

Общая архитектура Yii2



Климов П.В.

YiiSoft

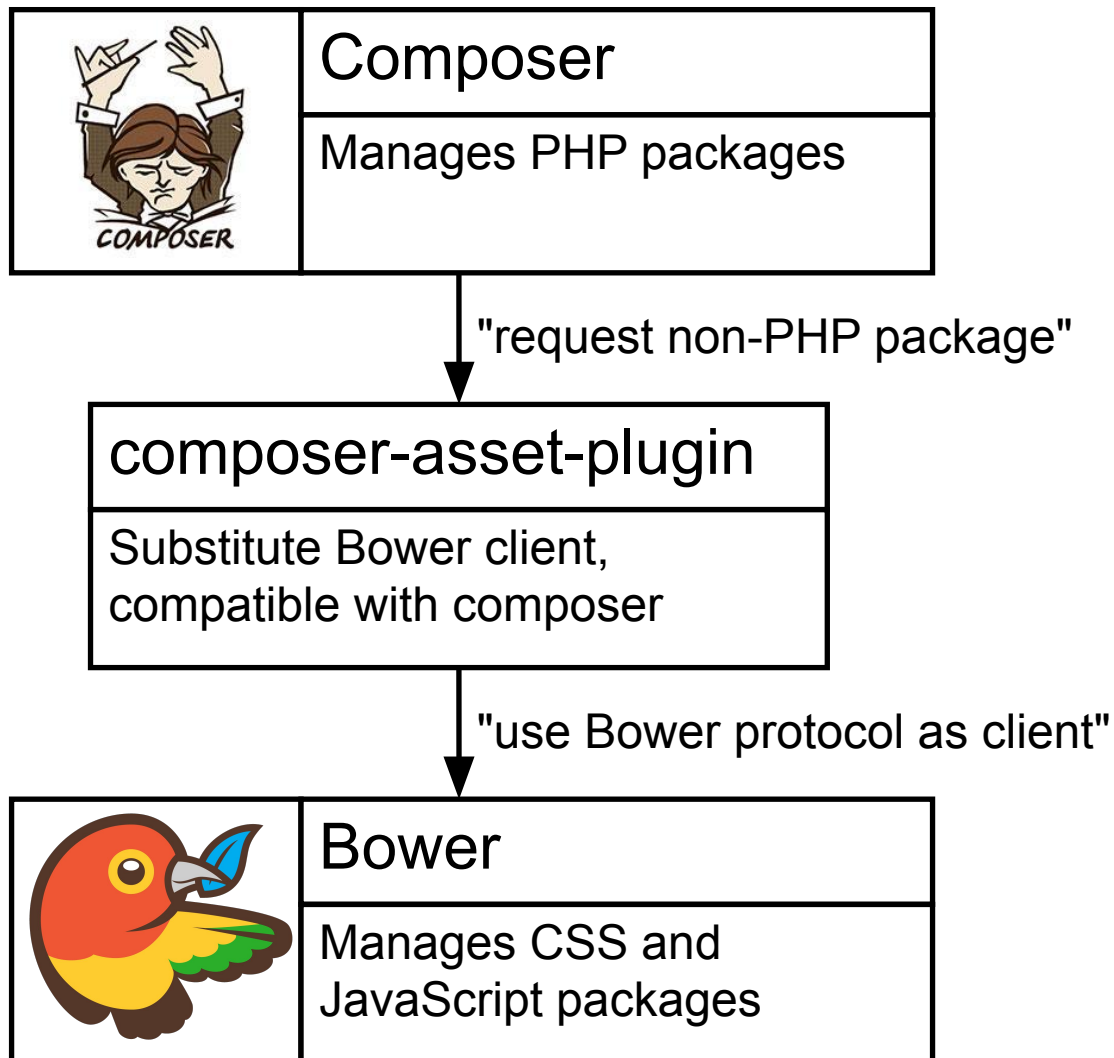
Область применения

- Построение функциональности не имеющей аналогов
- Новая интерпретация уже существующих бизнес процессов
- Стык или объединение различных областей и процессов
- Системы с повышенными требованиями к качеству

Основные характеристики

- РНР 5.4
- ООП
- Модульность
- Простота
- Высокое быстродействие

Управление зависимостями



Автозагрузка классов

// Autoload via Composer by PSR-4 :

```
$model = new yii\base\Model();
```

// Yii built-in autoload by PSR via alias:

```
Yii::setAlias('@app', '/var/www/myproject');
```

```
$someObj = new app\models\Customer();
```

// include '/var/www/myproject/models/Customer.php';

Магия в PHP

```
class Object
{
    public $publicProperty;
    private $_privateProperty;

    public function setPrivateProperty($value)
    {
        $this->_privateProperty = $value;
    }

    public function getPrivateProperty()
    {
        return $this->_privateProperty;
    }
}
```

```
class Object
{
    public function __get($propertyName)
    {
        $methodName = 'get' . $propertyName;
        if (method_exists($this, $methodName)) {
            return call_user_func([$this, $methodName]);
        } else {
            throw new Exception("Missing property {$propertyName}!");
        }
    }

    public function __set($propertyName, $value)
    {
        $methodName = 'set' . $propertyName;
        if (method_exists($this, $methodName)) {
            return call_user_func([$this, $methodName], $value);
        } else {
            throw new Exception("Missing property {$propertyName}!");
        }
    }
}
```

```
$object = new Object();
```

```
$object ->publicProperty = 'Public value';  
echo ($object->publicProperty);
```

```
$object->privateProperty = 'Private value';  
echo ($object->privateProperty);
```


Порождение объектов

```
function createObject(array $config)
{
    $className = $config['class'];
    if (empty($className)) {
        throw new Exception('Missing parameter "class"!');
    }
    unset($config['class']);

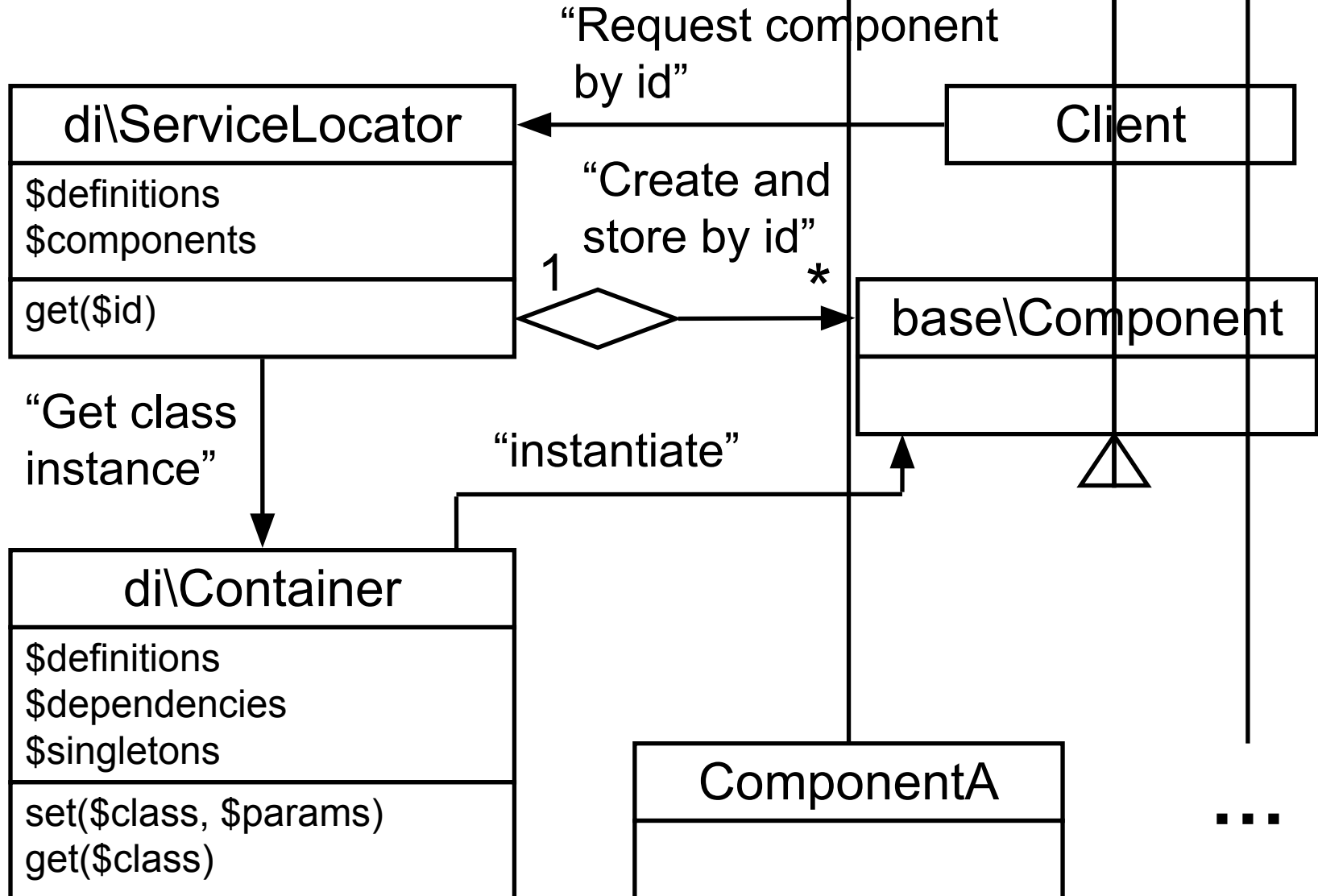
    $object = new $className();
    foreach ($config as $name=>$value) {
        $object->$name = $value; // Конфигурация
    }
    return $object;
}
```

