Spring BlazeDS Integration

Rossen Stoyanchev
SpringSource

About The Speaker

- Senior Consultant at SpringSource
- Spring Web Team
- Consulting & Training
- “Rich Web Applications With Spring” Course Lead

Topics

- **Flex**
- Spring
- Using Flex and Spring
- Spring/BlazeDS 1.0.0.M1
- Spring/BlazeDS 1.0.0.M2
- Looking Forward

Flex

- Highly-interactive user interfaces
  - Wide support (browser and desktop)
- ActionScript
  - A variant of JavaScript (ECMAScript compliant)
  - OO constructs and modularity like Java
- MXML
  - XML-based language for declarative UI
  - Compiles to ActionScript

Connecting Flex To Java

- Flex HTTPService or WebService
  - Similar to XHR requests
  - Consume plain XML or SOAP data
- Adobe BlazeDS (Open-Source)
  - Invoke methods on Java objects (Remoting)
  - Publish/subscribe to destinations (Messaging)

Adobe BlazeDS Serialization

- Binary encoding of data (AMF)
- No explicit mapping information required
- Java objects deserialized into dynamic ActionScript objects
- Equivalent ActionScript class annotated with Java class name

Adobe BlazeDS Messaging

- Client subscribes to destinations
- Push messaging from server to clients
- Client-to-client communication
- Integration with JMS

Using Adobe BlazeDS

- Configure MessageBrokerServlet
- Configure channels, endpoints, destinations
  - services-config.xml (main configuration)
  - proxy-config.xml (REST/SOAP destinations)
  - messaging-config.xml (destinations)
  - remoting-config.xml (Java remote services)
- Compile MXML and ActionScript artifacts supplying path to above configuration

Adobe LiveCycle Data Services

- Commercial offering
  - Start with BlazeDS and upgrade when needed
- Adds data management service option
  - Automatic synchronization of data between a Flex client and back-end services
  - JDBC/Hibernate integrations
  - Transaction management
- NIO based socket server for scaling

Topics

- Flex
- **Spring**
- Using Flex and Spring
- Spring/BlazeDS 1.0.0.M1
- Spring/BlazeDS 1.0.0.M2
- Looking Forward

Spring

- Powers many enterprise applications
- Reduces code dependencies
- Decouples applications from infrastructure
- Eases integration with enterprise services and best-of-breed Java technologies

