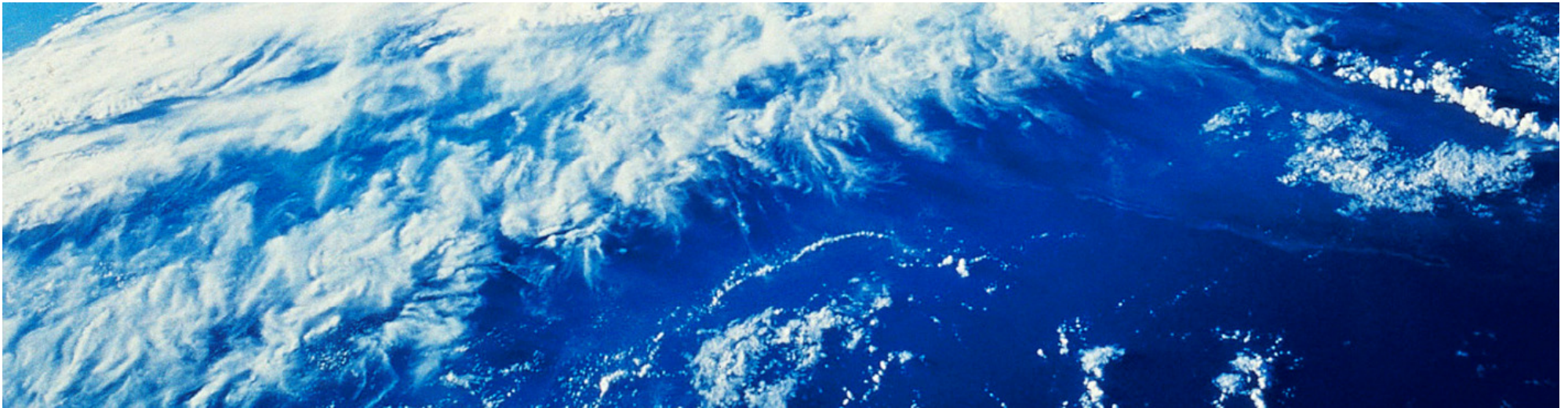


IBM WebSphere Commerce V7 FEP7

WebSphere Commerce Search



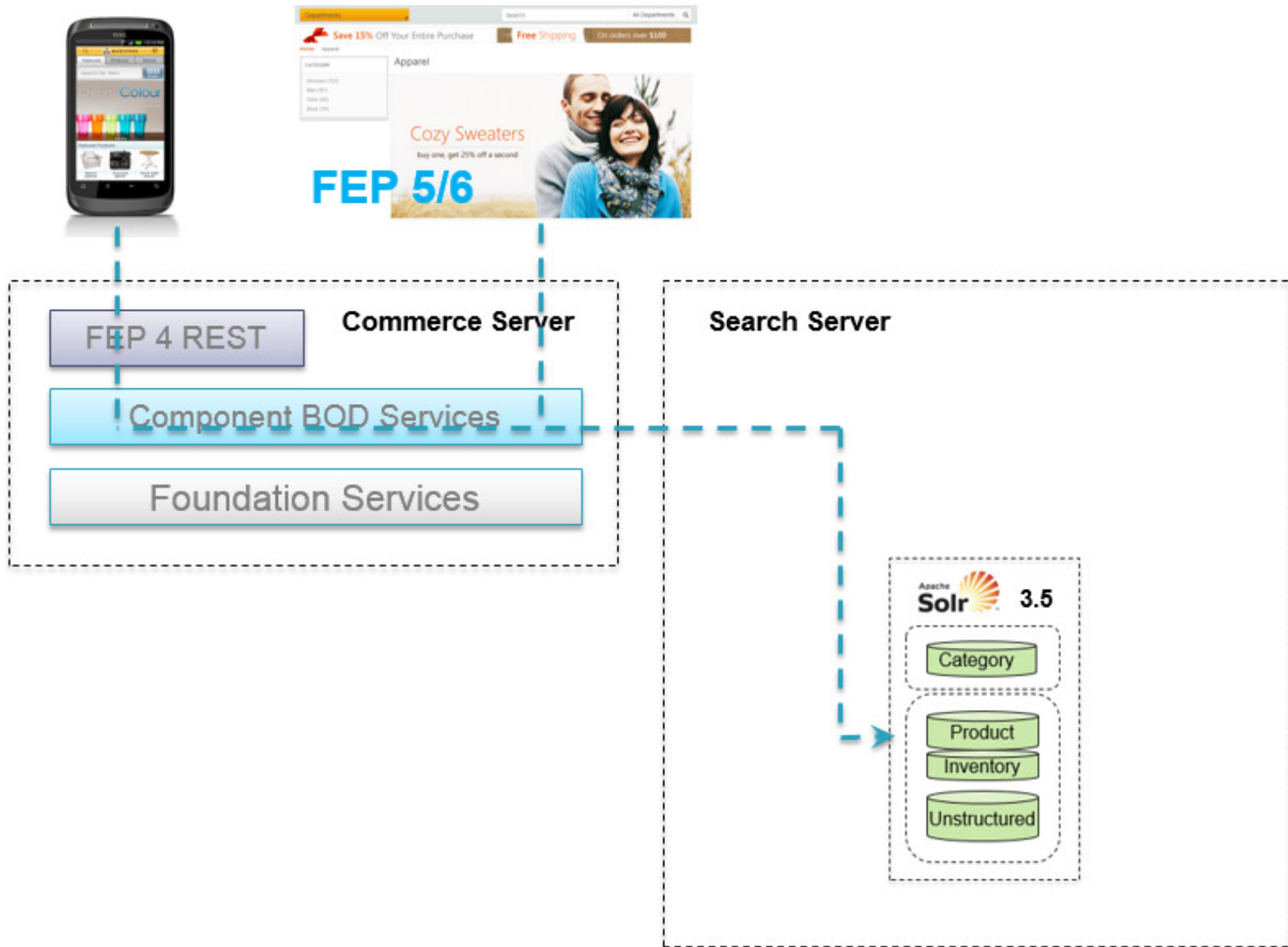
Agenda

- Architecture update
 - Search server architecture
 - REST services

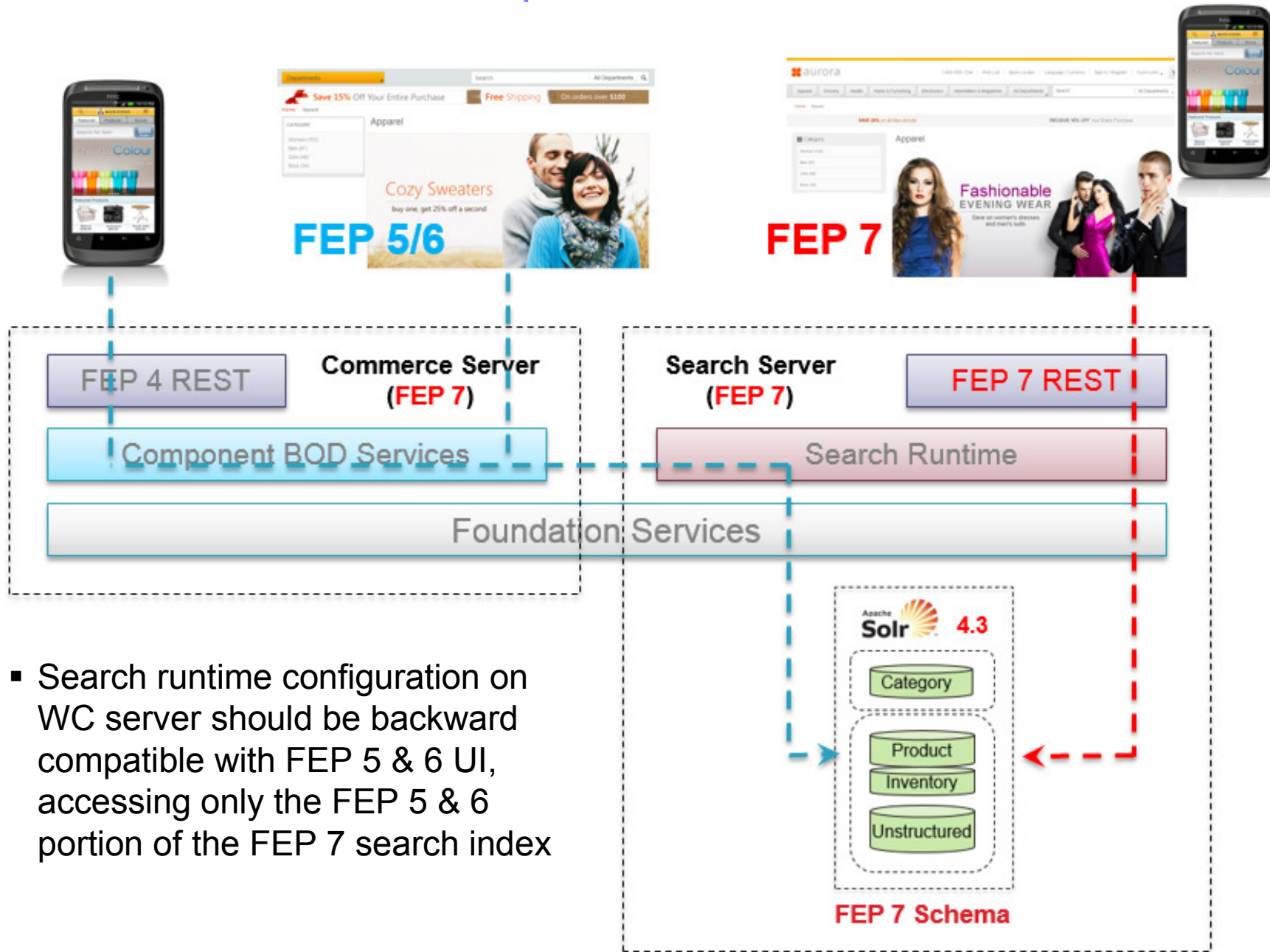
- Programming model update
 - Storefront interaction flow
 - Programming model compatibility
 - Index schema update
 - Customization points

