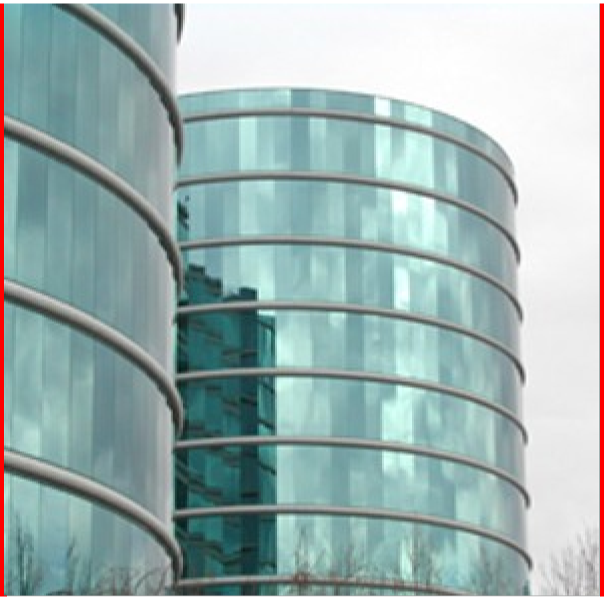


TDD on a Coherence project

Jon Hall - Oracle UK Consulting
jonathan.hall@oracle.com

Disclaimer

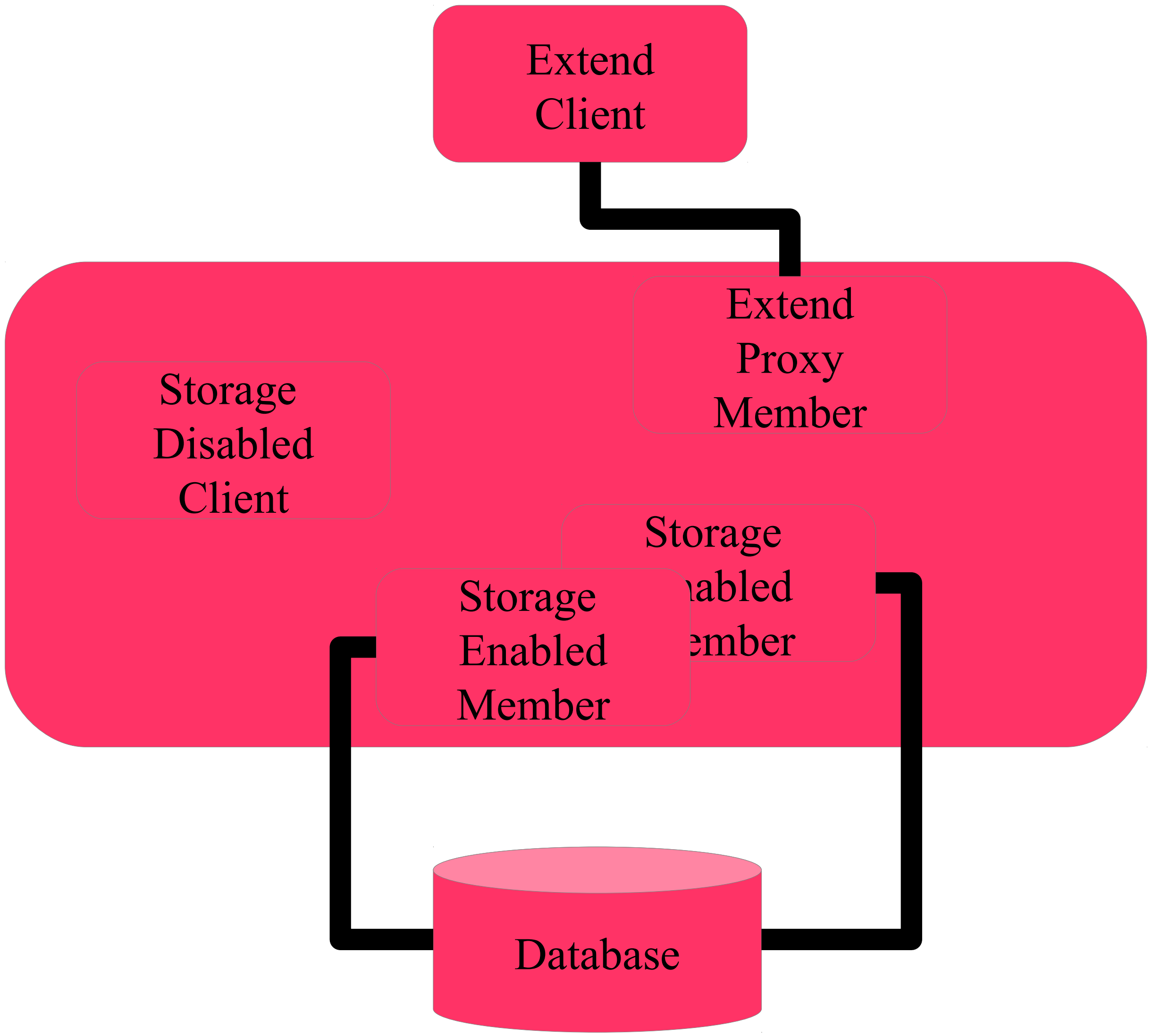
About me

- Pragmatic, hands-on, delivery focused Enterprise Developer/Architect with over 22 years IT experience
- Spring since 2005 and Maven since 2008 – like KISS
- Have worked for Oracle previously (part of Tangosol acquisition) – Coherence
- littlegrid Coherence test-support framework Developer (www.littlegrid.org)
- http://www.linked.com/jhall

Sample project overview

- Coherence
- Maven
- Spring 3
- JUnit
- Database
- In-process integration/functional testing

Sample project overview



Testing business/domain logic

- Think about how you approach the development, by developing test-first you're immediately using your API and also thinking about decoupling it from infrastructure.

- Shouldn't need to boot-strap queues, Spring, Coherence or need a database etc.

- For some 'service orchestration' you may need to use mocks or stubs – but try and avoid writing too much code for them – otherwise it becomes more than the class under test!

Sample project modules

https://bitbucket.org/littlegrid/sample-project - work in progress, but demonstrates concepts and techniques

sample-configuration

sample-database

sample-model

sample-persistence

sample-datagrid-common

sample-datagrid-cluster

sample-datagrid-cluster-integration-tests

sample-remote-client

sample-remote-client-integration-tests

Testing database access

Code example

- See sample-persistence module

Testing POF serialization

Code example

- See sample-datagrid-common module

Typical approaches to Coherence testing

…..

Typical approaches to Coherence testing

None – yes, honestly...

Typical approaches to Coherence testing

None – yes, honestly...

CacheFactory.getCache(“..) and the world of accidental clustering

System.setProperty(“...

CacheFactory.getCache(“..) and the world of the forgotten serializer/mismatch

Out of process testing – typically separate processes on the same machine

In-process testing – good fast compromise

ORACLE

Using Spring and Coherence (with the current version of Coherence)

SpringAwareCacheFactory

Supported method of integration between Coherence and Spring

http://docs.oracle.com/cd/E24290_01/coh.371/e22621/integratespring.htm

Coherence 3.4.x (or higher)

Spring 2.x (or higher)

