Intro to Grails

Craig Atkinson
Gr8Conf US 2014

About Me

- Principal Consultant at Object Partners, Inc.
- Grails 4 years
- @craigatk1
- craig.atkinson@objectpartners.com

Object Partners, Inc.

- Groovy, Java, mobile, open source
- Founded 1996
- ~100 senior consultants
  - Minneapolis, Omaha, Chicago, Denver
  - Average tenure >5 years

Object Partners

What is Grails?

Spring MVC + Hibernate + Groovy

Dependency management + CLI + build system

Plugin system w/ 1000+ plugins

Object Partners

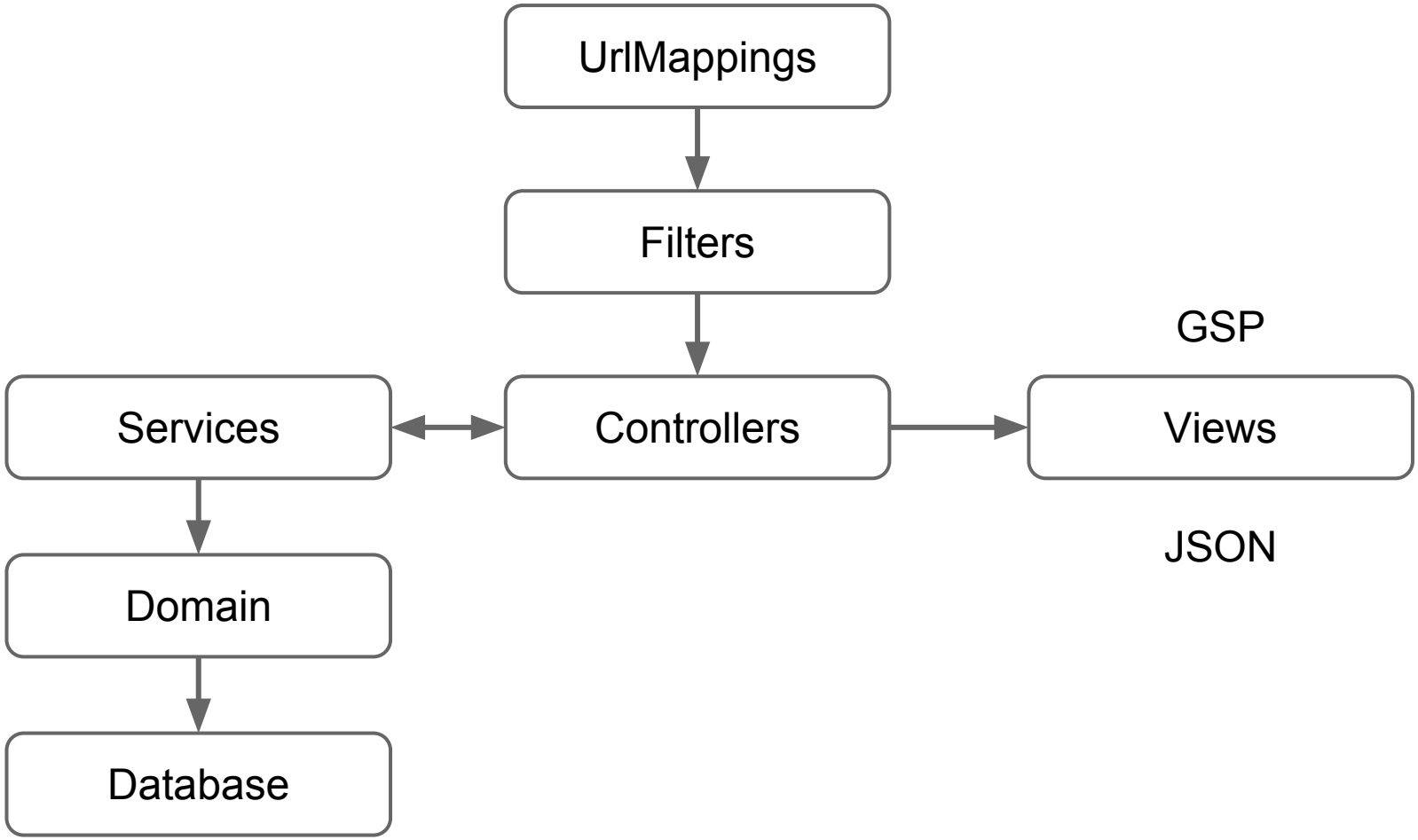
Running on JVM

- Reuse your existing Java code
- Leverage vast ecosystem of Java libraries
- Deploy to app servers such as Tomcat, etc.