- Deployment update
 - Advanced deployment configuration
 - Index lifecycle
 - Cache and invalidation
 - Logging and tracing

FEP 5 / 6 Search architecture



FEP 7 Search architecture update



- Search runtime configuration on WC server should be backward compatible with FEP 5 & 6 UI, accessing only the FEP 5 & 6 portion of the FEP 7 search index

FEP 7 Search architecture update



Store UI (Page Composer: widgets, layout)

Services

REST Services
CategoryHandler
ProductHandler
ContentHandler

Transaction Server

- Prices
- Marketing
- Promotion
- Wish list, carts
- Order Capture
- Payment

Commerce Server

Browse and Search Server

Search Server

- DynaCache, eXtremeScale
- Clusters (vertical & horizontal scaling)

Deployment

- 2 separate EARs (browse/search, transactions)

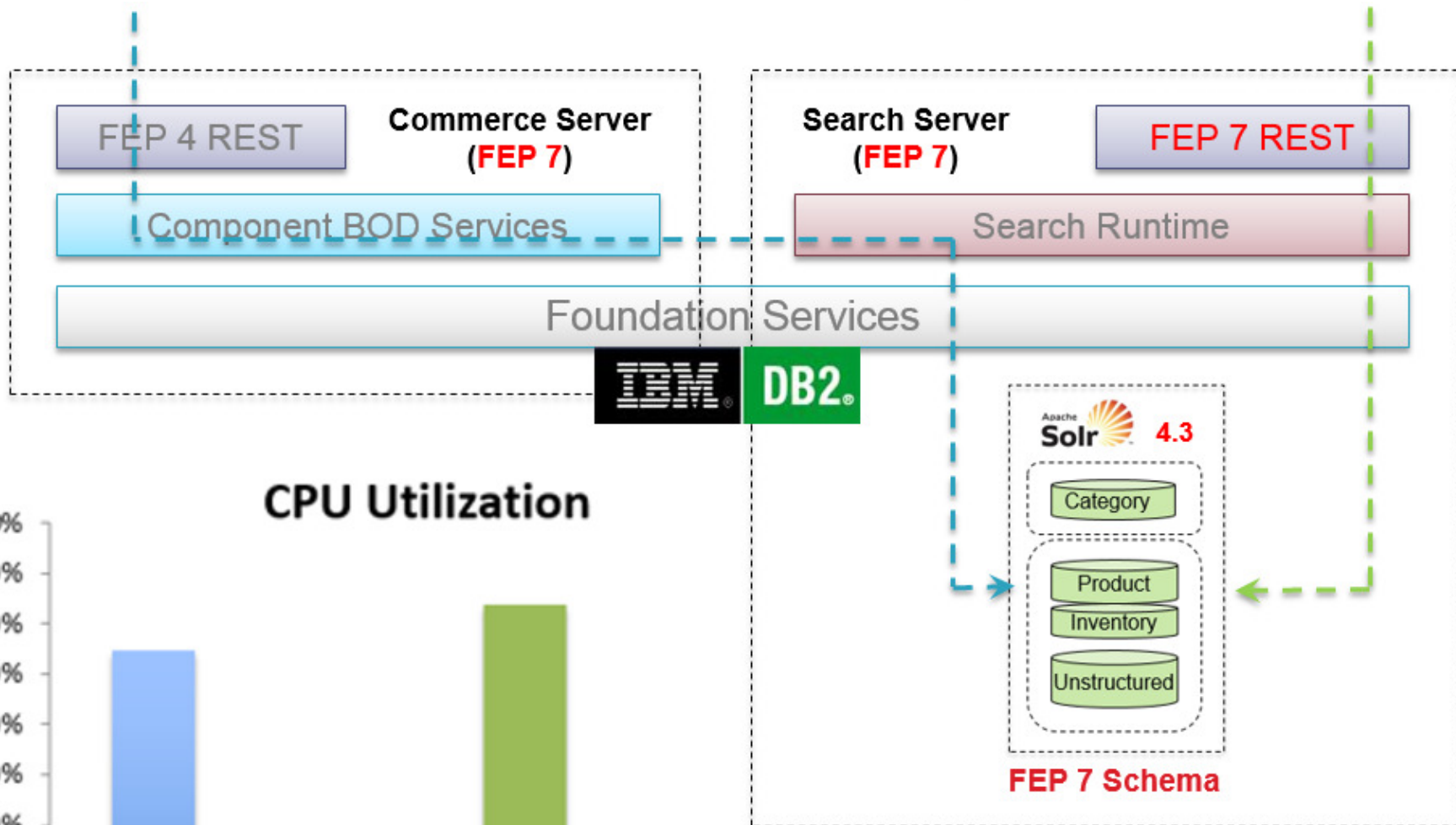
Lightweight services

- Search server REST services for storefront navigation

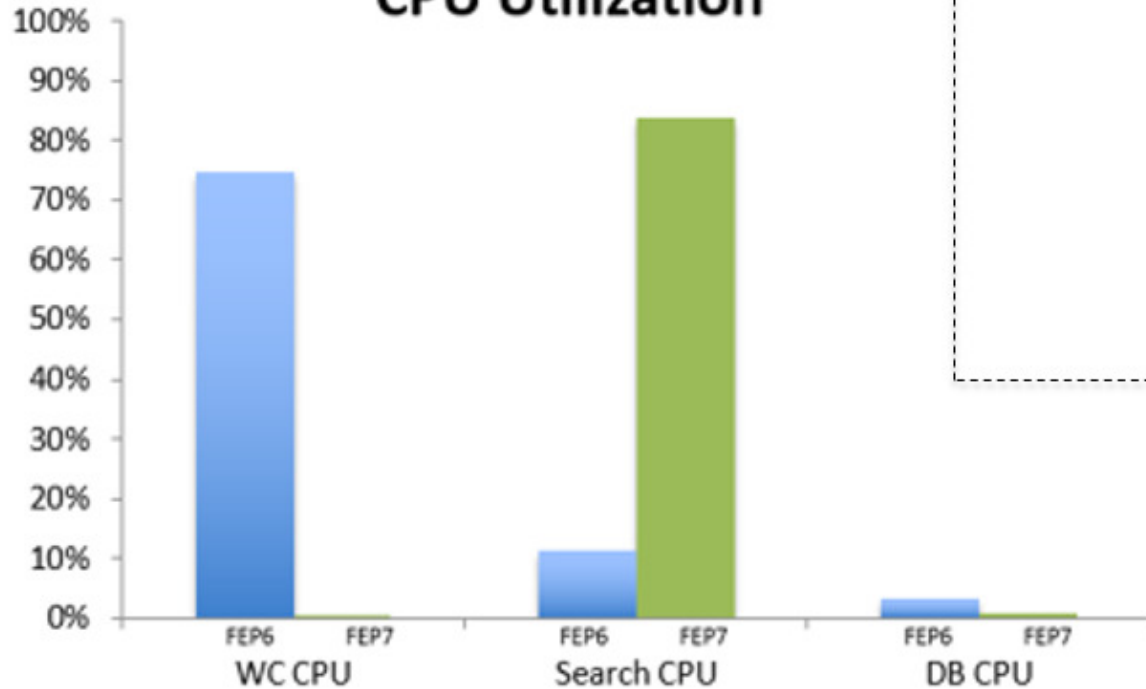
Scalability and Performance

- Scale-up independently for browse and transaction workload
- Lightweight Foundation for browse/search (e.g. No EJB container, No EMF)
- More fine grain and co-operative caching and invalidation between servers

Search REST service CPU utilization comparison



CPU Utilization



Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Storefront navigation features provided by Search server

