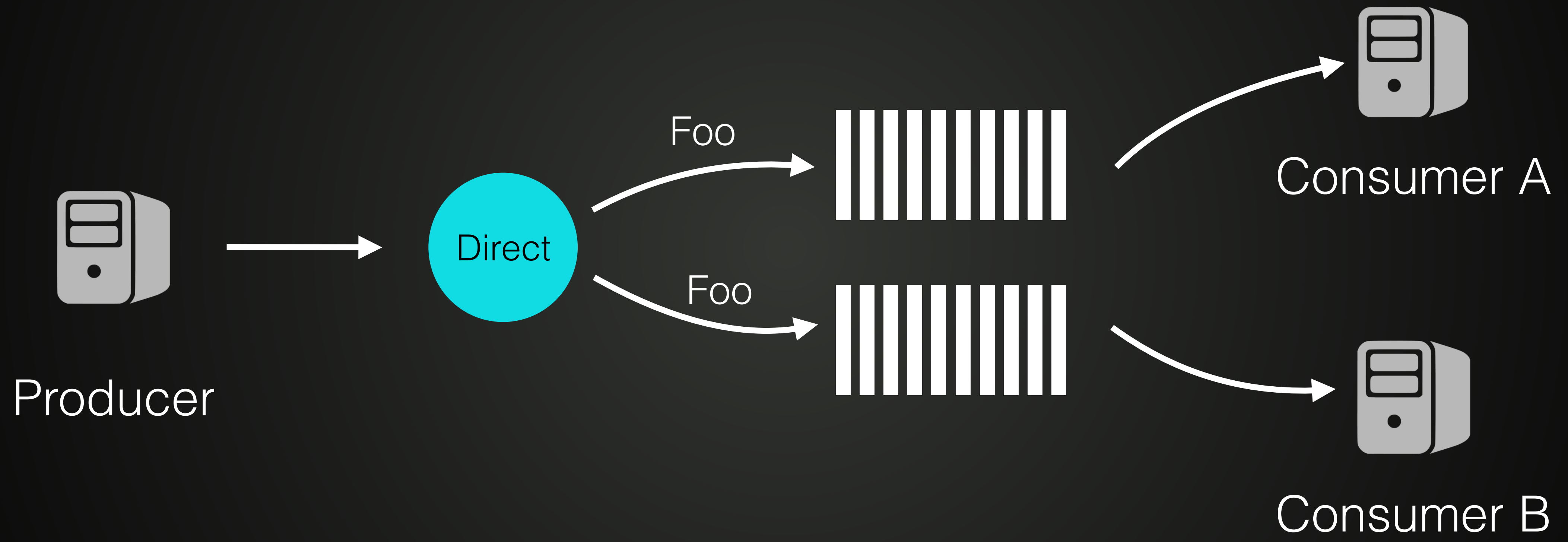
Publish/Subscribe



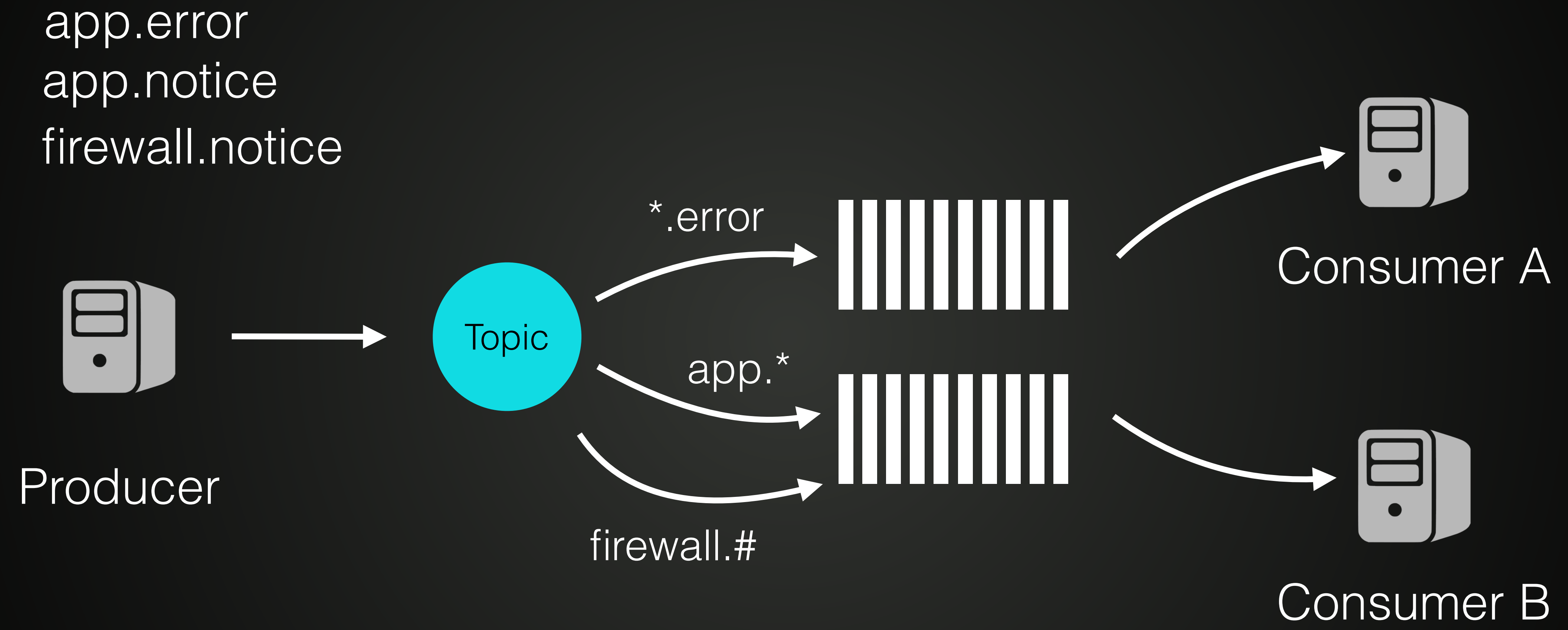
Routing



Routing

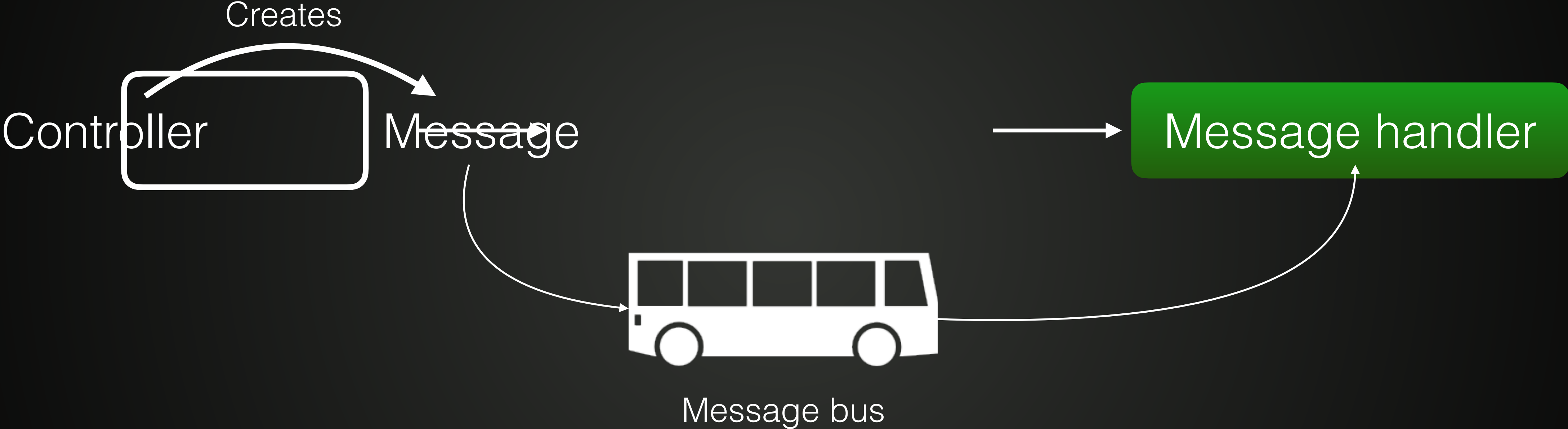


Topics



Let's see some code

```
$ composer req symfony/messenger
```



<?php

@tobiasnyholm

```
namespace App\Message\Command;
```

```
class SendNotification
```

```
{
```

```
    private $message;
```

```
    private $users;
```

```
    public function __construct(string $message, array $users)
```

```
    {
```

```
        $this->message = $message;
```

```
        $this->users = $users;
```

```
    }
```

```
    public function getMessage(): string
```

```
    {
```

```
        return $this->message;
```

```
    }
```

```
    public function getUsers(): array
```

```
    {
```

```
        return $this->users;
```

```
    }
```

```
}
```

<?php

@tobiasnyholm

```
namespace App\Controller;

use App\Message\Command\SendNotification;
use Symfony\Component\Messenger\MessageBusInterface;
// ...

class DefaultController
{
    public function index(MessageBusInterface $bus, Request $request)
    {
        $users = ['samuel', 'christelle'];
        $message = $request->query->get('message', 'Something. ');

        $bus->dispatch(new SendNotification($message, $users));

        return new Response('<html><body>OK.</body></html>');
    }
}
```

Symfony Exception

Symfony Docs

Symfony Support

NoHandlerForMessageException

HTTP 500 Internal Server Error

No handler for message "App\Message\Command\SendNotification".

Exception

Logs1

Stack Trace

Symfony\Component\Messenger\Exception\NoHandlerForMessageException

+ in vendor/symfony/messenger/Handler/Locator/ContainerHandlerLocator.php (line 36)

