Spring 2.0 and Beyond

Rod Johnson
CEO
Interface21

Topics

- What is Spring?
- Spring 2.0
  - Goals
  - Themes
- 2006 in Review
- What to expect in 2007…

What is Spring?

Much more than an IoC container…

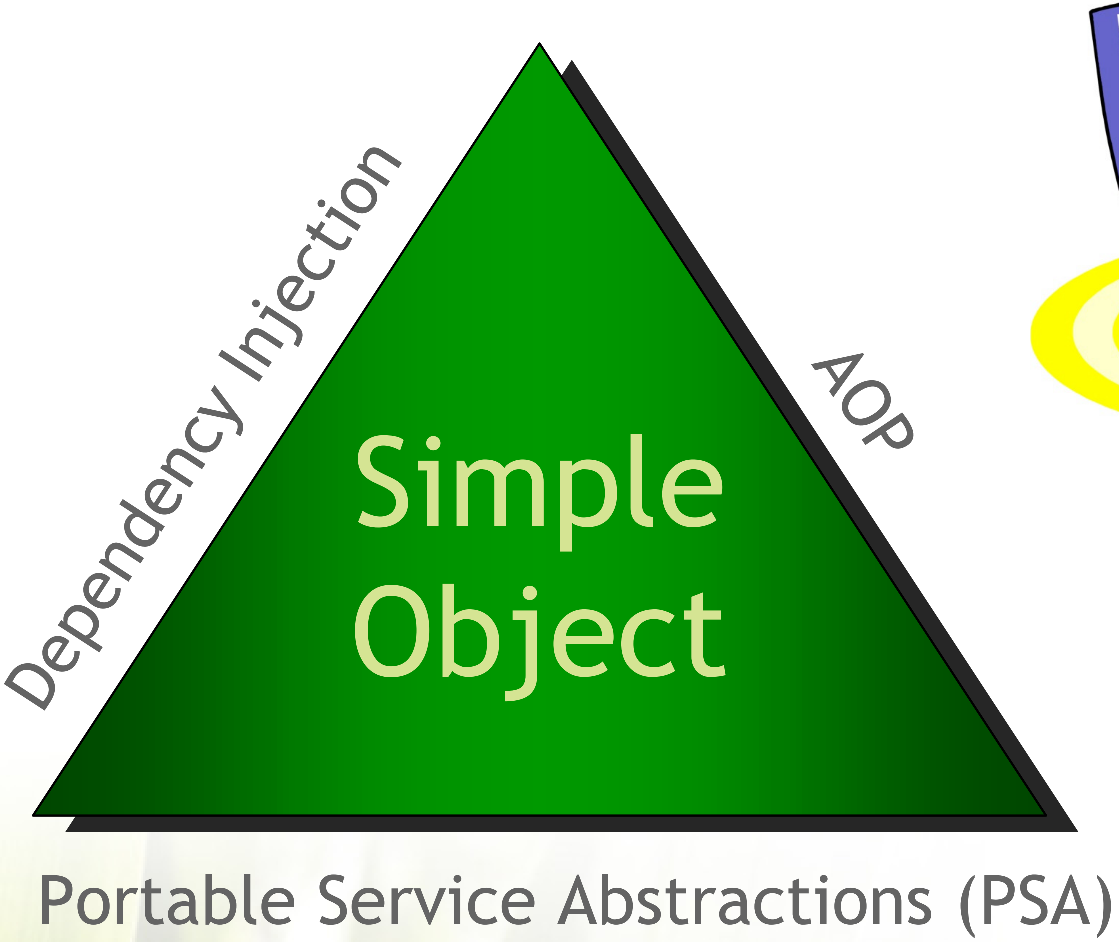
Core component model

+ Services
+ Patterns (Recipes)
+ Integration (Ecosystem)
+ Portability (Runs everywhere)

= Universal POJO programming model

- Embraced by more and more enterprise vendors
- Works on .NET, too

Enabling Technologies

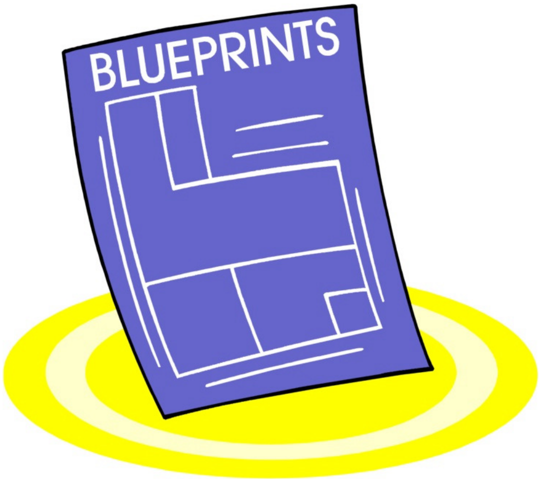


Dependency Injection

AOP

Simple Object

Portable Service Abstractions (PSA)



Spring 2.0 Goals

Spring 2.0 Core Goals

- Retain backward compatibility
- Continue realisation of core goals

- **Simplicity**
  - Make things easier
- **Power**
  - Make more things possible

Annotation Driven Transactions

```
@Transactional(readOnly=true)
interface TestService {

       @Transactional(readOnly=false,
             rollbackFor=DuplicateOrderIdException.class)
       void createOrder(Order order)
                    throws DuplicateOrderIdException;

       List queryByCriteria(Order criteria);
}
```

Annotation Driven Transactions

```
<bean
      class="org.springframework...DefaultAdvisorAutoProxyCreator"/>

<bean class="org.sfw...TransactionAttributeSourceAdvisor">
    <property name="transactionInterceptor
                    ref="transactionInter
</bea

<bea
                                       Transa
    <property name="transactionManager"
                    ref="transactionManag
    <property name="transactionAttributeSource">
        <bean class="org.sfw...AnnotationsTransactionAttributeSource">
             </bean>
    </property>
</bean>
```

```
<tx:annotation-driven/>
```

AOP: AspectJ Library Aspect

```
public aspect SystemArchitecture {

        public pointcut inDataAccessLayer() :
          within(org.springframework.samples.jpetstore.dao..*);

        public pointcut inDomainModel() :
          within(org.springframework.samples.jpetstore.domain.*);

        public pointcut inServiceLayer() :
          within(org.springframework.samples.jpetstore.service..*);

        ...
}
```

Use in Spring 2.0 XML

```
<aop:config>
    <aop:aspect ref="concurrentOperationExecutor">
    <aop:pointcut id="idempotentOperation"
          expression="..SystemArchitecture.inServiceLayer()"/>
    <aop:around
          pointcut-ref="idempotentOperation"
          method="doConcurrentOperation"/>
      </aop:aspect>
 </aop:config>
```

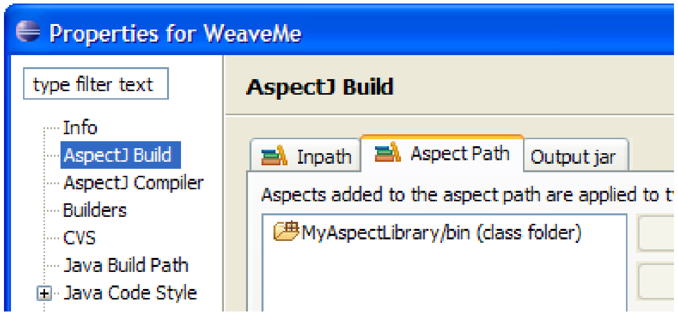
Use in @AspectJ Programming Model

```
@Aspect
public class ConcurrentOperationExecutor {

    @Around("..SystemArchitecture.inServiceLayer()")
    public Object doConcurrentOperation(
                   ProceedingJoinPoint pjp)
                   throws Throwable {
         // ...
    }
}
```

```
<aop:aspectj-autoproxy/>
```

Deployment choice: Compile with AspectJ

- Spring 2.0 offers a unique unified programming model for AOP
- We can compile the @AspectJ aspect with AJC or use AspectJ load time weaving



Aspects added to the aspect path are applied to t

MyAspectLibrary/bin (class folder)

Recap: Why AOP Matters

- Essential complement to DI to enable a POJO programming model
- *For more than mere interception*
  - Simple, not *simplistic*
- Both parts of the same big picture
- Go to Ramnivas’s talk to learn more…

AOP in Practice

- *Using AOP gives us competitive advantage in our content management solution. It’s helped us get a sophisticated product to market quickly. We’ve found the AOP capabilities of the Spring component model a key architectural enabler.*
  - John Newton
    - CTO and Founder, Alfresco

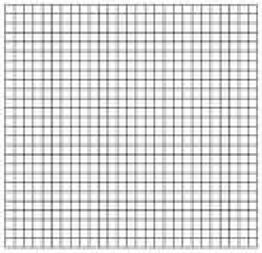
Spring 2.0: Building out the Spring Component Model

- Doesn't impose old assumptions
- Can form the basis for new kinds of applications
- **Enables the future, not tied to the past**
- Spring 2.0 added IoC container enhancements to facilitate build out

- New frontiers
  - Grid
  - Multi-language support
  - Batch
  - J2ME?

Grid-enabled POJOs with Spring

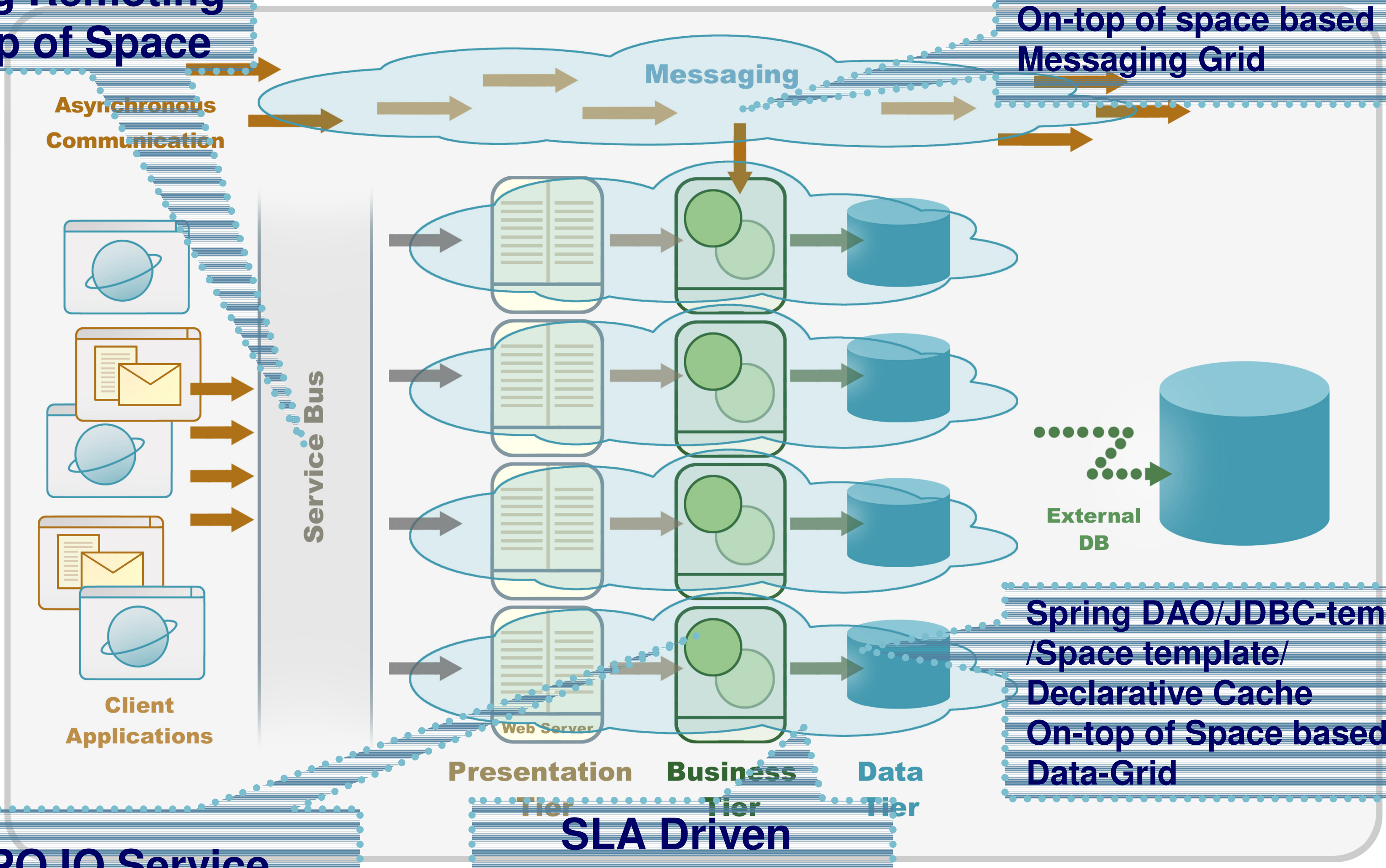
- Old monolithic application server model didn’t meet the needs of 2002-era applications well
  - Less and less appropriate as time goes on
  - In both the web and enterprise space
- Grid vendors see Spring POJO-based programming model as the ideal to scale out
  - Oracle Coherence
  - GigaSpaces
  - Terracotta
  - BEA WebLogic RealTime Server



Spring and Space-Based Architecture (SBA): How do they fit together?