Search features	FEP 7 Search REST services
Keyword Search	Processed only on Search server
Category Browse	Processed only on Search server
Faceted Navigation	Processed only on Search server (cacheable)
Relevancy Boosting	Processed only on Search server (cacheable)
Catalog Filter	Processed only on Search server (pre-compiled in WC)
Contract Entitlement	Processed only on Search server (cacheable)
Contract Price	Storefront UI calls Commerce server
Inventory	Processed only on Search server
Search Term Associations	Processed only on Search server (cacheable)
Search Rules	Search calls Commerce (cacheable)
Extended Sites	Processed only on Search server
Workspace	Processed only on Search server

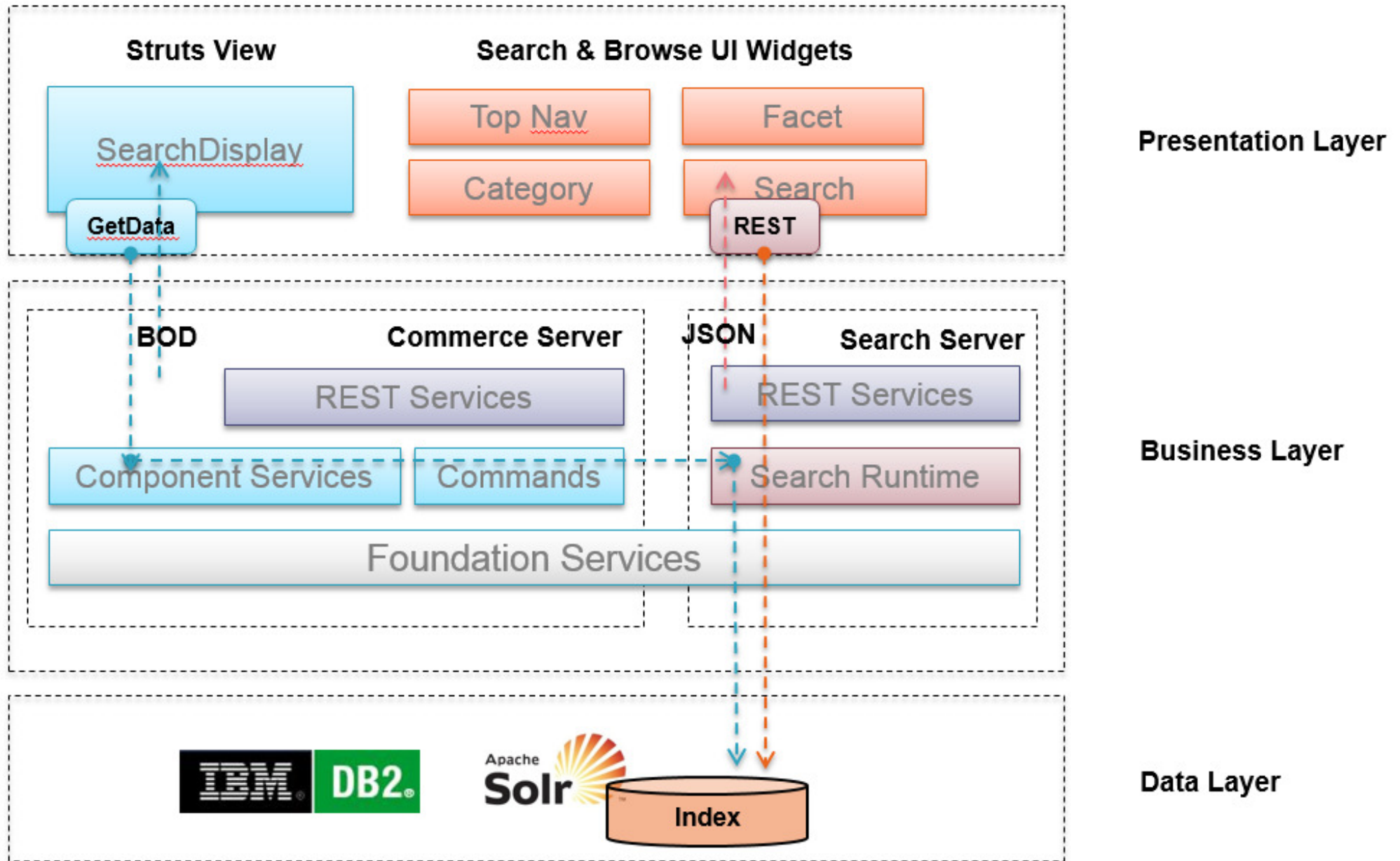
Agenda

- Architecture update
 - Search server architecture
 - REST services

- Programming model update
 - Storefront interaction flow
 - Programming model compatibility
 - Index schema update
 - Customization points

- Deployment update
 - Advanced deployment configuration
 - Index lifecycle
 - Cache and invalidation
 - Logging and tracing

FEP 7 Search interaction flow at storefront



FEP 7 Search REST services for storefront from Search server

- ProductViewHandler – keyword search, browsing, faceted navigation
 - store/{storeId}/productview/bySearchTerm/{searchTerm}
 - store/{storeId}/productview/byId/{productId}
 - store/{storeId}/productview/byIds
 - store/{storeId}/productview/{partNumber}
 - store/{storeId}/productview/byCategory/{categoryId}

- CategoryViewHandler – navigation menu
 - store/{storeId}/categoryview/byId/{categoryId}
 - store/{storeId}/categoryview/byIds
 - store/{storeId}/categoryview/{categoryIdentifier}
 - store/{storeId}/categoryview/@top
 - store/{storeId}/categoryview/byParentCategory/{parentCategoryId}

- SiteContent – unstructured search, auto suggest
 - store/{storeId}/sitecontent/webContentsBySearchTerm/{searchTerm}
 - store/{storeId}/sitecontent/keywordSuggestionsByTerm/{term}
 - store/{storeId}/sitecontent/categorySuggestions
 - store/{storeId}/sitecontent/brandSuggestions
 - store/{storeId}/sitecontent/webContentSuggestions

Example: JSTL taglib and JSP dot notation comparison

FEP 5 / 6:

```

<wcf:getData type="com.ibm.commerce.catalog.facade.datatypes.CatalogNavigationViewType" var="catalogNavigationView"
  expressionBuilder="${navigationView}" scope="request" varShowVerb="showCatalogNavigationView"
  maxItems="${pageSize}" recordSetStartNumber="${beginIndex}" scope="request">
  <wcf:param name="searchProfile" value="${searchProfile}" />
  <wcf:param name="searchTerm" value="${newSearchTerm}" />
  <wcf:param name="intentSearchTerm" value="${intentSearchTerm}" />
  <wcf:param name="searchType" value="${searchType}" />
  <wcf:param name="search.
    <c:set var="globalcategories" value="${catalogNavigationView.facetView}" scope="request"/>
    <c:set var="globalfacets" value="${catalogNavigationView.facetView}" scope="request"/>
    <c:set var="globalresults" value="${catalogNavigationView.catalogEntryView}" scope="request"/>
    <c:set var="globalbreadcrumbs" value="${catalogNavigationView.breadCrumbTrailView}" scope="request"/>
    <c:set var="globalreport" value="${catalogNavigationView.previewReport}" scope="request"/>
    <c:set var="spellcheck" value="${catalogNavigationView.metadata.spellcheck}" scope="request"/>
    <c:set var="metaData" value="${catalogNavigationView.metadata.metaData}" scope="request"/>
  </wcf:getData>
  <!-- Global Results will contain only one element -->
  <c:forEach var="catEntry" items="${globalresults}" varStatus="status">
    <c:set var="catEntryIdentifier" value="${catEntry.uniqueID}"/>
  </c:forEach>