Spring Projects

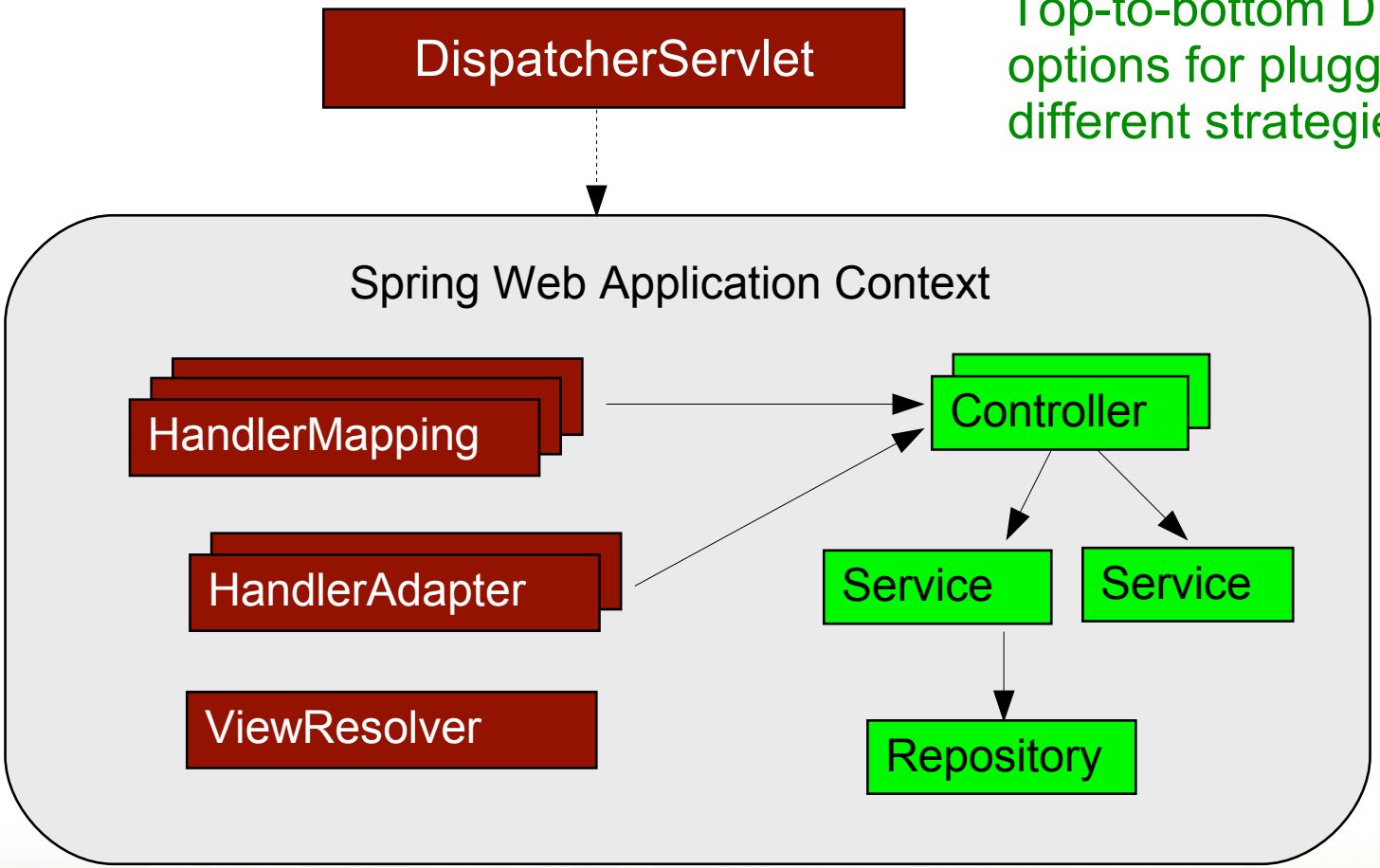
- Open-source projects
- A common approach
  - Flexible design
  - Ease of testing
  - Independence from infrastructure
  - Non-invasive, AOP-style services
- Developers accustomed to the "Spring Way" of doing things

Spring Batch, Spring Dynamic Modules, Spring Faces, Spring Integration, Spring JavaScript , Spring Web Flow, Spring Web Services, ...