- Spring is about application programming model
- Provides a POJO abstraction that decouples your code from platform code
- More general than Java EE programming models
  - No assumptions about underlying environment
  - Can accommodate radically different platforms underneath
- Goal
  - Decouple programming from deployment infrastructure
  - Allow choice of most appropriate runtime without re-architecture of business logic
  - One of Spring’s main purposes is to facilitate choice

Spring on SBA – non intrusive SBA



Spring Remoting
On-top of Space

JMS Template/Mule-ESB
On-top of space based
Messaging Grid

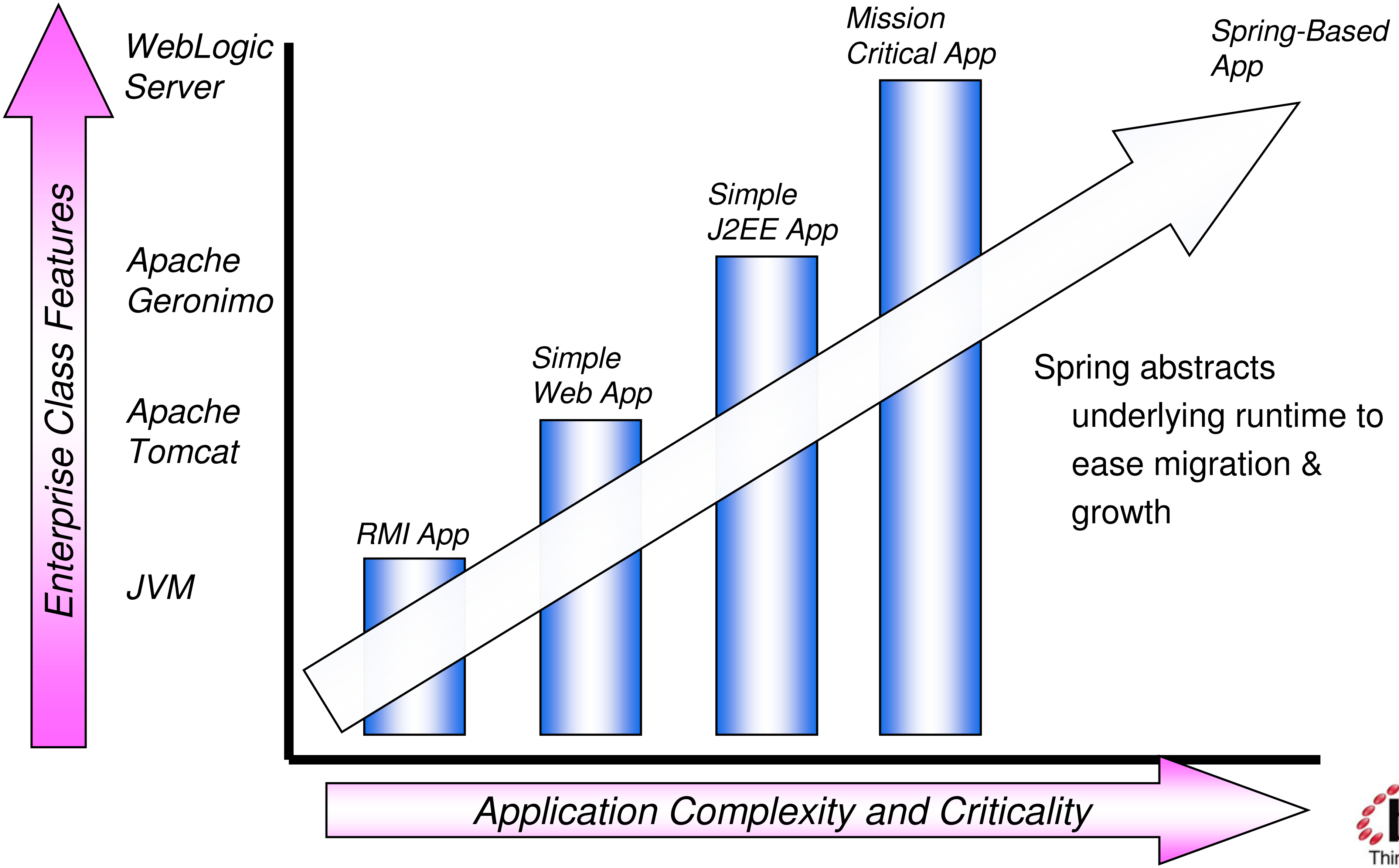
Spring DAO/JDBC-template
/Space template/
Declarative Cache
On-top of Space based
Data-Grid

POJO Service

SLA Driven
Spring Container

Spring Enables Applications To Evolve

- Spring prevents platform lock-in during design phase



Spring abstracts underlying runtime to ease migration & growth

Building out the Spring component model: Multi-language support

- Spring 2.0 offers a unique *cross language* component model
  - DI, AOP can apply to objects written in other languages
- "lang" namespace makes it easy to define beans written in languages other than Java
  - Java interface is the interoperability standand
  - JRuby
  - Groovy
  - BeanShell
  - integrate your own favourite

Multi-language support

```
<lang:jruby id="messenger"
     refresh-check-delay="5000"
     script-source=
       "classpath:org/xyz/messenger/Messenger.rb"
     script-interfaces="org.xyz…Messenger">
  <lang:property name="message"
                value="Hello World!"/>
</lang:jruby>

<bean id="messengerClient" class="…">
  <property name="messenger" ref="messenger"/>
</bean>
```

Multi-language support

```
require 'java'
include_class
   'org.springframework.scripting.Messenger'

class RubyMessenger < Messenger

 def setMessage(message)
  @@message = message
 end

 def getMessage
  @@message
 end
end
#RubyMessenger.new – optional
```

2006 - Year in Review

Themes of 2006

- Final releases
  - Acegi Security for Spring
  - Spring 2.0
  - Spring Web Flow
- First release
  - Spring Web Services
    - Contract-based Web Services
- Growing adoption
  - Spring became ubiquitous, especially in the enterprise
  - Forrester Report:
    - *A majority of [enterprise Java] users reported using Spring*

May: Pitchfork Project
Not the band: *Project Pitchfork*

- Interface21 and BEA collaboration
  - Build new features of Java EE 5 programming model (injection and interception) on top of Spring
  - Used in core of WebLogic 10 and useable separately
- Provides WebLogic with enhanced time to market
- Underpinned by solid, proven Spring implementation
  - Not experimental
- Exposes Spring programming model so that developers don’t hit a wall when Java EE injection and interception is not enough

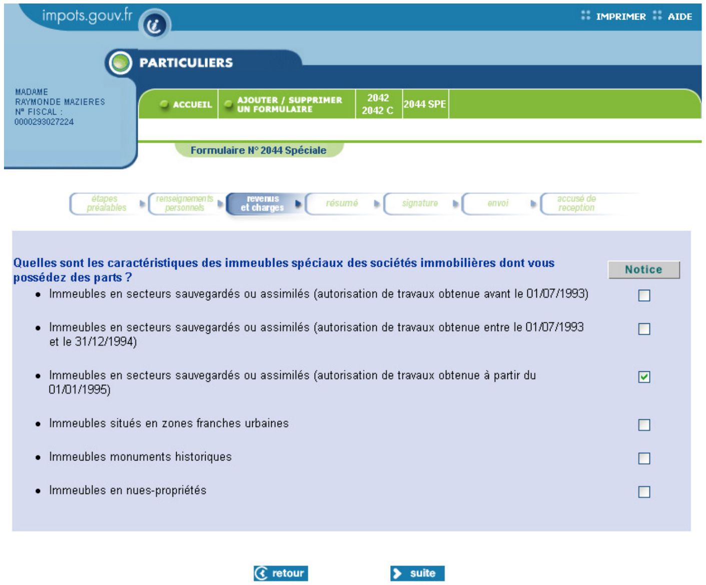
May: Oracle Fusion Middleware Building on Spring

- Also in a JavaOne Keynote, senior Oracle executive Thomas Kurian described how Oracle are also using Spring in their next generation middleware

- Oracle are using Spring in numerous applications…

June

- French Online Taxation web interface fully rolled out, based on Spring
- Handles taxation requirements for 34 million taxpayers
- French Minister of Finance described project as a “true triumph of IT”



MADAME
RAYMONDE MAZIERES
N° FISCAL :
0000293027224

Quelles sont les caractéristiques des immeubles spéciaux des sociétés immobilières dont vous possédez des parts ?

- Immeubles en secteurs sauvegardés ou assimilés (autorisation de travaux obtenue avant le 01/07/1993) □
- Immeubles en secteurs sauvegardés ou assimilés (autorisation de travaux obtenue entre le 01/07/1993 et le 31/12/1994) □
- Immeubles en secteurs sauvegardés ou assimilés (autorisation de travaux obtenue à partir du 01/01/1995) ☑
- Immeubles situés en zones franches urbaines □
- Immeubles monuments historiques □
- Immeubles en nues-propriétés □

Spring has had a significant impact on the productivity of our J2EE developments. Thanks to its simple yet powerful programming model we were able to significantly improve time to market and build better quality solutions.

Thomas van de Velde
Lead Java Architect
Accenture Delivery Architectures

Spring-based application Processes almost all UK Inter-bank payments at Voca

Part of UK’s Critical National Infrastructure

Voca process Direct Debits, Direct Credits and Standing Orders to move money between banks

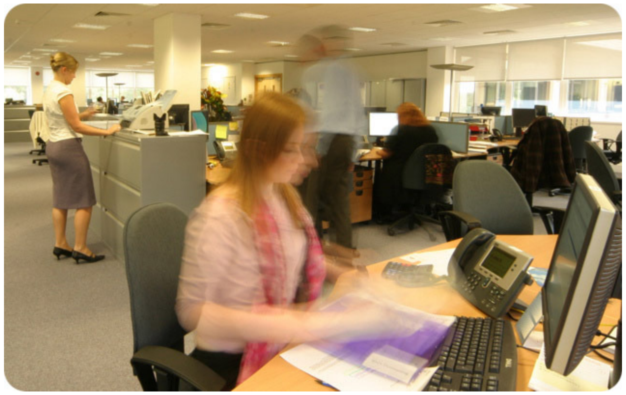
Over 5 billion transactions worth €4.5 trillion in 2005

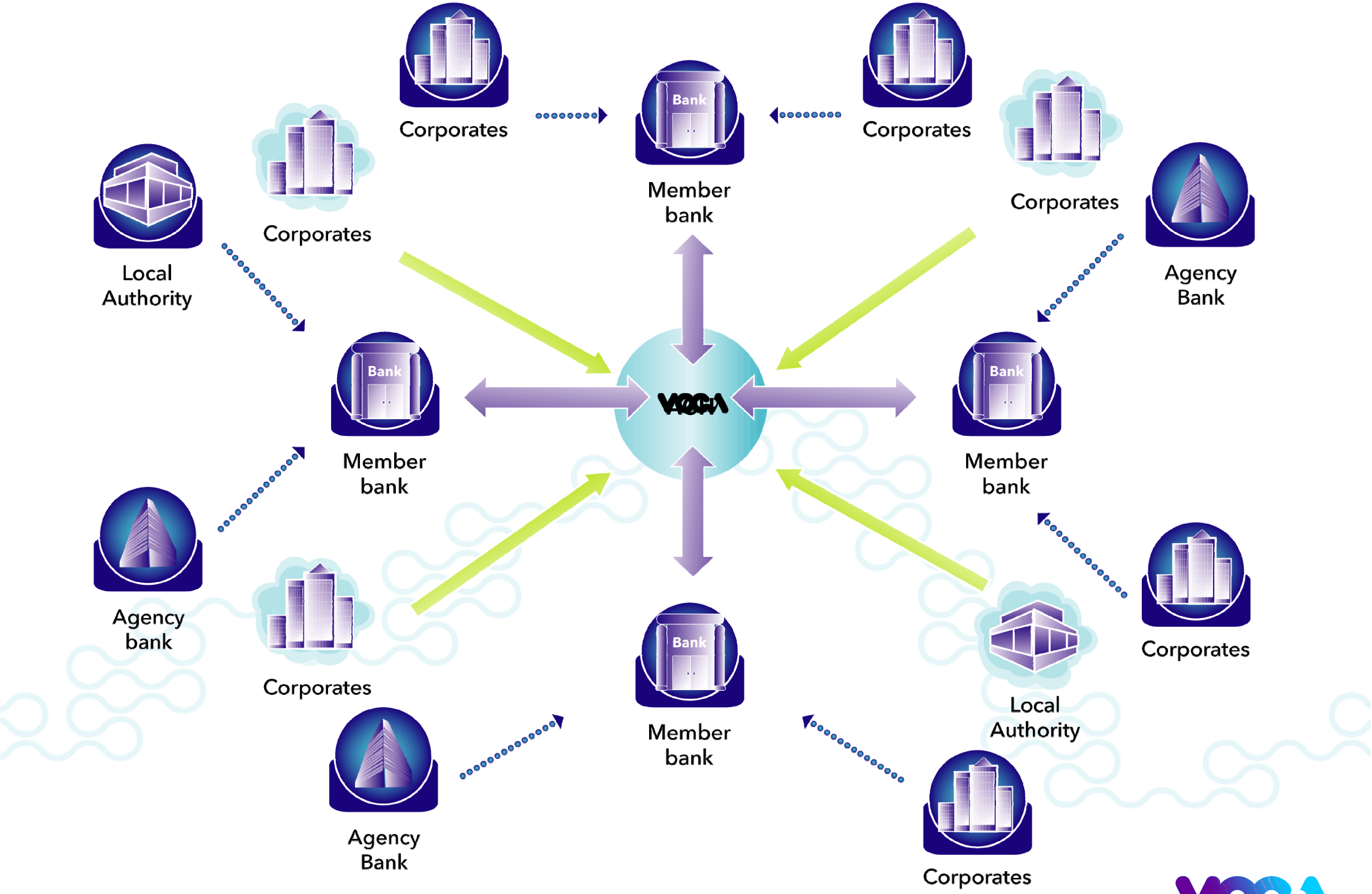
Some 15% of Europe’s Direct Debits and Direct Credits are handled by Voca

Over 70% of the UK population use Direct Debits to pay household bills; Direct Credits are used to pay over 90% of UK salaries