Convention over Configuration

- Sensible defaults
- Minimal annotations
- No XML required

Object Partners



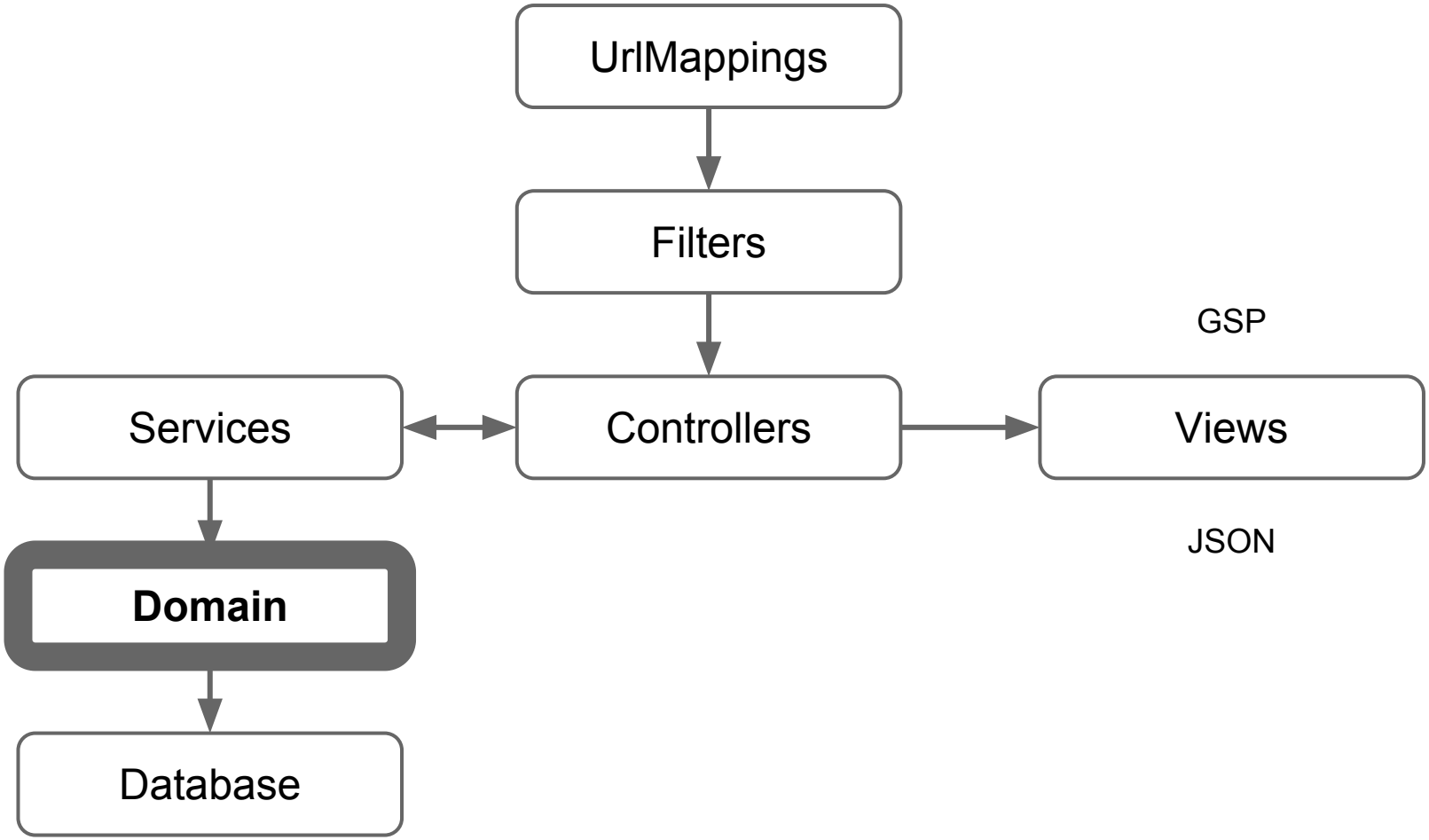
Object Partners

Our First Grails app

```
> grails create-app <app-name>
```

```
> grails create-app grails-intro
```

Object Partners



Object Partners

Domain

- Grails Object Relational Mapper (GORM)
- Sensible conventions for table & column mapping
  - Can override for legacy schemas

```
> grails create-domain-class Person
```

Object Partners

Domain Class

```
class Person {
  String firstName
  String middleName
  String lastName

  static constraints = {
    firstName(nullable: false, blank: false)
    middleName(nullable: true)
    lastName(nullable: false, blank: false)
  }
}
```

Creating Domain Instance

```
Person newPerson = new Person(
  firstName: 'Jim',
  lastName: 'Smith'
)
newPerson.save()
```

```
Person invalidPerson = new Person(
  firstName: 'Jim'
)
assert !invalidPerson.validate() // no last name
```

