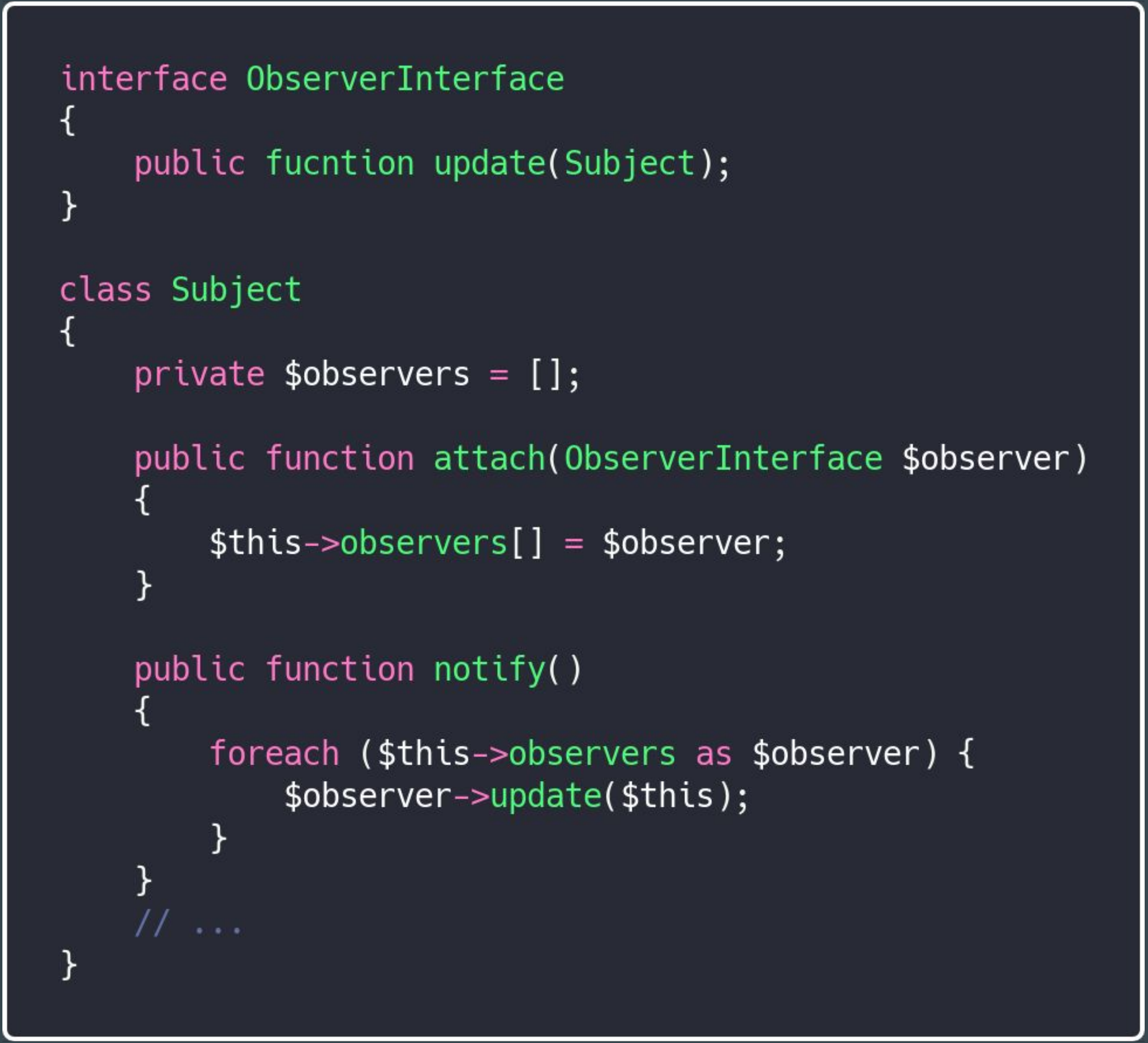
Events

Goals

1. Extend the service without inheritance
2. Horizontal extendability

Observer pattern



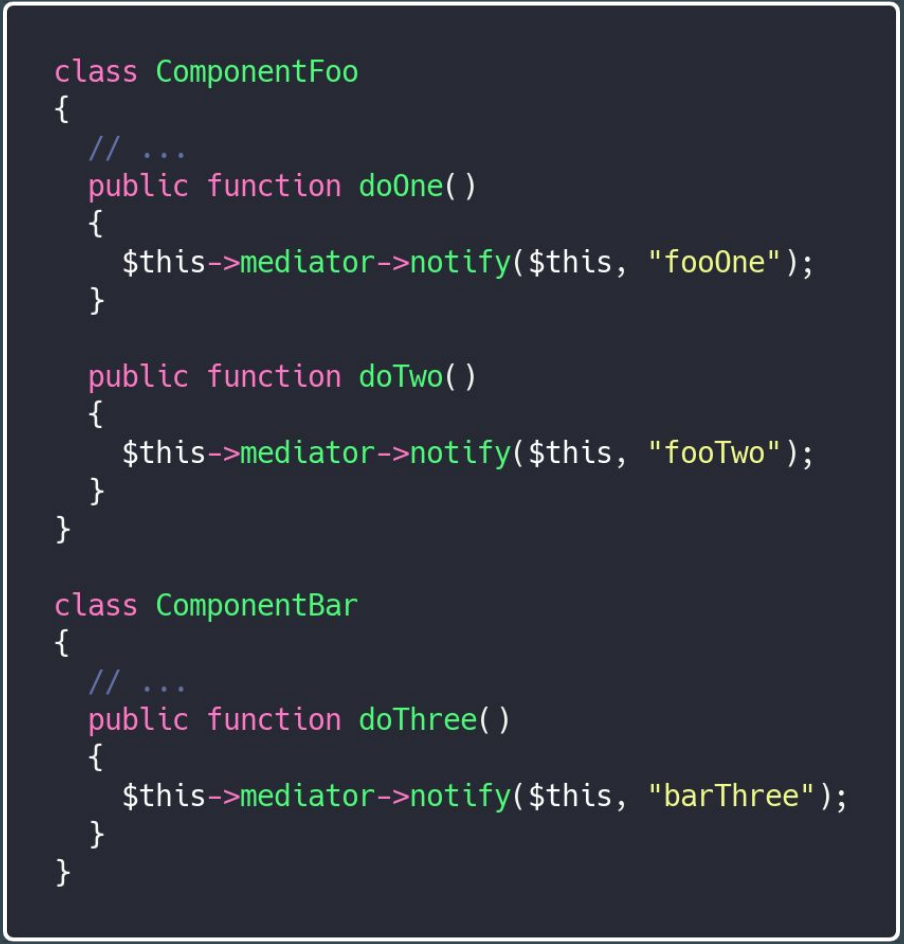
```
interface ObserverInterface
{
    public fucntion update(Subject);
}

class Subject
{
    private $observers = [];

    public function attach(ObserverInterface $observer)
    {
        $this->observers[] = $observer;
    }

    public function notify()
    {
        foreach ($this->observers as $observer) {
            $observer->update($this);
        }
    }
    // ...
}
```

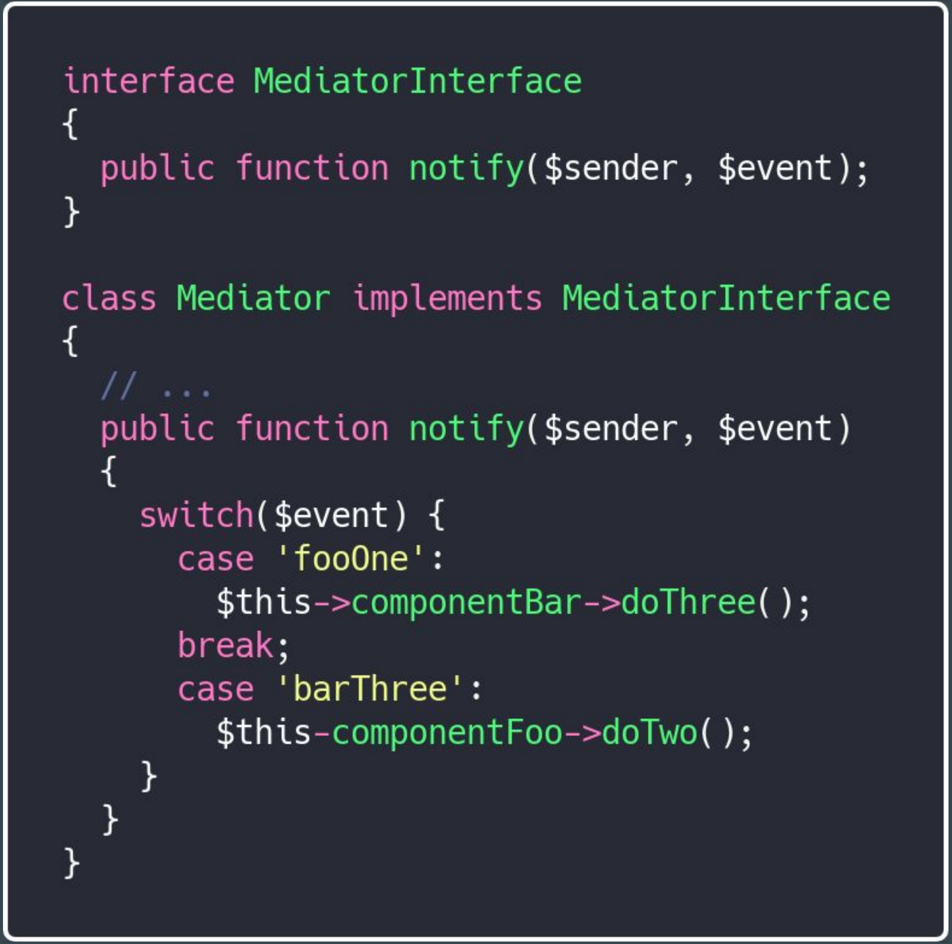
Mediator pattern



```
class ComponentFoo
{
  // ...
  public function doOne()
  {
    $this->mediator->notify($this, "fooOne");
  }

  public function doTwo()
  {
    $this->mediator->notify($this, "fooTwo");
  }
}

class ComponentBar
{
  // ...
  public function doThree()
  {
    $this->mediator->notify($this, "barThree");
  }
}
```



```
interface MediatorInterface
{
  public function notify($sender, $event);
}

class Mediator implements MediatorInterface
{
  // ...
  public function notify($sender, $event)
  {
    switch($event) {
      case 'fooOne':
        $this->componentBar->doThree();
      break;
      case 'barThree':
        $this-componentFoo->doTwo();
    }
  }
}
```