+ in vendor/symfony/messenger/Middleware/HandleMessageMiddleware.php -> resolve (line 33)

+ in vendor/symfony/messenger/MessageBus.php -> handle (line 74)

+ in vendor/symfony/messenger/Asynchronous/Middleware/SendMessageMiddleware.php -> Symfony\Component\Messenger\{closure} (line 57)

+ in vendor/symfony/messenger/MessageBus.php -> handle (line 74)

+ in vendor/symfony/messenger/Middleware/LoggingMiddleware.php -> Symfony\Component\Messenger\{closure} (line 36)

+ in vendor/symfony/messenger/MessageBus.php -> handle (line 74)

+ in vendor/symfony/messenger/MessageBus.php -> Symfony\Component\Messenger\{closure} (line 47)

+ in vendor/symfony/messenger/TraceableMessageBus.php -> dispatch (line 38)

- TraceableMessageBus -> dispatch (object(SendNotification)) in src/Controller/Startpage.php (line 44)

39. */

40. public function notify(Request \$request)

41. {

42. \$users = ['samuel', 'christelle'];

500

@ app_startpage_notify

3434 ms

12.0 MB

1

8 in 2.23 ms

75 ms

1

Sf 4.2.0-DEV

```
<?php

namespace App\MessageHandler;

use App\Message\Command\SendNotification;

class SendNotificationHandler
{
    public function __invoke(SendNotification $message)
    {
        foreach ($message->getUsers() as $user) {
            echo "Send notification to... ".$user."\n";
        }
    }
}
```

```
# config/services.yaml
services:
    App\Message\CommandHandler\SendNotificationHandler:
        tags:
            - messenger.message_handler
```

```
# - - - - -
```

```
# config/services.yaml
services:
    App\Message\CommandHandler\:
        resource: '../src/Message/CommandHandler/*'
        tags: ['messenger.message_handler']
```

<

>

127.0.0.1:8000/notify

+

Send notification to... samuel Send notification to... christelle OK.

