Action-Domain-Responder

Refining MVC

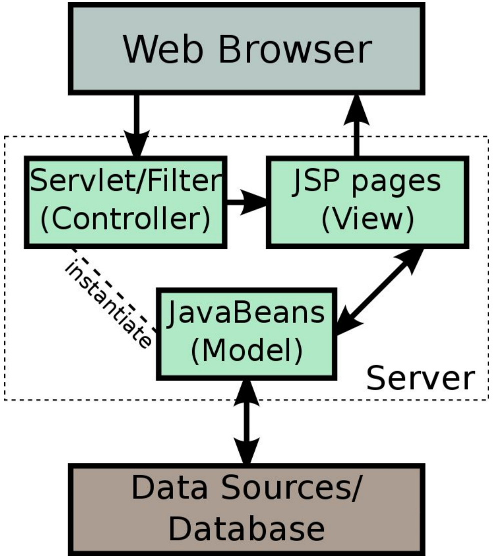
Nigel Dunn

ON | CORPS

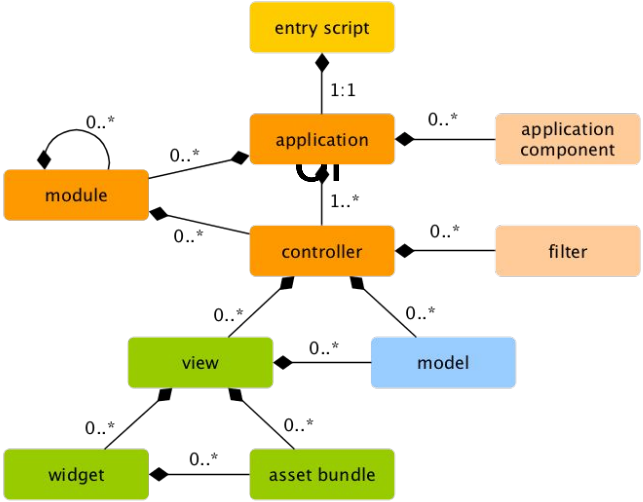
Why refine MVC?

Which MVC?

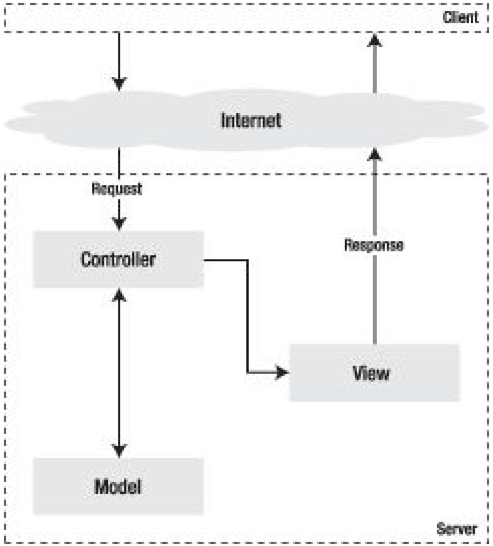
JSP v2



Yii 2



Symfony v1



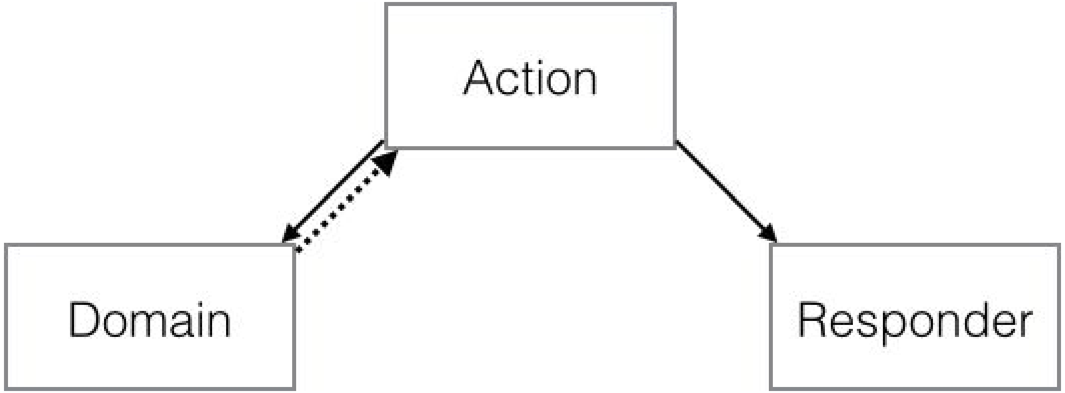
What is ADR?

The pattern

Action ≈ Controller

Domain ≈ Model

Responder ≈ View



Responder ≈ View

```
class TicketResponder
{
    public function index(Response $response, array $data) : Response
    {
        …
    }
}
```

- Returns HTTP response
- Domain payload passed in
- Likely to be one class per responder

Moving away from Views

- More than a templating engine
- Handles all responses - errors and redirects as well as successes
- Handles content negotiation
- No access to Model

Domain ≈ Model

```
class TicketService
{
    public function getTickets() : array
    {
        ...
    }
}
```

- Returns payload as a data structure
- Returns status of operation (not shown here)
- Multi-layered

Moving away from Models

- More than a database layer
- Models are so last century

Action ≈ Controller

```
$app->get('/tickets', function (Request $request, Response $response) {
    $tickets = $this->ticket_service->getTickets();
    return $this->ticket_responder->index(
        $response,
        ["tickets" => $tickets]
    );
});
```

- Typically would be class of its own
- Minimal logic

Moving away from Controller

- No longer a catch-all class with many action & supporting methods
- Doesn’t get involved with HTTP headers or non-2xx responses
- Doesn’t get involved with returning JSON/XML data

Resources

Details
https://github.com/pmjones/adr

Talk
https://vimeo.com/106771285

Implementing Slim ‘first application’ with ADR
https://www.slimframework.com/docs/cookbook/action-domain-responder.html

Recap

Questions?