Common Spring Idioms

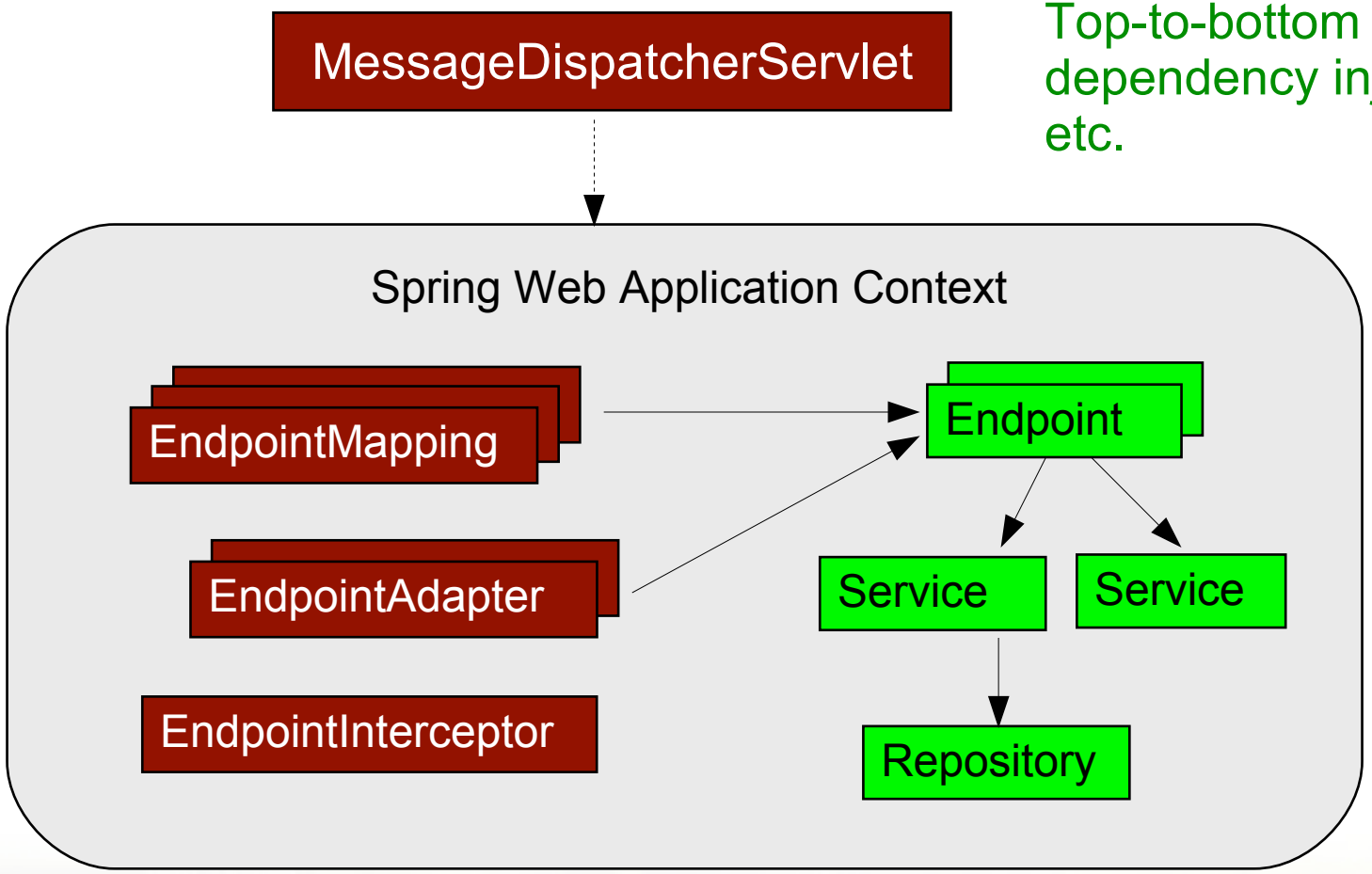
- Custom XML namespaces
  - Succinct, expressive configuration
- Factory beans
  - Code-free initialization of infrastructure
- Service exporters (JMS, JMX, WS)
  - Transparently expose existing services
- Many more...