GORM Methods

```
.save()
.get(id)
.list()
.delete()
.count()
.validate()

// and many more
```

Object Partners

Domain Object Relationships

- One-one = domain class field
  - Address address
- One-many
  - static hasMany = [addresses: Address]
  - static belongsTo = [person: Person]

```
class Address {
  String street
  String city
  String state
  String postalCode
}
```

```
class Person {
  Address address

  String firstName
  String middleName
  String lastName
  Integer age
}
```

Multiple Addresses

```
class Person {
  static hasMany = [addresses: Address]

  String firstName
  String middleName
  String lastName
  Integer age
}
```

Object Partners

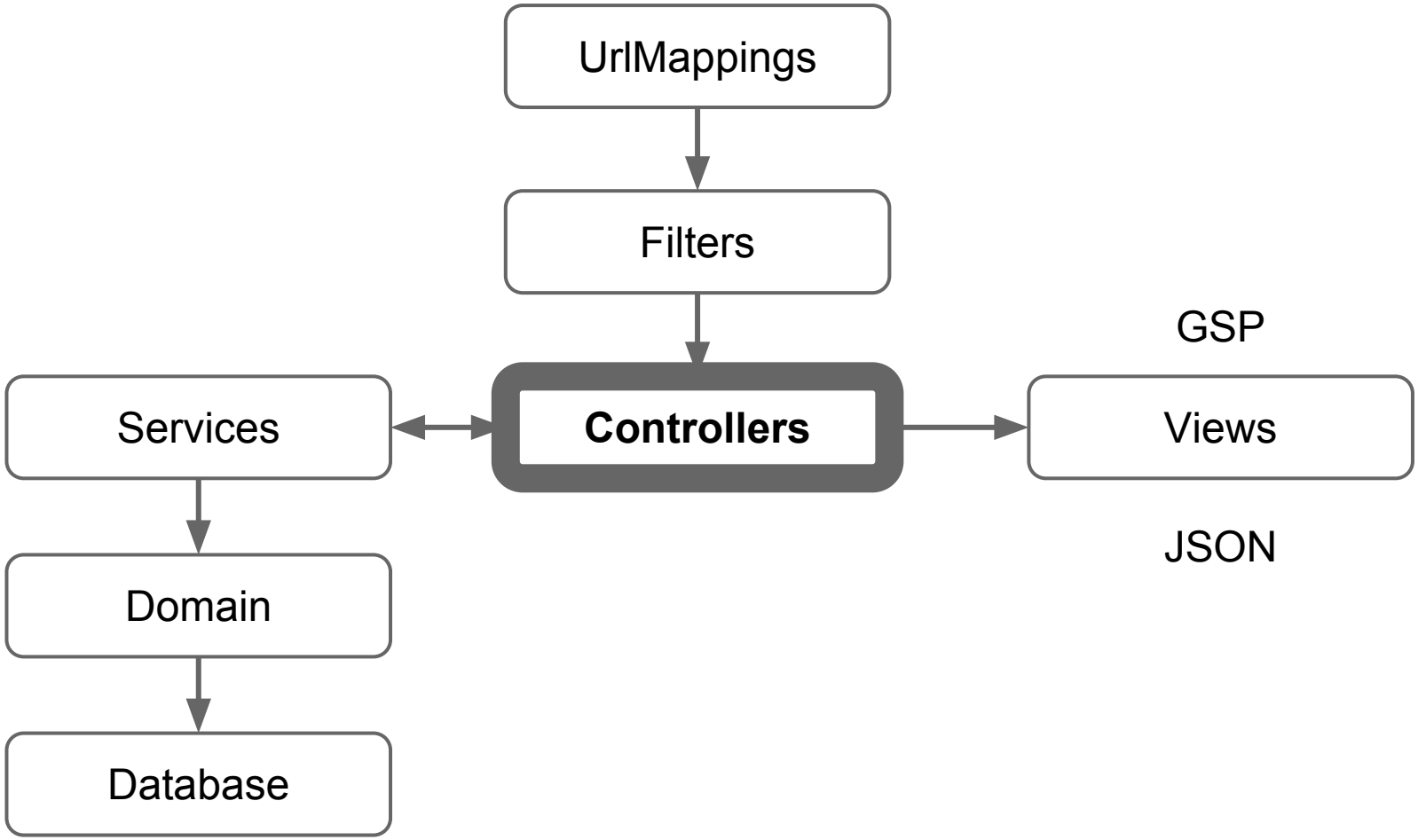
Mapping to Existing Databases

- Customize table & column mappings

```
class User {
  String username

  static mapping = {
    table 'app_user'
    username column: 'user_name'
  }
}
```

Object Partners



Object Partners

Controllers

1. Parse input

   Query parameters, form submission, JSON, etc.