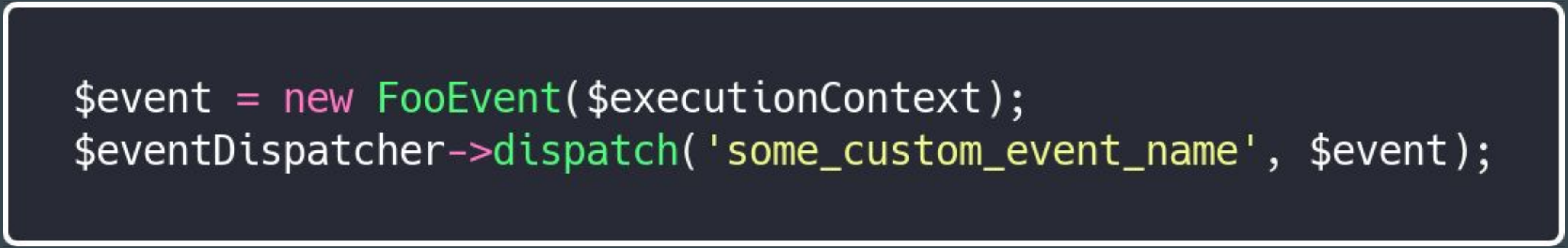
Events

- Event
- GenericEvent
- FormEvent
- OnFlushEventArgs, EventArgs

EventDispatcher

- Symfony EventDispatcher
- Doctrine EventManager

Dispatching events

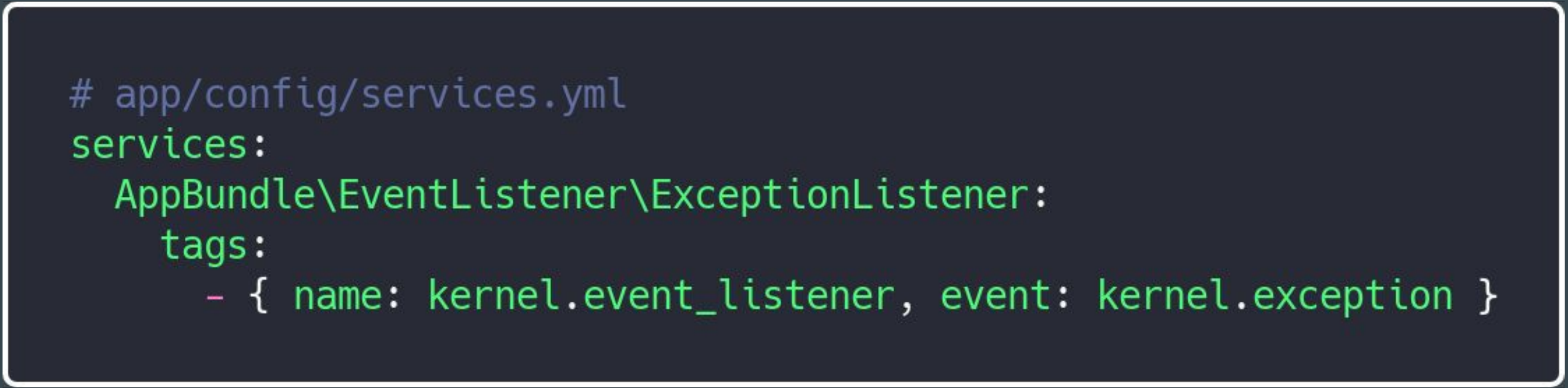


```
$event = new FooEvent($executionContext);
$eventDispatcher->dispatch('some_custom_event_name', $event);
```

Examples:

- Data is the context: AbstractEngine
- Service is the context: UnitOfWork
- Organizing events: FormEvents

Listening Events



```yaml
# app/config/services.yml
services:
  AppBundle\EventListener\ExceptionListener:
    tags:
      - { name: kernel.event_listener, event: kernel.exception }
```

- Default **method** will be onKernelException
- Priority
  - default is 0
  - 255 to -255
  - higher is earlier
- Example: ProductVisibilityRestrictionListener, services.yml

Event Subscriber

- Symfony EventSubscriberInterface
- Doctrine EventSubscriber

Performance Issues!

Overriding Listeners

- $event->stopPropagation()
- Override class name parameter
- Create a service with the same name
- Remove original listener with the CompilerPass
- Replace original listener with the CompilerPass

Debugging

- debug:event-dispatcher
- debug:event-dispatcher kernel.exception