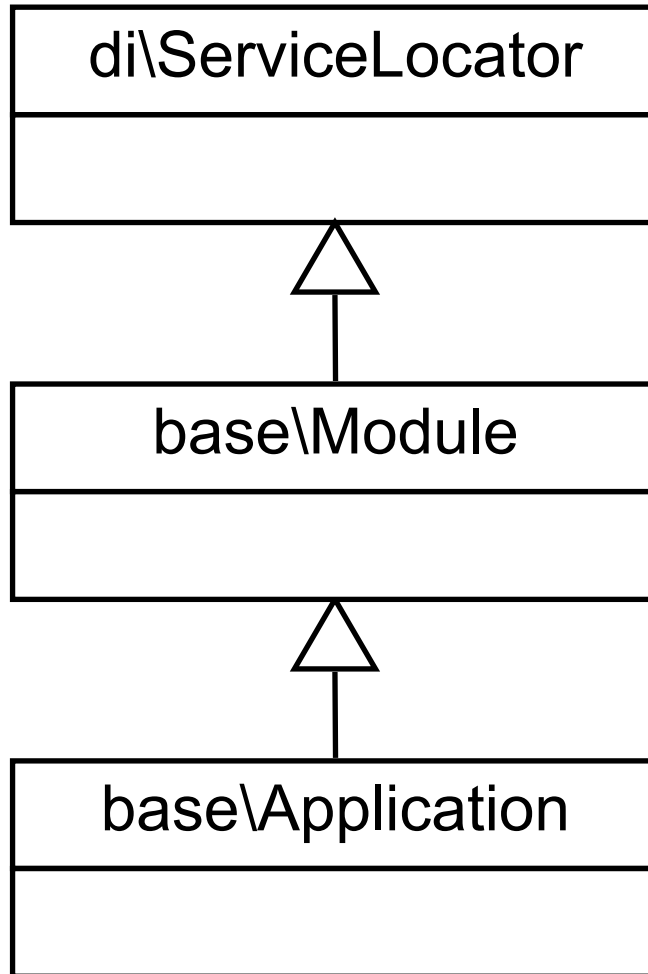
Задание любого объекта через массив:

```
$config = [  
    'class' => 'yii\web\UrlManager',  
    'enablePrettyUrl' => true,  
    'showScriptName' => false,  
    'rules' => [  
        '/' => 'site/index',  
        '<controller:\w+>/<id:\d+>*' => '<controller>/view',  
    ],  
];
```

```
$object = Yii::createObject($config);
```

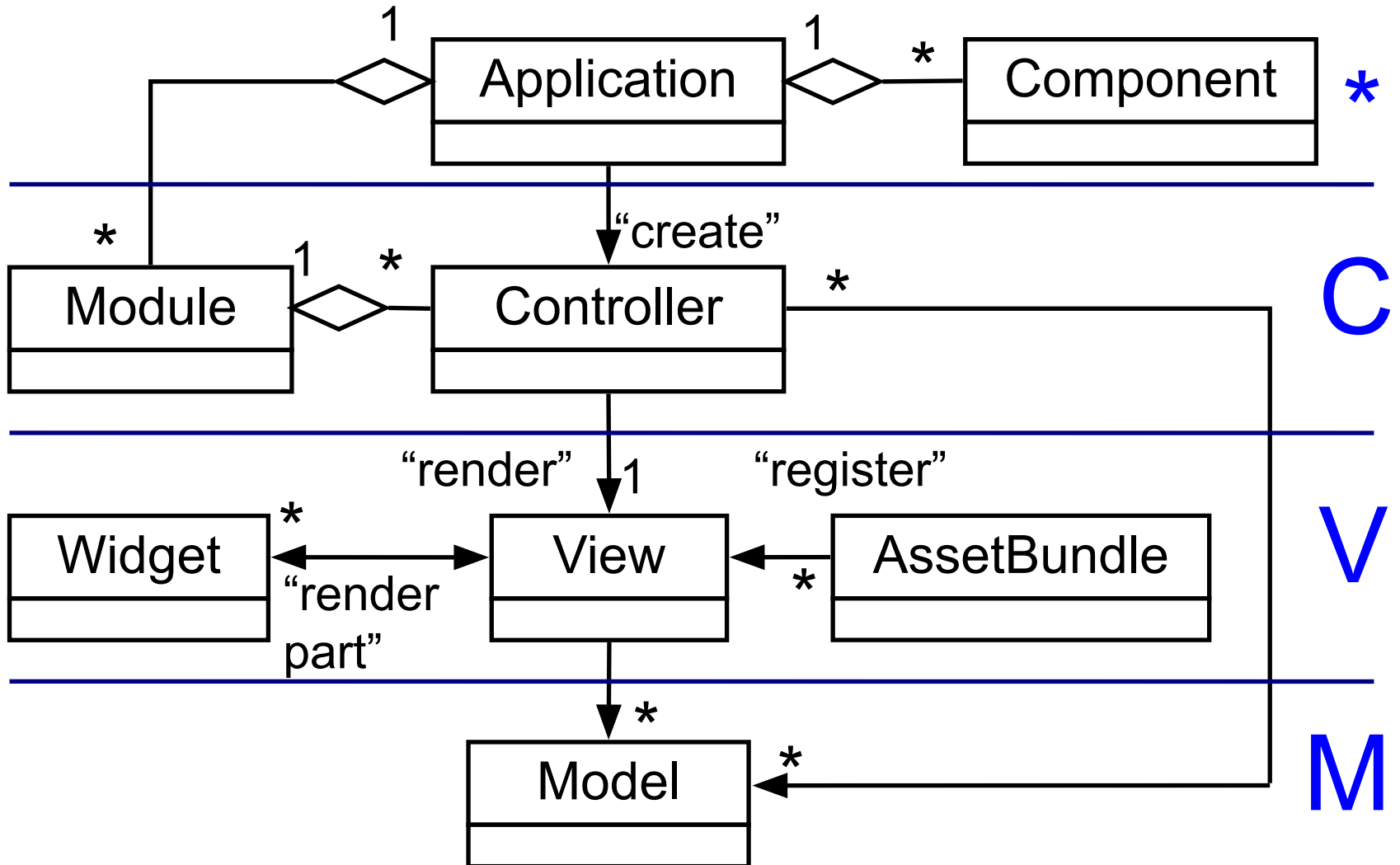
Dependency Injection (DI)



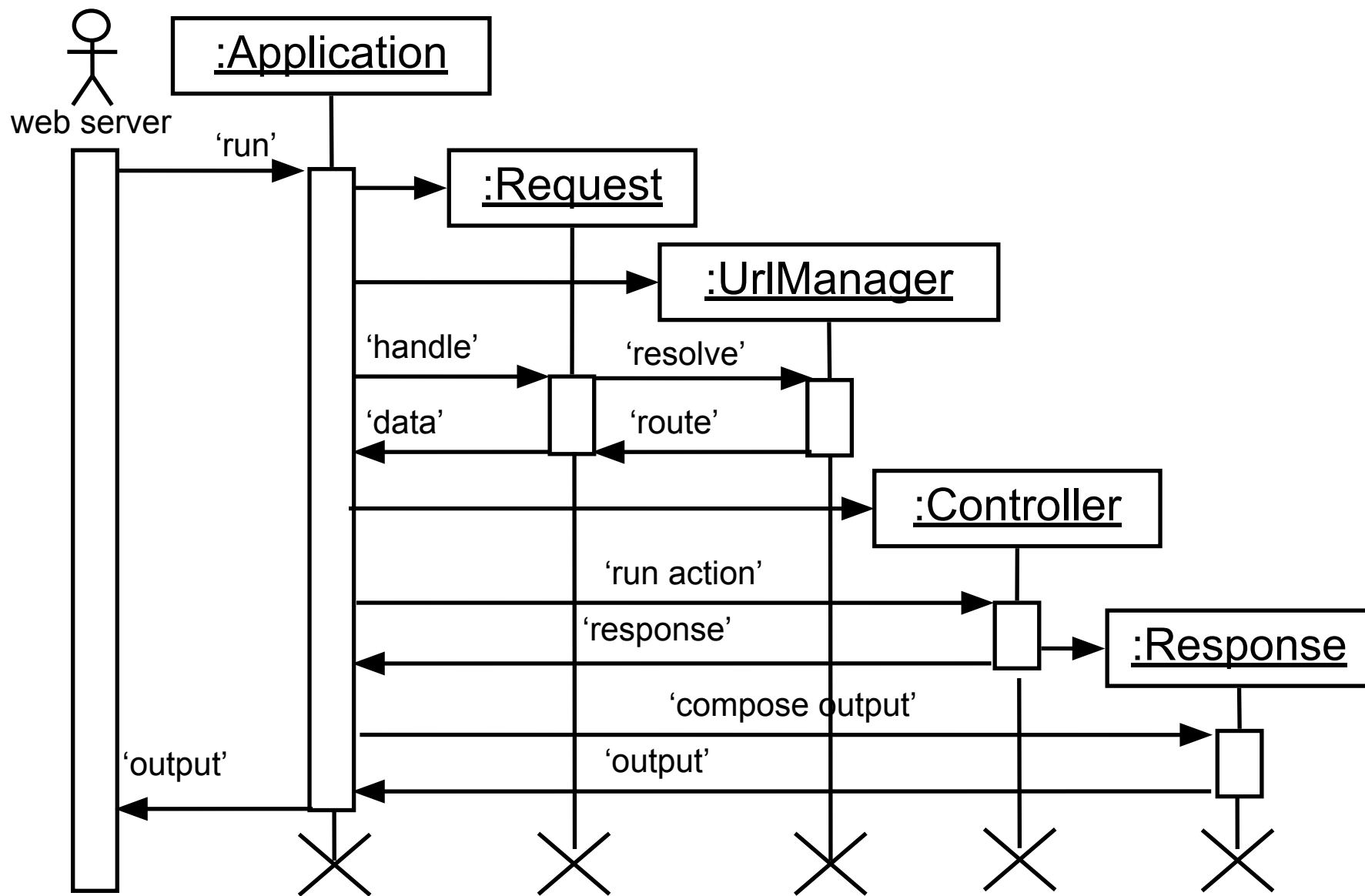


```
$config = [  
    'name' => 'My Web Application',  
    ...  
    'components' => [  
        'user' => [  
            'enableAutoLogin' => true,  
        ],  
        ...  
    ],  
];  
(new yii\web\Application($config))->run();  
...  
$application = Yii::$app;  
$user = Yii::$app->get('user');
```

MVC в Yii2



Маршрутизация web запроса



Построение отображения

```
class View extends Component
{
    public function renderFile($viewFile, $data = null)
    {
        require($viewFile);
    }
}
```