```

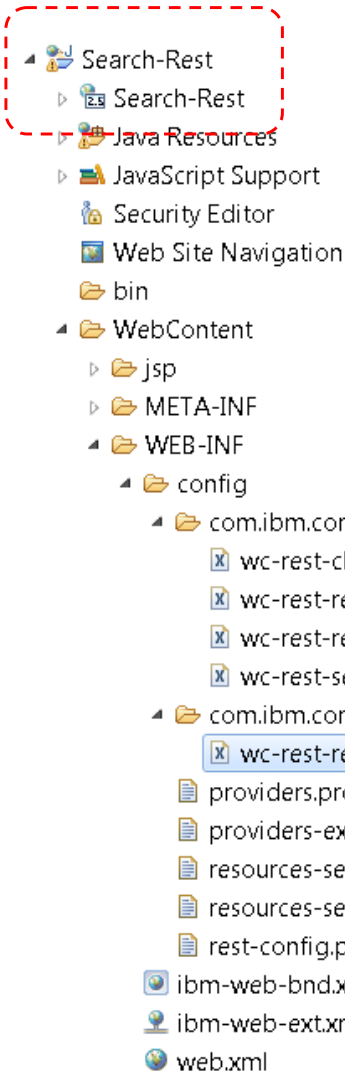
FEP 7:

```

<wcf:rest var="catalogNavigationView1" url="${searchHostNamePath}${searchContextPath}/store/${WCParam.storeId}/productview/bySearchTerm/${searchTerm}" format="json" >
  <wcf:param name="pageSize" value="${pageSize}" />
  <wcf:param name="pageNumber" value="${pageNumber + 1}" />
  <wcf:param name="profileName" value="${searchProfile}" />
  <wcf:param name="intentSearchTerm" value="${intentSearchTerm}" />
  <wcf:param name="searchType" value="${searchType}" />
  <wcf:param name="searchSource" value="S" />
  <wcf:param name="originalSearchTerm" value="${ori
    <c:set var="globalcategories" value="${catalogNavigationView1.facetView}" scope="request"/>
    <c:set var="globalfacets" value="${catalogNavigationView1.facetView}" scope="request"/>
    <c:set var="globalresults" value="${catalogNavigationView1.catalogEntryView}" scope="request"/>
    <c:set var="globalbreadcrumbs" value="${catalogNavigationView1.breadCrumbTrailView}" scope="request"/>
    <c:set var="globalreport" value="${catalogNavigationView1.metadata.previewReport}" scope="request"/>
    <c:set var="mpe_id" value="${catalogNavigationView1.metadata.espot}" scope="request" />
    <c:set var="intv_id" value="${catalogNavigationView1.metadata.activity}" scope="request" />
    <c:set var="experimentId" value="${catalogNavigationView1.metadata.experiment}" scope="request" />
    <c:set var="testElementId" value="${catalogNavigationView1.metadata.testelement}" scope="request" />
  </wcf:rest>
  <!-- Global Results will contain only one element -->
  <c:forEach var="catEntry" items="${globalresults}" varStatus="status">
    <c:set var="catEntryIdentifier" value="${catEntry.uniqueID}"/>
  </c:forEach>

```

Example: customizing FEP 7 Search REST service



REST configuration (wc-rest-resourceconfig.xml):

```

<ResourceConfig>
  <Resource name="productview">
    <GetUri uri="store/{storeId}/productview/byMyCategory/{categoryId}"
            description="A custom version of ProductView by categoryid based on below search profile."
            searchProfile="IBM findProductsByCategory"/>
  </Resource>
</ResourceConfig>
  
```

1. REST configuration (wc-rest-resourceconfig.xml)
 - URL to resource
 - Search profile

2. REST resource handler
 - Resource handler method using Java annotation
 - URL parameter to control parameter mapper (wc-component.xml)

Example: customizing FEP 7 Search REST service

REST resource handler:

```

@Path("/{storeId}/productview")
@ClassDescription("This custom class provides RESTful services to get the ProductView details.")
@Encoded
public class MyProductViewHandler extends ProductViewHandler {

    @GET
    @Path("byMyCategory/{categoryId}")
    @Produces({ MediaType.APPLICATION_ATOM_XML, MediaType.APPLICATION_JSON,
        MediaType.APPLICATION_XML, MediaType.APPLICATION_XHTML_XML })
    @Authentication(NO_AUTHENTICATION_HTTP)
    public Response findProductsByCategory(@PathParam("STOREID") String storeId,
        @PathParam("categoryId") String categoryId) {

        // store/{storeId}/productview/byMyCategory/{categoryId}
    }
}

```

Control parameter mapper (wc-component.xml):

```

<_config:valuemapping externalName="SearchControlParameterMapping" internalName="SearchControlParameterMapping">
  <_config:valuemap externalValue="searchType" internalValue="_wcf.search.type" />
  <_config:valuemap externalValue="searchSource" internalValue="_wcf.search.source" />
  <_config:valuemap externalValue="orderBy" internalValue="_wcf.search.sort" />
  <_config:valuemap externalValue="pageNumber" internalValue="_wcf.search.page.number" />
  <_config:valuemap externalValue="pageSize" internalValue="_wcf.search.page.size" />
  <_config:valuemap externalValue="categoryId" internalValue="_wcf.search.category" />
  <_config:valuemap externalValue="catalogId" internalValue="_wcf.search.catalog" />
  <_config:valuemap externalValue="langId" internalValue="_wcf.search.language" />
  <_config:valuemap externalValue="storeId" internalValue="_wcf.search.store.online" />
  <_config:valuemap externalValue="physicalStoreIds" internalValue="_wcf.search.store.physical" />
  <_config:valuemap externalValue="contractId" internalValue="_wcf.search.contract" />
  <_config:valuemap externalValue="currency" internalValue="_wcf.search.currency" />
  <_config:valuemap externalValue="returnFields" internalValue="_wcf.search.internal.response.fields" />
  <_config:valuemap externalValue="responseFormat" internalValue="_wcf.search.internal.response.format" />
  <_config:valuemap externalValue="responseTemplate" internalValue="_wcf.search.internal.response.template" />
  <_config:valuemap externalValue="facet" internalValue="_wcf.search.facet" />
  <_config:valuemap externalValue="facetLimit" internalValue="_wcf.search.facet.field.limit" />