200

@ app_startpage_notify

3083 ms

8.0 MB

2

4

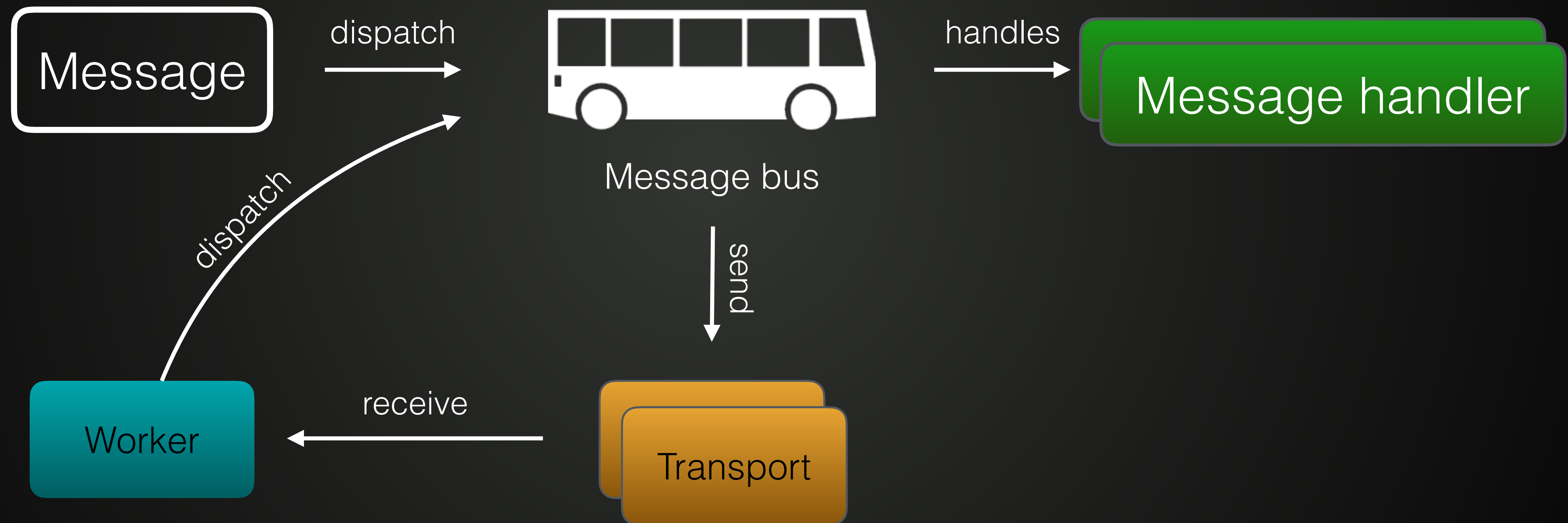
2 ms

1

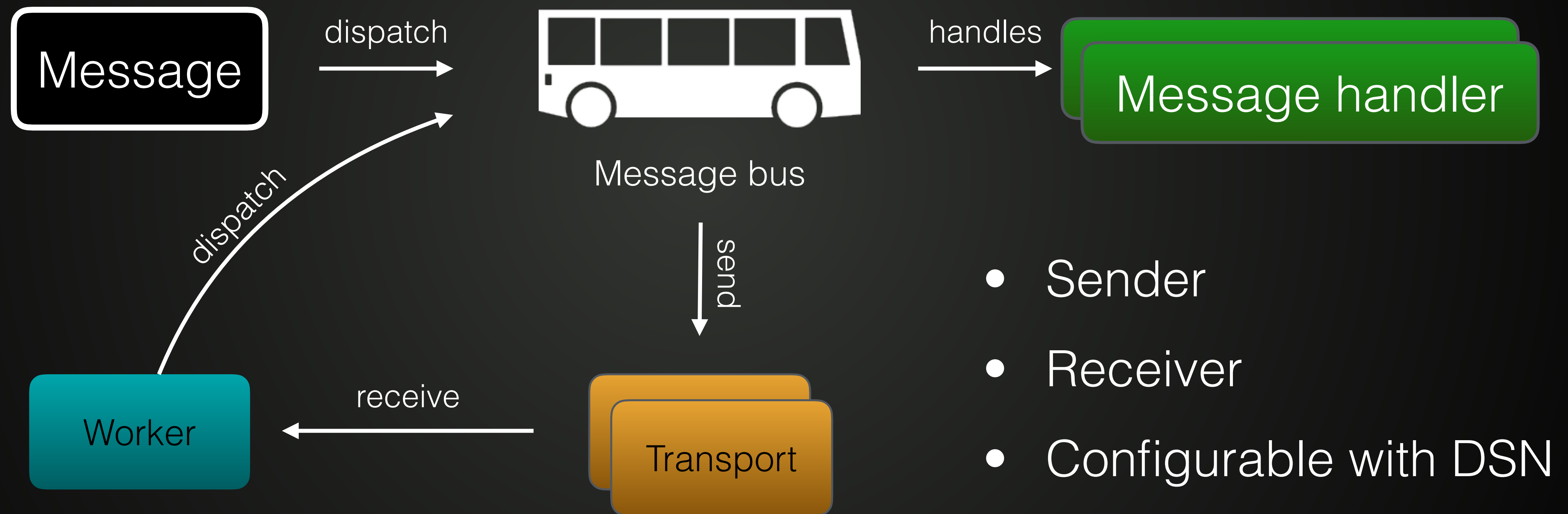
sf 4.2.0-DEV

X

Transports



Transports



- Sender
- Receiver
- Configurable with DSN
- Encode/Decode

```
# config/packages/messenger.yaml
framework:
    messenger:
        transports:
            default: '%env(MESSENGER_DSN) %'

        routing:
            'App\Message\Command\SendNotification': default

# -----

# .env
MESSENGER_DSN=amqp://guest:guest@localhost:5672/%2f/messages
```

<

>

127.0.0.1:8000/notify

OK.