Spring MVC Configuration



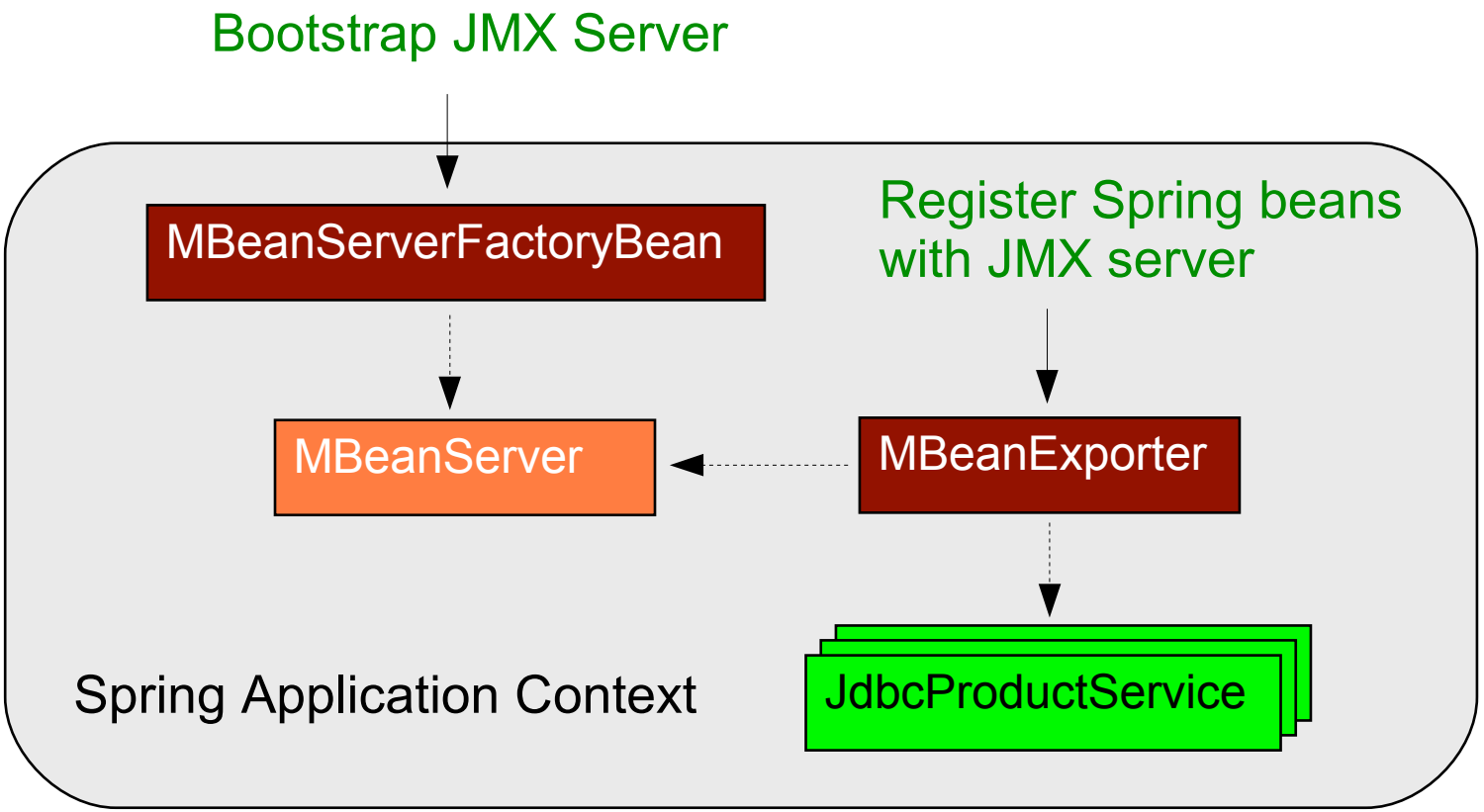
Top-to-bottom DI, lots of options for plugging in different strategies!

Spring WS Configuration



Top-to-bottom dependency injection, etc.

Exposing Spring-Managed Objects As JMX MBeans



Topics

- Flex
- Spring
- **Using Flex and Spring**
- Spring/BlazeDS 1.0.0.M1
- Spring/BlazeDS 1.0.0.M2
- Looking Forward

Existing Spring/BlazeDS Integration

- SpringFactory in services-config.xml
- Provide mapping between remoting destinations and spring bean names
- SpringFactory performs lookup in Spring application context

Existing Integration Sample Configuration

```
<factories>
    <factory id="spring" class="flex.samples.factories.SpringFactory"/>
</factories>

<destination id="mortgageService">
    <properties>
        <factory>spring</factory>
        <source>mortgageBean</source>
    </properties>
</destination>
```

Existing Spring/BlazeDS Integration Disadvantages

- Dependency lookups not natural to Spring applications
- Extra configuration required to look up the Spring beans you already have
- A barrier for deeper integration with Spring Security, Spring JMS, etc.