<html>

<body>

<h1>Data <?php **echo** \$data; ?></h1>

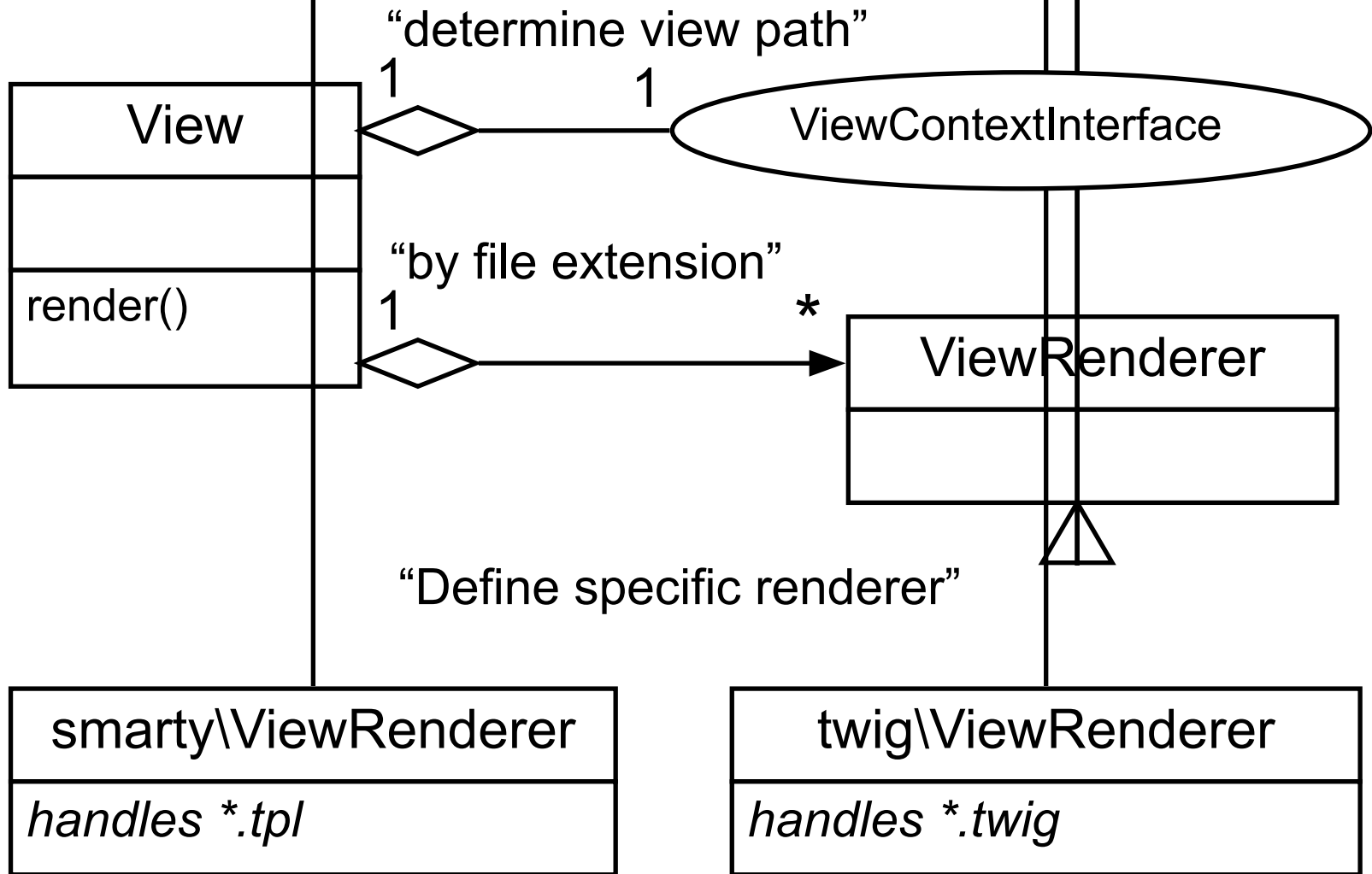
<hr />

<?php \$this->render('main_menu.php'); ?>

</body>

</html>

Структура отображения



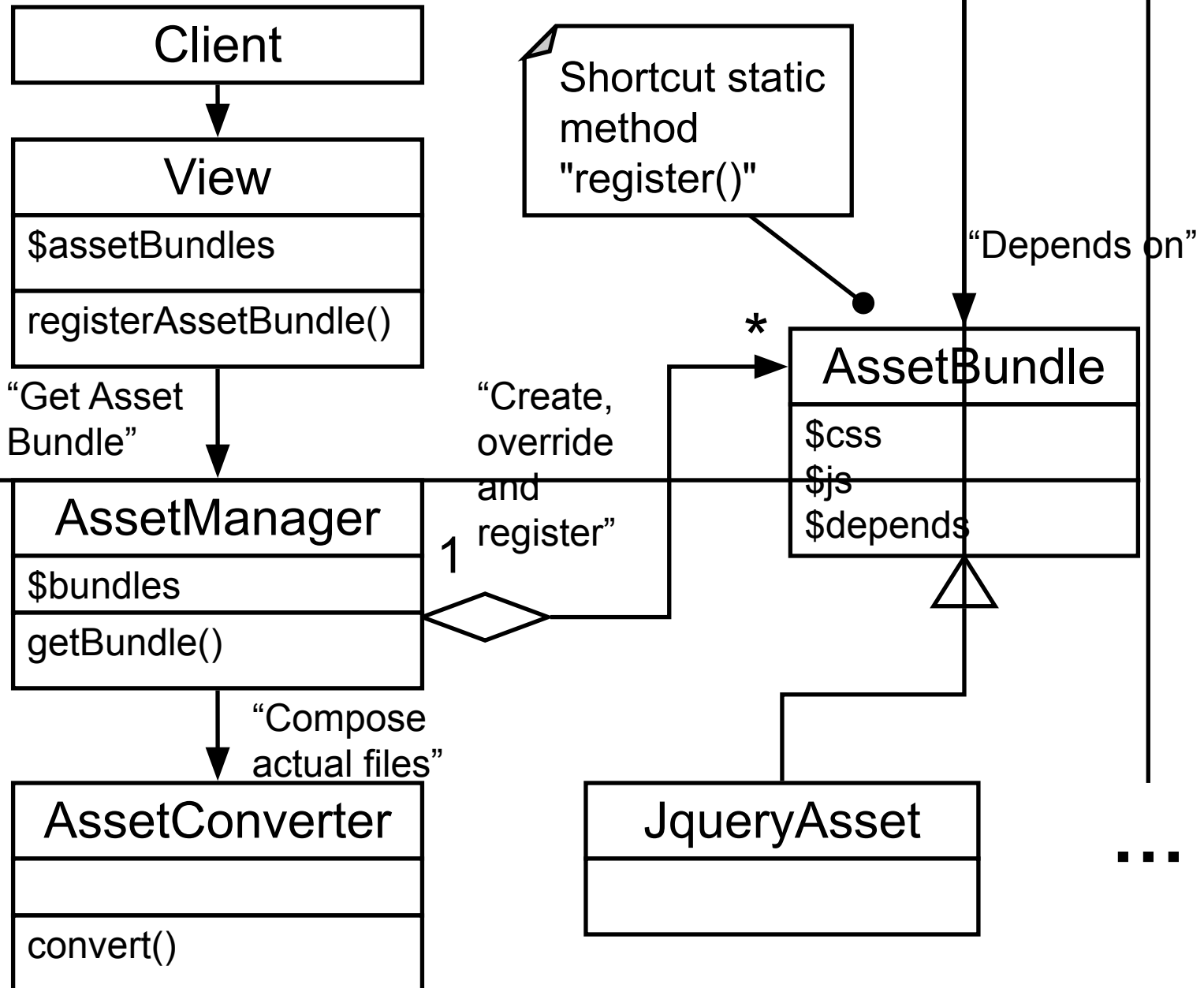
Виджет (Widget)

```
<?php echo GridView::widget([
    'dataProvider' => $dataProvider,
    'options' => ['class' => 'detail-grid-view table-responsive'],
    'filterModel' => $searchModel,
    'columns' => [
        'time',
        'level',
        'category',
        'message',
    ],
]); ?>
```

Total **10** items.

#	Time	Level	Category	Message
		<input type="text"/>	<input type="text"/>	<input type="text"/>
1	14:39:35.570	trace	yii\base\Module::getModule	Loading module: gii
2	14:39:35.570	trace	yii\base\Application::bootstrap	Bootstrap with yii\gii\Module::bootstrap()
3	14:39:35.573	trace	yii\web\UrlManager::parseRequest	Pretty URL not enabled. Using default URL pa

