



Better Code Design in
PHP

ConFoo Montreal

Anna Filina | February 2025

Agenda

 Common pitfalls

 Project

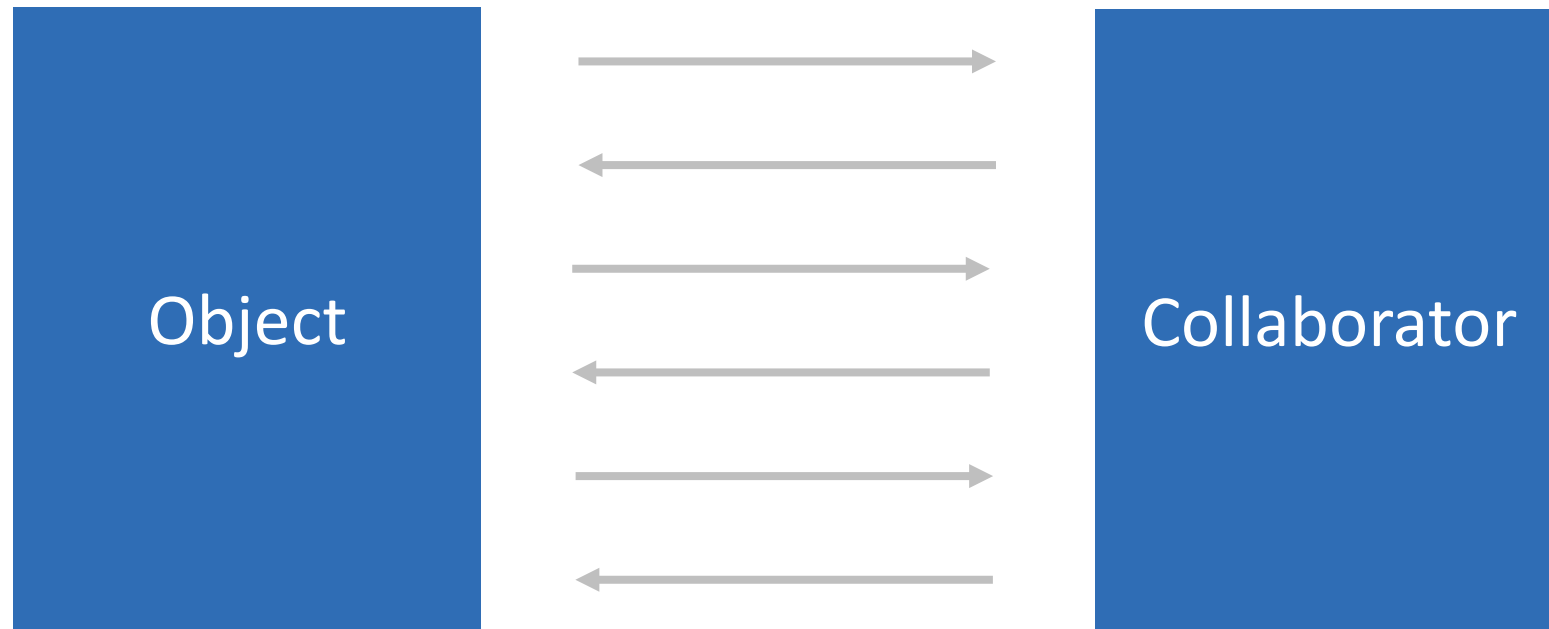
Anna Filina



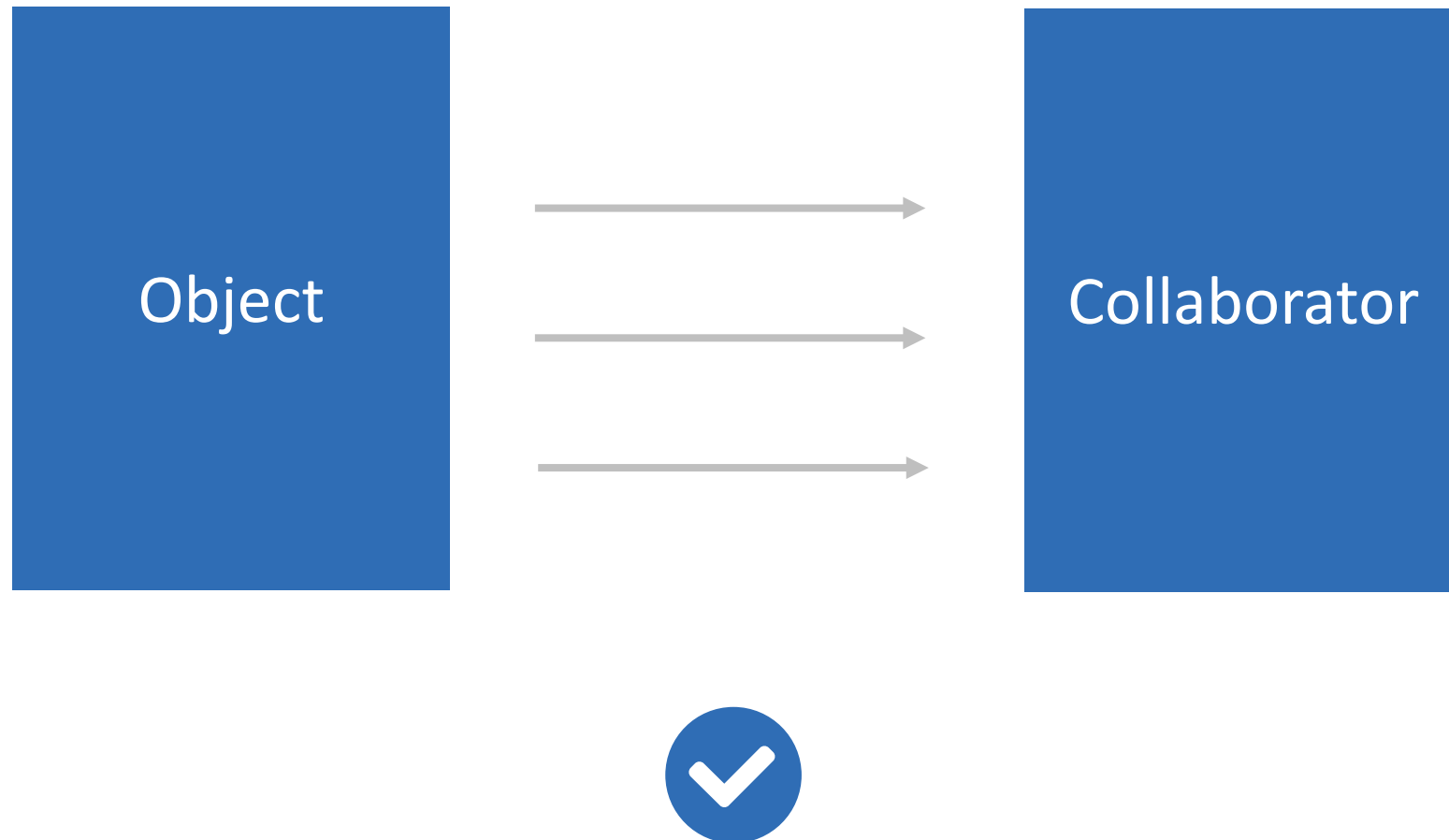
- Coding since 1997
- PHP, Java, C#, etc.
- Zend Professional Services
- I fix legacy
- I write tests
- I teach developers