2. Delegate to services

   Business logic, database access, etc.

3. Render response

   GSP, JSON, etc.

Object Partners

Controller Actions

- Controllers contain one or more actions
- Actions are any public method in controller

URL Convention

/controller/action

```
class UserController {
  def signup() {
    ...
  }
}
```

/user/signup

URL Mappings

grails-app/conf/UrlMappings.groovy

```
class UrlMappings {

  static mappings = {
    "/$controller/$action?/$id?(.$format)?"

    "/"(view:"/index")
    "500"(view:'/error')
  }
}
```

ID in URL

/controller/action/id

```
class PersonController {
  def show(Long id) {
    ...
  }
}
```

/person/show/1

Object Partners

Parameter Data Binding

- Access request data in actions
- Same action code handles different types of requests
  - URL query parameters
  - Form submissions
  - API call (JSON or XML)

Object Partners

Typed Action Arguments

- Automatic conversion to boolean, number, date, etc.
- Even automatically load domain object from database

```
class ReportController {
  def createReport(String title, Boolean editable) {
    ...
  }

  def viewReport(Report report) {
    // Param name is 'report.id'
    ...
  }
}
```

Object Partners

Command Object

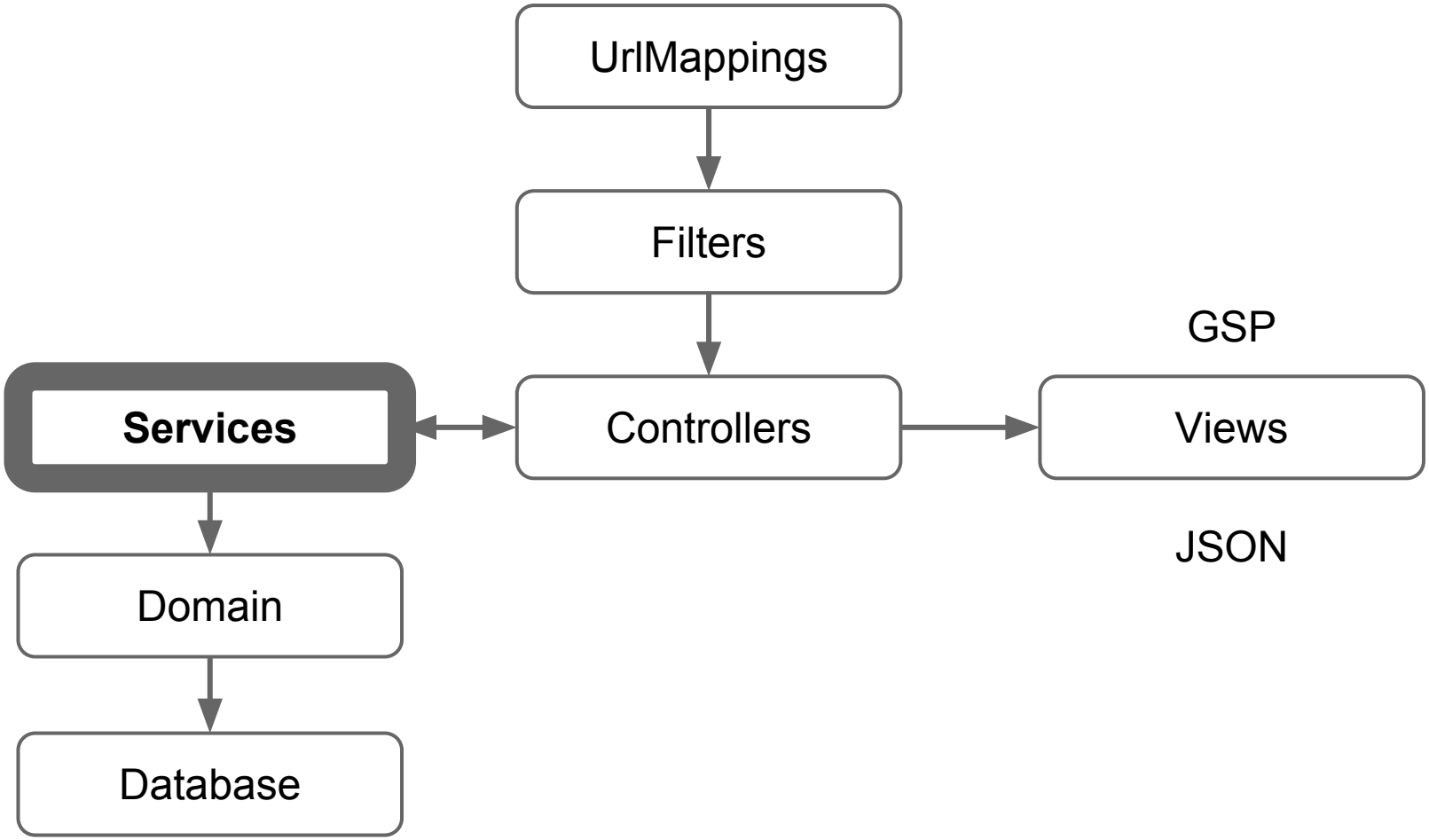
- Action method arguments can become unwieldy when many parameters
- Command object is class with fields for each parameter
- Supports server-side parameter validation