Over 80 million items on a peak day

Voca have never lost a payment

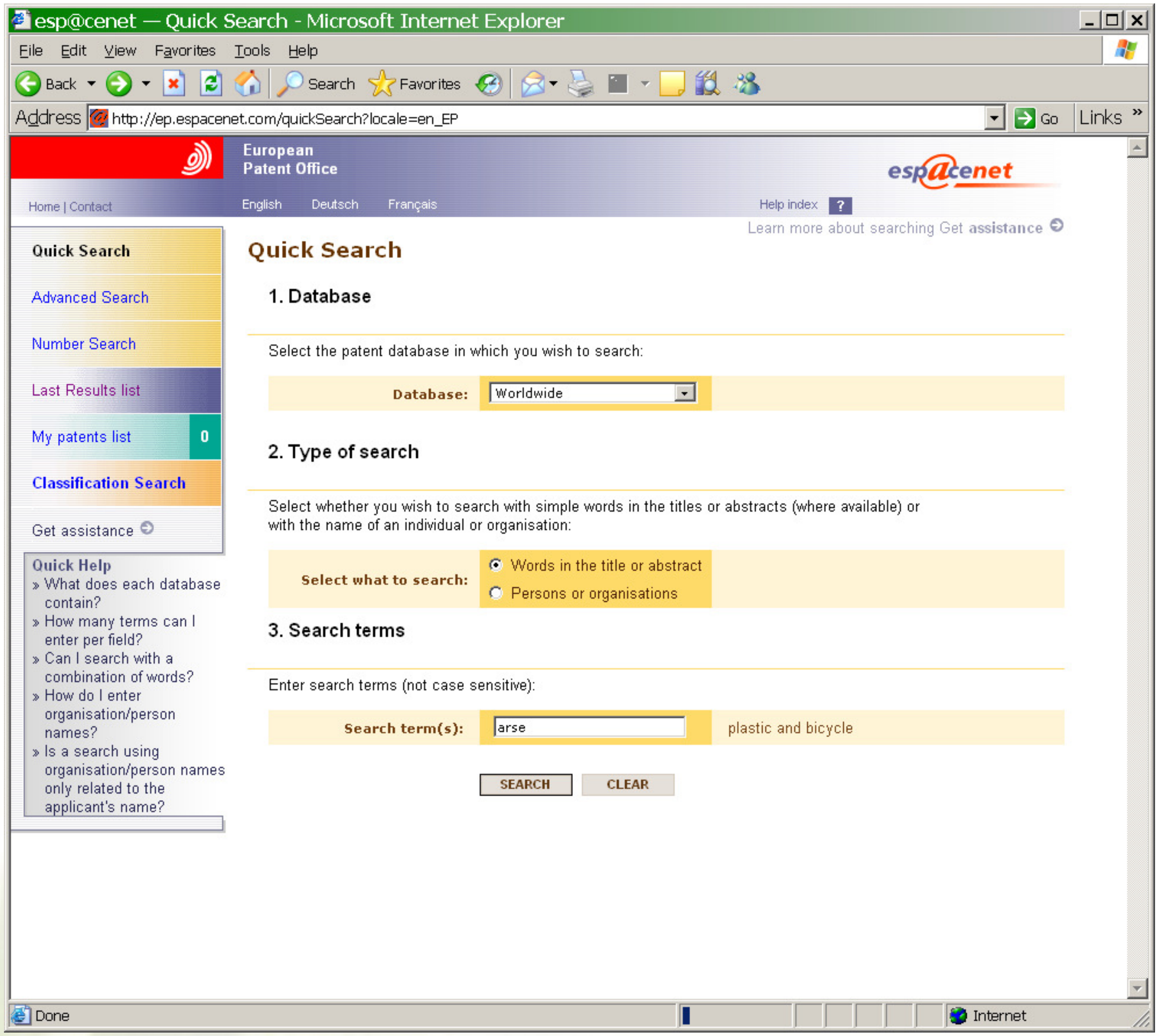



Corporates
Member bank
Corporates
Corporates
Local Authority
Corporates
Agency Bank
Member bank
VOCA
Member bank
Agency bank
Corporates
Corporates
Local Authority
Member bank
Agency Bank
Corporates

VOCA

European Patent Office: Spring Success Story

- Uses Spring and Spring Web Flow extensively
- Achieving excellent throughput and performance
- Much lower hardware utilization than with previous J2EE architecture



Learn more about searching Get assistance

Quick Search

Quick Help
» What does each database contain?
» How many terms can I enter per field?
» Can I search with a combination of words?
» How do I enter organisation/person names?
» Is a search using organisation/person names only related to the applicant's name?

1. Database

Select the patent database in which you wish to search:

Database: Worldwide

2. Type of search

Select whether you wish to search with simple words in the titles or abstracts (where available) or with the name of an individual or organisation:

Select what to search:
● Words in the title or abstract
○ Persons or organisations

3. Search terms

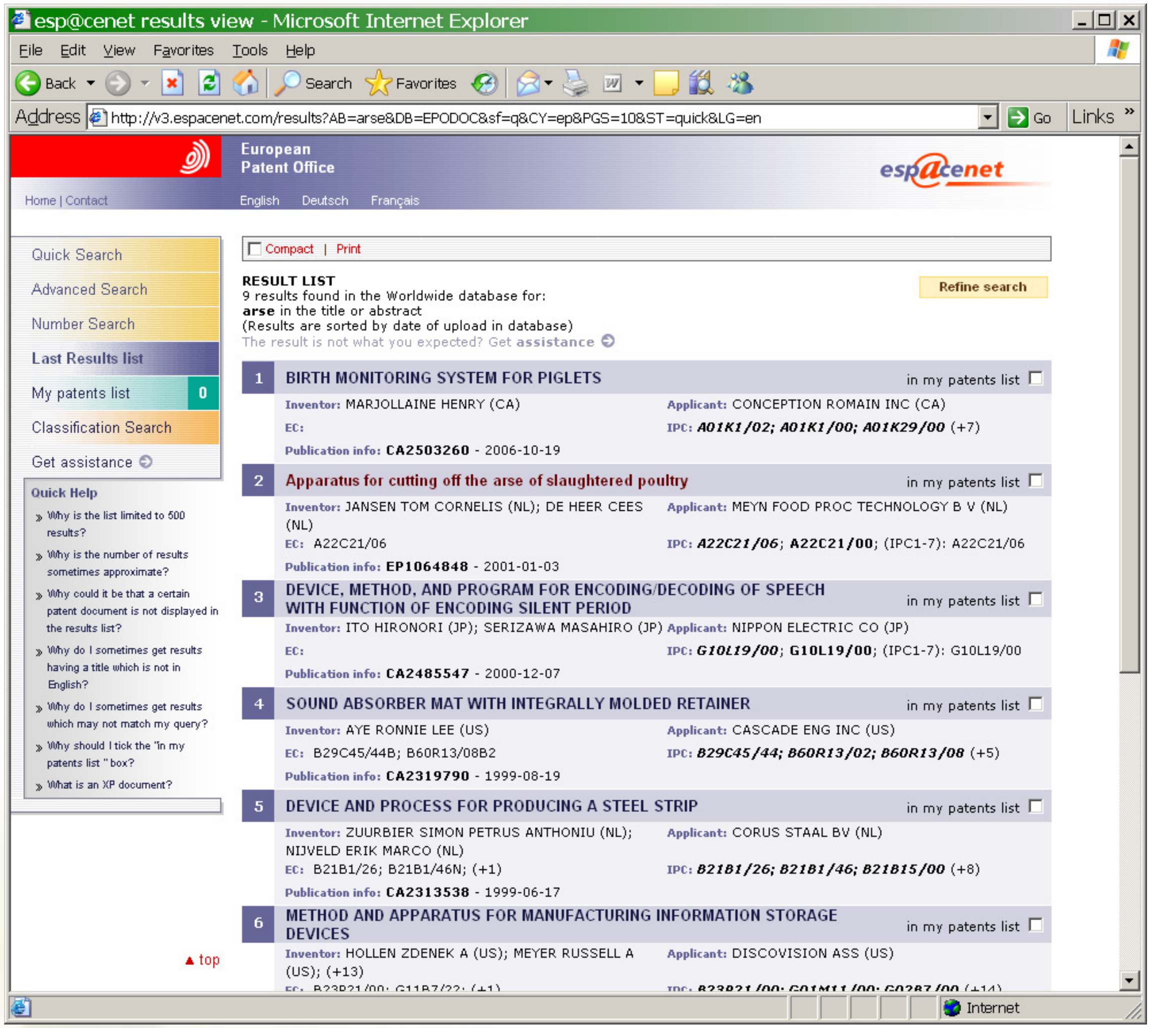
Enter search terms (not case sensitive):

Search term(s): arse plastic and bicycle

SEARCH CLEAR

Worldwide Patent Search

- Spring end to end
- Uses Spring MVC



http://v3.espacenet.com/results?AB=arse&DB=EPODOC&sf=q&CY=ep&PGS=10&ST=quick&LG=en

Quick Help

- Why is the list limited to 500 results?
- Why is the number of results sometimes approximate?
- Why could it be that a certain patent document is not displayed in the results list?
- Why do I sometimes get results having a title which is not in English?
- Why do I sometimes get results which may not match my query?
- Why should I tick the "in my patents list " box?
- What is an XP document?

RESULT LIST
9 results found in the Worldwide database for:
arse in the title or abstract
(Results are sorted by date of upload in database)
The result is not what you expected? Get **assistance**

1 BIRTH MONITORING SYSTEM FOR PIGLETS — in my patents list □
Inventor: MARJOLLAINE HENRY (CA) — Applicant: CONCEPTION ROMAIN INC (CA)
EC: — IPC: ***A01K1/02; A01K1/00; A01K29/00*** (+7)
Publication info: **CA2503260** - 2006-10-19

2 Apparatus for cutting off the arse of slaughtered poultry — in my patents list □
Inventor: JANSEN TOM CORNELIS (NL); DE HEER CEES (NL) — Applicant: MEYN FOOD PROC TECHNOLOGY B V (NL)
EC: A22C21/06 — IPC: ***A22C21/06***; **A22C21/00**; (IPC1-7): A22C21/06
Publication info: **EP1064848** - 2001-01-03

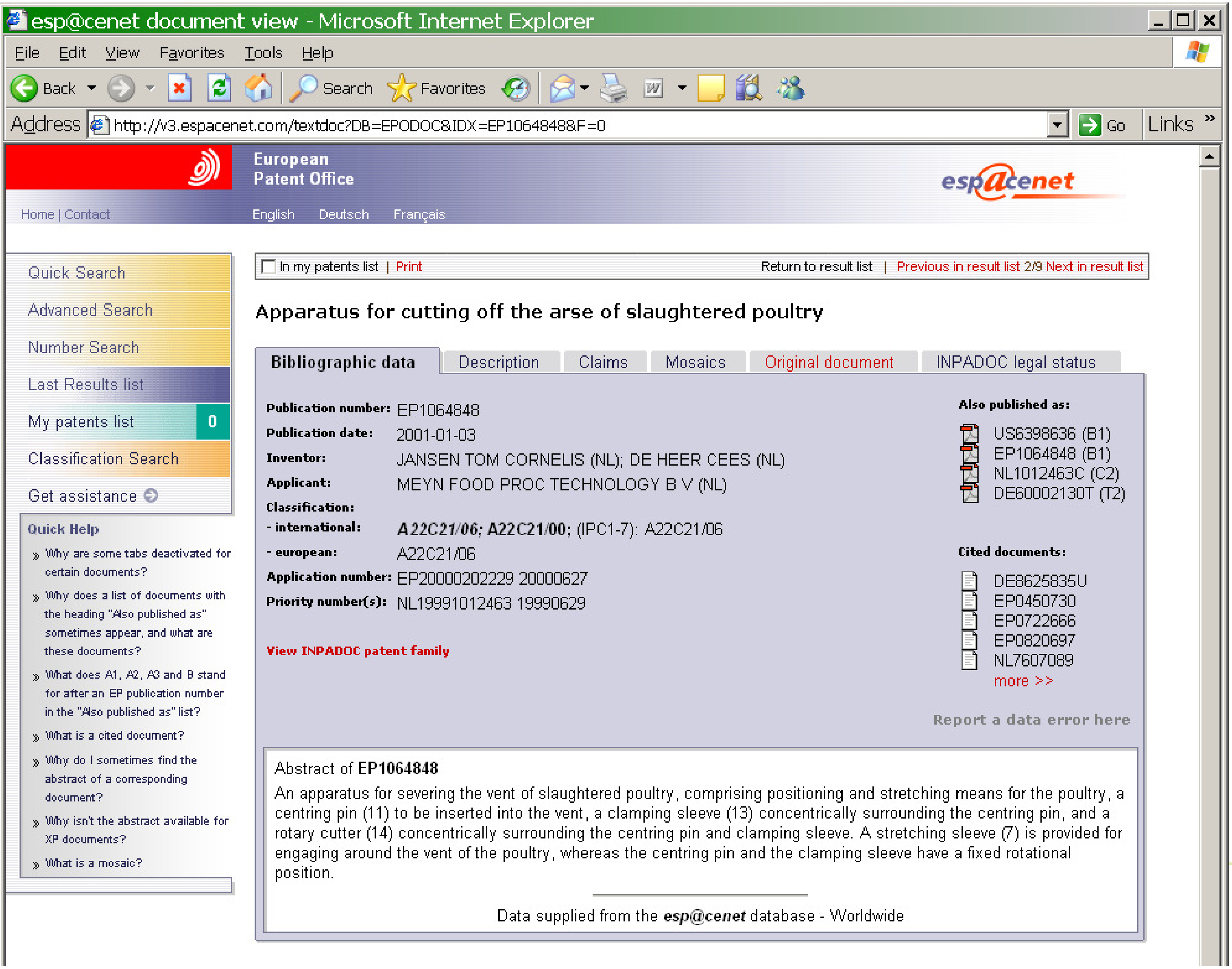
3 DEVICE, METHOD, AND PROGRAM FOR ENCODING/DECODING OF SPEECH WITH FUNCTION OF ENCODING SILENT PERIOD — in my patents list □
Inventor: ITO HIRONORI (JP); SERIZAWA MASAHIRO (JP) — Applicant: NIPPON ELECTRIC CO (JP)
EC: — IPC: ***G10L19/00***; **G10L19/00**; (IPC1-7): G10L19/00
Publication info: **CA2485547** - 2000-12-07