Common Pitfalls

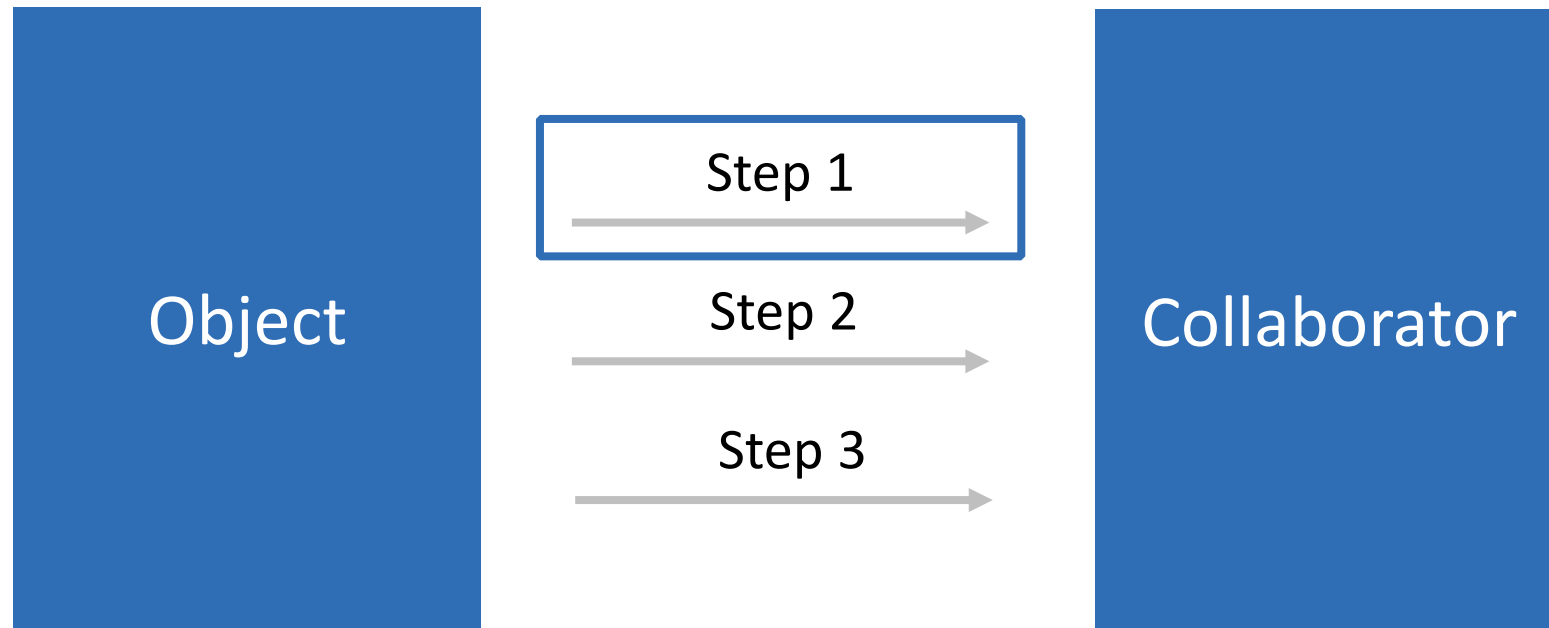
Back-and-Forth



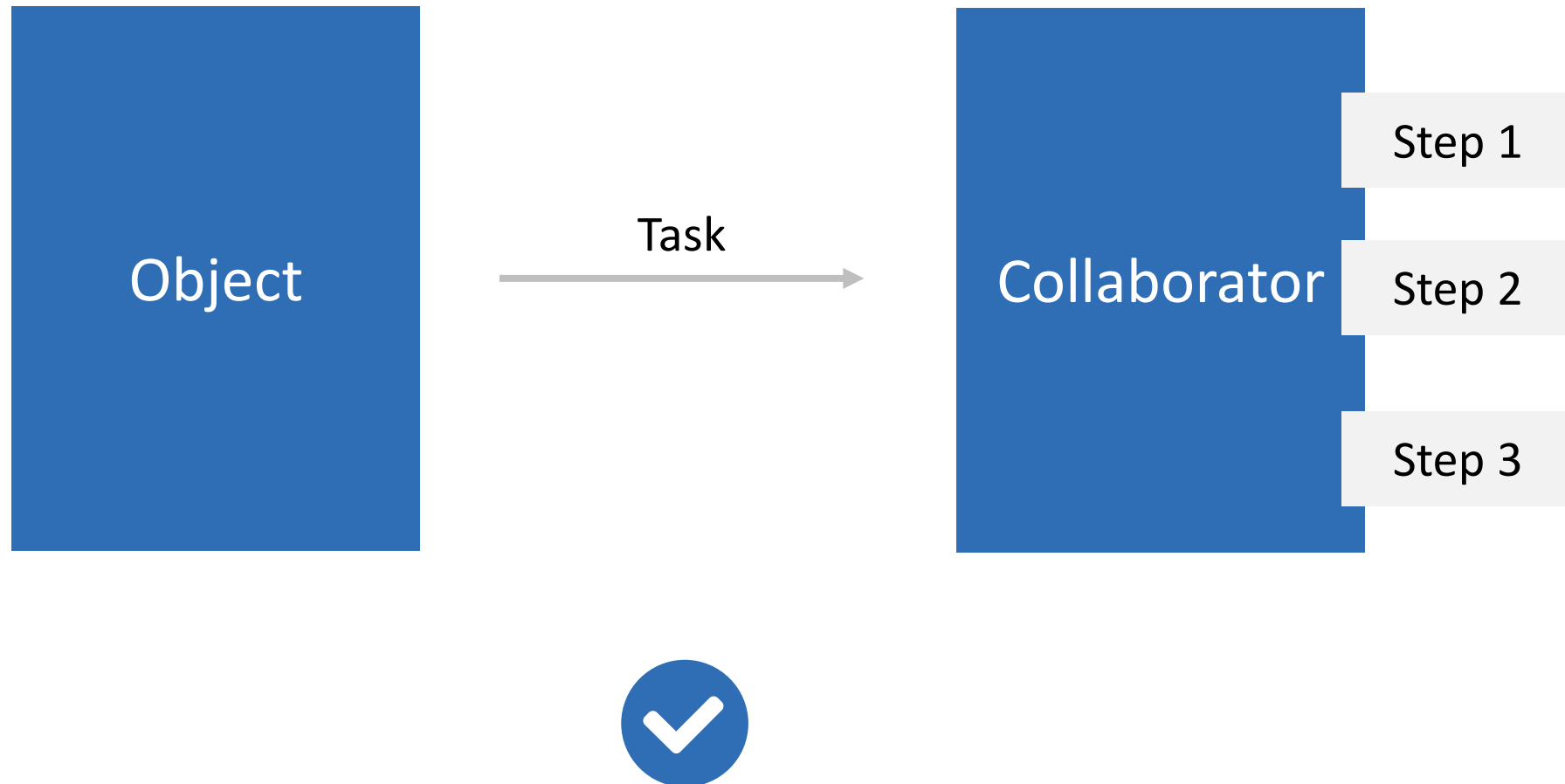
One-Way



Chatty



Single Call



What Drives My Design

- Avoid complexity until I need it.
- Can I understand the *business* use case?

Calculating Invoice Taxes

Concise,
business
language

vs

Verbose, low-
level data
manipulation

Hard to Read

```
$list = $this->repository->fetch(['column1' => $value . $seq . $suffix]);  
$filteredList = $this->filter->removeType($list, 2);  
$sortedList = $this->sorter->sort($filteredList, ['integers' => true]);  
$this->database->persist('table1', $sortedList);
```

Objects should speak business,
not technical.