Object Partners

```
class ReportController {
  def createReport(NewReportCommand cmd) {
    ...
  }
}

class NewReportCommand {
  String title
  String content
  String owner
  Boolean editable
  ReportStatus status // Can also bind Enums
}
```

Object Partners



Object Partners

Services

- All classes under grails-app/services
  - Automatically available for dependency injection
- Business logic
- Transactional database access

Object Partners

Database Queries

- Dynamic finders
- ‘where’ queries
- Criteria queries
- Hibernate Query Language (HQL)
- SQL

Object Partners

Dynamic Finders

- Automatically created for all one and two field combinations

```
Person.findByFirstName("Jim")

Person.findByFirstNameAndLastName("Jim", "Smith")

Person.findAllByLastName("Smith")
```

Object Partners

Criteria Queries

Groovy DSL for Hibernate criteria queries

```
Person.withCriteria {
    eq('lastName', personLastName)

    address {
        eq('city', cityName)
    }
}
```

‘where’ Queries

Queries written with standard Groovy code

```
Person.where {
    lastName == 'Smith' &&
    address.city == 'St. Paul' &&
    age > 35
}.list()
```

Object Partners

Hibernate Query Language (HQL)

- SQL-like syntax
- Automatically converts results to domain objects

```
Person.executeQuery("select person from Person person \
    inner join person.address as address \
    where person.age > 35 and \
      person.lastName = 'Smith' and
      address.city = 'St. Paul'")
```

Raw SQL

```
def sessionFactory // Injected by Grails

List<String> findLastNamesWithAge(Integer age) {
  def currentSession = sessionFactory.currentSession

  def sqlQuery = currentSession.createSQLQuery(
    "select p.last_name from person p where p.age = :age")

  sqlQuery.setInteger('age', age)

  return sqlQuery.list()
}
```

Object Partners

Services Automatically Injected

- Available in controllers, filters, taglibs, other services
- Injected by camelCase class name

```
class UserController {
  def userService // Not typed
  // or
  AdminService adminService // Typed service
}
```