Asset Management



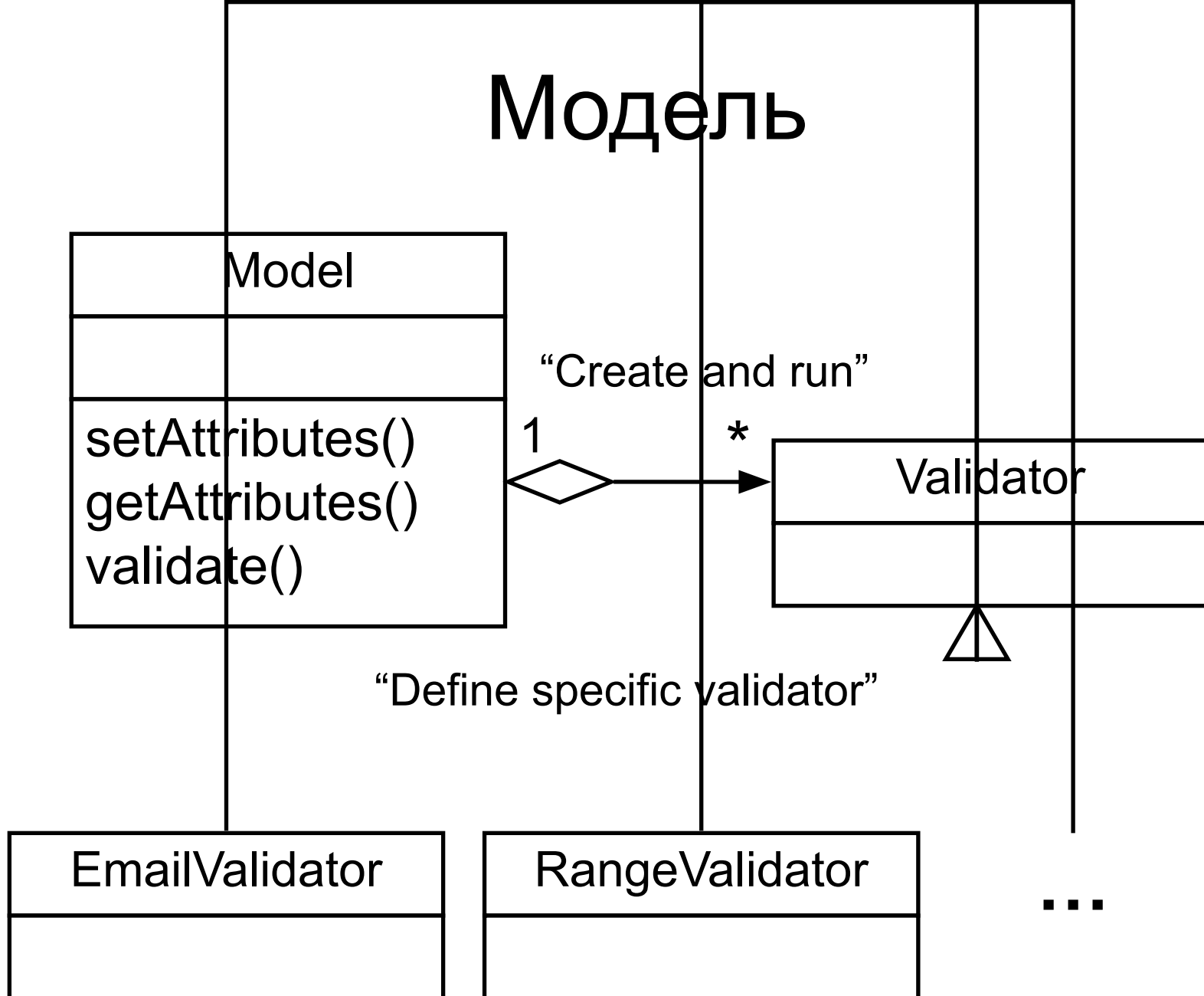
```
class YiiAsset extends AssetBundle
```

```
{  
    public $sourcePath = '@yii/assets';  
    public $js = [  
        'yii.js',  
    ];  
    public $depends = [  
        'yii\web\jQueryAsset',  
    ];  
}
```

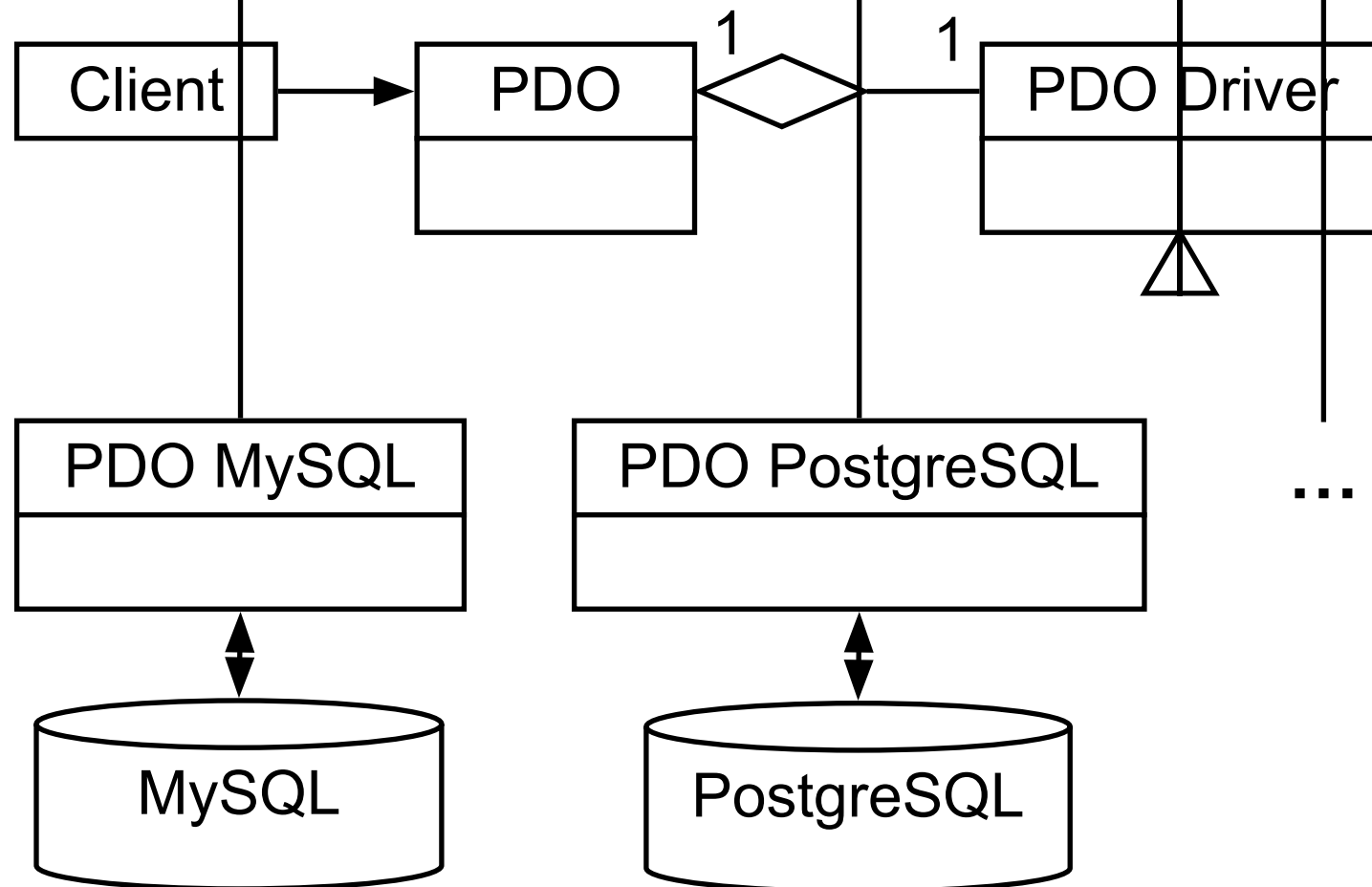
```
class JQueryAsset extends AssetBundle
```

```
{  
    public $sourcePath = '@bower/jquery/dist';  
    public $js = [  
        'jquery.js',  
    ];  
}
```