More Readable

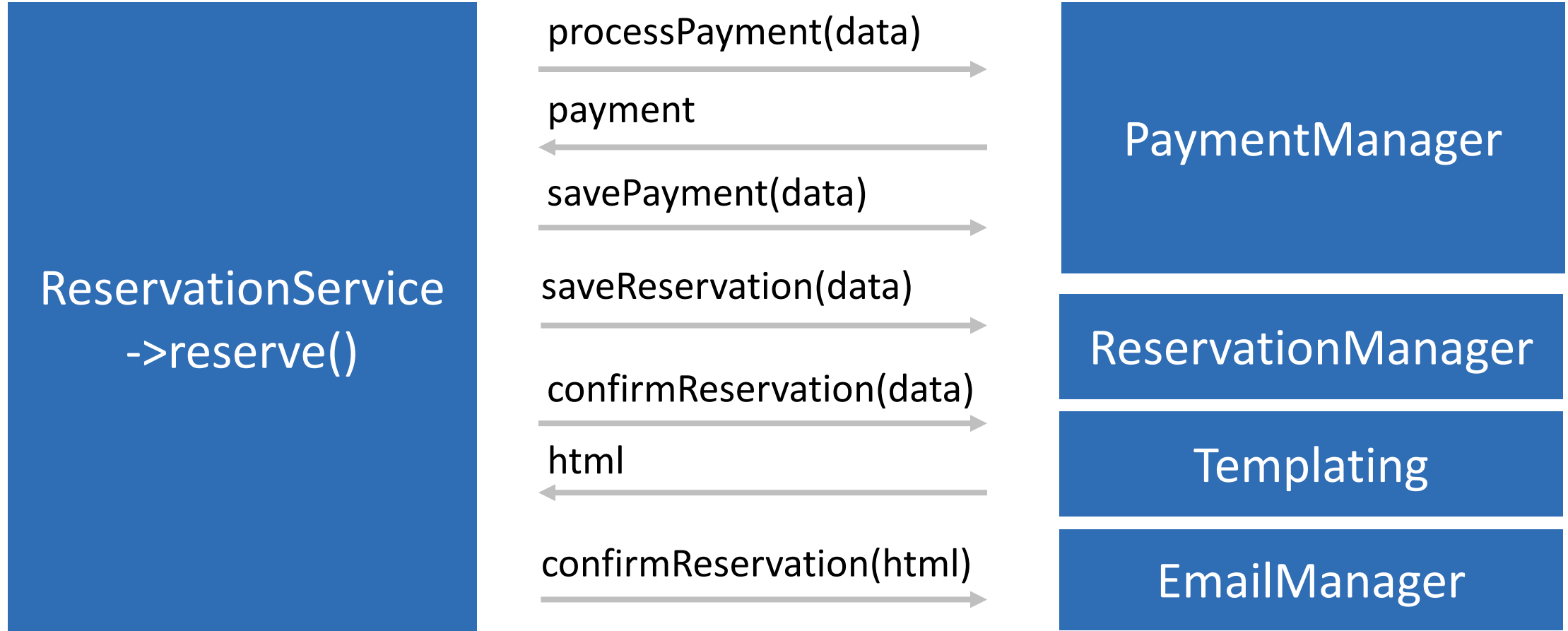
```
$taxTotals = $this->taxes  
->calculateForLineItems($order->lineItems);
```

Project

Short-Term Rental Marketplace

- Search properties for rent by city, date and number of guests.
- Reserve & pay.

Reservation



Reservation

ReservationService
->reserve()

processPayment(data)

payment

savePayment(data)

saveReservation(data)

confirmReservation(data)

html

confirmReservation(html)