4 SOUND ABSORBER MAT WITH INTEGRALLY MOLDED RETAINER — in my patents list □
Inventor: AYE RONNIE LEE (US) — Applicant: CASCADE ENG INC (US)
EC: B29C45/44B; B60R13/08B2 — IPC: ***B29C45/44; B60R13/02; B60R13/08*** (+5)
Publication info: **CA2319790** - 1999-08-19

5 DEVICE AND PROCESS FOR PRODUCING A STEEL STRIP — in my patents list □
Inventor: ZUURBIER SIMON PETRUS ANTHONIU (NL); NIJVELD ERIK MARCO (NL) — Applicant: CORUS STAAL BV (NL)
EC: B21B1/26; B21B1/46N; (+1) — IPC: ***B21B1/26; B21B1/46; B21B15/00*** (+8)
Publication info: **CA2313538** - 1999-06-17

6 METHOD AND APPARATUS FOR MANUFACTURING INFORMATION STORAGE DEVICES — in my patents list □
Inventor: HOLLEN ZDENEK A (US); MEYER RUSSELL A (US); (+13) — Applicant: DISCOVISION ASS (US)
EC: B23B21/00; G11B7/22; (+1) — IPC: ***B23B21/00; G01M11/00; G02B7/00*** (+14)

Hours of Spring-powered Fun!



http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=EP1064848&F=0

Quick Help

- Why are some tabs deactivated for certain documents?
- Why does a list of documents with the heading "Also published as" sometimes appear, and what are these documents?
- What does A1, A2, A3 and B stand for after an EP publication number in the "Also published as" list?
- What is a cited document?
- Why do I sometimes find the abstract of a corresponding document?
- Why isn't the abstract available for XP documents?
- What is a mosaic?

□ In my patents list | Print   Return to result list | Previous in result list 2/9 Next in result list

Apparatus for cutting off the arse of slaughtered poultry

Bibliographic data | Description | Claims | Mosaics | Original document | INPADOC legal status

Publication number: EP1064848
Publication date: 2001-01-03
Inventor: JANSEN TOM CORNELIS (NL); DE HEER CEES (NL)
Applicant: MEYN FOOD PROC TECHNOLOGY B V (NL)
Classification:
- international: ***A22C21/06; A22C21/00;*** (IPC1-7): A22C21/06
- european: A22C21/06
Application number: EP20000202229 20000627
Priority number(s): NL19991012463 19990629

View INPADOC patent family

Also published as:

US6398636 (B1)
EP1064848 (B1)
NL1012463C (C2)
DE60002130T (T2)

Cited documents:

DE8625835U
EP0450730
EP0722666
EP0820697
NL7607089
more >>

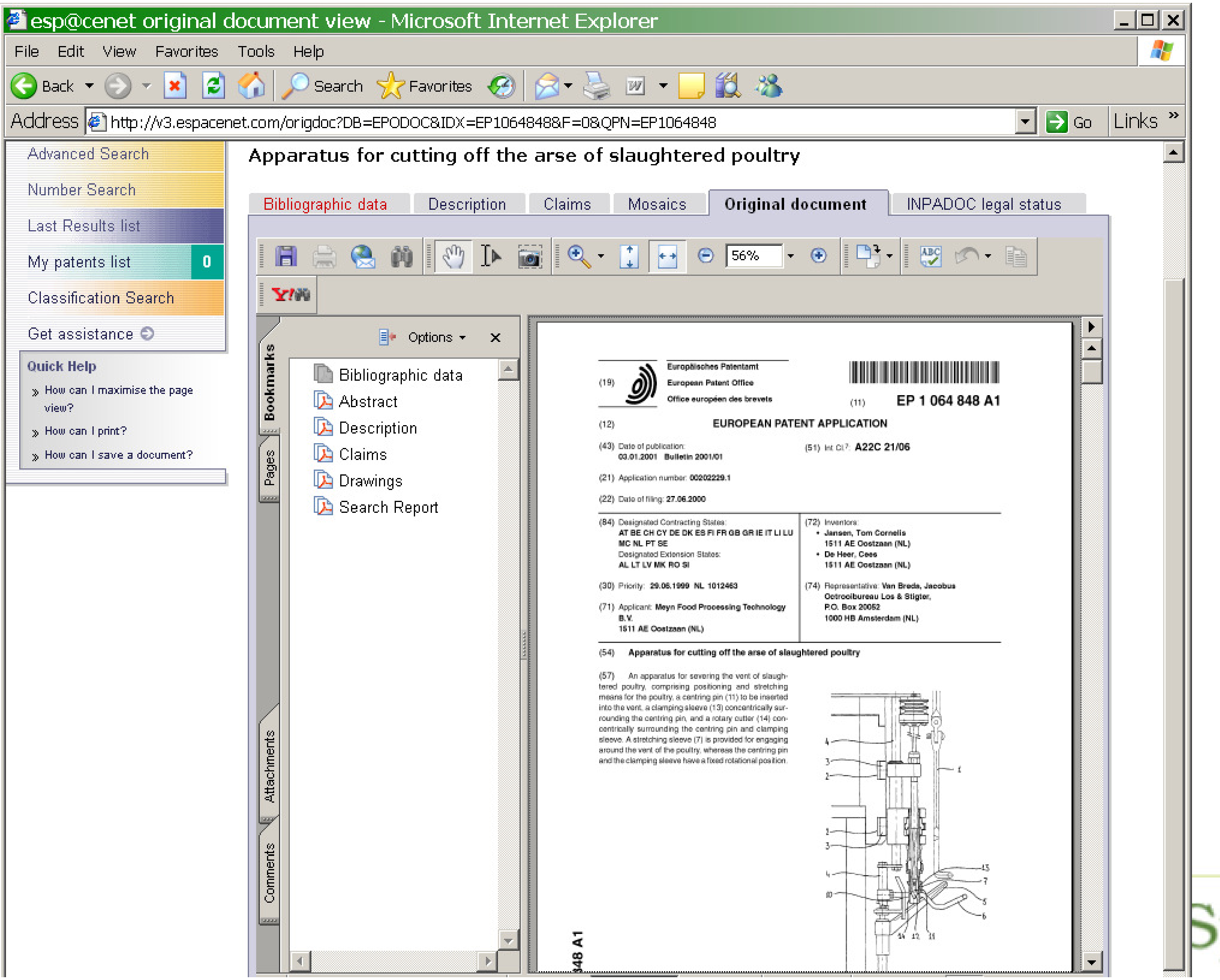
Report a data error here

Abstract of **EP1064848**

An apparatus for severing the vent of slaughtered poultry, comprising positioning and stretching means for the poultry, a centring pin (11) to be inserted into the vent, a clamping sleeve (13) concentrically surrounding the centring pin, and a rotary cutter (14) concentrically surrounding the centring pin and clamping sleeve. A stretching sleeve (7) is provided for engaging around the vent of the poultry, whereas the centring pin and the clamping sleeve have a fixed rotational position.

Data supplied from the *esp@cenet* database - Worldwide

Full Document Retrieval



Quick Help

- How can I maximise the page view?
- How can I print?
- How can I save a document?

Apparatus for cutting off the arse of slaughtered poultry

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 064 848 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
03.01.2001 Bulletin 2001/01

(51) Int Cl.⁷: A22C 21/06

(21) Application number: 00202229.1

(22) Date of filing: 27.06.2000

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(72) Inventors:
- Jansen, Tom Cornelis
1511 AE Oostzaan (NL)
- De Heer, Cees
1511 AE Oostzaan (NL)

(30) Priority: 29.06.1999 NL 1012463

(74) Representative: Van Breda, Jacobus
Octrooibureau Los & Stigter,
P.O. Box 20052
1000 HB Amsterdam (NL)

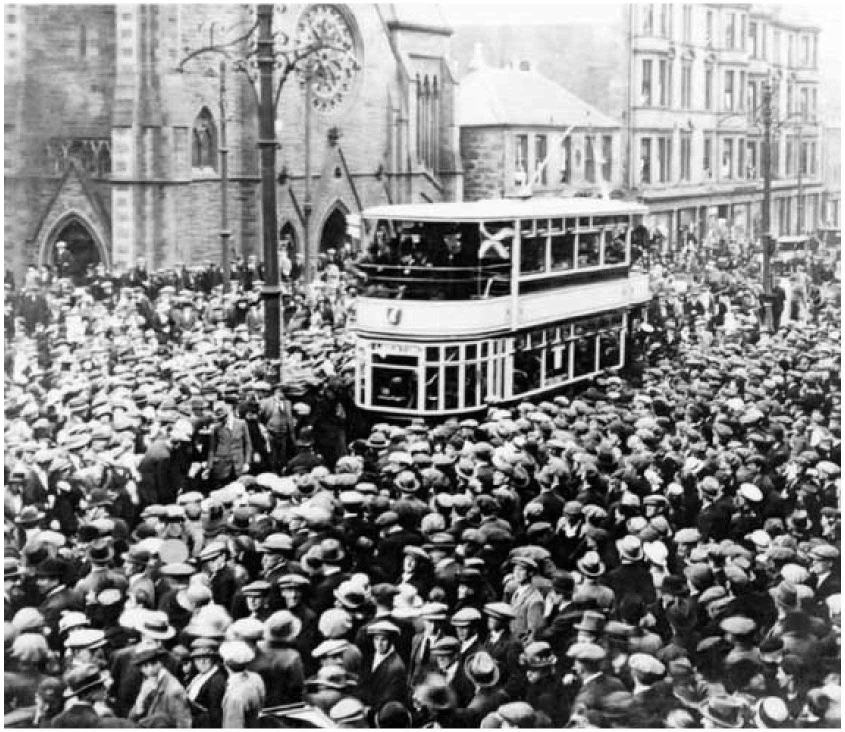
(71) Applicant: Meyn Food Processing Technology B.V.
1511 AE Oostzaan (NL)

(54) Apparatus for cutting off the arse of slaughtered poultry

(57) An apparatus for severing the vent of slaughtered poultry, comprising positioning and stretching means for the poultry, a centring pin (11) to be inserted into the vent, a clamping sleeve (13) concentrically surrounding the centring pin, and a rotary cutter (14) concentrically surrounding the centring pin and clamping sleeve. A stretching sleeve (7) is provided for engaging around the vent of the poultry, whereas the centring pin and the clamping sleeve have a fixed rotational position.

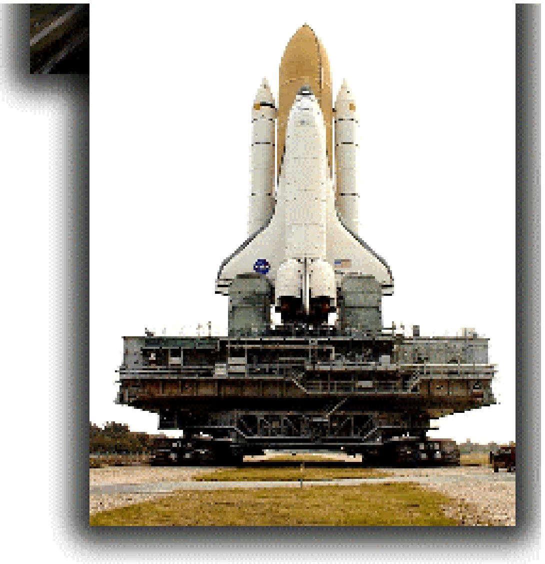
September: 1 million downloads

- The Spring Framework passed 1 million downloads from its primary web site
  - Excludes other download sources
  - Excludes Acegi Security and sister projects
- Took down Interface21 servers
  - It's OK, they're not running Spring
- Powered on to 1.2 million downloads and beyond quickly



Themes for 2007...

Themes for 2007



- Spring 2.0 was a *launchpad*, as well as an end in itself
- Building out the Spring component model
  - Spring OSGi
- The Spring Portfolio

OSG - *what?*

				1 s					
		1 o	p	e	n				
				r					
				v					
3 i	n	i	t	i	a	t	i	v	e
				c					
	2 g	a	t	e	w	a	y		
				s					

INTERFACE21

OSG*injection*

The Dynamic Module System for Java

Spring

OSGi: A Module System…

- Partition a system into modules
  - "bundles"
- Strict visibility rules
- Resolution process
  - satisfies dependencies of a module
- Understands versioning!

...and it's Dynamic!

- Modules can be
  - installed
  - started
  - stopped
  - uninstalled
  - updated
- …at runtime!