```


Example: customizing FEP 7 Search REST service

REST resource handler:

```

@Path("/{storeId}/productview")
@ClassDescription("This custom class provides RESTful services to get the ProductView details.")
@Encoded
public class MyProductViewHandler extends ProductViewHandler {

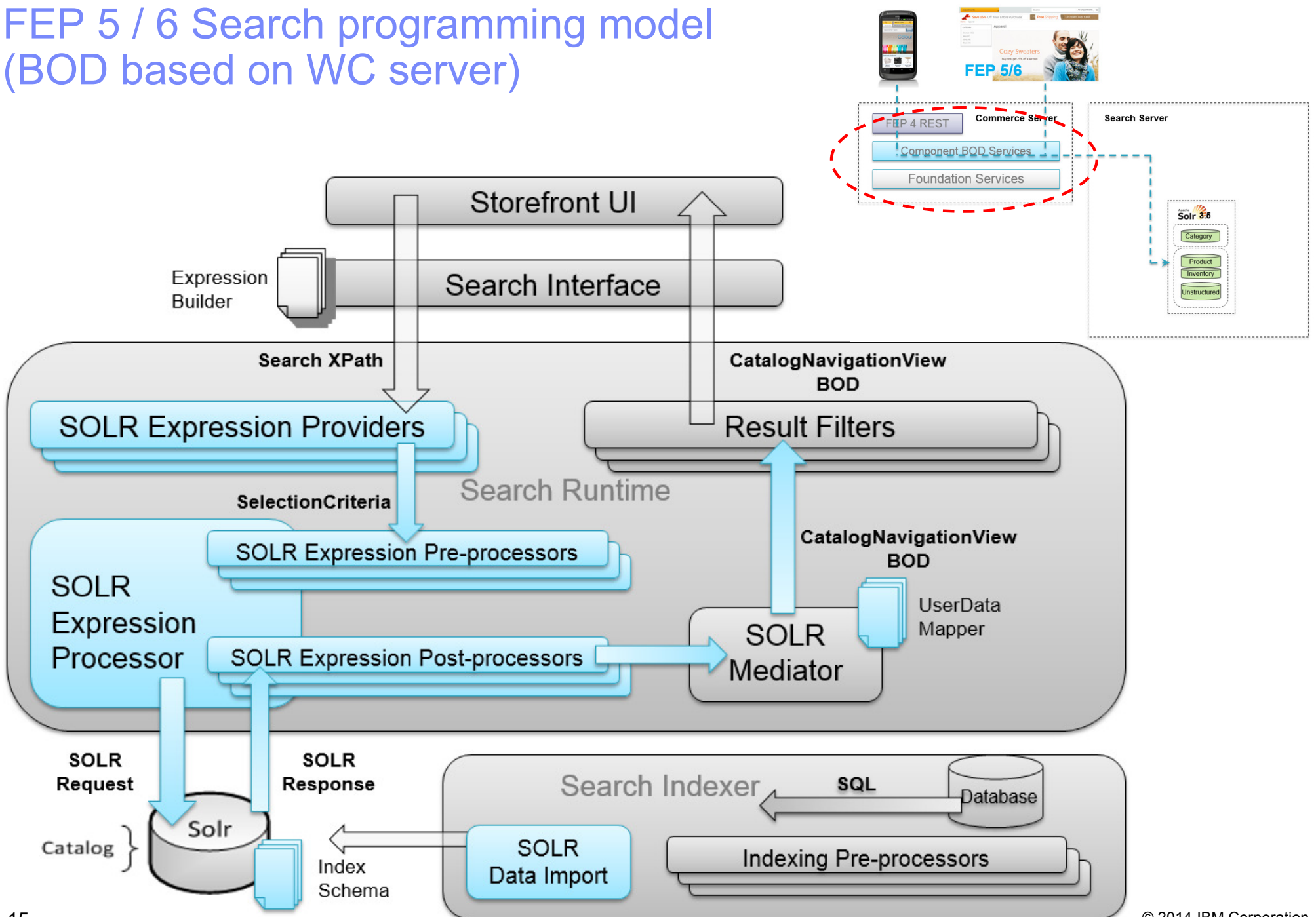
    @GET
    @Path("byMyCategory/{categoryId}")
    @Produces({ MediaType.APPLICATION_ATOM_XML, MediaType.APPLICATION_JSON,
        MediaType.APPLICATION_XML, MediaType.APPLICATION_XHTML_XML })
    @Authentication(NO_AUTHENTICATION_HTTP)
    public Response findProductsByCategory(@PathParam(STOREID) String storeId,
        @PathParam("categoryId") String categoryId) {

        // store/{storeId}/productview/byMyCategory/{categoryId}

        Response result = null;
        SearchCriteria searchCriteria = null;
        try {
            // Initialize SearchCriteria object
            ① searchCriteria = initSearchCriteria(storeId, GET_METHOD,
                COMPONENT_ID, RESOURCE_NAME,
                "store/{storeId}/productview/byMyCategory/{categoryId}");
            // Assign input parameter into SearchCriteria
            searchCriteria.setControlParameterValue(
                SearchServiceConstants.CTRL_PARAM_SEARCH_CATEGORY, categoryId);
            // Perform the service request and return the response in the
            // appropriate format.
            ② result = performSearch(searchCriteria);
        } catch (Exception e) {
            result = generateResponseFromRespData(searchCriteria, null, e);
        }
        return result;
    }
}

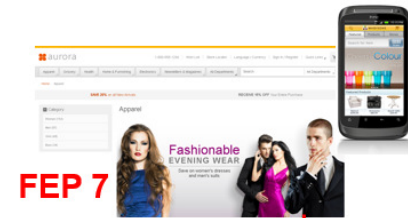
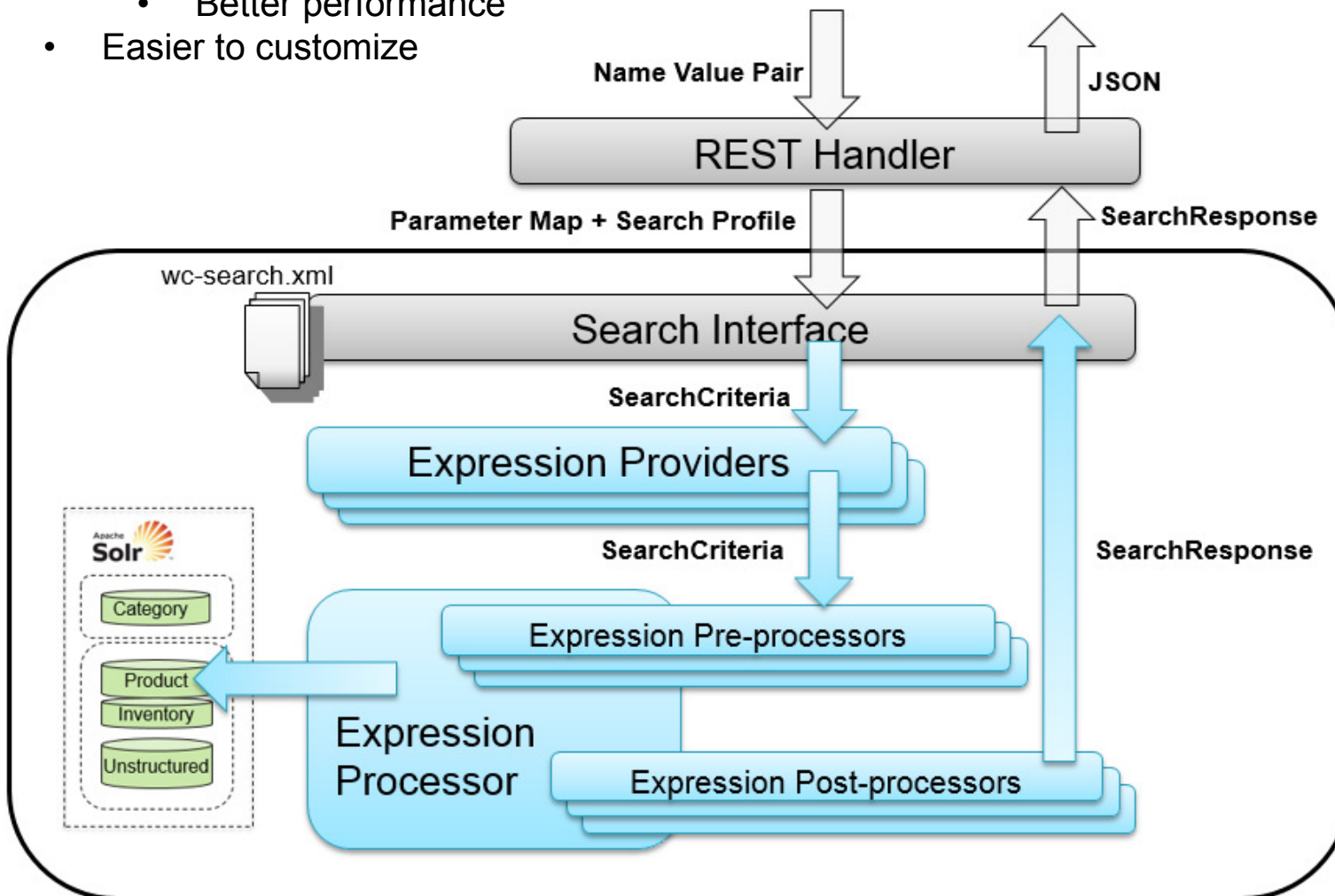
```

FEP 5 / 6 Search programming model (BOD based on WC server)

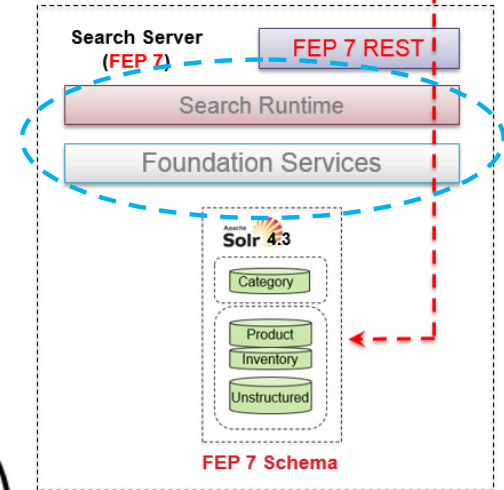


FEP 7 Search programming model update (Simplified, lightweight on Search server)

- Shorter overall code path
 - No EJB, No EMF / SDO
 - Lighter memory footprint
 - Better performance
- Easier to customize