200

@ app_startpage_notify

3164 ms

8.0 MB

2

34

2 ms

1

4.2.0-DEV

```
$ bin/console messenger:consume-messages
```



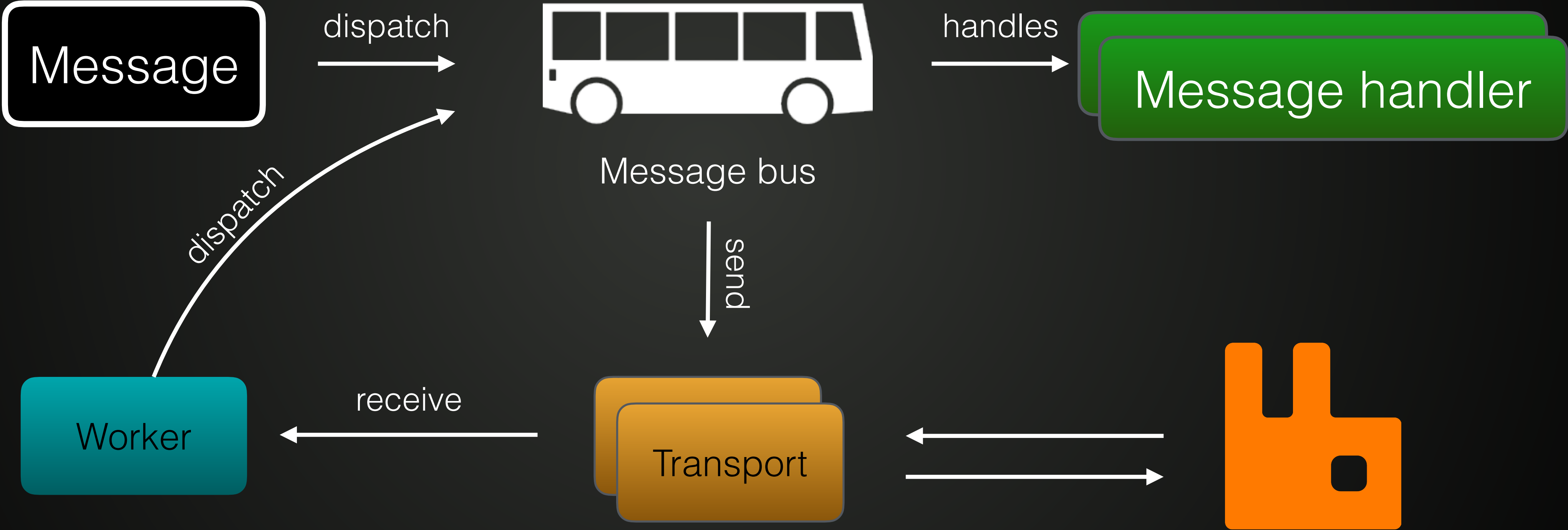
A terminal window titled "message-component-demo — bin/console messenger:consume-messages —...". The prompt is "[tobias@Tobiass-iMac:~/Workspace/PHPStorm/message-component-demo on master]". The command "% bin/console messenger:consume-messages" has been executed, resulting in two lines of output: "Send notification to... samuel" and "Send notification to... christelle". A cursor is visible on the line following the second output. In the top right corner of the terminal area, there is a purple star icon and a green plus sign icon.

```
[tobias@Tobiass-iMac:~/Workspace/PHPStorm/message-component-demo on master]
[% bin/console messenger:consume-messages
Send notification to... samuel
Send notification to... christelle
█
```



A terminal window titled "message-component-demo — bin/console messenger:consume-messages de...". The prompt is "[tobias@Tobiass-iMac:~/Workspace/PHPStorm/message-component-demo on master]". The command entered is "% bin/console messenger:consume-messages default --limit 100 --time-limit 300". A cursor is visible on the line below the command.

```
message-component-demo — bin/console messenger:consume-messages de...  
[tobias@Tobiass-iMac:~/Workspace/PHPStorm/message-component-demo on master]  
[% bin/console messenger:consume-messages default --limit 100 --time-limit 300 ]  
█
```