Spring and OSGi: Complementary Technologies

- Both are the best at what they do
  - Injection/AOP component model
  - Dynamic runtime infrastructure
- Both run *everywhere*
- Little overlap
- Natural to combine dynamic power of OSGi with ease of use of Spring component model
- Spring/OSGi integration may make its way into OSGi specifications

Spring OSGi - Project goals

- Use Spring container to configure modules (bundles)
- Make it easy to publish and consume services
  - across a number of bundles
- Enable applications to be coded without dependency on OSGi APIs
  - easy unit and integration testing
- Provide the needed bundles and infrastructure to deploy OSGi based applications to application servers

Project collaborators

- **Led by Interface21**
- Committers from BEA and Oracle also active on the project
- Input to the specification and direction from
  - OSGi Alliance (technical director and CTO)
  - BEA, Oracle, IBM
  - Eclipse Equinox
  - Felix
  - and many individuals

Spring modules

- Spring modules packaged as OSGi bundles
  - spring-core
  - spring-beans
  - spring-aop
  - etc.
- All the necessary import and export package headers defined
- Enables an OSGi application to import and use Spring packages and services
- Currently done in Spring-OSGi project
  - spring module jars will come this way "out of the box" in Spring 2.1

Spring Makes it Easy! - Exporting a Service

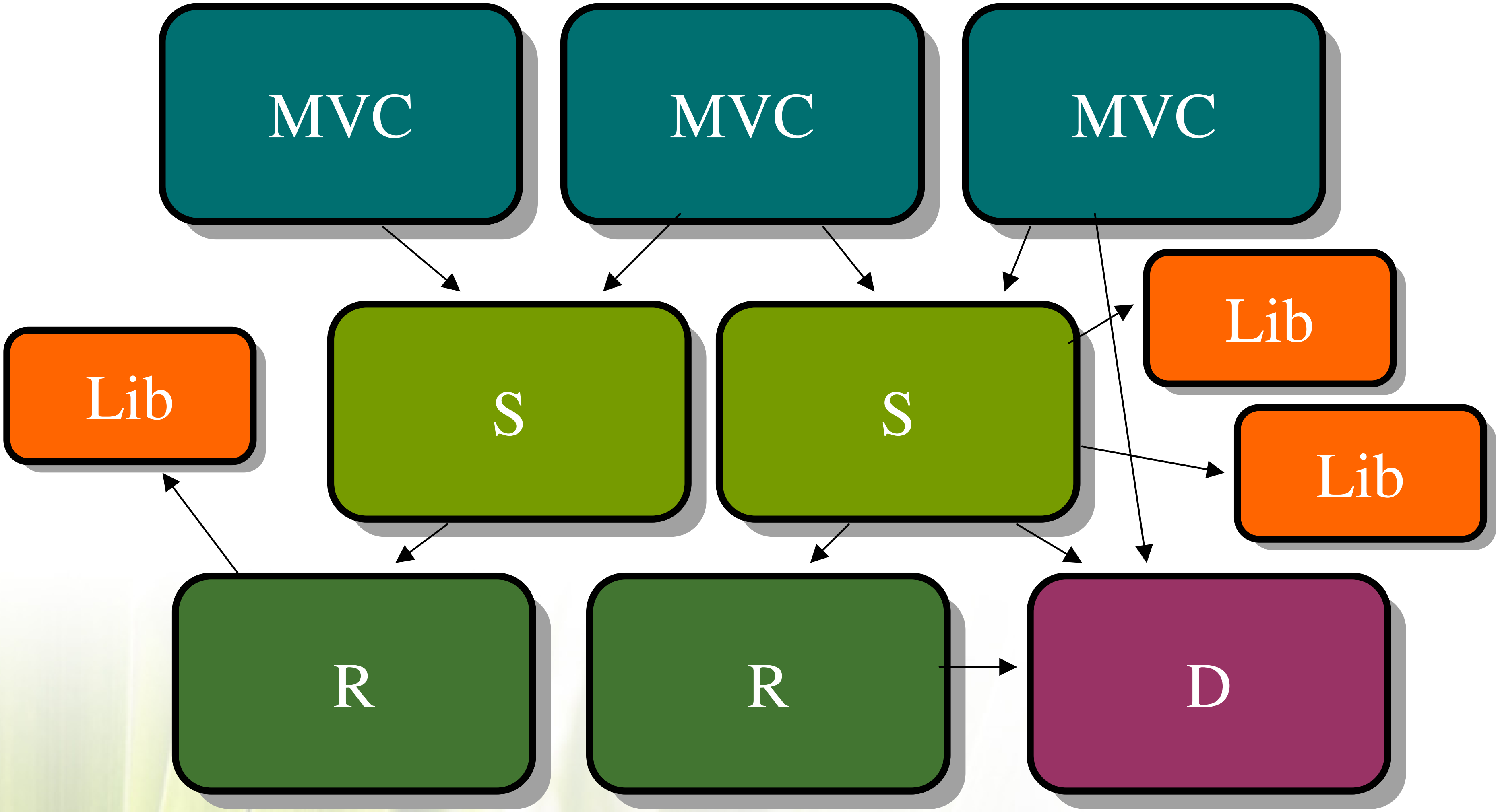
```
<osgi:service id="simpleServiceOsgi"
  ref="simpleService"
  interface=
     "org.sfw.osgi.samples.ss.MyService"/>
```

Spring Makes it Easy! - Importing a Service

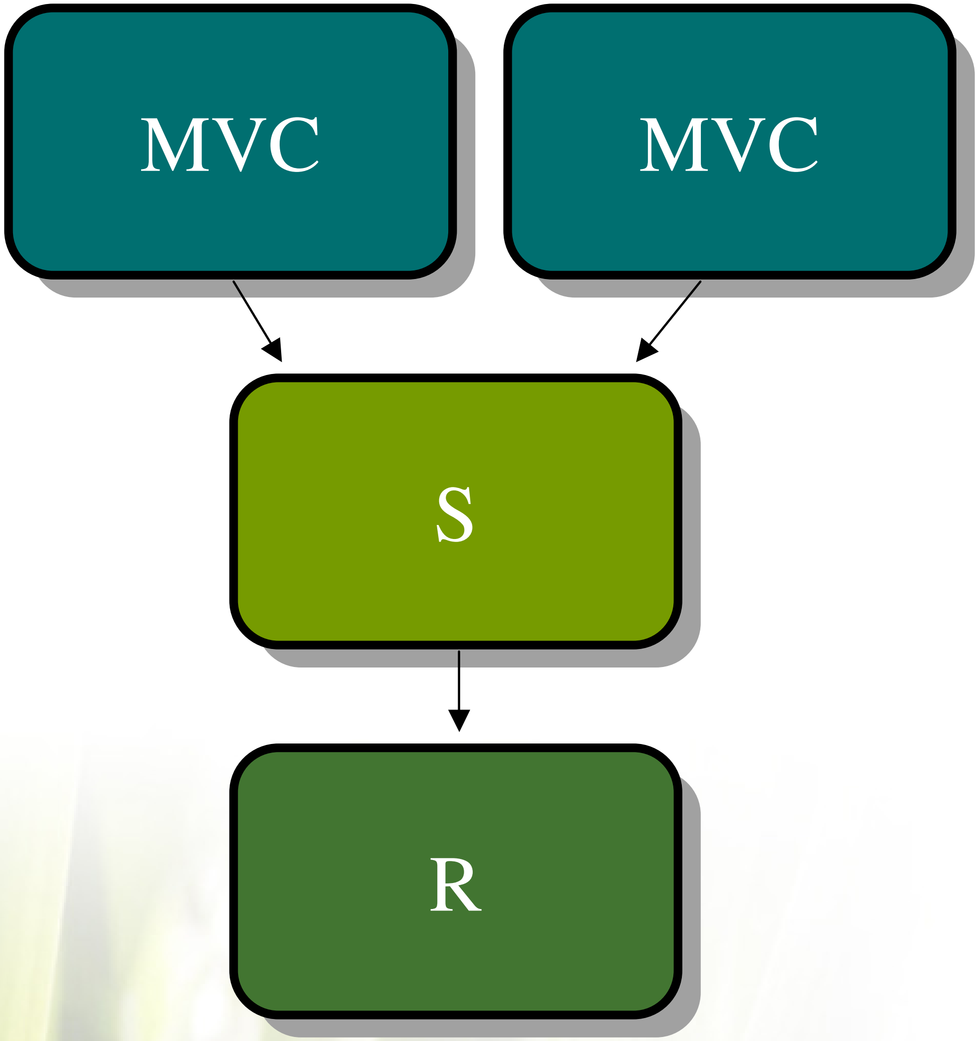
```
<osgi:reference id="aService"
  interface=
     "org.sfw.osgi.samples.ss.MyService"/>
```

Visibility

Each bundle is a segregated class space

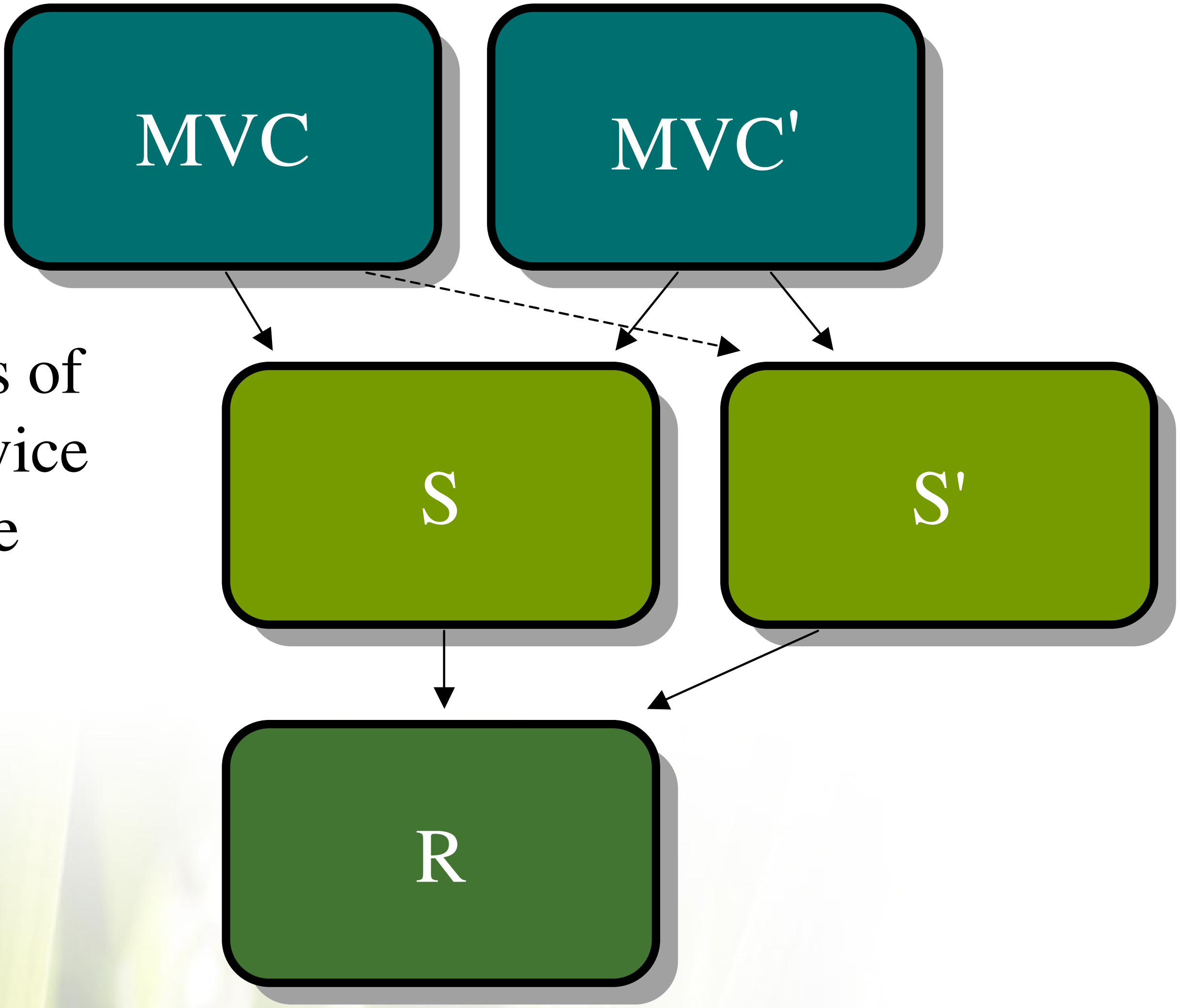


Versioning



Versioning

Two versions of the same service types… at the same time!