Topics

- Flex
- Spring
- Using Flex and Spring
- **Spring/BlazeDS 1.0.0.M1**
- Spring/BlazeDS 1.0.0.M2
- Looking Forward

Spring Web

- Spring aims to be agnostic to the chosen client technology and architecture
  - REST with Spring MVC
  - SOAP with Spring Web Services
  - JSF with Spring Web Flow and Spring Faces
  - Spring JavaScript (Dojo)
- Simplify integration and promote best practices

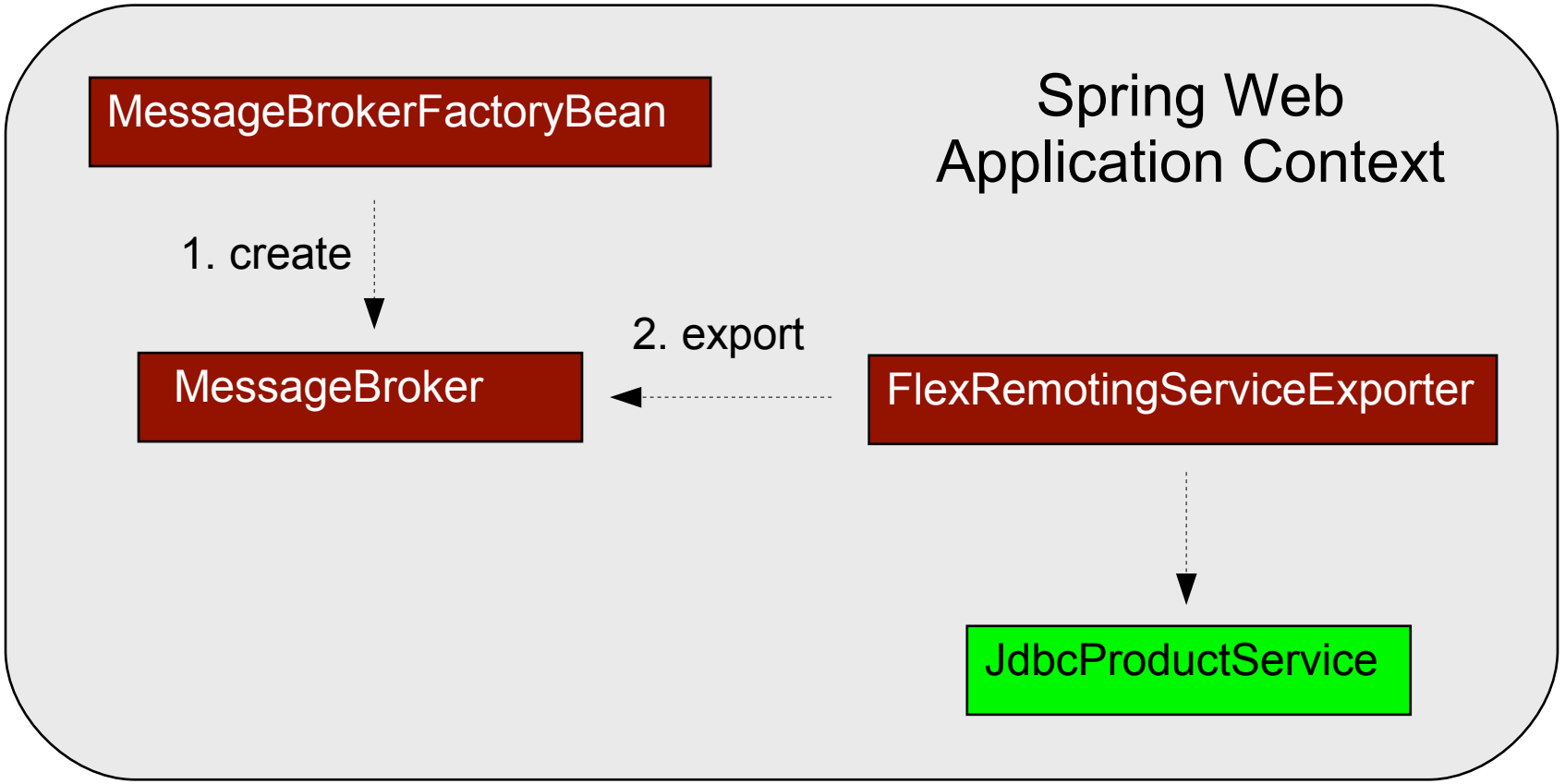
Spring BlazeDS

- An open-source Spring project
- Provides a path to integrating the open-source BlazeDS with Spring
- A joint partnership between SpringSource and Adobe