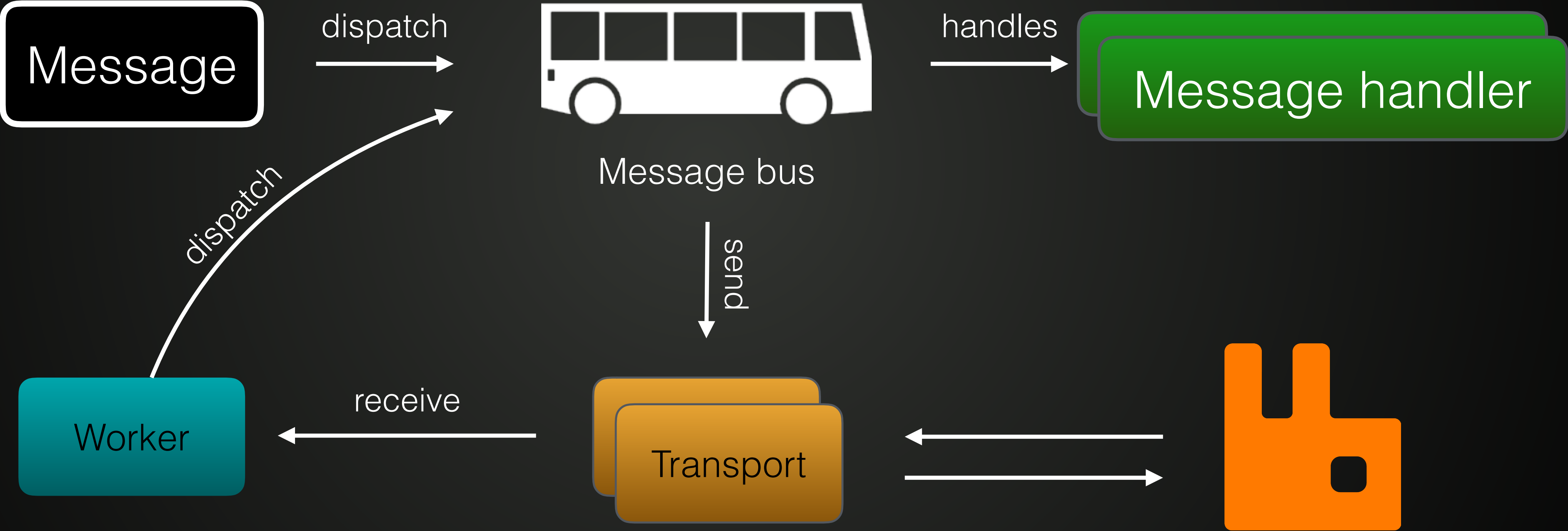


Enqueue is production ready, battle-tested messaging solution for PHP. Provides a common way for programs to create, send, read messages.

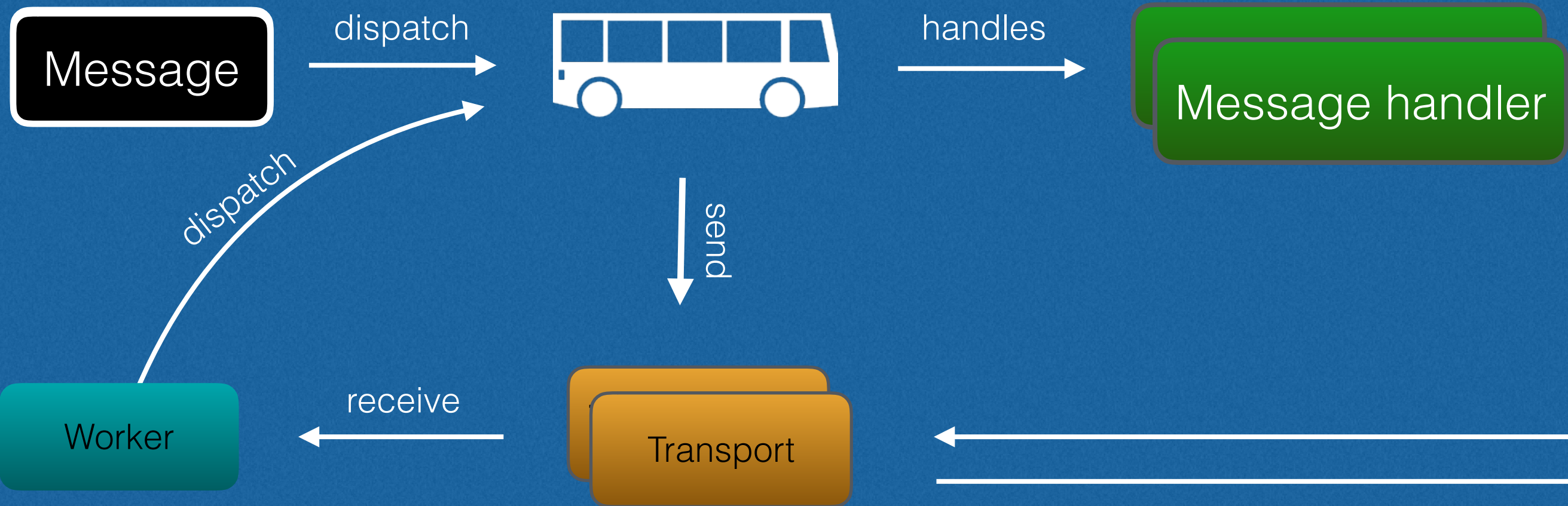
Supports major brokers such as RabbitMQ, Kafka, Amazon SQS, Google PubSub, Redis etc.