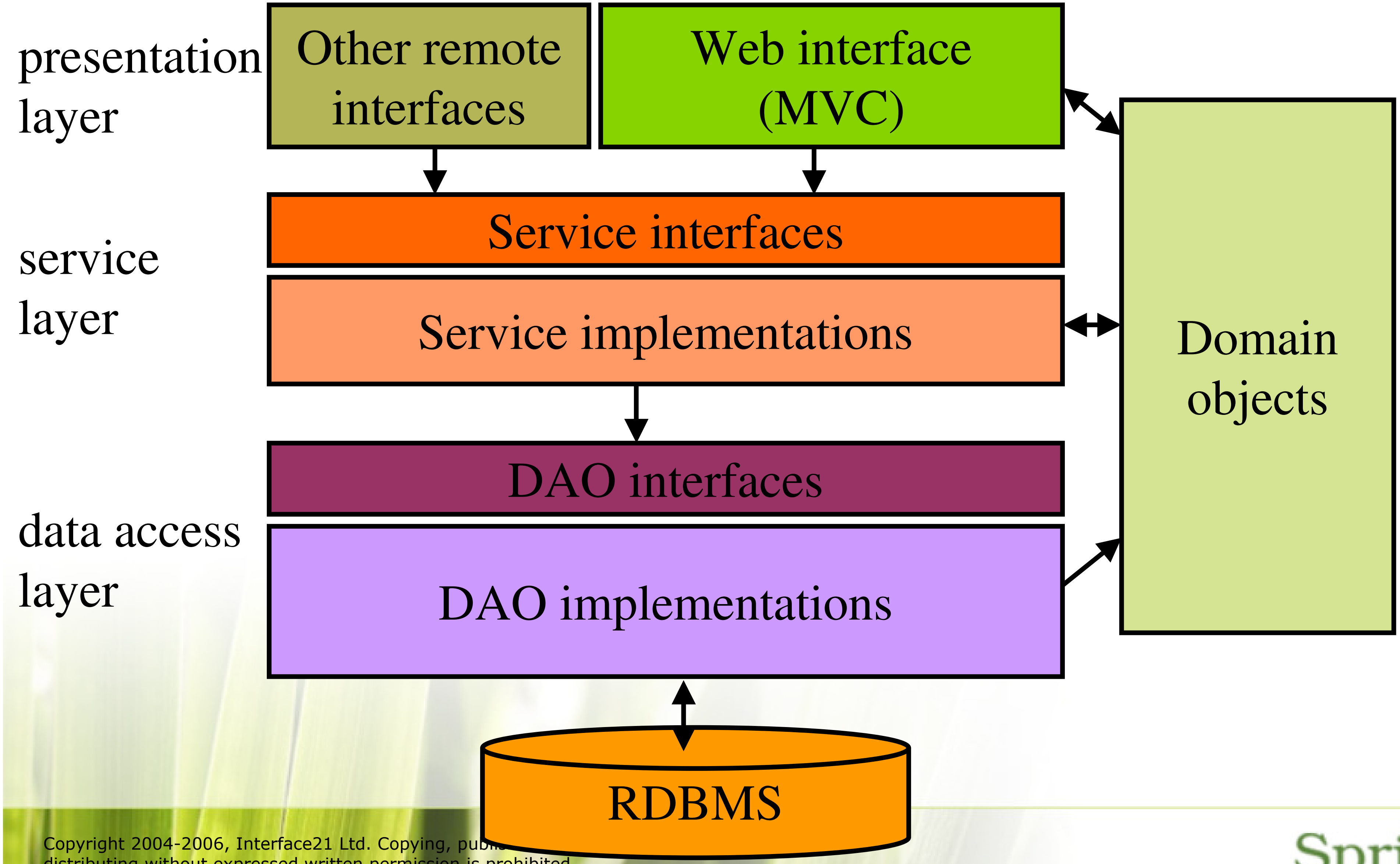
The Big Picture

Spring

The Spring Portfolio

- **Not just about the Spring *Framework***
- The Spring component model and Spring “recipes” provide a solid basis for a rich and growing ecosystem
- Much progress in last 12 months

Typical application layering



presentation layer

Other remote interfaces

Web interface (MVC)

service layer

Service interfaces

Service implementations

Domain objects

data access layer

DAO interfaces

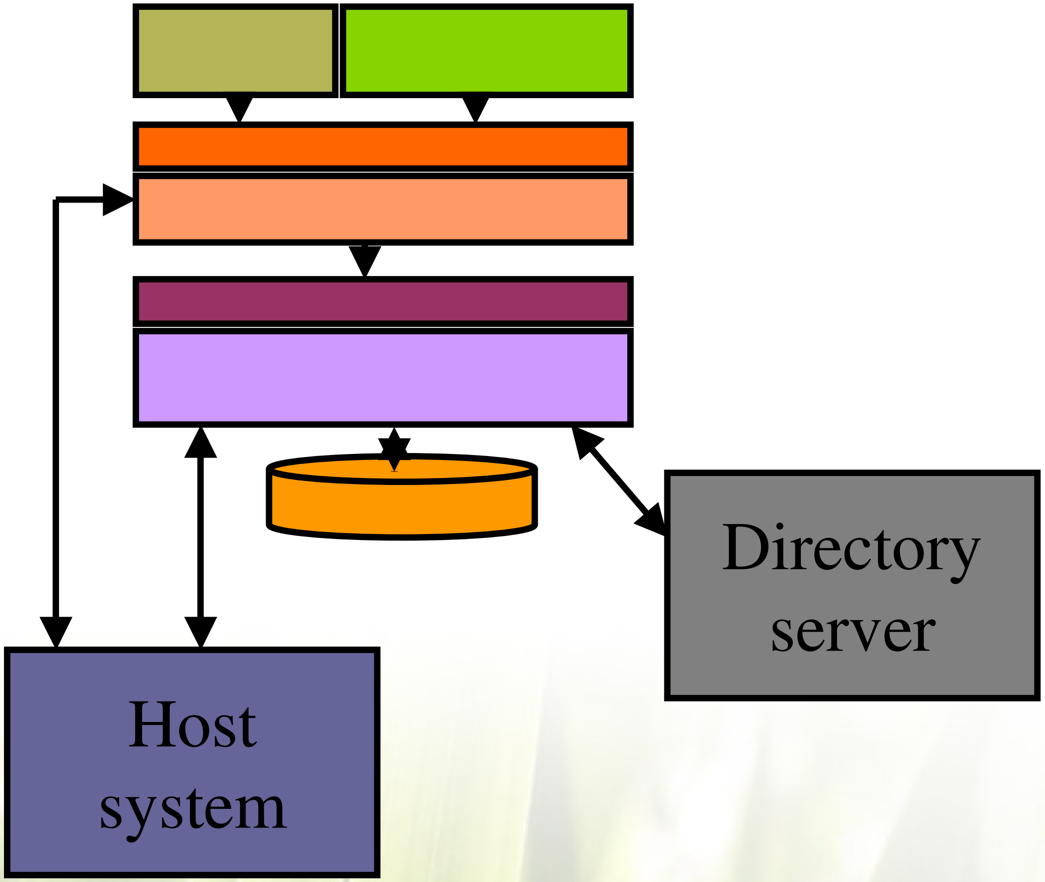
DAO implementations

RDBMS

Data access layer

- JDBC, iBATIS, JDO, Hibernate, TopLink,
- Introduction of JPA support in Spring 2.0
  - Open JPA, Toplink Essentials, Hibernate
- Further enhancements planned for Spring 2.1
  - simple: JPA namespace
  - powerful: support for common non-standard features
    - e.g. criteria API
    - e.g. user types

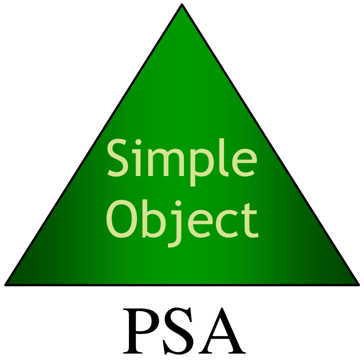
Other back-ends...



- Spring also has support for **accessing enterprise systems**
  - CICS
  - IMS
  - …
  - *JCA CCI: CciTemplate*
- **Spring LDAP** project adds support for interacting with LDAP servers

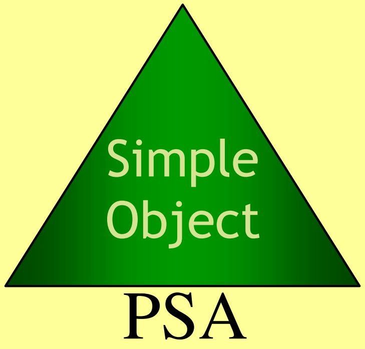
Spring LDAP

- www.springframework.org/ldap
- LdapTemplate
  - resource acquisition and release
  - iteration
  - exception translation
    - NamingException -> DataAccessException
- Utility classes
  - filters, LDAP paths and attributes



LdapTemplate

```
public List findByLastName(String lastName) {
  String filterExpr = createFilter(lastName);
  return ldapTemplate.search("",filterExpr,
    new AttributesMapper() {
      public Object mapFromAttributes(
        Attributes attrs) throws NamingException {
        Person person = new Person();
        person.setFullName(
          (String)attrs.get("cn").get());
        person.setLastName(
          (String)attrs.get("sn").get());
        …
        return person;
      }
    });
  }
}
```

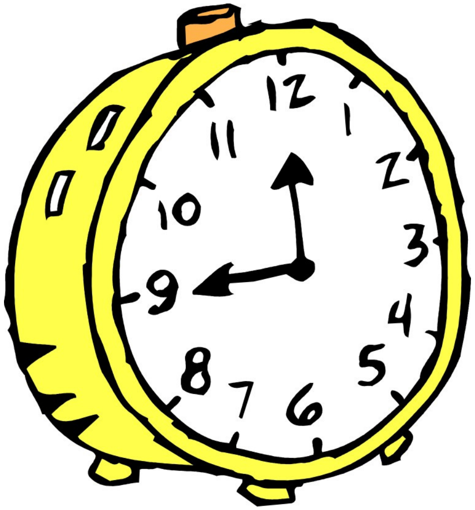


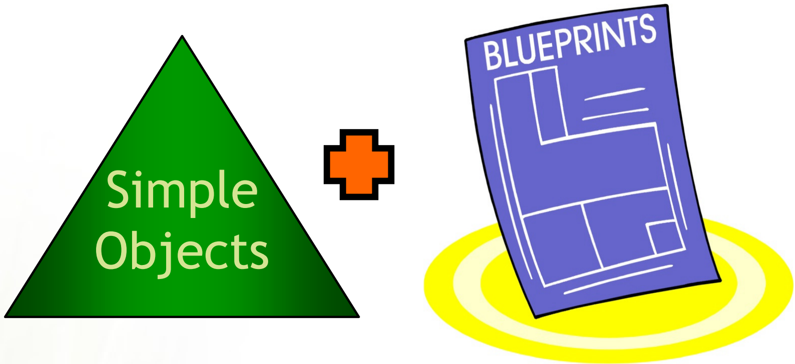
Service layer

- You all know how to define business services and configure them with Spring…
- In this section:
  - scheduling
  - asynchronous tasks
  - messaging
  - ESB integration
  - clustering

Scheduling

- JDK Timer and **Quartz**
- Run jobs in background
  - cron expressions
  - simple triggers
- Method invoking job detail factory bean makes executing operation on a bean trivial

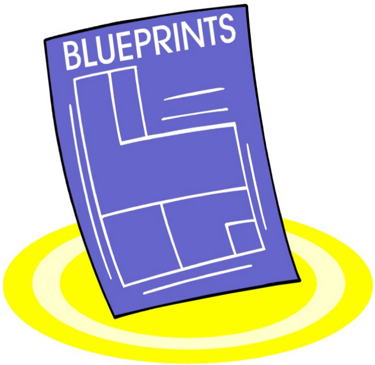




Asynchronous tasks

- Asychronous tasks run by TaskExecutor
  - kicked off programmatically
  - or by scheduler

- Simple configuration change to switch from thread pool to commonj work manager….



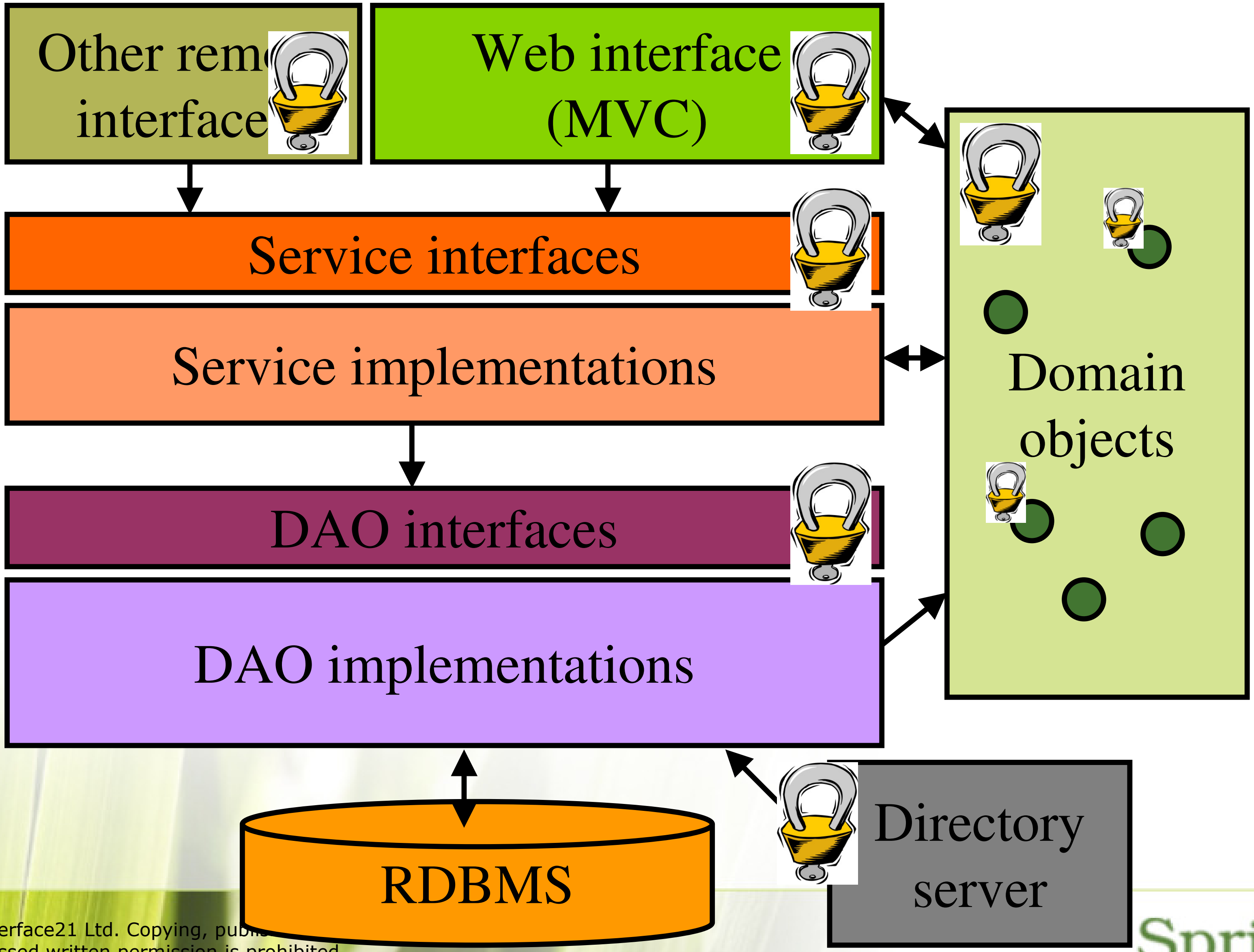
Messaging

- Spring 2.0 extended Spring's JMS support
  - JmsTemplate + …
  - Message-driven pojo's
    - fully JEE compliant
    - participate in transaction
    - process message and send reply in one unit of work
- For XML payloads
  - **Spring Web Services** offers OXM framework
  - www.springframework.org/spring-ws