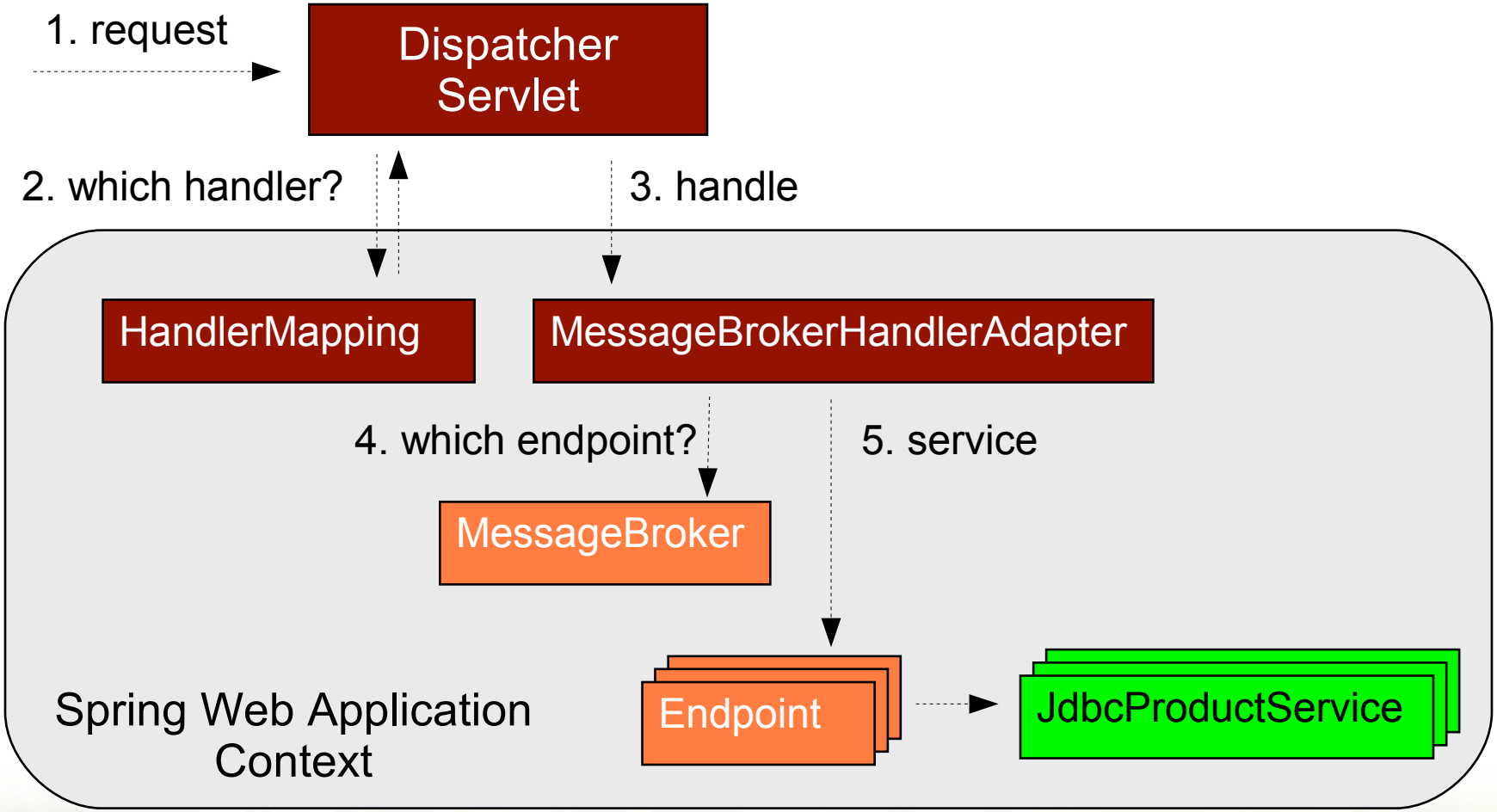
Spring BlazeDS 1.0.0 M1

- A foundational release (Dec 2008)
- Spring MVC DispatcherServlet
- Spring-managed MessageBroker
  - MessageBrokerFactoryBean
- Transparently register Spring beans
  - FlexRemotingServiceExporter