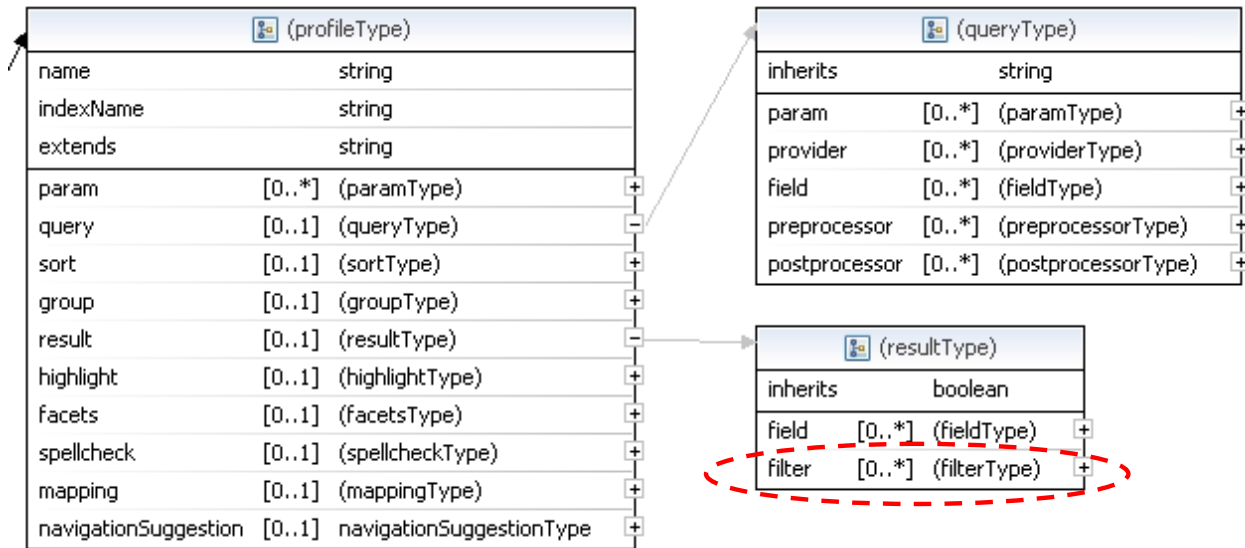


FEP 7

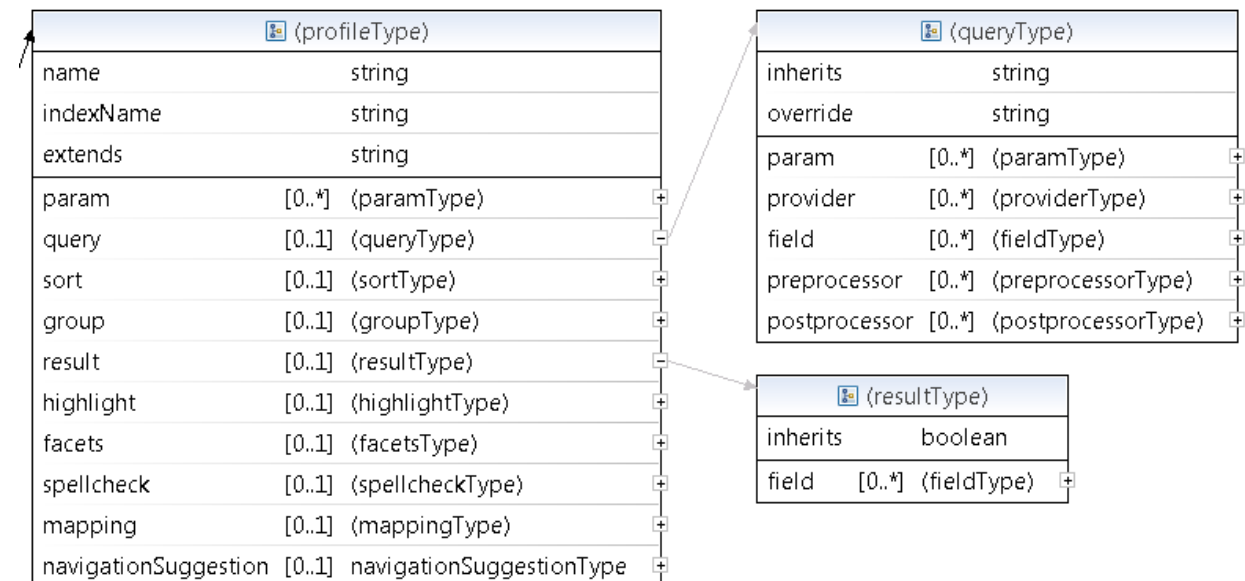


Search profile definition comparison

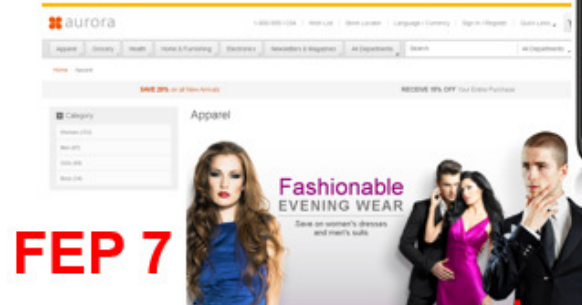
FEP 5 / 6:



FEP 7:

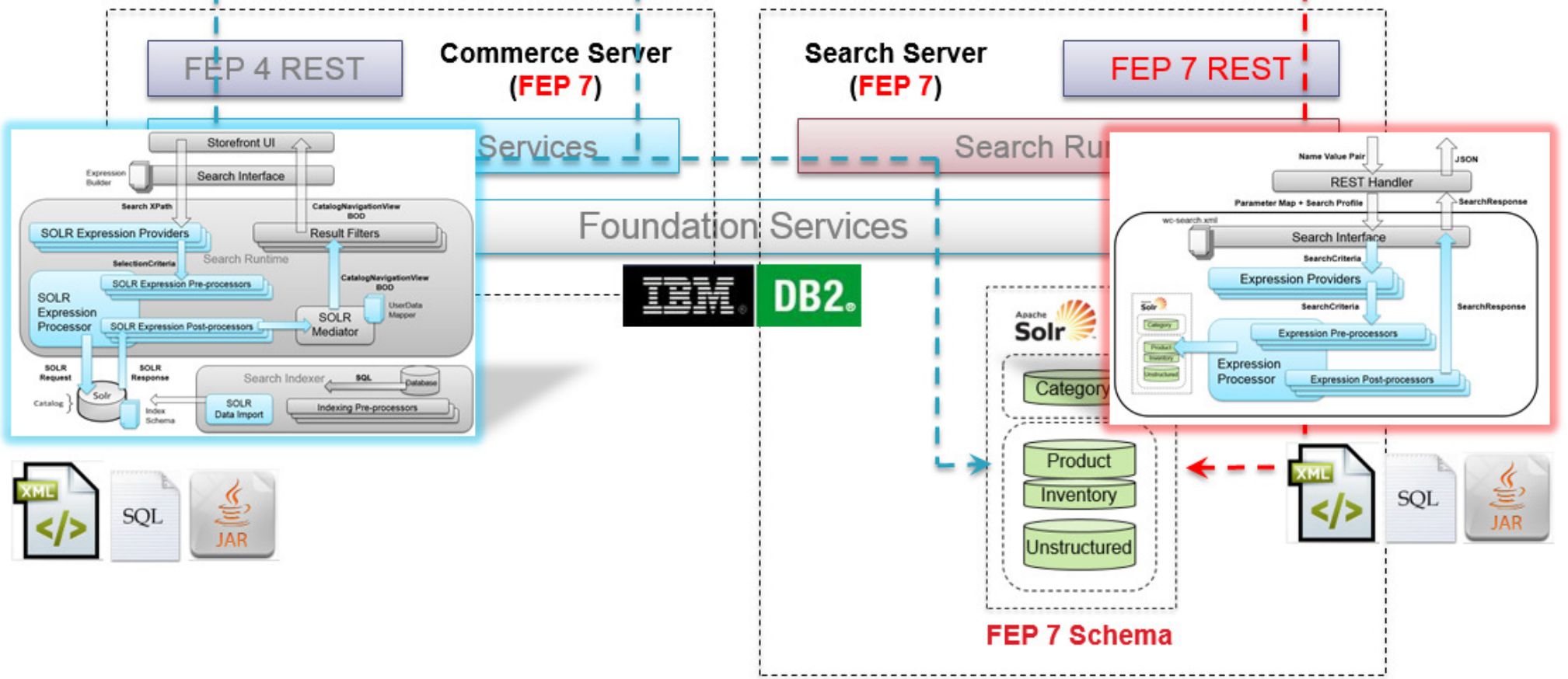


FEP 7 Search compatibility



FEP 5/6

FEP 7

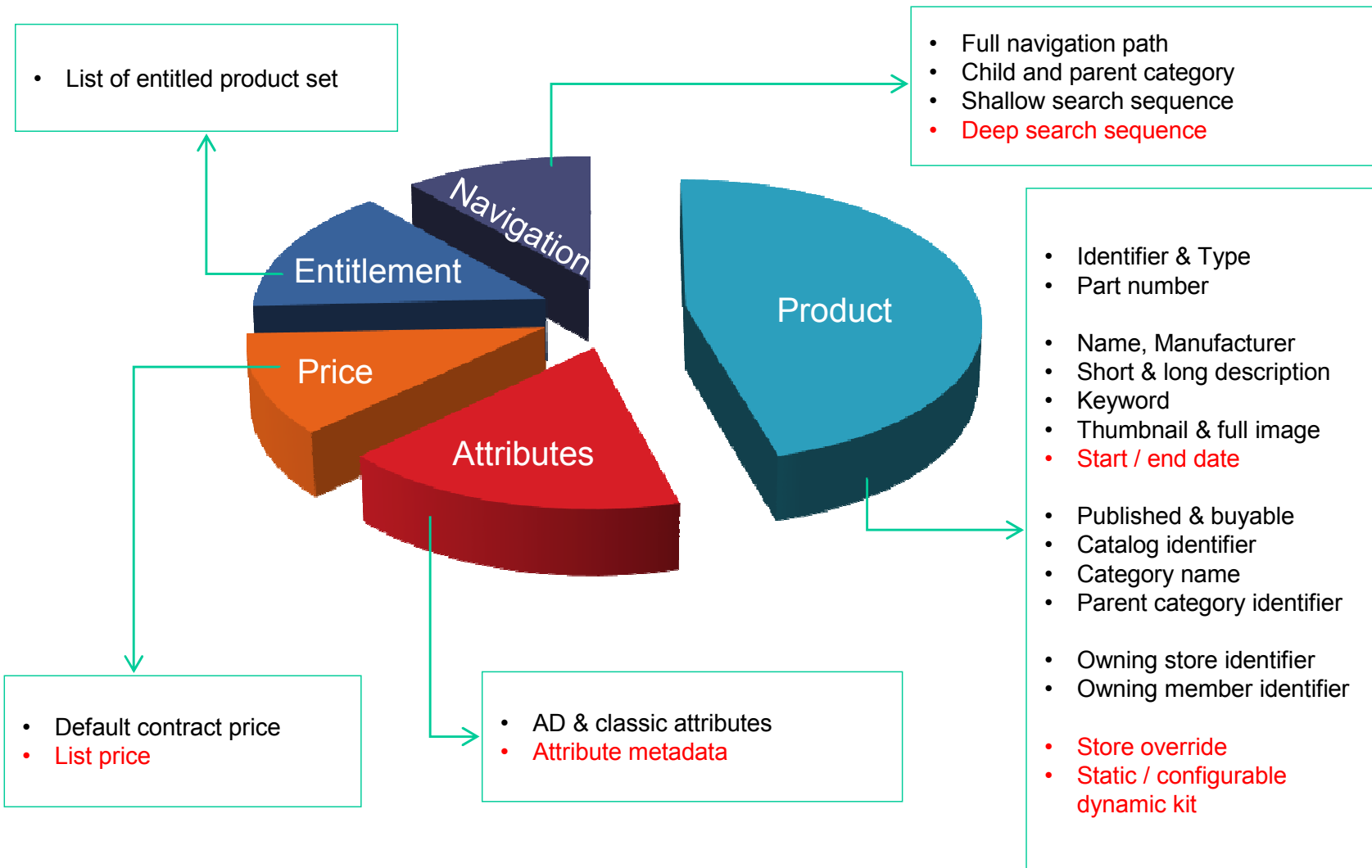


Additional programming considerations on Search server

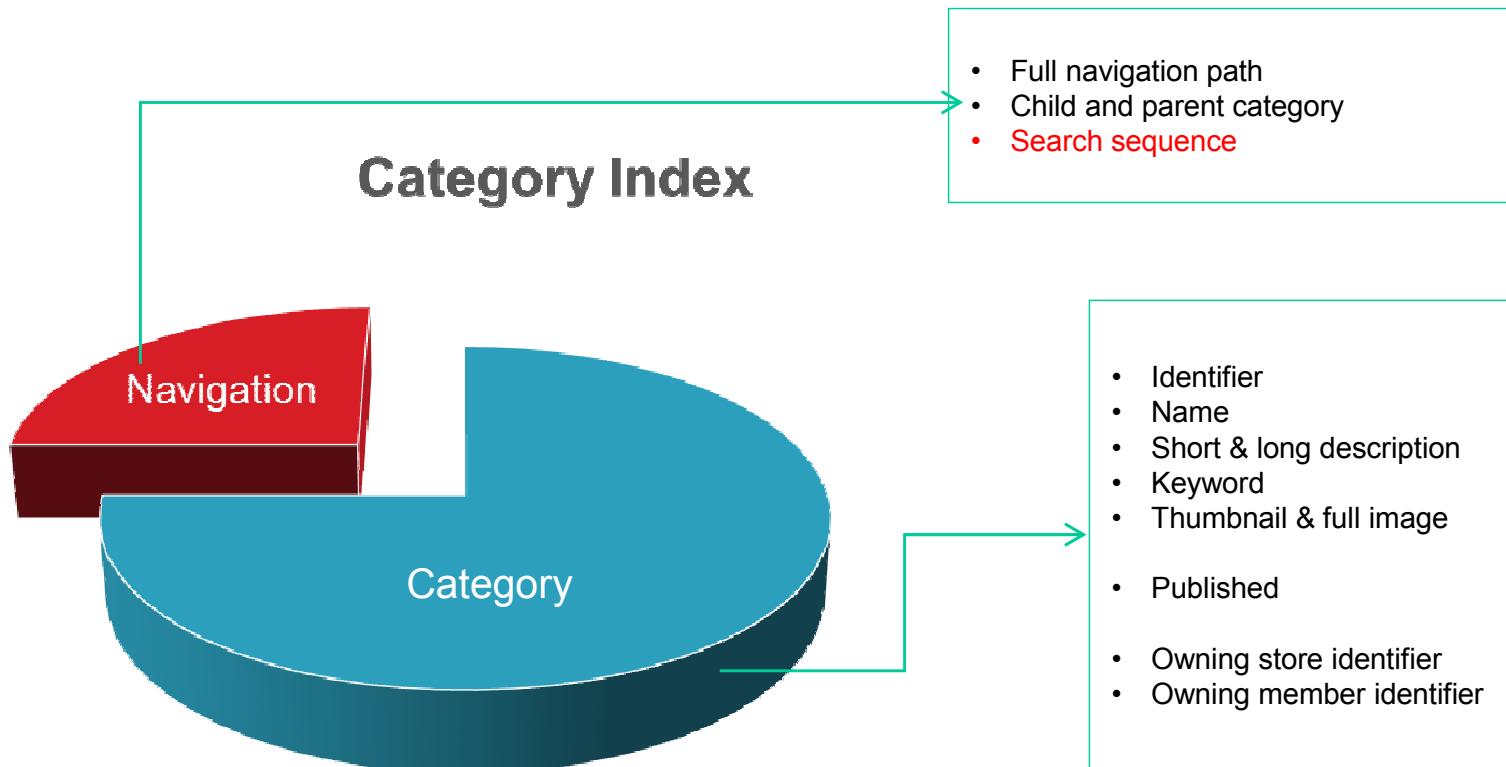
- Search REST services are stateless
 - Shopper's session context should be maintained from the storefront UI
 - Business context must be passed in explicitly as URL parameters, otherwise Search runtime will fallback to store level defaults, e.g. catalogId, contractId, langId, currency
- Search server only uses simple data object model
 - Avoid using EJB or AccessBean since all server logics are executed only in the Search server's web container
 - Use REST services to make remote calls and avoid using SDO and component services since no EMF compatible runtime is available on Search server
- Re-use complex custom logic on WC
 - Avoid remote call to WC unless absolutely necessary
 - Use caching strategies to offload remote calls to avoid web container thread deadlock from happening
 - Use DSL JDBC query service to look up additional business data directly from database
 - Re-use locally cached data as much as possible
- Classloader note for runtime environment
 - Indirectly class dependency at runtime may throw ClassNotFoundException even the same code compiles and runs inside of the Toolkit environment
 - Verify all import statements in custom code and all its dependent classes' import statements to ensure classes declared are in the Search server's classpath at runtime

FEP 7 search index schema design update

Product Index



FEP 7 search index schema design update



Summary: customization points

- Index
 - Index schema – local index, extension index
 - Indexer – pre-processor + DIH, dataload
- Runtime
 - Search runtime (search profile) – expression providers, query pre- and post-processor
 - REST services for searching and browsing
- User interface
 - Management Center (search rules)
 - Storefront
- Deployment
 - Network topology
 - Index lifecycle flow
 - Caching and invalidation

Agenda

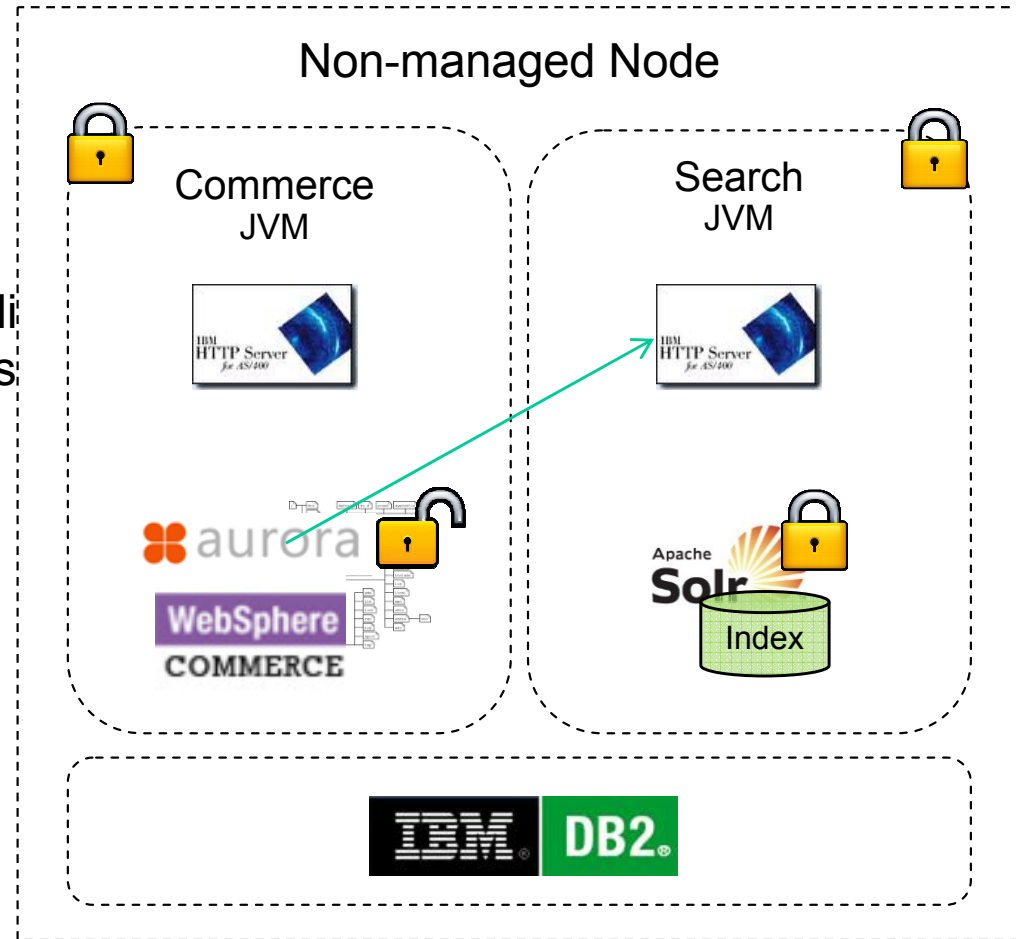
- Architecture update
 - Search server architecture
 - REST services