Acegi Security for Spring

- Soon to be rebranded Spring Security
- Leading enterprise-grade Java security framework
- Handles real world complexity
- Mature and proven

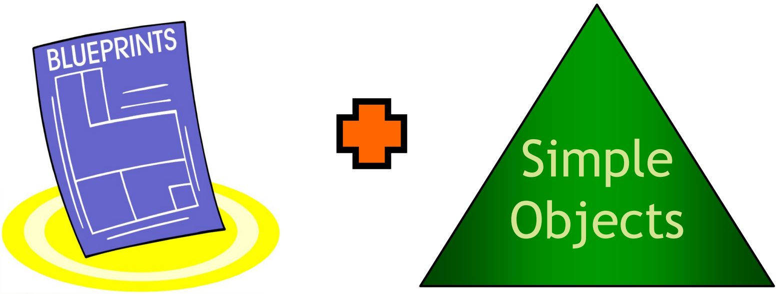
Security – impacts all layers



Presentation Layer: Spring Web Flow

- www.springframework.org/webflow
  - controlled navigation

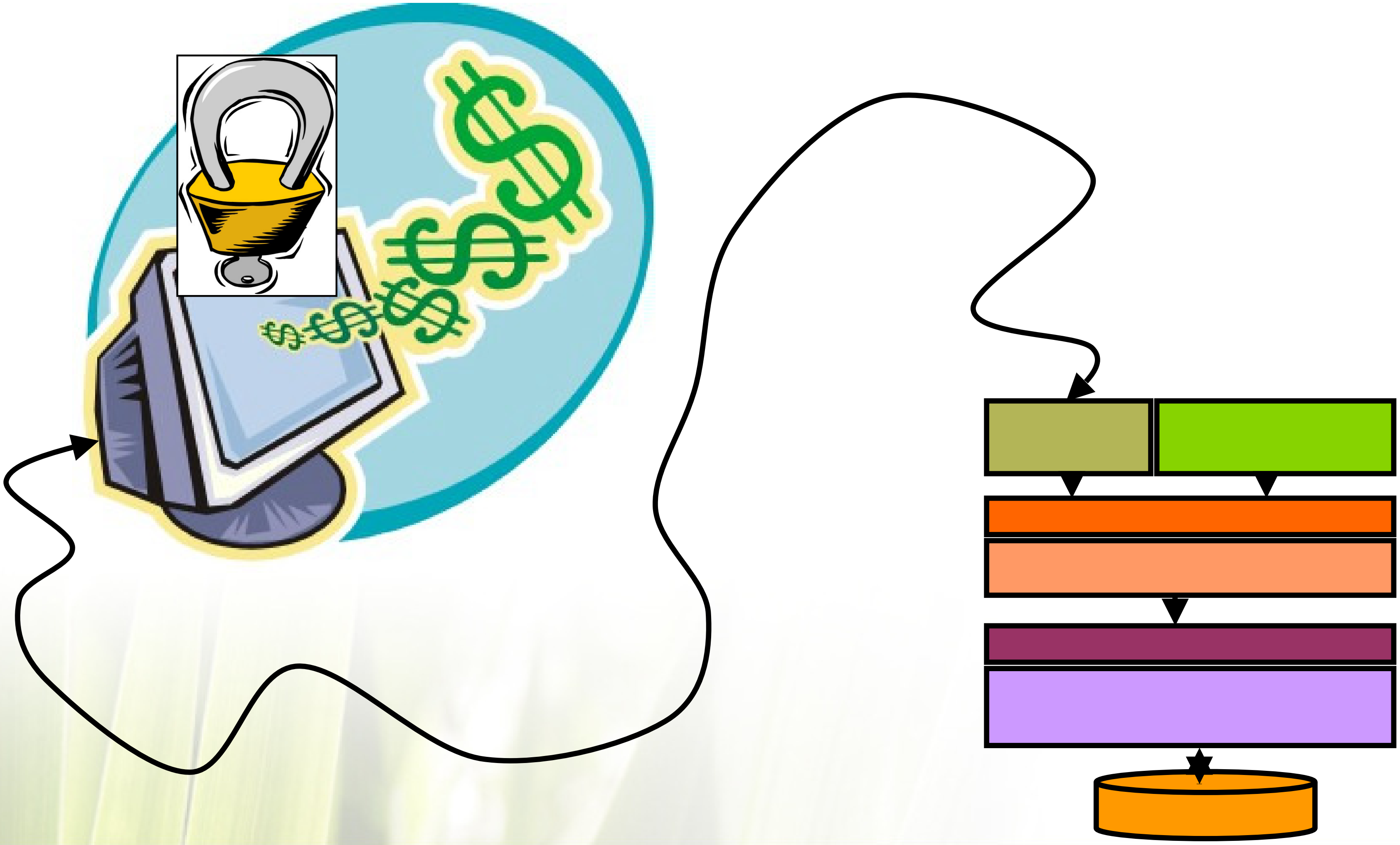
- Model web interaction as a **flow**
- Declare states and the transitions between them
  - view state (user's bit)
  - action state (your bit)
  - decision state (what next..?)
  - sub-flow state (abstraction)
  - end state



Spring Web Flow

- Controller logic greatly simplified
- Highly sophisticated conversational management
- Flows can be composed
- Servlet, portlet, multiple web frameworks
- Automatic post-redirect-get
- Browser buttons "just work"
  - back, history, refresh etc.
- Task data lifecycle management
  - request, flash, flow, conversation scopes

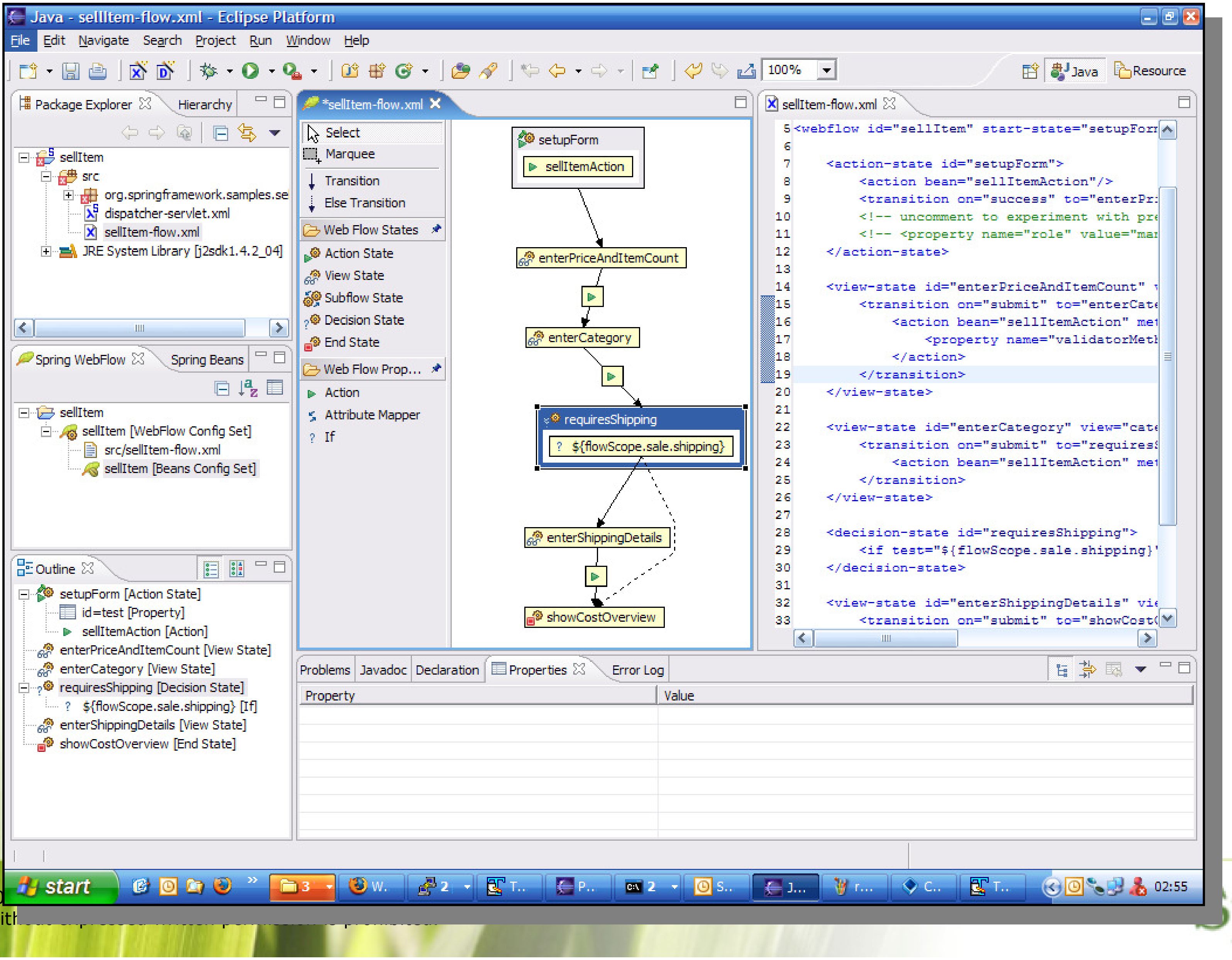
Presentation Layer: Spring Rich Client



Tooling: Spring IDE

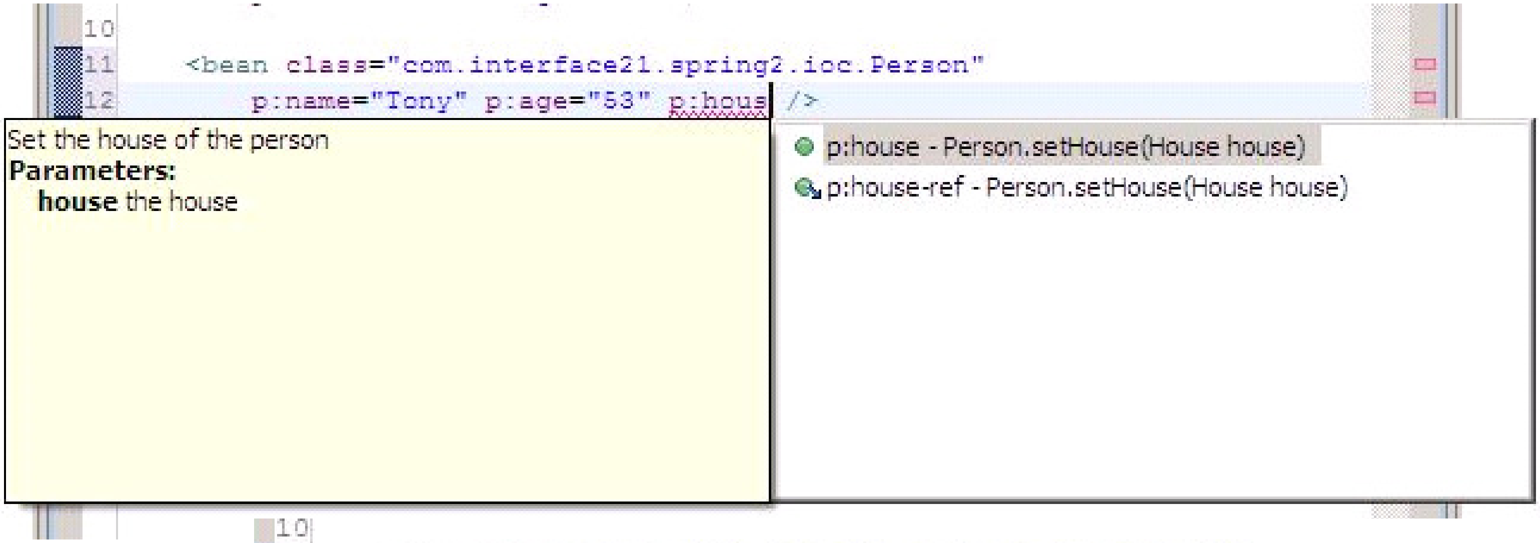
- http://springide.org
- Eclipse plugin for Spring development
  - Spring 2.0 preview release available this week!
- Features coming in next release:
  - Spring AOP cross-reference information
  - **Spring WebFlow support**

Spring IDE: WebFlow editor



```
<webflow id="sellItem" start-state="setupForr

    <action-state id="setupForm">
        <action bean="sellItemAction"/>
        <transition on="success" to="enterPri
        <!-- uncomment to experiment with pre
        <!-- <property name="role" value="mar
    </action-state>

    <view-state id="enterPriceAndItemCount" v
        <transition on="submit" to="enterCate
            <action bean="sellItemAction" met
                <property name="validatorMeth
            </action>
        </transition>
    </view-state>

    <view-state id="enterCategory" view="cate
        <transition on="submit" to="requires
            <action bean="sellItemAction" met
        </transition>
    </view-state>

    <decision-state id="requiresShipping">
        <if test="${flowScope.sale.shipping}'
    </decision-state>

    <view-state id="enterShippingDetails" vie
        <transition on="submit" to="showCost(
```