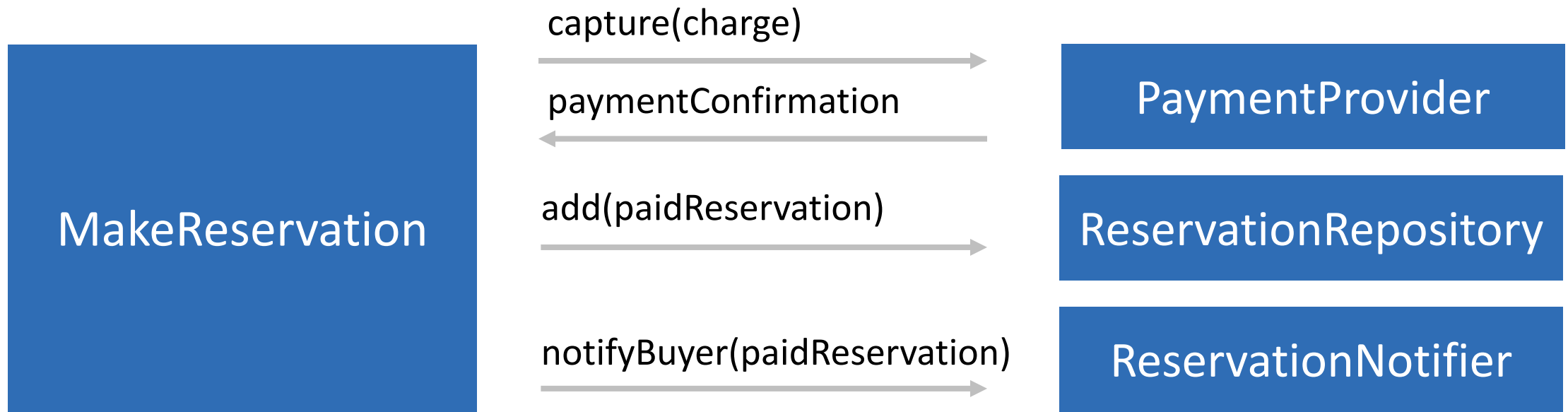
PaymentManager

ReservationManager

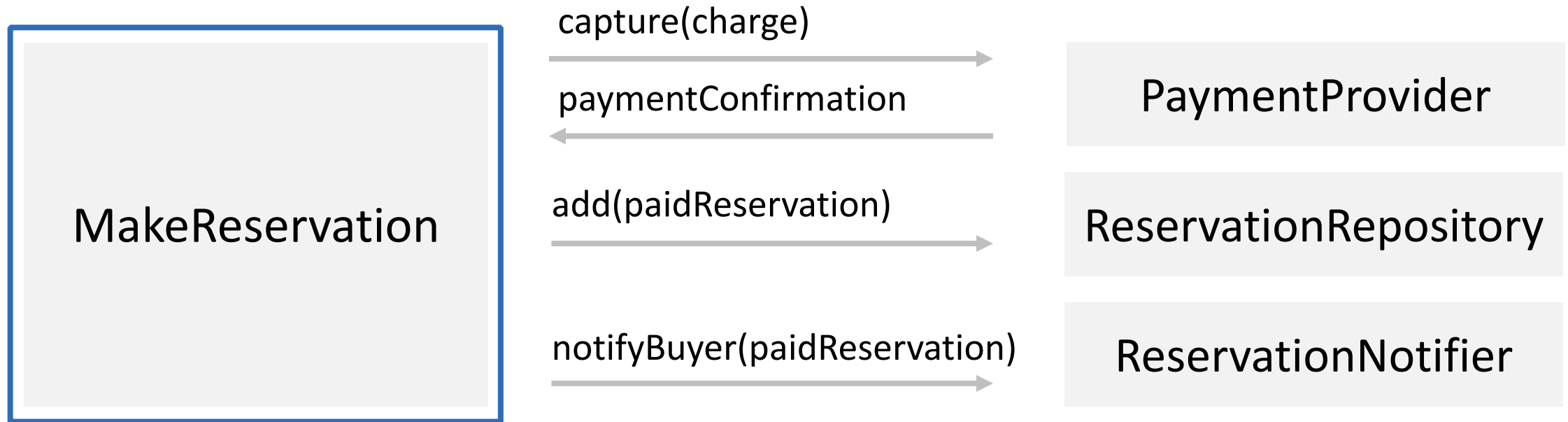
Templating

EmailManager

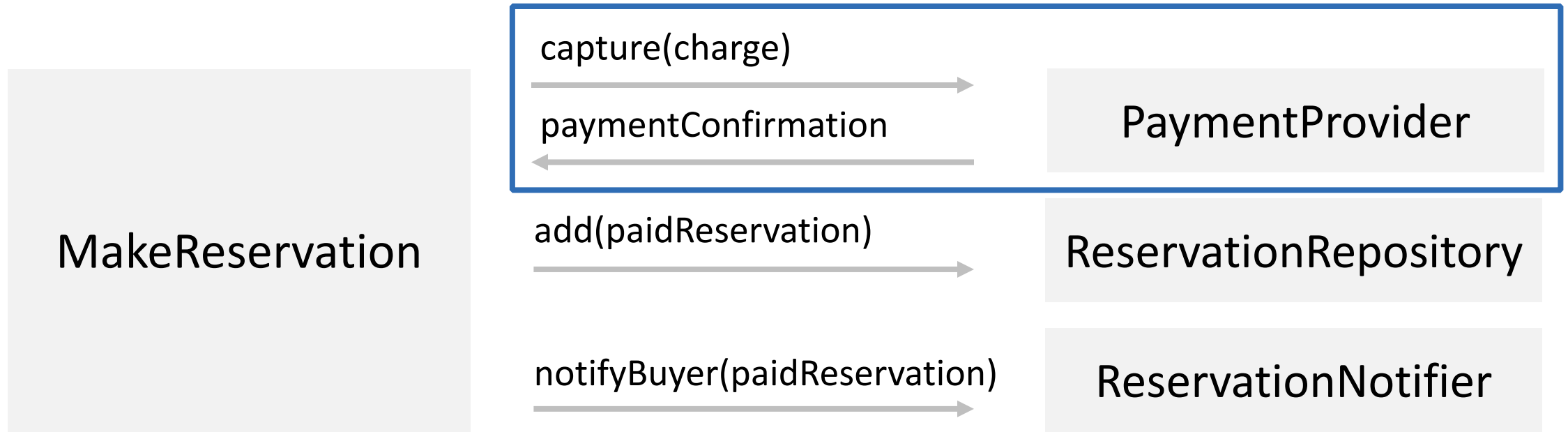
Business Language



Business Language

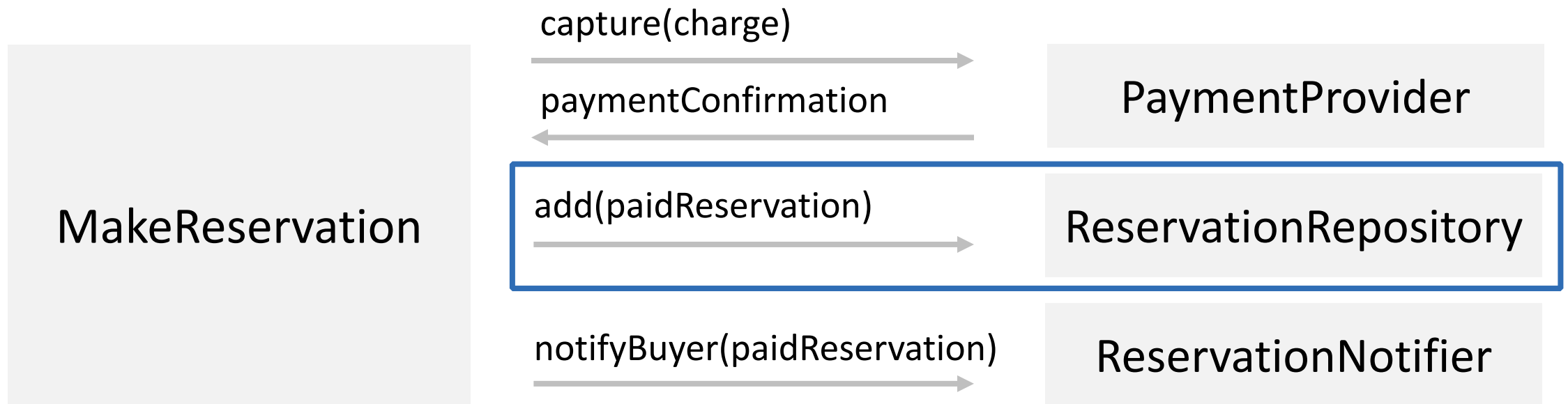


Business Language

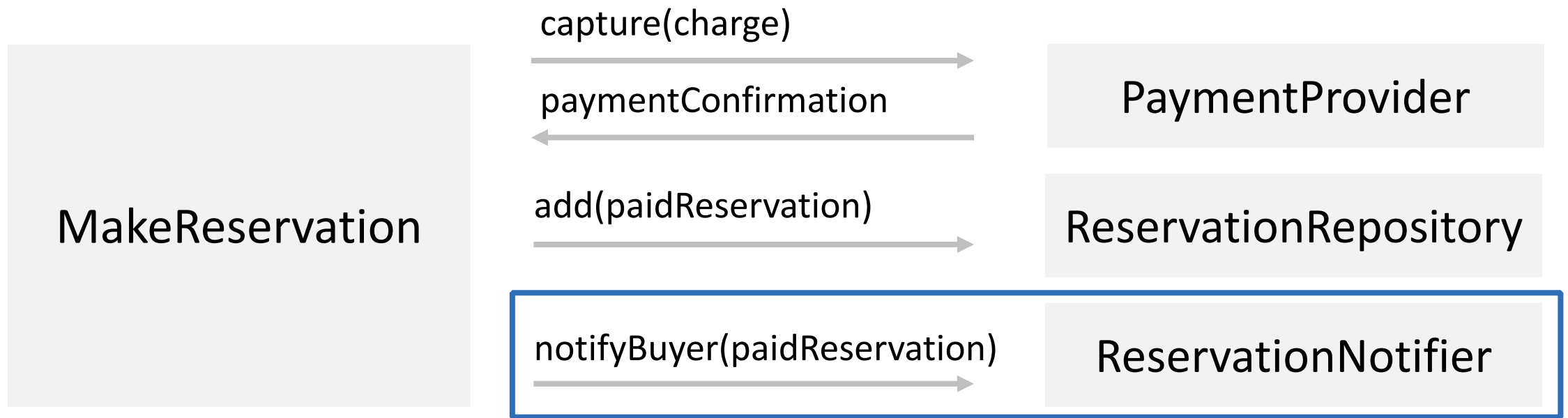


Example: 4ZLWJM

Business Language



Business Language



Good design needs no diagrams.

Start in English

Scenario: Guest can make a reservation

Given I selected dates for a property

When I submit valid payment details

Then I should receive a reservation confirmation via e-mail

Search Scenario

Scenario: Find property if the city, guests and dates match

Given a property "Lviv for 4" in Lviv with 3 beds for 4 guests

And the property is available on Jan 1 to Jan 31

When I search a property for 2 adults in Lviv on Jan 1 to Jan 7

Then I should see "Lviv for 4" in the results

Simple Code

- No need for framework.
- No need to design the database.
- No need to have a UI.
- Code the business rules.