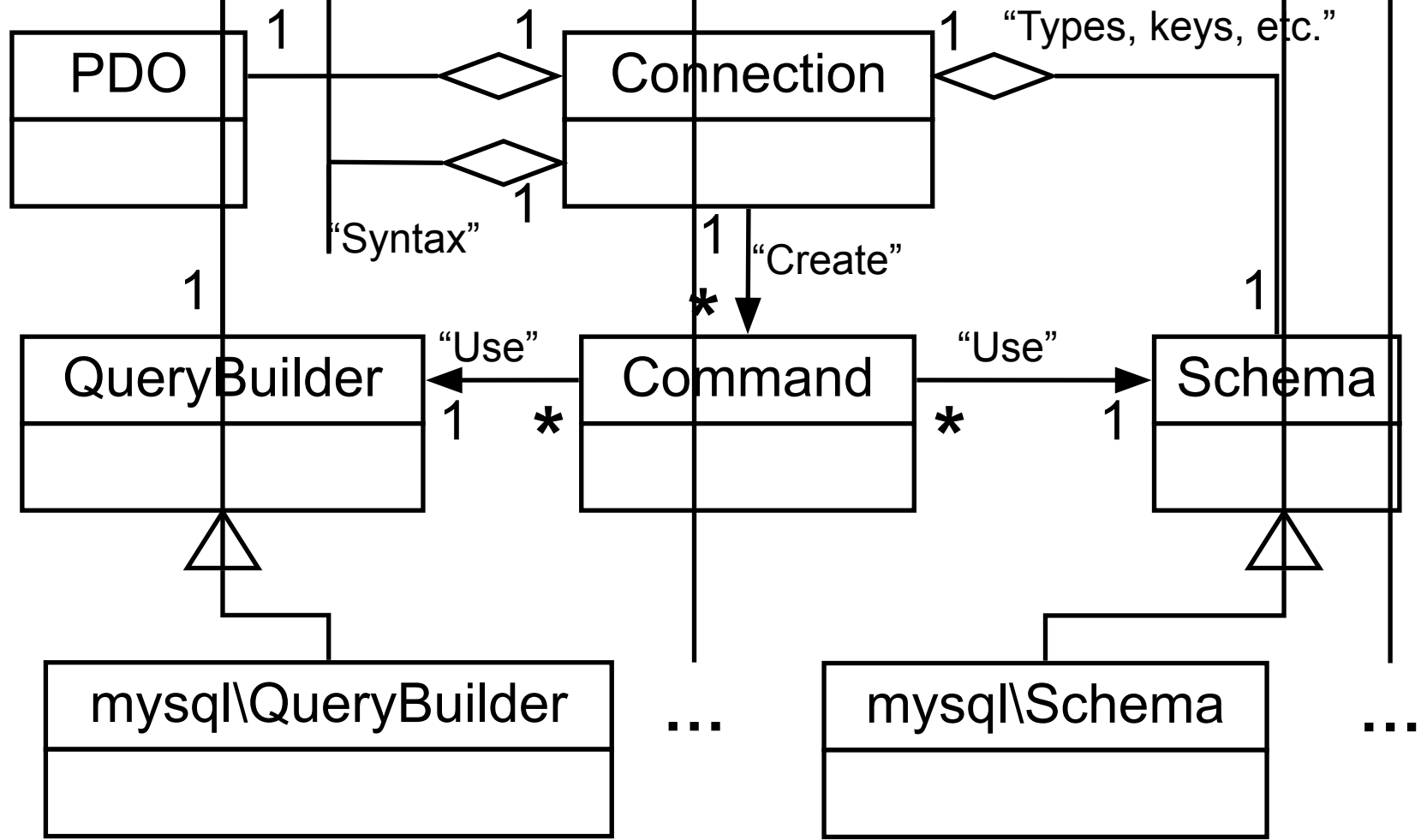
Модель



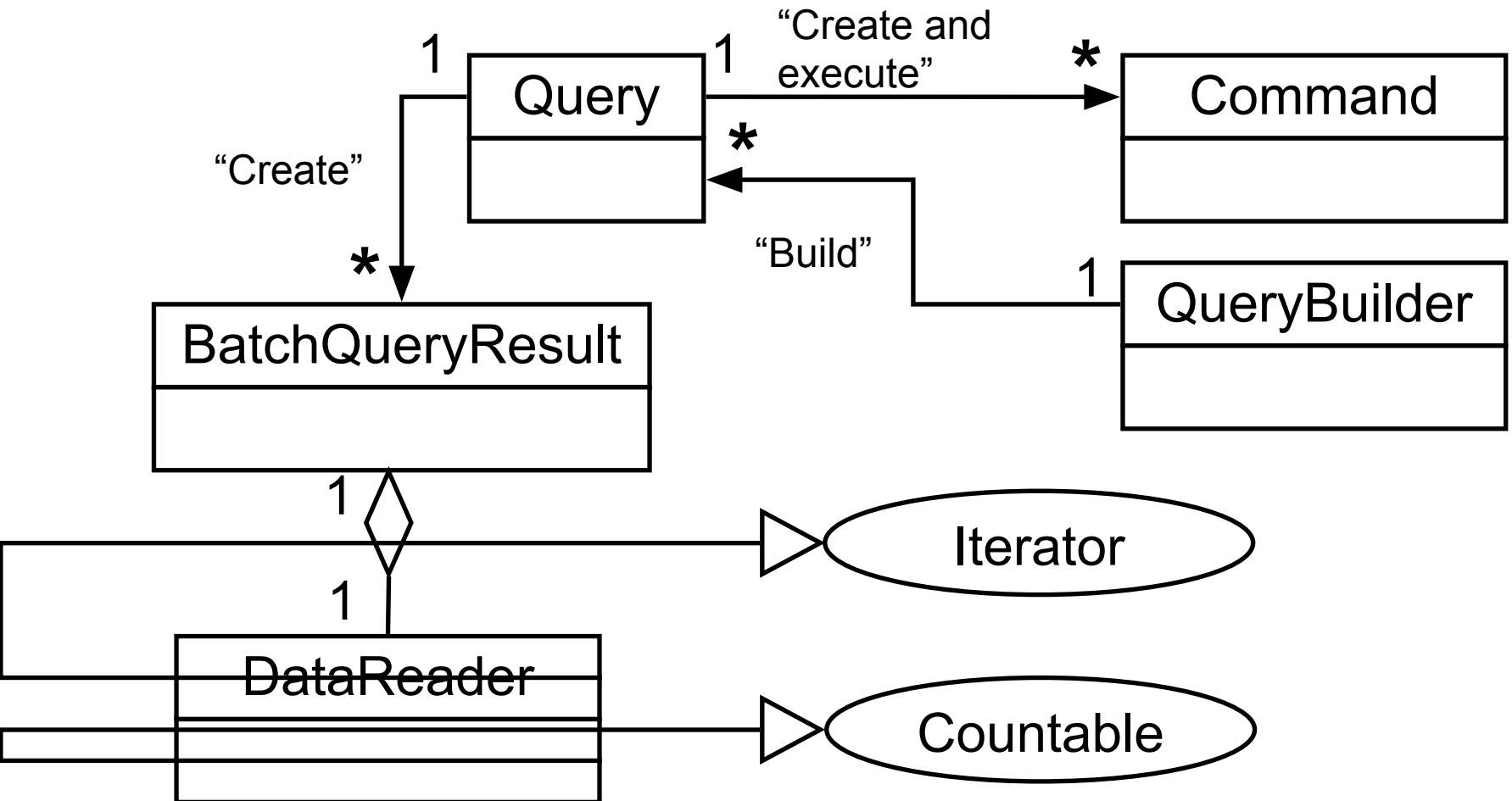
Доступ к базе данных через PDO



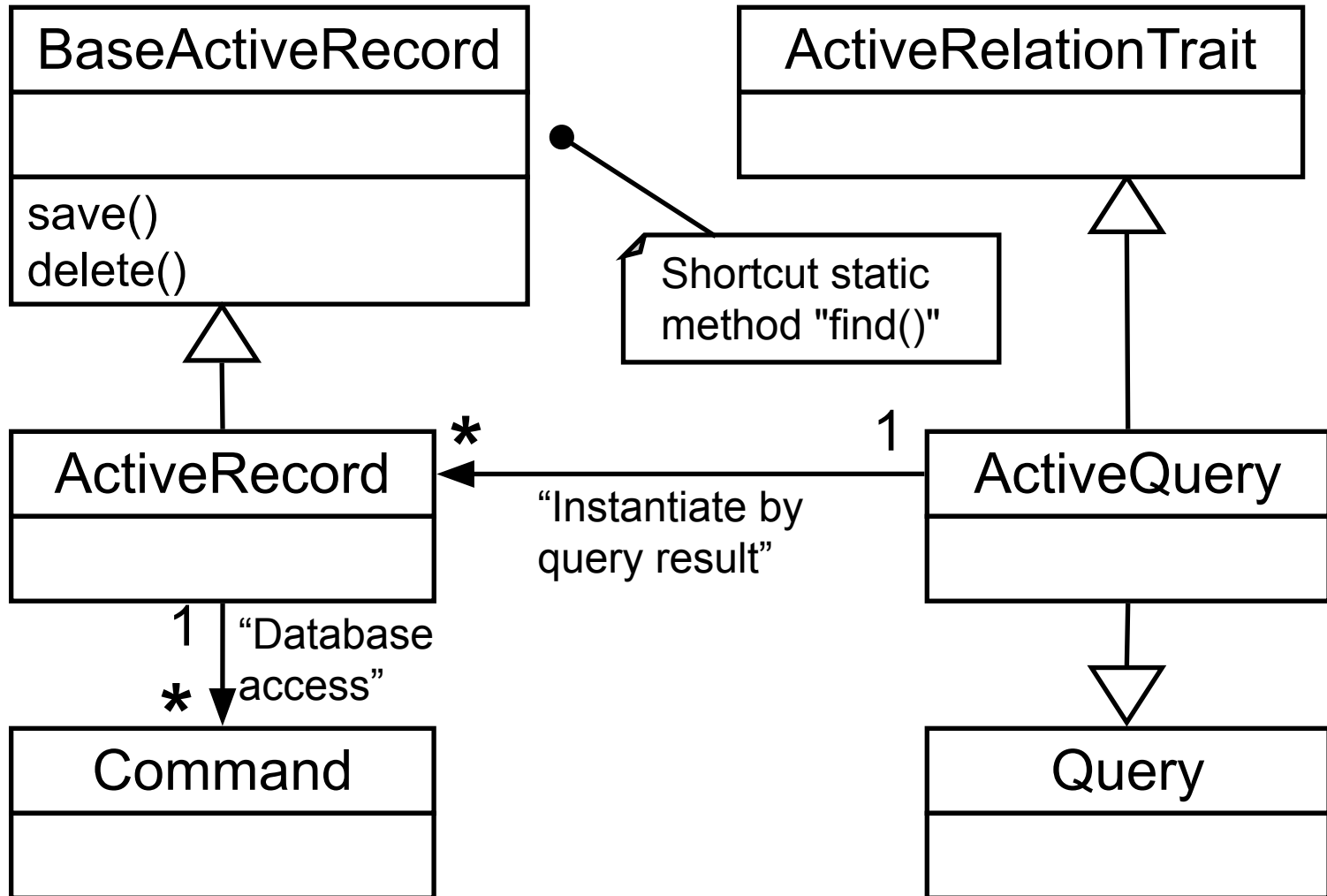
Абстракция базы данных



Select Query Shortcut



Active Record



// Выборка записей:

```
$allUsers = User::find()->all();
```

// Вставка новой записи:

```
$newUser = new User();
```

```
$newUser->name = 'new user';
```

```
$newUser->save();
```

// Обновление существующей записи:

```
$existingUser = User::find()->where(['name'=>'testuser'])->one();
```

```
$existingUser->email = 'newemail@domain.com';
```

```
$existingUser->save();
```

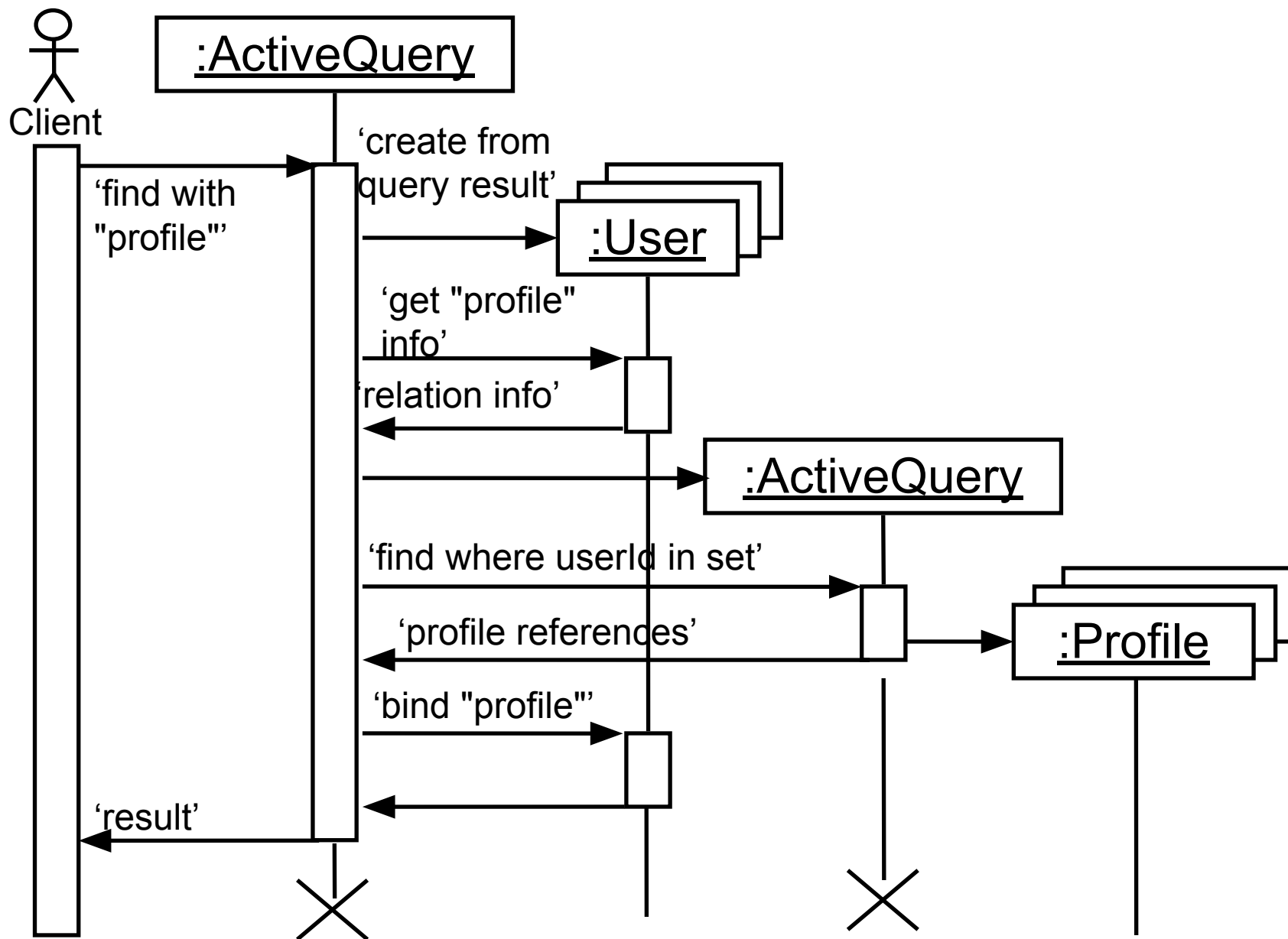
// Отложенная загрузка отношения:

```
$bio = $existingUser->profile->bio;
```