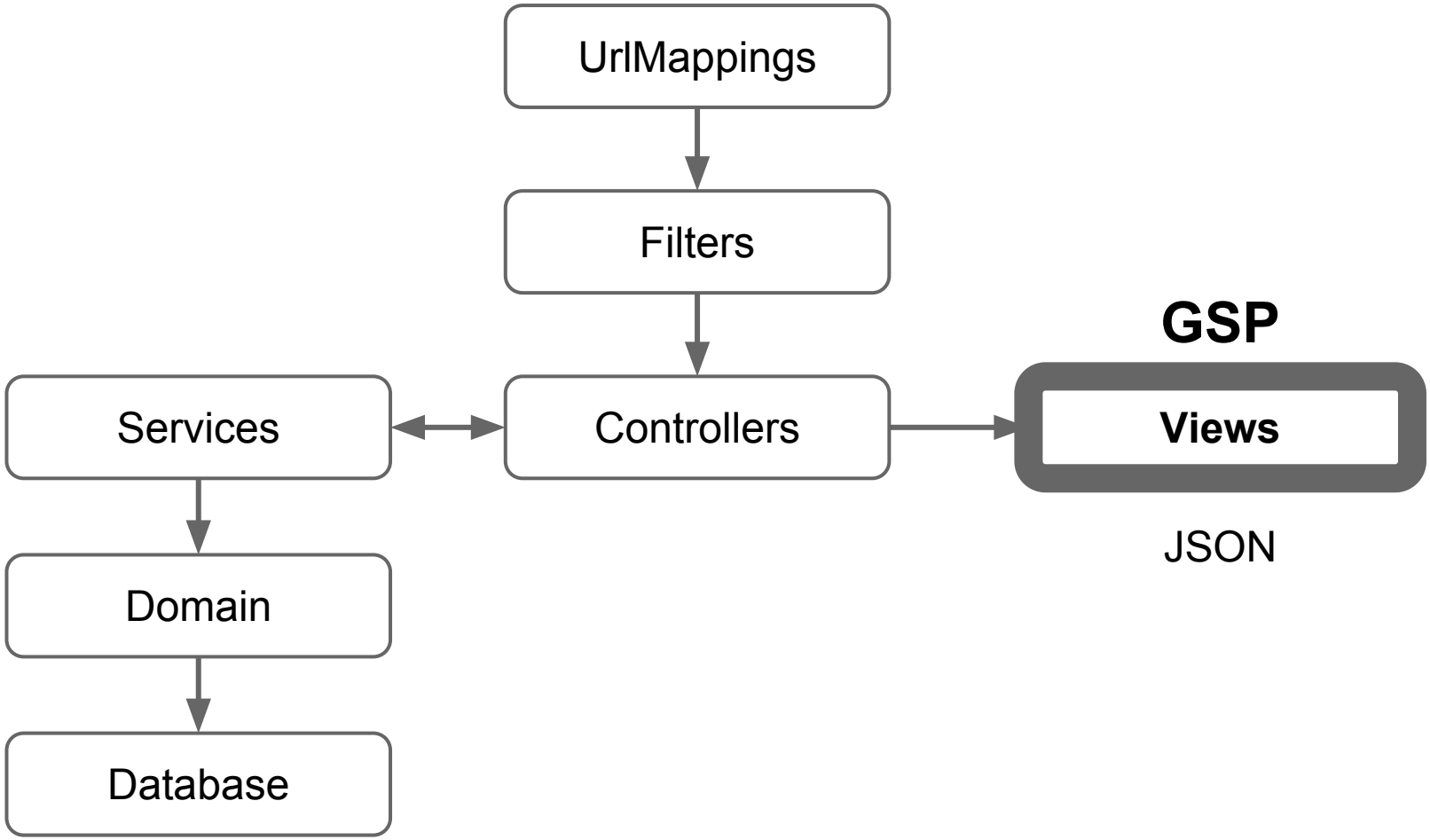
Inject Other Classes

grails-app/conf/spring/resources.groovy

```
beans = {
  beanName(BeanClass)
}
```

```
class MyController {
  BeanClass beanName
}
```

Object Partners



Groovy Server Page (GSP)

- Dynamically generate HTML
- Combine markup with data and code to render HTML back to browser

```
<div>
  <h3>${sectionHeading}</h3>

  <p>${sectionContent}</p>
</div>
```

Templates

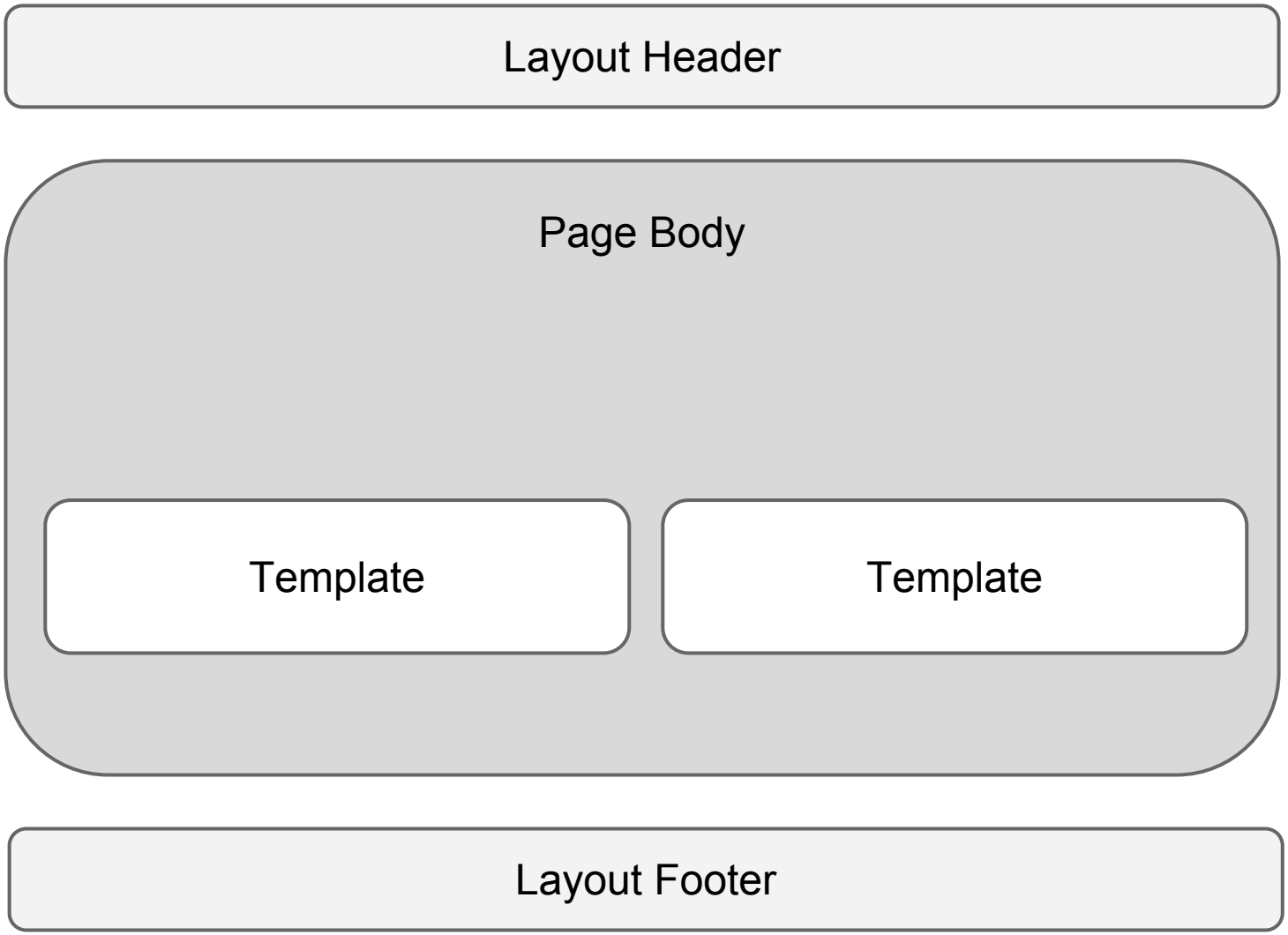
- Split GSPs into multiple templates
  - Share common sections among pages
  - Break up large pages

```
<g:render template="myTemplate"
          model="${[param1: value1, param2: value2]}" />
```