600+ GitHub stars
100k downloads and growing





App 1



App 2

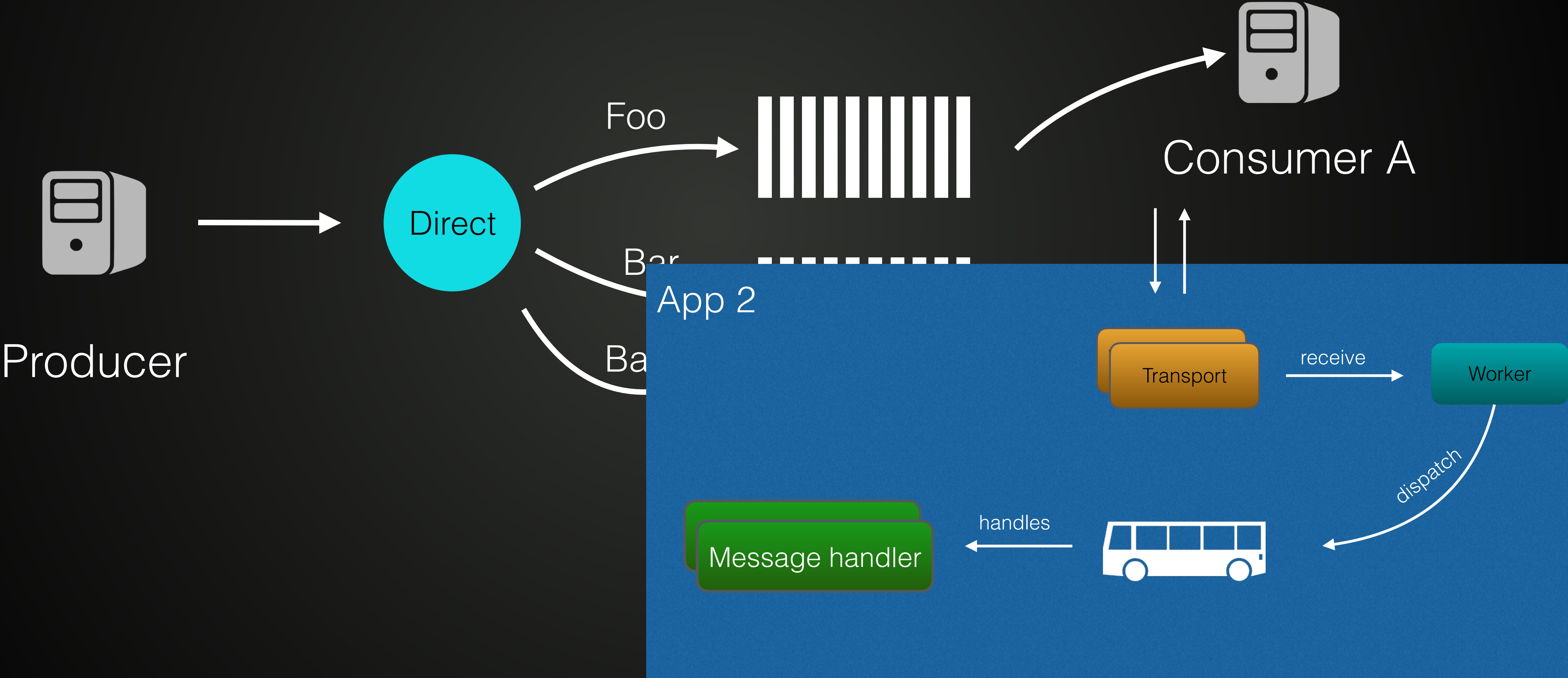


```
# config/packages/messenger.yaml
framework:
    messenger:
        transports:
            default: '%env(MESSENGER_DSN) %'
            enqueue: '%env(ENQUEUE_DSN) %'

        routing:
            'App\Message\Command\SendNotification': [default, enqueue]

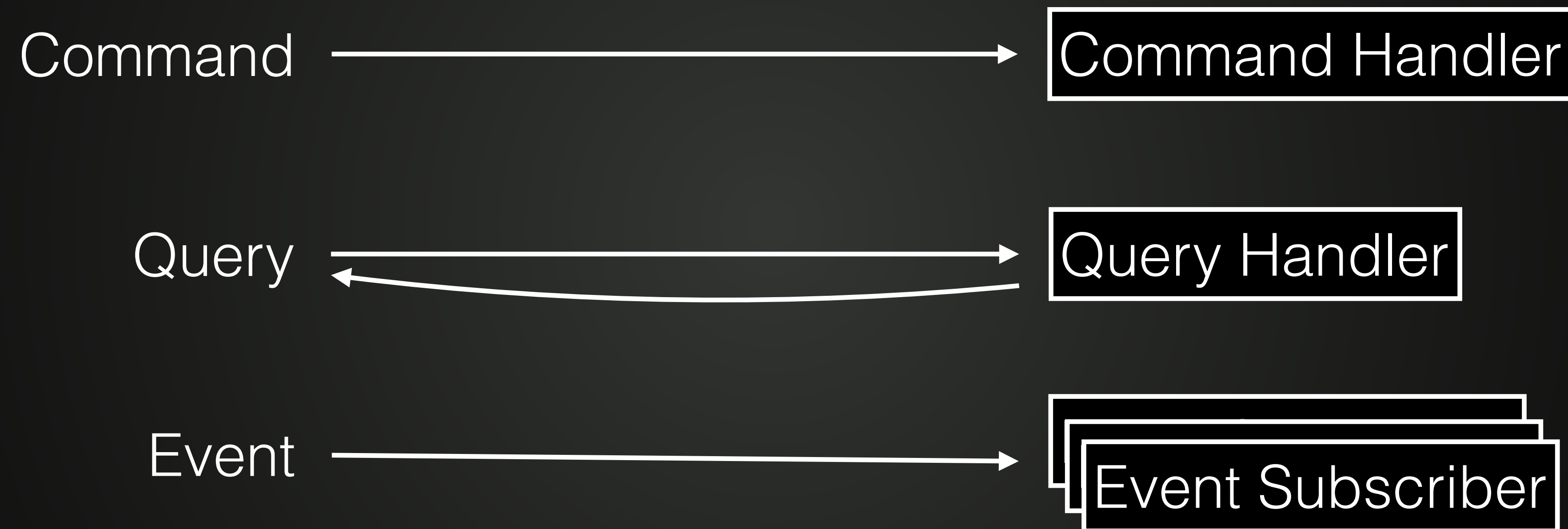
# - - - - -

# .env
MESSENGER_DSN=amqp://guest:guest@localhost:5672/%2f/messages
ENQUEUE_DSN=enqueue://acme
```



Multiple busses

Multiple busses



Multiple busses

```
# config/packages/messenger.yaml
framework:
    messenger:
        # The bus that is going to be injected when injecting MessageBusInterface:
        default_bus: messenger.bus.commands

        # Create buses
        buses:
            messenger.bus.command: ~
            messenger.bus.query: ~
            messenger.bus.event: ~
```