// Жадная загрузка отношения:

```
$allUsers = User::find()->with('profile')->all();
```

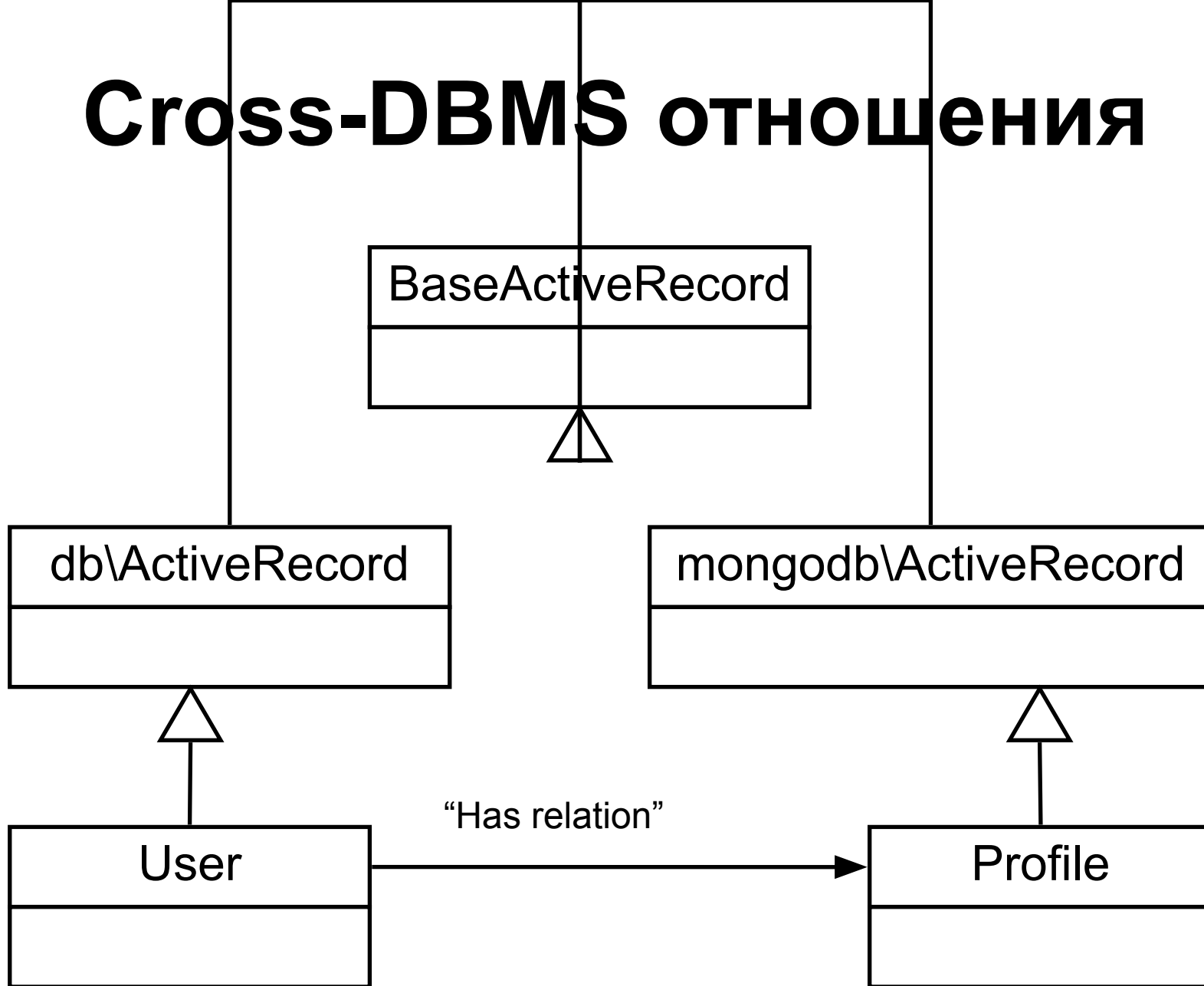
«Жадная» загрузка отношений



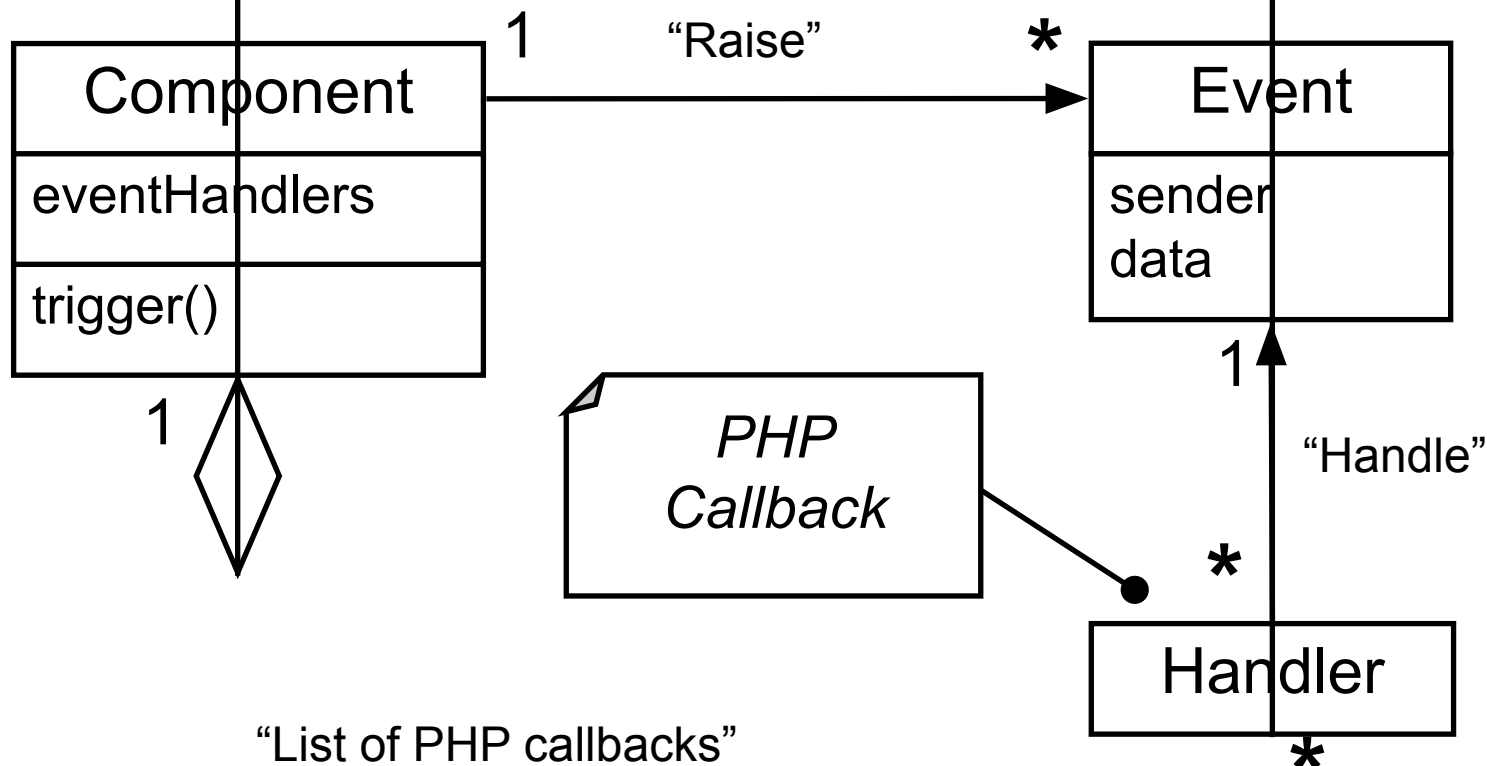
NOSQL Active Record

- MongoDB
- Redis
- ElasticSearch
- Sphinx

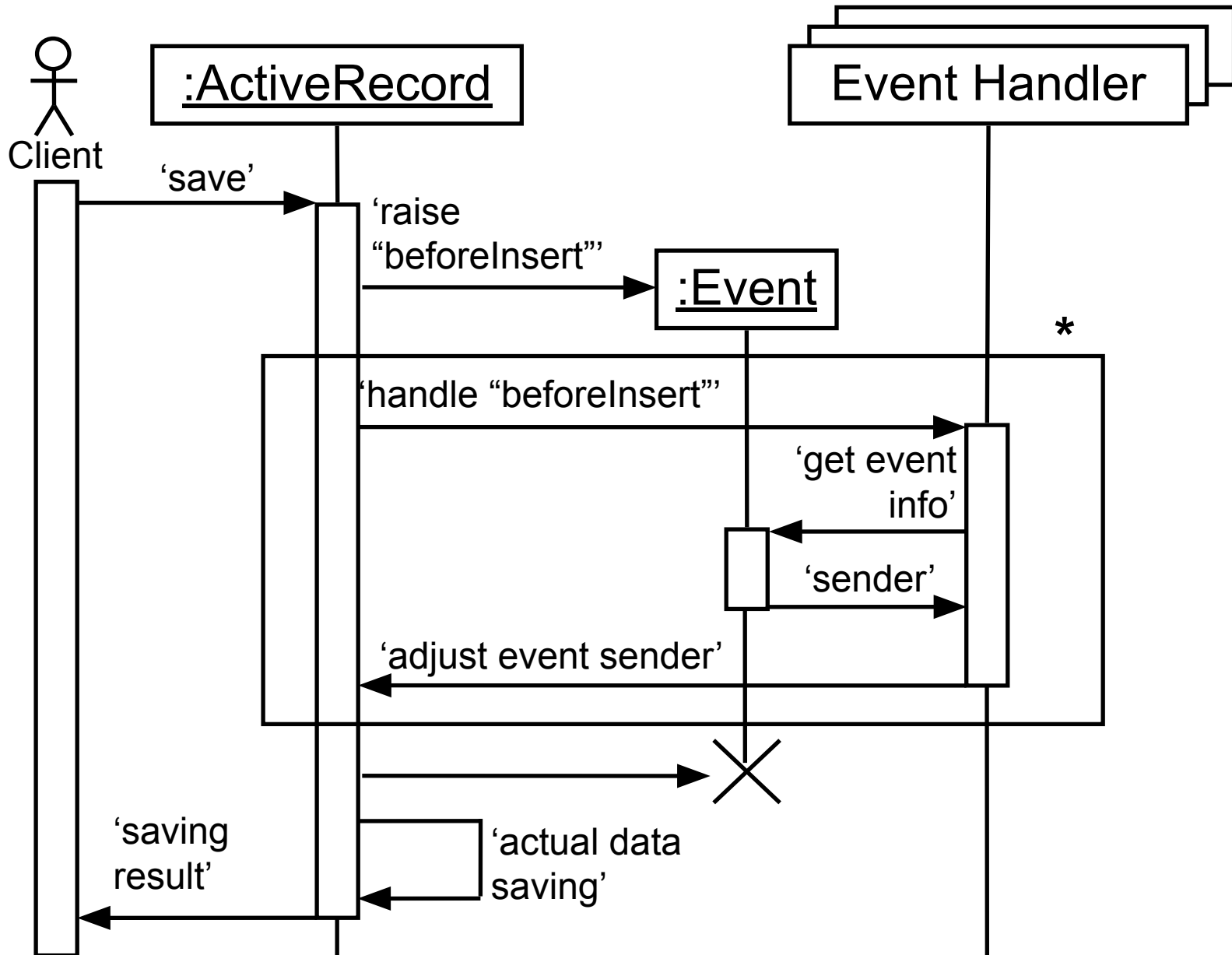
Cross-DBMS отношения



События (Events) в Yii



Обработка события

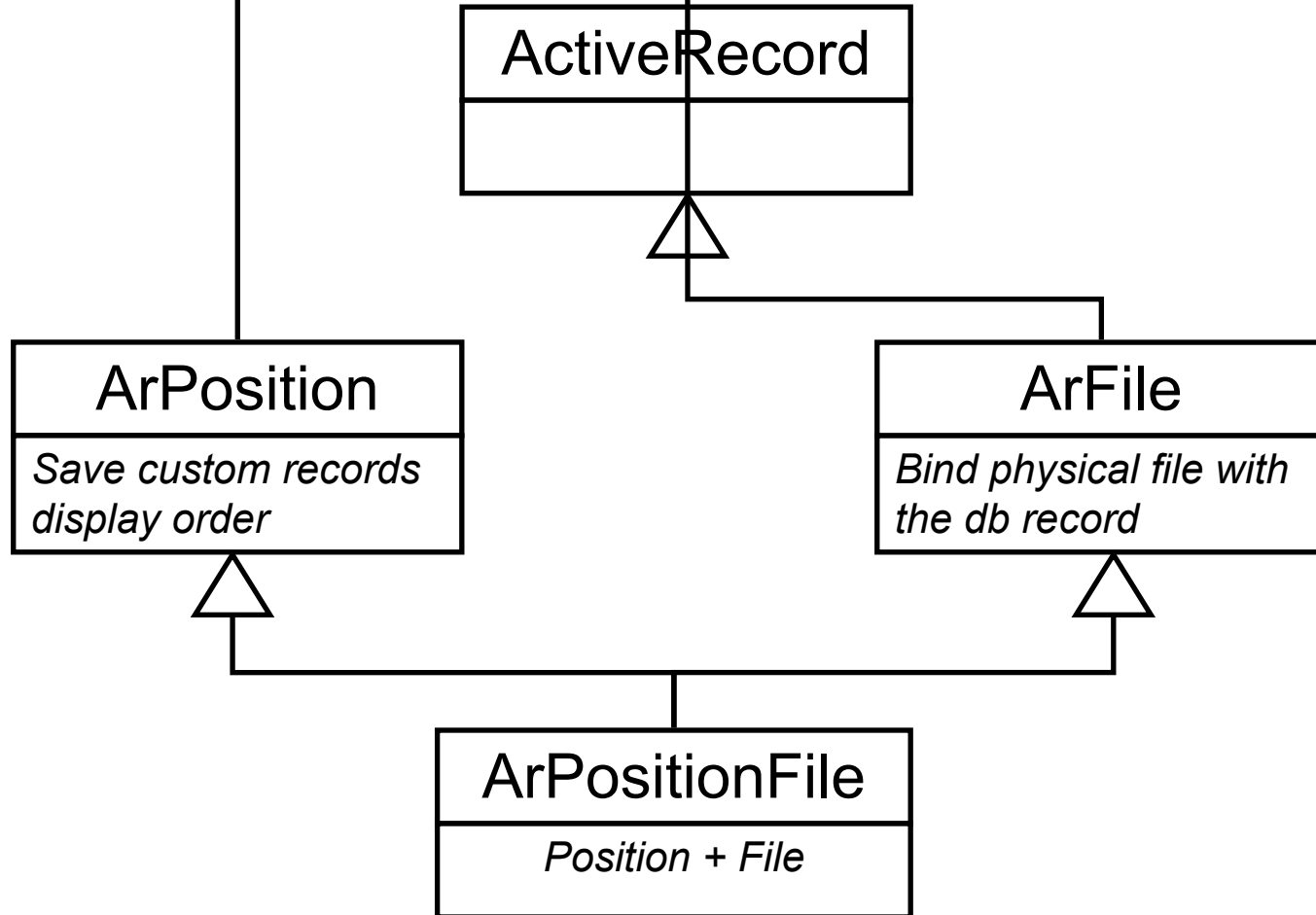


```