Layouts

- Include common, site-wide markup in one place
  - Page header, footer, etc.
- Wrap GSP in layout

```
<head>
  <meta name="layout" content="main"/>
</head>
```


Layout Header
Page Body
Template
Template
Layout Footer

Minimize Code in View

- Put code logic in Taglibs
- Built-in and custom

Built-In Taglibs

- Forms, links, conditionals, formatting, i18n messages, etc.

```
<g:form controller="invoice" action="submit">
    <g:textField name="invoiceName" />
    <g:submitButton value="Create" />
</g:form>
```

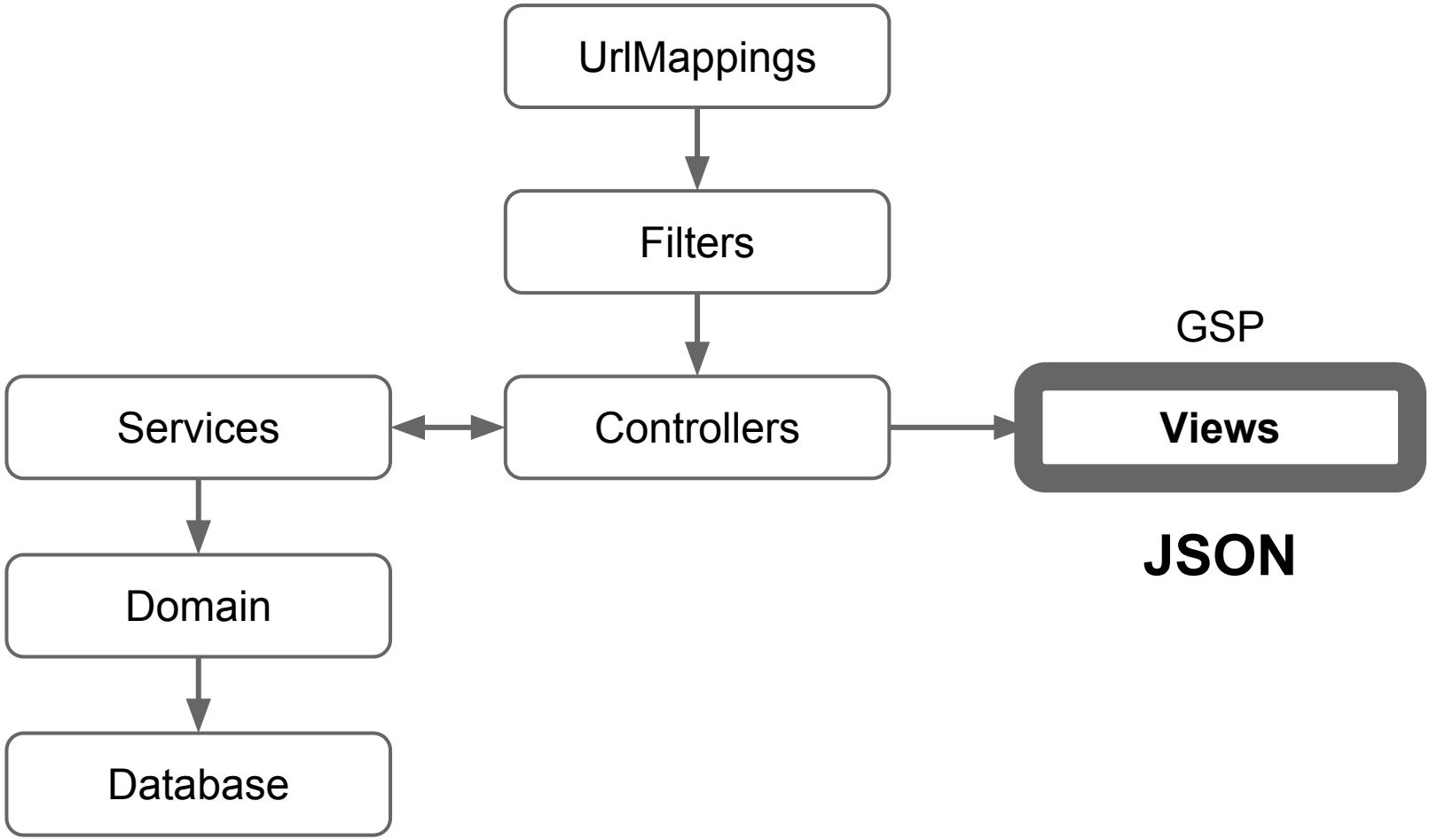
```
<g:if test="${myVal == someVal}">
    <p>My paragraph to display</p>
</g:if>
```