Initialization At Startup



Request Handling At Runtime



Benefits Of New Approach

- Top-to-bottom dependency injection
- DI for the MessageBroker instance
- Flexible mapping of requests
- Familiar Spring MVC programming model

JDBC DAO Method Before

```
public Product getProduct(int productId) {
        Product product = new Product();
        Connection c = null;
        try {
                c = dataSource.getConnection();
                PreparedStatement ps = c.prepareStatement(
                        "SELECT * FROM product WHERE id=?");
                ps.setInt(1, productId);
                ResultSet rs = ps.executeQuery();
                if (rs.next()) {
                        product = new Product();
                        product.setProductId(rs.getInt("id"));
                        product.setName(rs.getString("name"));
                        product.setDescription(rs.getString("description"));
                        product.setImage(rs.getString("image"));
                        product.setCategory(rs.getString("category"));
                        product.setPrice(rs.getDouble("price"));
                        product.setQty(rs.getInt("qty"));
                }
        } catch (Exception e) {
                e.printStackTrace();
                throw new RuntimeException(e);
        }
        return product;
}
```

JDBC DAO Method After

```
public List<Product> getProducts() {
    return template.query("SELECT * FROM product ORDER BY name", productMapper);
}
```

Demo

Topics

- Flex
- Spring
- Using Flex and Spring
- Spring/BlazeDS 1.0.0.M1
- **Spring/BlazeDS 1.0.0.M2**
- Looking Forward

Spring BlazeDS 1.0.0 M2

- Simplified XML namespace-based configuration
- Integration with Spring Security
- New integration-test module using FlexUnit

Flex Custom Namespace

- Create MessageBroker

```
<flex:message-broker />
```

Exporting Flex Destinations

- Option 1

```
<flex:remoting-destination ref="productService" />

<bean id="productService" class="flex.spring.JdbcProductDAO" >
    <constructor-arg ref="dataSource"/>
</bean>
```

- Option 2

```
<bean id="productService" class="flex.spring.JdbcProductDAO" >
    <flex:remoting-destination/>
     <constructor-arg ref="dataSource"/>
</bean>
```

Export Via Annotations

```
@Component
@RemotingDestination(id="foo", channels="my-amf, my-amf2")
public class FooService {

    @RemotingInclude
    public String bar() {
        return "bar";
    }

    @RemotingExclude
    public String baz() {
        return "baz";
    }
}
```

Securing Channels & Paths

```
<flex:message-broker>
    <flex-secured>
        <flex-secured-channel access="ROLE_USR"channel="my-amf"  />
        <flex-secured-endpoint-path
            access="ROLE_USR" pattern="**/protected/**"  />
    <flex-secured>
</flex:message-broker>
```

Method-Level Security

```
<bean id="productService" class="FooService" >
    <flex:remoting-destination/>
    <security-intercept-methods>
        <security-protect access="ROLE_USER" method="*" />
    </security-intercept-methods>
</bean>
```

Demo

Build Environment

- FlexBuilder
- Command line mxmlc compiler
  - Ant tasks
  - Maven plugins
- Mxmlc schema
  - Check out xsd4xml project on Google Code

Topics

- Flex
- Spring
- Using Flex and Spring
- Spring/BlazeDS 1.0.0.M1
- Spring/BlazeDS 1.0.0.M2
- **Looking Forward**

Roadmap

- Towards RC1 and 1.0 Final for the end of March
- Support for using Spring JMS to route messages to JMS MOM
- Support for routing messages via Spring Integration channels

Resources

Spring BlazeDS Forum:

http://forum.springframework.org

Joint Adobe and SpringSource Webinar:

http://tinyurl.com/dlyp6c

Spring BlazeDS Test Drive:

http://tinyurl.com/ca6gnl

Thank You For Attending!