function handleBeforeInsert(Event $event)
{
    $sender = $event->sender;
    // Изменяем состояние отправителя события:
    $sender->create_date = date('Y-m-d', strtotime('NOW'));
}
```

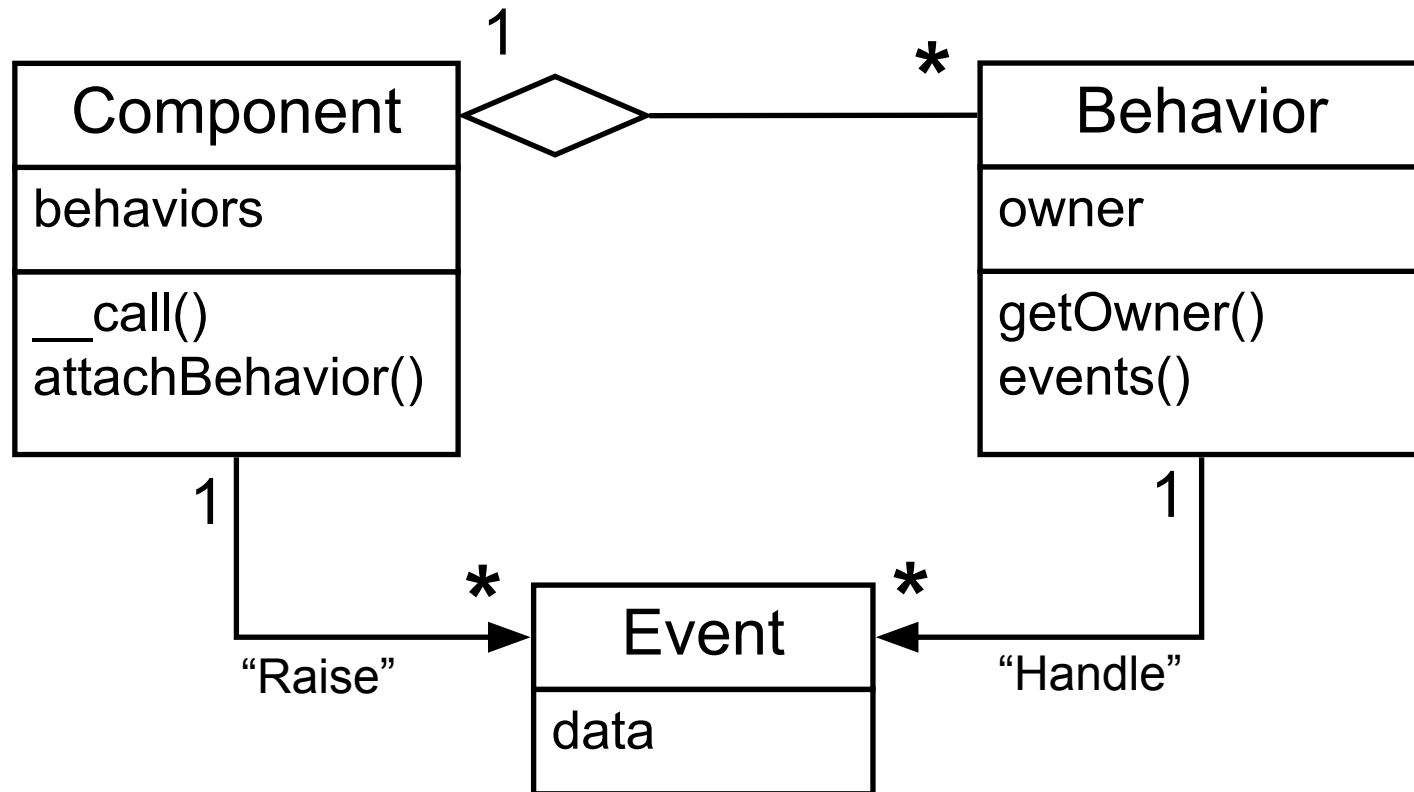
```
$user = new User();
// Назначаем обработчик события:
$user->on('beforeInsert', 'handleBeforeInsert');
$user->name = 'test name';
$user->save();
```

```
echo $user->create_date; // Вывод: '2015-03-21'
```