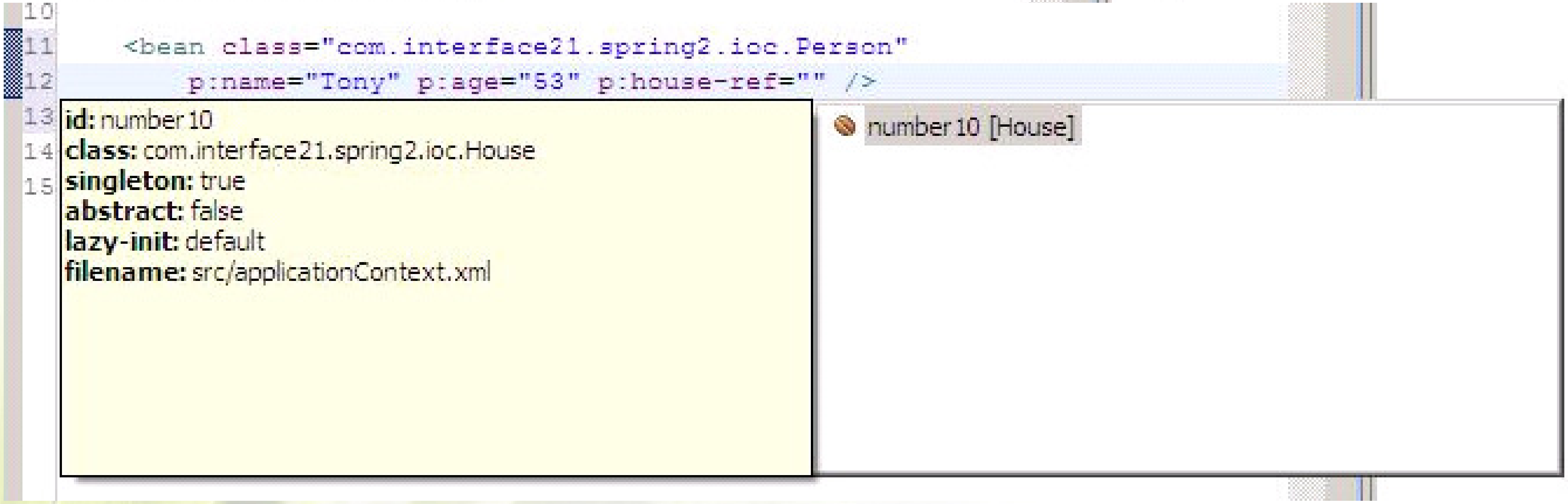
Spring IDE : p namespace



```
<bean class="com.interface21.spring2.ioc.Person"
    p:name="Tony" p:age="53" p:hous />
```

Set the house of the person
Parameters:
house the house

p:house - Person.setHouse(House house)
p:house-ref - Person.setHouse(House house)

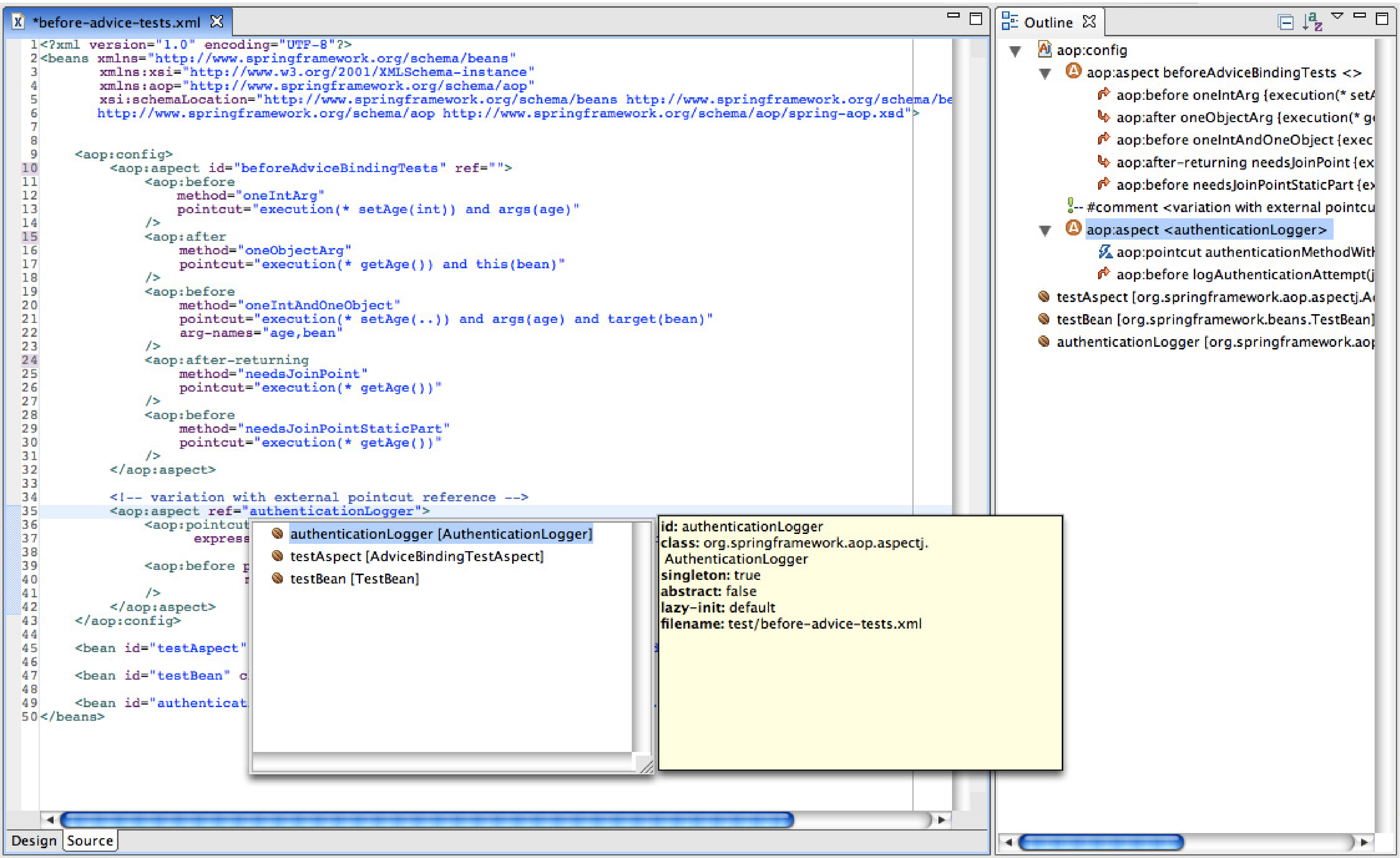


```
<bean class="com.interface21.spring2.ioc.Person"
    p:name="Tony" p:age="53" p:house-ref="" />
```

id: number 10
class: com.interface21.spring2.ioc.House
singleton: true
abstract: false
lazy-init: default
filename: src/applicationContext.xml

number 10 [House]

Spring IDE : AOP support



```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:aop="http://www.springframework.org/schema/aop"
        xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/be
       http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop.xsd">


    <aop:config>
        <aop:aspect id="beforeAdviceBindingTests" ref="">
            <aop:before
                method="oneIntArg"
                pointcut="execution(* setAge(int)) and args(age)"
            />
            <aop:after
                method="oneObjectArg"
                pointcut="execution(* getAge()) and this(bean)"
            />
            <aop:before
                method="oneIntAndOneObject"
                pointcut="execution(* setAge(..)) and args(age) and target(bean)"
                arg-names="age,bean"
            />
            <aop:after-returning
                method="needsJoinPoint"
                pointcut="execution(* getAge())"
            />
            <aop:before
                method="needsJoinPointStaticPart"
                pointcut="execution(* getAge())"
            />
        </aop:aspect>

        <!-- variation with external pointcut reference -->
        <aop:aspect ref="authenticationLogger">
            <aop:pointcut
                    express

            <aop:before p
                        r
            />
        </aop:aspect>
    </aop:config>

    <bean id="testAspect"

    <bean id="testBean" c

    <bean id="authenticat
</beans>
```

authenticationLogger [AuthenticationLogger]
testAspect [AdviceBindingTestAspect]
testBean [TestBean]

id: authenticationLogger
class: org.springframework.aop.aspectj.
AuthenticationLogger
singleton: true
abstract: false
lazy-init: default
filename: test/before-advice-tests.xml

```
aop:config
    aop:aspect beforeAdviceBindingTests <>
        aop:before oneIntArg {execution(* setA
        aop:after oneObjectArg {execution(* ge
        aop:before oneIntAndOneObject {exec
        aop:after-returning needsJoinPoint {ex
        aop:before needsJoinPointStaticPart {ex
    -- #comment <variation with external pointcu
    aop:aspect <authenticationLogger>
        aop:pointcut authenticationMethodWit
        aop:before logAuthenticationAttempt(j
testAspect [org.springframework.aop.aspectj.A
testBean [org.springframework.beans.TestBean]
authenticationLogger [org.springframework.ao
```

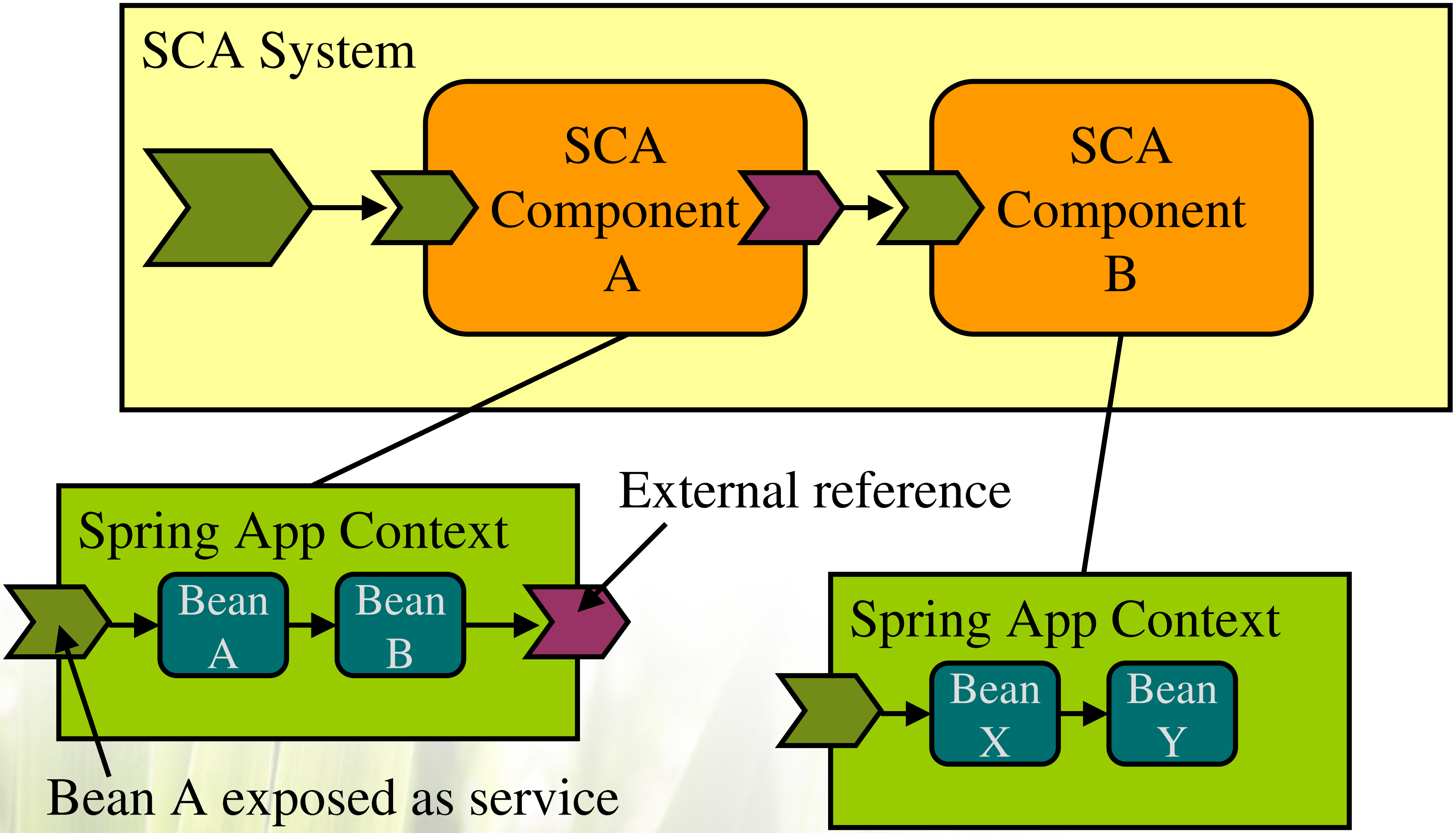
The Open SOA Collaboration

- BEA
- IBM
- Oracle
- SAP
- Sun
- **Interface21**
- Red Hat
- Cape Clear Software
- IONA
- Primeton Technologies
- Sybase
- Siemens
- Software AG
- TIBCO
- Rogue Wave Software

Service Component Architecture

- Assembly model
  - service components, references, wires
- Policy framework
  - QoS etc.
- Service implementation and client API
  - **Spring**
  - Java
  - C++
  - BPEL
  - EJB
- Bindings
  - Web Services, JMS, JCA

Any Spring Application is “SCA-ready”...



Not just about the Spring Framework

- Spring Framework
- Spring WebFlow
- Spring Web Services
- Spring Security
- Spring Rich Client
- Spring LDAP
- Spring IDE
- Spring OSGi
- Spring Modules
- Spring.NET
- Pitchfork

- AspectJ
- Mule
- Terracotta
- Tangosol Coherence
- GigaSpaces
- DWR
- JSON-RPC
- SCA / OSOA
- OSGi

Integrated Release Train

- In 2007 we will be bringing the portfolio closer together
- Integrated releases of Spring Portfolio products
  - “Release Train”
- Value added integration
- One stop shop
  - It all works together
  - No need to waste time integrating open source products

Summary

- Spring 2.0 was a major enhancement – Upgrade now!
    - Backward compatible
    - But added many new features
- But not an end in itself
- In the next year we will see…
    - The Spring component model tackle new frontiers
    - The Spring Portfolio grow and integrate further

Spring 2.0 and Beyond

Rod Johnson
CEO
Interface21