Cache Store: Spring on server-side

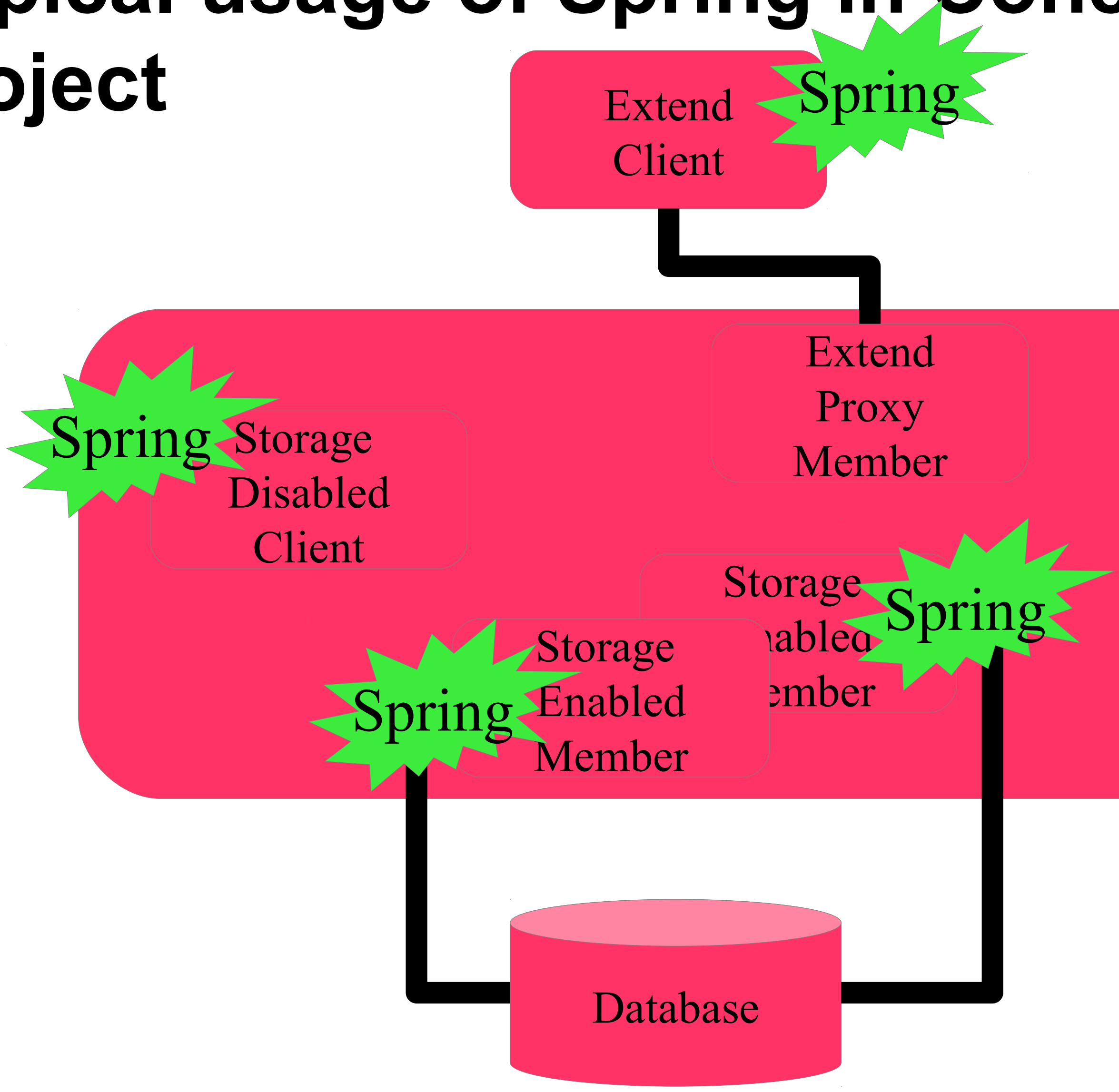
Configurable-cache-factory-config in override file

```
<cachestore-scheme>
    <class-scheme>
        <class-name>spring-bean:accountCacheStore</class-name>
       </class-scheme>

<bean id="accountCacheStore"
  class="...account.repository.impl.AccountCacheStoreImpl"
          scope="prototype">
```

ORACLE

Typical usage of Spring in Coherence project



Spring wired Default Cache Server

This is a technique that is used quite a bit, it goes something like this:

```
public static void main(final String[] args) {
    bootStrapApplicationContext();
    DefaultCacheServer.main(args);
}

private static void bootStrapApplicationContext() {
    // Cause Spring to boot-strap and initialise the app context
    ApplicationContextHolder.getApplicationContext();
}
```

Testing Coherence server-side code and configuration

Code example

- with Spring
- without Spring

Testing Coherence Extend client-side code and configuration

Member left event

Spring configuration

Named Cache: Spring on client-side (Extend/storage disabled client)

Coherence's named cache is typically wired into a repository or service

```
<bean id="accountCache"
      class="com.tangosol.net.CacheFactory"
      factory-method="getCache">
    <constructor-arg value="account"/>
</bean>
```

Using CI (Continuous Integration)

Setting up for concurrent builds (if required)

Quick straw poll of CI servers

- Bamboo
- Hudson
- Jenkins
- TeamCity
- Other?

Moving beyond CI to CD (Continuous Deployment)

Bringing it all together...

Who's doing it?

Who's trying to do CD in CI? ...and so your CI cycles are long?

Thoughts? Other ideas? What do you do?

Thank you!