From Gherkin to Code

Scenario: Find property if the city, guests and dates match

Given a property "Lviv for 4" in Lviv with 3 beds for 4 guests

And the property is available on Jan 1 to Jan 31

When I search a property for 2 adults in Lviv on Jan 1 to Jan 7

Then I should see "Lviv for 4" in the results

From Gherkin to Design

When I search a property for 2 adults in Lviv on Jan 1 to Jan 7
Then I should see "Lviv for 4" in the results



First Code

```
$response = [  
  [  
    'name' => 'Lviv for 4',  
    'price' => '$345 CAD',  
  ],  
];
```

Request/Response



Handle Function

```
public function handle(array $request): array
{
    return [
        [
            'name' => 'Lviv for 4',
            'price' => '$345 CAD',
        ]
    ];
}
```

Handler Class

When I search a property [...]

```
final readonly class PropertySearchHandler
{
    public function handle(array $request): array
    {
        return [
            [
                'name' => 'Lviv for 4',
                'price' => '$345 CAD',
            ]
        ];
    }
}
```


Execute

```
// public/index.php
```

```
$url = parse_url($_SERVER['REQUEST_URI']);
```

```
$response = match ($url['path']) {  
    '/' => (new PropertySearchHandler())->handle($_GET),  
};
```

```
header('Content-Type: application/json');  
echo json_encode($response);
```

PHP Server

```
php -S localhost:8000 -t public/
```

Output

The screenshot shows a web browser's developer console with the following elements:

- Address Bar:** Contains navigation icons (back, forward, refresh) and the address `localhost:8000`.
- Tab Bar:** Shows three tabs: `JSON` (selected), `Raw Data`, and `Headers`.
- Toolbar:** Includes buttons for `Save`, `Copy`, `Collapse All`, `Expand All`, and a `Filter JSON` dropdown.
- JSON Data:** The console displays a JSON object with the following structure:

```
{  "0": {    "name": "Lviv for 4",    "price": "$345 CAD"  }}
```

**“A complex system that works is
invariably found to have evolved from a
simple system that worked.”**

”

— John Gall

Classes

```
return [  
    new Property('Lviv 4', '$345 CAD'),  
];
```

Too reusable becomes unusable.

Property

```
final readonly class Property
{
    public function __construct(
        public string $name,
        public string $price,
    )
    {
    }
}
```

Usage

```
return match ($request['city']) {  
    'Lviv' => [  
        new Property('Lviv for 4', '$345 CAD'),  
    ],  
    'Odesa' => [  
        new Property('Odesa 4', '$310 CAD'),  
        new Property('Luxury Odesa 4', '$525 CAD'),  
    ],  
};
```

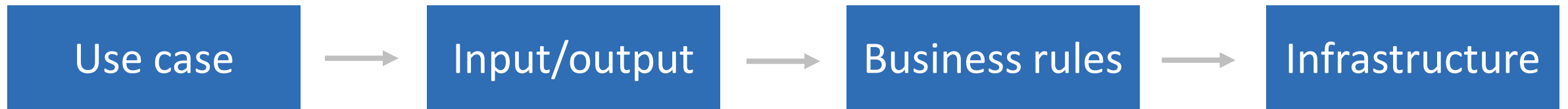

Output

The screenshot shows a web browser's developer console with the following elements:

- Address Bar:** Contains navigation icons (back, forward, refresh) and the URL `localhost:8000/?city=Odesa`.
- Console Tabs:** Three tabs are visible: **JSON** (selected), **Raw Data**, and **Headers**.
- Console Actions:** A row of buttons includes **Save**, **Copy**, **Collapse All**, **Expand All**, and a **Filter JSON** dropdown.
- JSON Output:** The console displays a JSON array with two objects. The first object is expanded, showing its properties.

```
{
  "0": {
    "name": "Odesa 4",
    "price": "$310 CAD"
  },
  "1": {
    "name": "Luxury Odesa 4",
    "price": "$525 CAD"
  }
}
```

My Coding Order



SearchCriteria

```
$criteria = new SearchCriteria($request['city']);  
return match ($criteria->city) {  
    // ...  
};
```

SearchCriteria

```
final readonly class SearchCriteria
{
    public function __construct(
        public string $city, // can be validated
    )
    {
    }
}
```

Problem with City

```
return match ($criteria->city) {  
    'Lviv' => [  
        new Property('Lviv for 4', '$345 CAD'),  
    ],  
    'Odesa' => [  
        new Property('Odesa 4', '$310 CAD'),  
        new Property('Luxury Odesa 4', '$525 CAD'),  
    ],  
};
```

Coordinates

```
$coordinates = match ($criteria->city) {  
    'Lviv' => new Coordinates(49.8397, 24.0297),  
    'Odesa' => new Coordinates(46.4825, 30.7233),  
};
```

Coordinates

```
$coordinates = match ($criteria->city) {  
    'Lviv' => new Coordinates(49.8397, 24.0297),  
    'Odesa' => new Coordinates(46.4825, 30.7233),  
};  
  
return match ($coordinates->latitude) {  
    49.8397 => [  
        new Property('Lviv for 4', '$345 CAD'),  
    ],  
    46.4825 => // ...,  
};
```

Output

The screenshot shows a web browser's developer console with the address bar displaying `localhost:8000/?city=Odesa`. The console is set to 'JSON' view. The output is an array of two objects. The first object has a `name` of `"Odesa 4"` and a `price` of `"$310 CAD"`. The second object has a `name` of `"Luxury Odesa 4"` and a `price` of `"$525 CAD"`.

```
{
  "name": "Odesa 4",
  "price": "$310 CAD"
},
{
  "name": "Luxury Odesa 4",
  "price": "$525 CAD"
}
```