Custom Taglibs

```
UserService userService

def currentUsername = { attrs, body ->
  if (userService.isLoggedIn()) {
    out << userService.currentUser.username
  } else {
    out << ""
  }
}
```

Object Partners



Object Partners

APIs

- Grails 2.3+
- `@Resource`, `RestfulController`, etc.

@Resource API

```
import grails.rest.Resource

@Resource(uri='/person', formats=['json'])
class Person {
    String firstName
    String lastName
}
```

List all: `GET /person`

Read one: `GET /person/$id`

Create: `POST /person`

Update: `PUT /person/$id`

Delete: `DELETE /person/$id`

Object Partners

RestfulController

- Add custom actions to built-in CRUD operations

```
class UserController extends RestfulController {
  UserController() {
    super(User)
  }

  def resetPassword(String username) {
    ...
  }
}
```

Object Partners

Testing

- Tests are first-class citizens
  - Unit and integration tests built in
  - Functional tests through plugins
- Spock is now default test runner

Object Partners

Spock

- Default test runner in Grails 2.3+
- BDD-style test case structure
- Powerful mocking support
- Easy data-driven testing

Object Partners

Unit Testing

- @TestFor(Class) for Grails classes
  - Controllers, services, domain, etc.

Object Partners

```
@Mock(Person)
@TestFor(PersonController)
class PersonControllerSpec extends Specification {

  def 'should create new person'() {
    when:
    // 'controller' is instance of PersonController
    controller.createPerson('Jim', 'Smith')

    then:
    assert Person.findByFirstName('Jim')?.lastName == 'Smith'
  }
}
```

Object Partners

Integration Testing

- In-memory H2 database
- Great for testing DB access in services
- Automatically rolls back any DB changes between tests


```
class PersonServiceIntegrationSpec extends IntegrationSpec {
  PersonService personService // Automatically injected by Grails

  def 'should find last names of people with given age'() {
    given:
    Person person = new Person(firstName: Jim, lastName: 'Smith',
        age: 37).save()

    when:
    List<String> results = personService.findLastNamesWithAge(37)

    then:
    assert results == [person.lastName]
  }
}
```

Object Partners

Web Functional Testing

- In-memory H2 database
- Geb for browser functional testing
  - Grails plugin for Geb
- Examples of Grails + Geb
  - github.com/geb/geb-example-grails
  - github.com/craigatk/geb-example

Object Partners

Development Environment

- CLI works with Windows, Mac, Linux
- Full-featured IDE
  - Groovy/Grails Tool Suite
  - IntelliJ IDEA
- Text editor of your choice
  - Vim, Emacs, Sublime, etc.

Object Partners

Learn More

- Grails documentation
- StackOverflow
- Books
  - Grails in Action, 2nd edition
  - Definitive Guide to Grails 2
  - Programming Grails (more advanced)

Q & A

Thanks for attending!