Multiple busses

```
# config/packages/messenger.yaml
framework:
    messenger:
        # The bus that is going to be injected when injecting MessageBusInterface:
        default_bus: messenger.bus.commands

        # Create buses
        buses:
            messenger.bus.command:
                middleware:
                    - messenger.middleware.exactly_one_handler
                    - messenger.middleware.validation
                    - 'App\Middleware\EventStoreMiddleware'
            messenger.bus.query:
                middleware:
                    - messenger.middleware.exactly_one_handler
                    - messenger.middleware.validation
            messenger.bus.event:
                middleware:
                    - messenger.middleware.allow_no_handler
                    - messenger.middleware.validation
```

Multiple busses

```
services:
  _defaults:
    # ...

  bind:
    $commandBus: '@messenger.bus.command'
    $queryBus: '@messenger.bus.query'
    $eventBus: '@messenger.bus.event'
```

```
<?php

namespace App\Message\QuaryHandler;

use App\Message\Quary\GetUserForNotification;

class GetUsersForNotificationHandler
{
    public function __invoke(GetUserForNotification $query)
    {
        $users = $this->getUsers($query->getSubject());

        → return $users;
    }

    private function getUsers(string $subject)
    {
        // TODO Decide upon $subject
        return ['tobias', 'maria', 'samuel'];
    }
}
```

Send events from handler

```
class SendNotificationHandler
{
    private $eventBus;

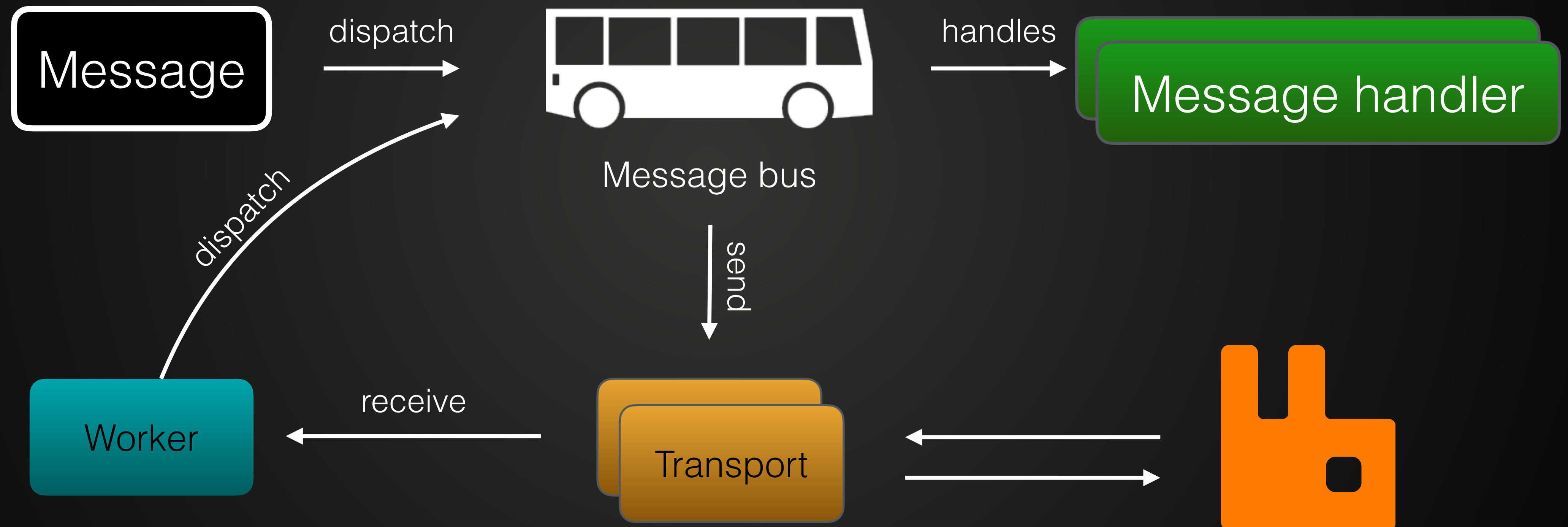
    public function __construct(MessageBusInterface $eventBus)
    {
        $this->eventBus = $eventBus;
    }

    public function __invoke(SendNotification $message)
    {
        foreach ($message->getUsers() as $user) {
            echo "Send notification to... ".$user."\n";
        }

        $this->eventBus->dispatch(new NotificationSent());
    }
}
```