Проблема множественного наследования



Поведение (Behavior)



Расширение интерфейса

```
class Component
{
    private $_behaviors = [ ];

    public function __call($method, $arguments)
    {
        // Ищем недостающий метод среди поведений:
        foreach ($this->_behaviors as $behavior) {
            if (method_exists($behavior, $method)) {
                return $behavior->$method($arguments);
            }
        }
        throw new Exception("Missing method {$method}");
    }
    ...
}
```

```
class ArBehaviorExample extends Behavior
{
    public function behaviorMethod()
    {
        $this->owner->create_date = date('Y-m-d', strtotime('NOW'));
    }
}
```

```
$user = new User();
// Добавляем поведение:
$behavior = new ArBehaviorExample();
$user->attachBehavior($behavior);
```

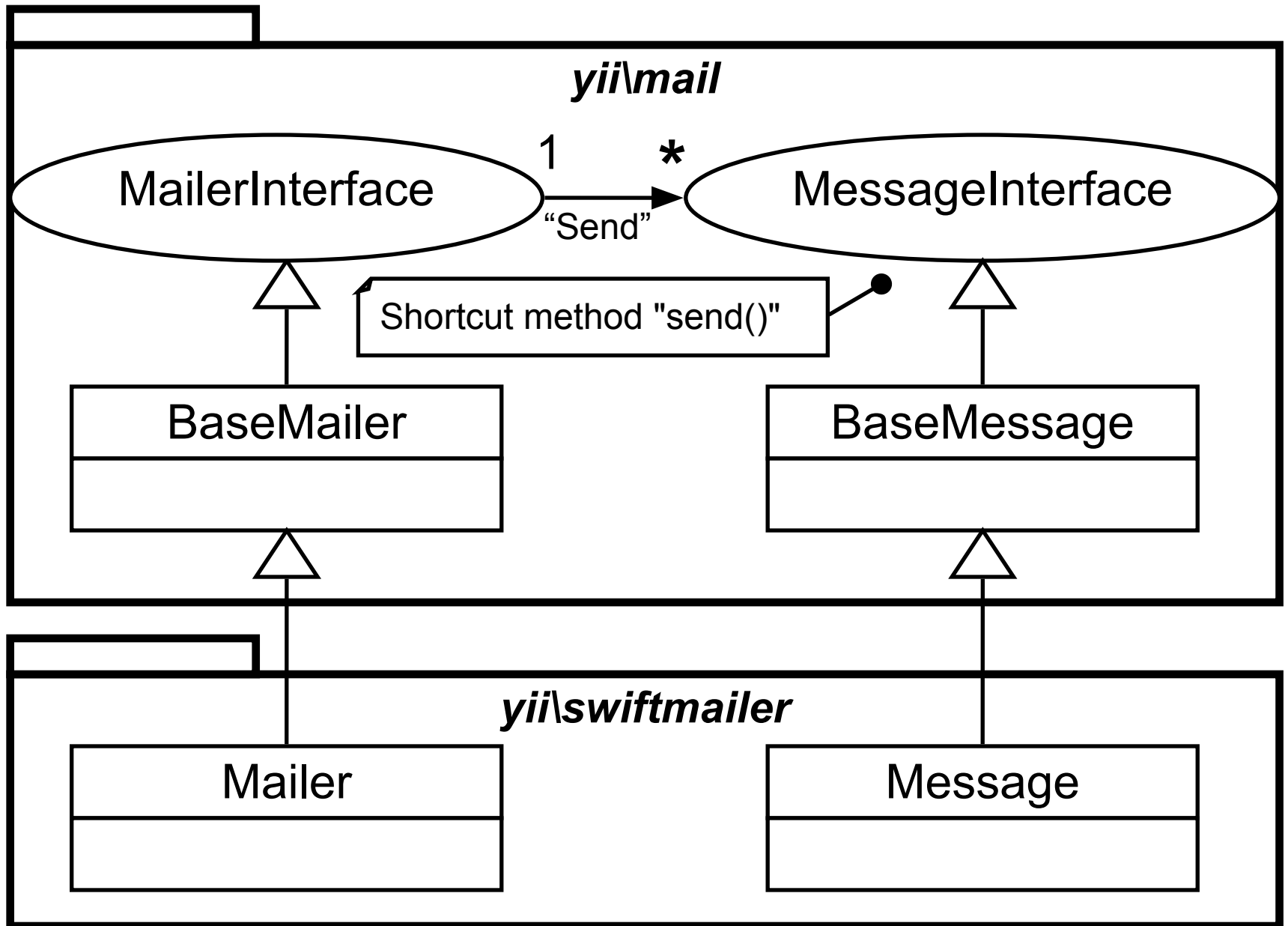
```
// Вызываем метод поведения:
$user->behaviorMethod();
echo $user->create_date; // Вывод: '2015-03-21'
```

Обработка событий

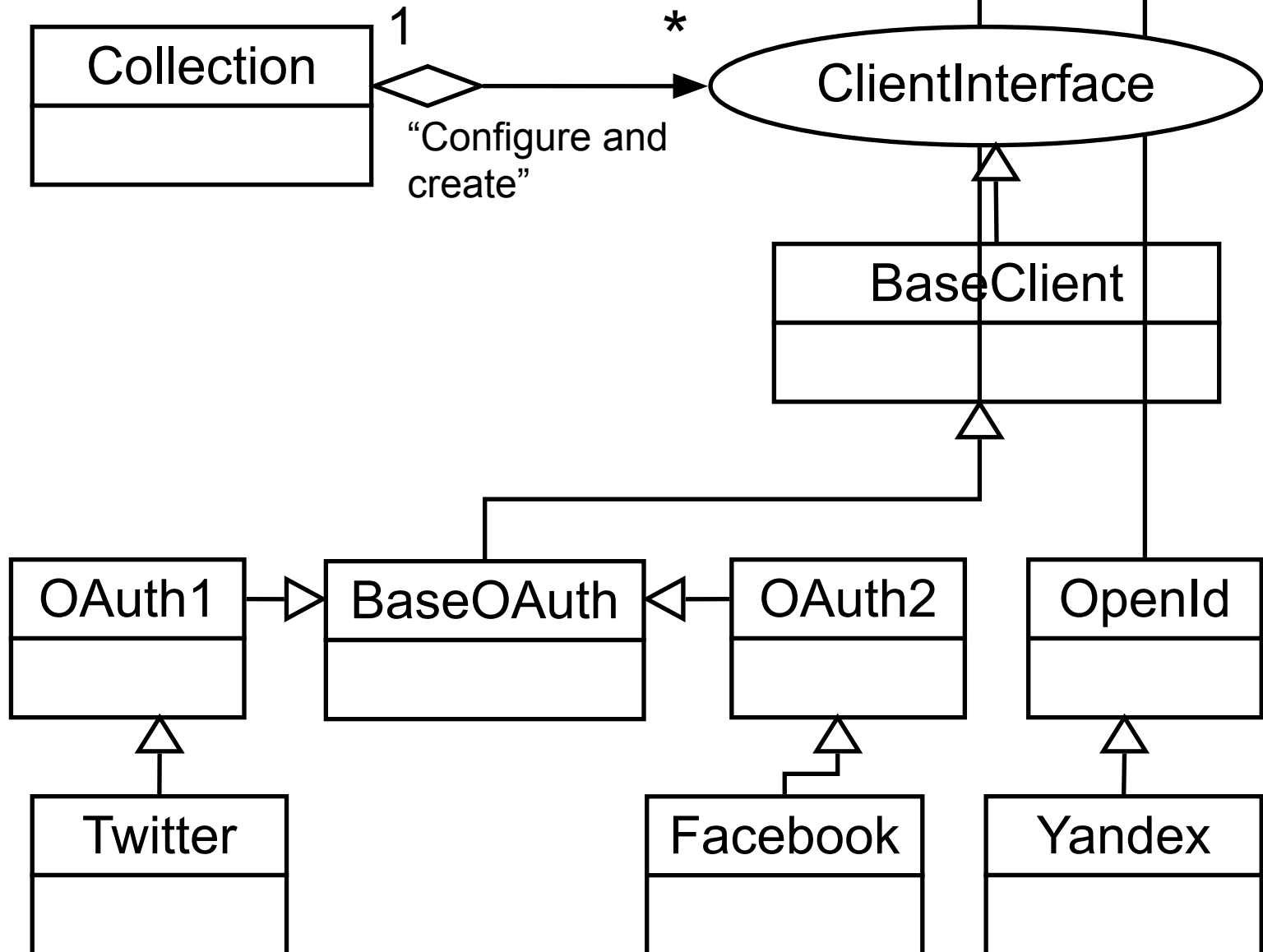
```
class ExampleBehavior extends Behavior
{
  public function events()
  {
    return [
      'beforeInsert' => 'handleBeforeInsert',
      'afterInsert' => 'handleAfterInsert'
    ];
  }

  public function handleBeforeSave(Event $event)
  {
    // Обработка события «beforeInsert»
  }
  ...
}
```

Отправка электронной почты



Расширение «AuthClient»



Дополнительные расширения

- Gii
- Debug
- Bootstrap
- JUI
- Codeception
- Imagine
- ApiDoc

Yii2

- Composer + Bower
- Компонентная структура и DI
- MVC
- Управление «assets»
- PDO и Active Record
- NOSQL и Active Record
- Cross-DBMS отношения
- События
- Поведения
- Стандартные расширения

ZFort приглашает на работу:



Senior Magento
Developer /
TechLead



Middle PHP
CMS Developer



Middle
ASP.NET
Developer



Middle
FrontEnd
Developer



Senior
Project
Manager