Improvements

```
$coordinates = match ($criteria->city) {  
    'Lviv' => new Coordinates(49.8397, 24.0297),  
    'Odesa' => new Coordinates(46.4825, 30.7233),  
};
```

```
return match ($coordinates->latitude) {  
    49.8397 => [  
        new Property('Lviv for 4', '$345 CAD'),  
    ],  
    46.4825 => // ...,  
};
```

SQL

```
SELECT name, price, ST_DISTANCE_SPHERE  
(  
    POINT(:search_longitude, :search_latitude),  
    POINT(longitude, latitude)  
) * .001 AS `distance_in_km`  
FROM `properties`  
HAVING `distance_in_km` <= :max_distance;
```

SQL

```
SELECT name, price, ST_DISTANCE_SPHERE  
(  
    POINT(:search_longitude, :search_latitude),  
    POINT(longitude, latitude)  
) * .001 AS `distance_in_km`  
FROM `properties`  
HAVING `distance_in_km` <= :max_distance;
```

Usage

```
return $this->propertyRepository->findWithinRadius(  
    $coordinates,  
    self::DEFAULT_RANGE  
);
```

Problem with City Names

springfield|



Springfield MA, USA



Springfield MO, USA



Springfield Park Saint-Hubert, Longueuil, QC

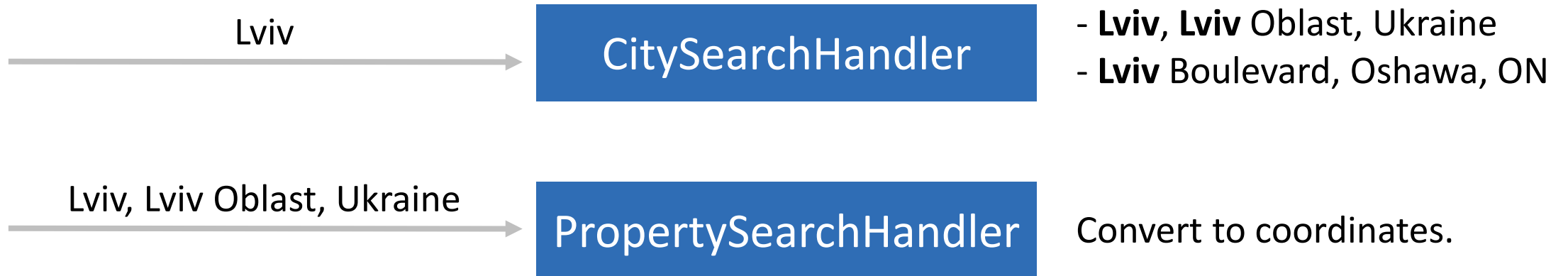


Springfield VT, USA



Springfield IL, USA

Problem



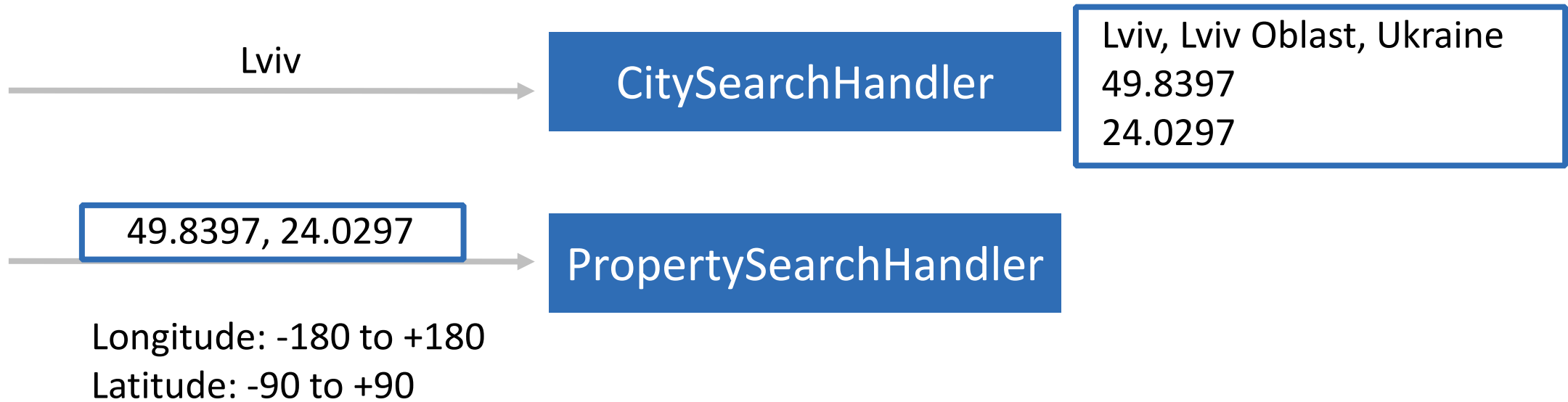
Reorganize

```
public function handle(array $request): array
{
    $searchCriteria = SearchCriteria::fromRequest($request);

    // ...

    return $this->propertyRepository->find($searchCriteria);
}
```

Problem



Reorganize

```
final readonly class SearchCriteria
{
    public function construct(
        public string $city,
    )
    {
    }
}
```

Code is simply a model of
the business rules and knowledge.

PERFORCE

Thank You

@afilina@phpc.social

@afilina

afilina@perforce.com