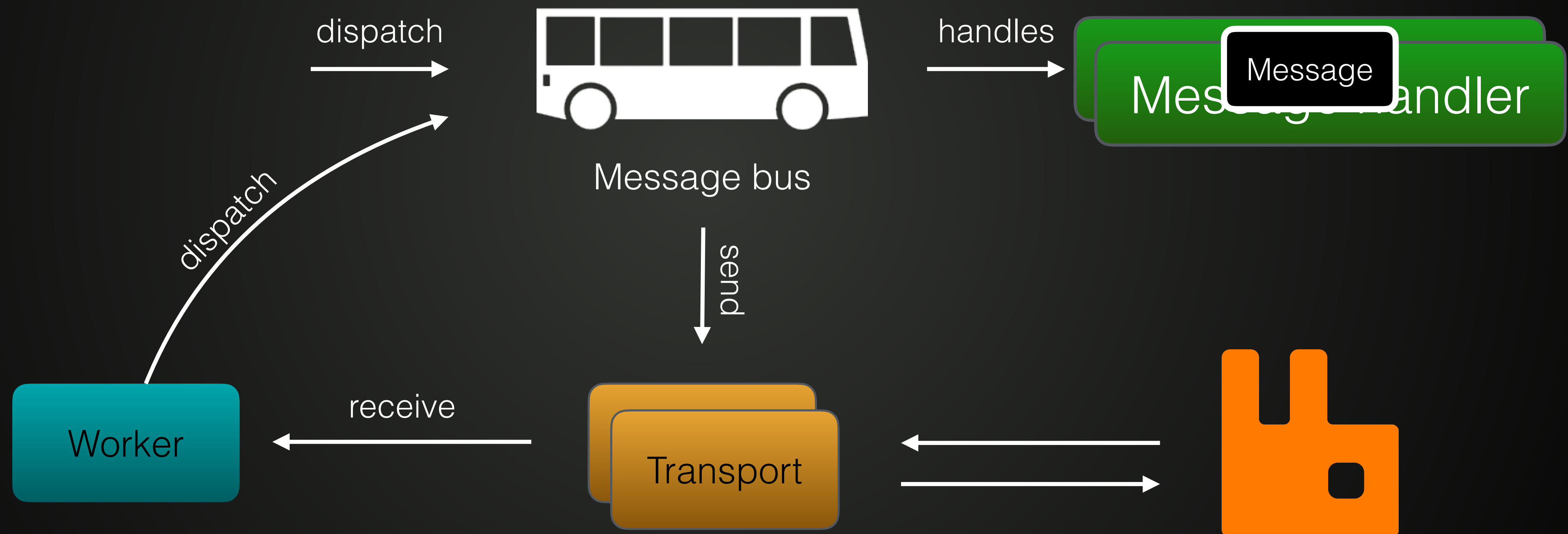
Handle failure

Requeue



Error

Requeue



Create new entities

```
use Symfony\Component\Validator\Constraints as Assert;
```

@tobiasnyholm

```
class CreateUser
```

```
{
```

```
    /**
```

```
     * @Assert\Uuid()
```

```
     * @Assert\NotBlank()
```

```
     */
```

```
private $uuid;
```

```
    /**
```

```
     * @Assert\NotBlank()
```

```
     * @Assert\Length(min=3)
```

```
     */
```

```
private $username;
```

```
public function __construct(string $uuid, string $username)
```

```
{
```

```
    $this->uuid = $uuid;
```

```
    $this->username = $username;
```

```
}
```

```
public function getUuid(): string
```

```
{
```

```
    return $this->uuid;
```

```
}
```

```
public function getUsername(): string
```

```
{
```

```
    return $this->username;
```

```
<?php

namespace App\Controller;

use App\Message\SendNotification;
use Symfony\Component\Messenger\MessageBusInterface;
// ...

class UserController
{
    public function create(MessageBusInterface $bus, Request $request)
    {
        $uuid = (string) Uuid::uuid4();
        $bus->dispatch(new CreateUser($uuid, $request->request->get('username', '')));

        return new Response('See /api/user/' . $uuid);
    }
}
```



```
use Symfony\Component\Validator\Constraints as Assert;
```

@tobiasnyholm

```
class CreateUser
```

```
{
```

```
    /**
```

```
     * @Assert\Uuid()
```



```
     * @Assert\NotBlank()
```



```
    */
```

```
private $uuid;
```

```
    /**
```

```
     * @Assert\NotBlank()
```



```
     * @Assert\Length(min=3)
```



```
    */
```

```
private $username;
```

```
public function __construct(string $uuid, string $username)
```

```
{
```

```
    $this->uuid = $uuid;
```

```
    $this->username = $username;
```

```
}
```

```
public function getUuid(): string
```

```
{
```

```
    return $this->uuid;
```

```
}
```

```
public function getUsername(): string
```

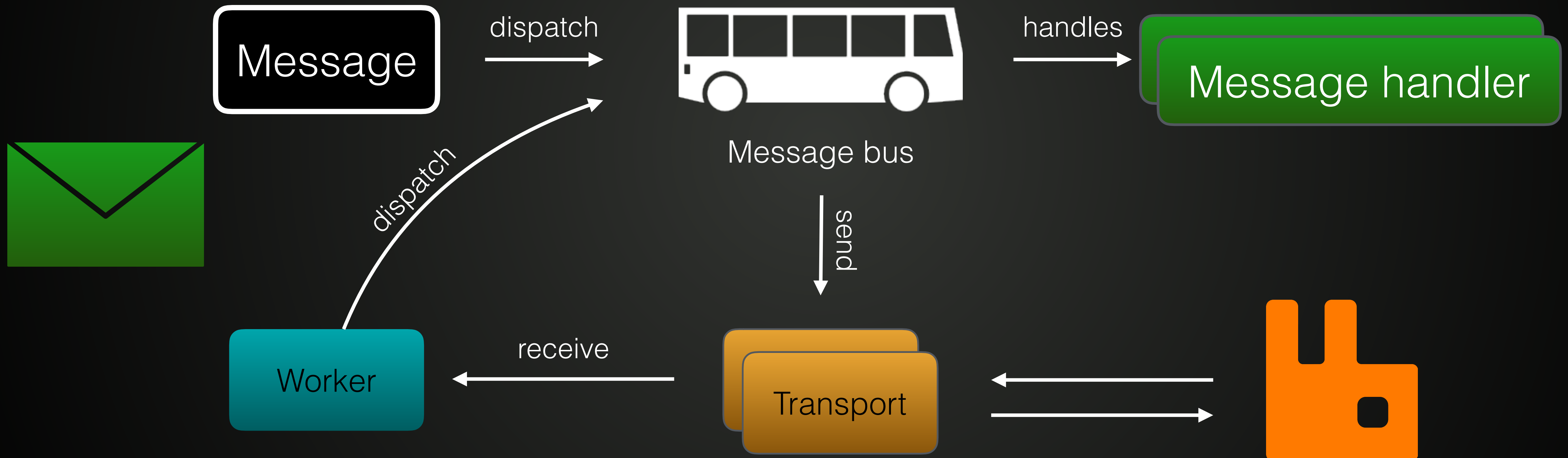
```
{
```

```
    return $this->username;
```


Questions?

<https://joind.in/talk/bf257>

Envelope



Questions?

<https://joind.in/talk/bf257>