- Programming model update
 - Storefront interaction flow
 - Programming model compatibility
 - Index schema update
 - Customization points

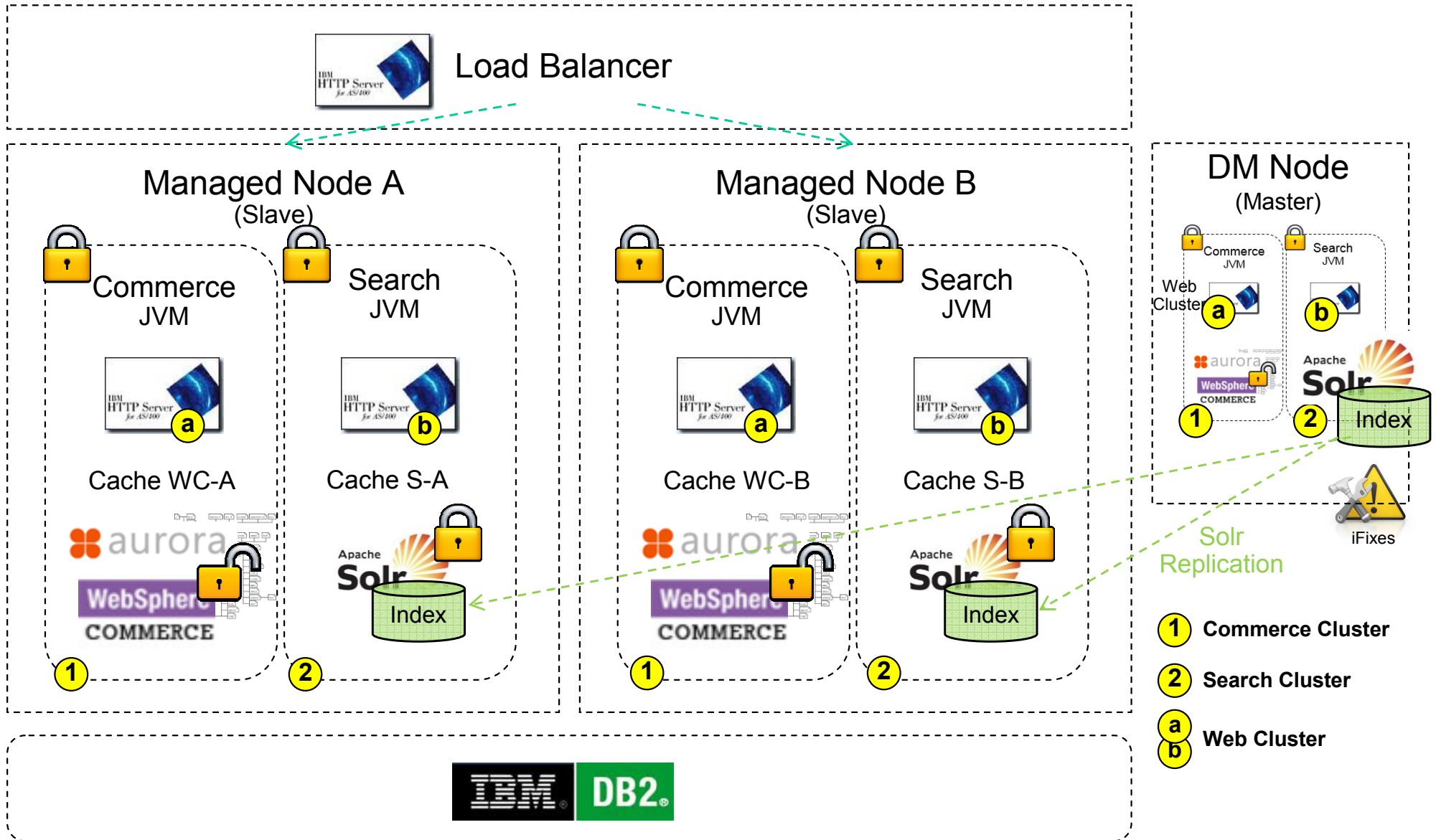
- Deployment update
 - Advanced deployment configuration
 - Index lifecycle
 - Cache and invalidation
 - Logging and tracing

Standard deployment configuration

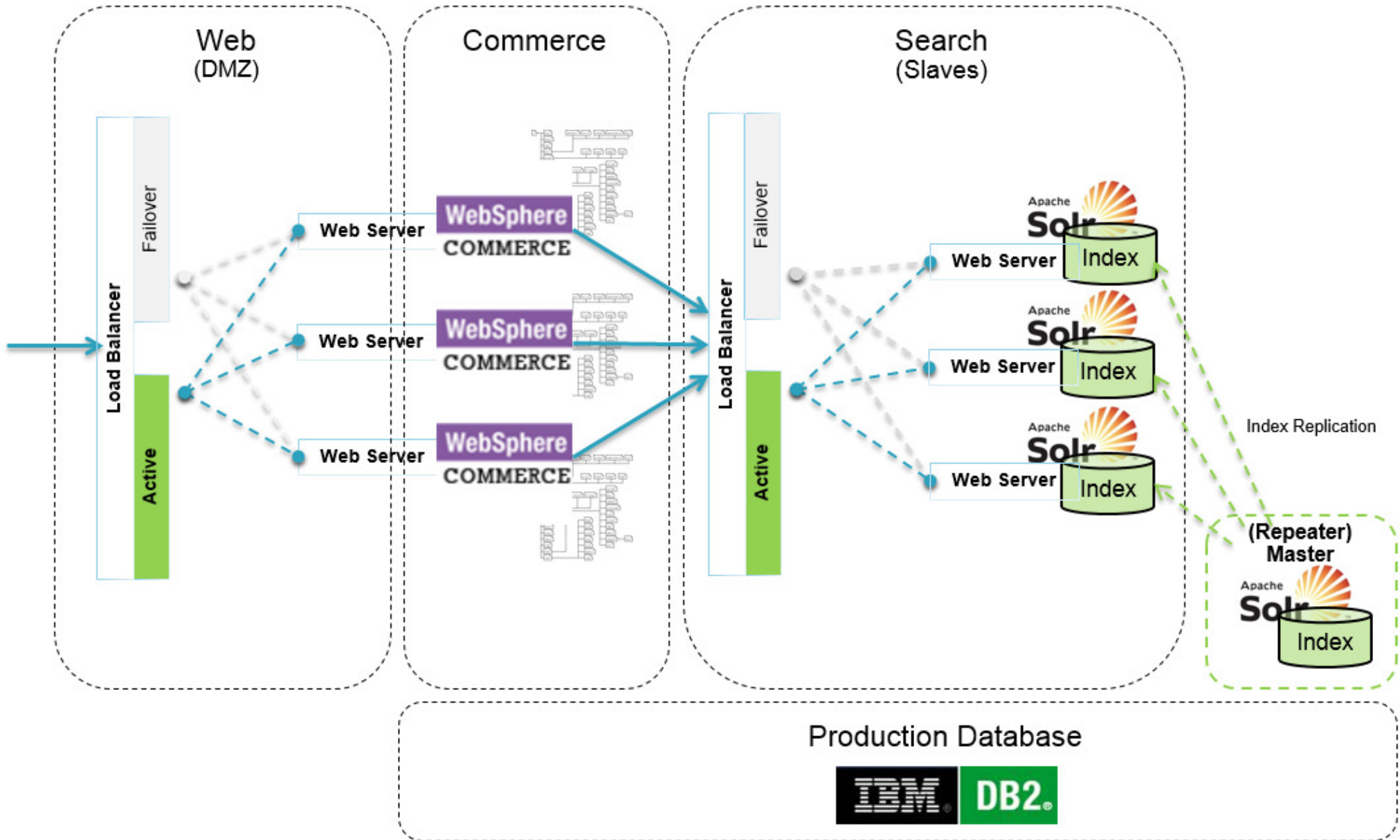
- **Standard** (Runtime OTB)
 - Dual WAS profile (WC, Search) on same (non-managed) node
 - Separate JDBC datasource and separate JVM heap space
 - Common database – DB2, Oracle, IBMi
 - Index created locally and configurations will be synchronized between two WAS profiles
 - WAS name binding used for defining hostname and port numbers
 - WAS security
 - WC – only Administrative enabled
 - Search – both Administrative and Application enabled (except for searching)
 - A self-signed certificated is imported by default; import your own CA-signed certificate in production environments
 - Same iFixes can be applied to both WC and Search



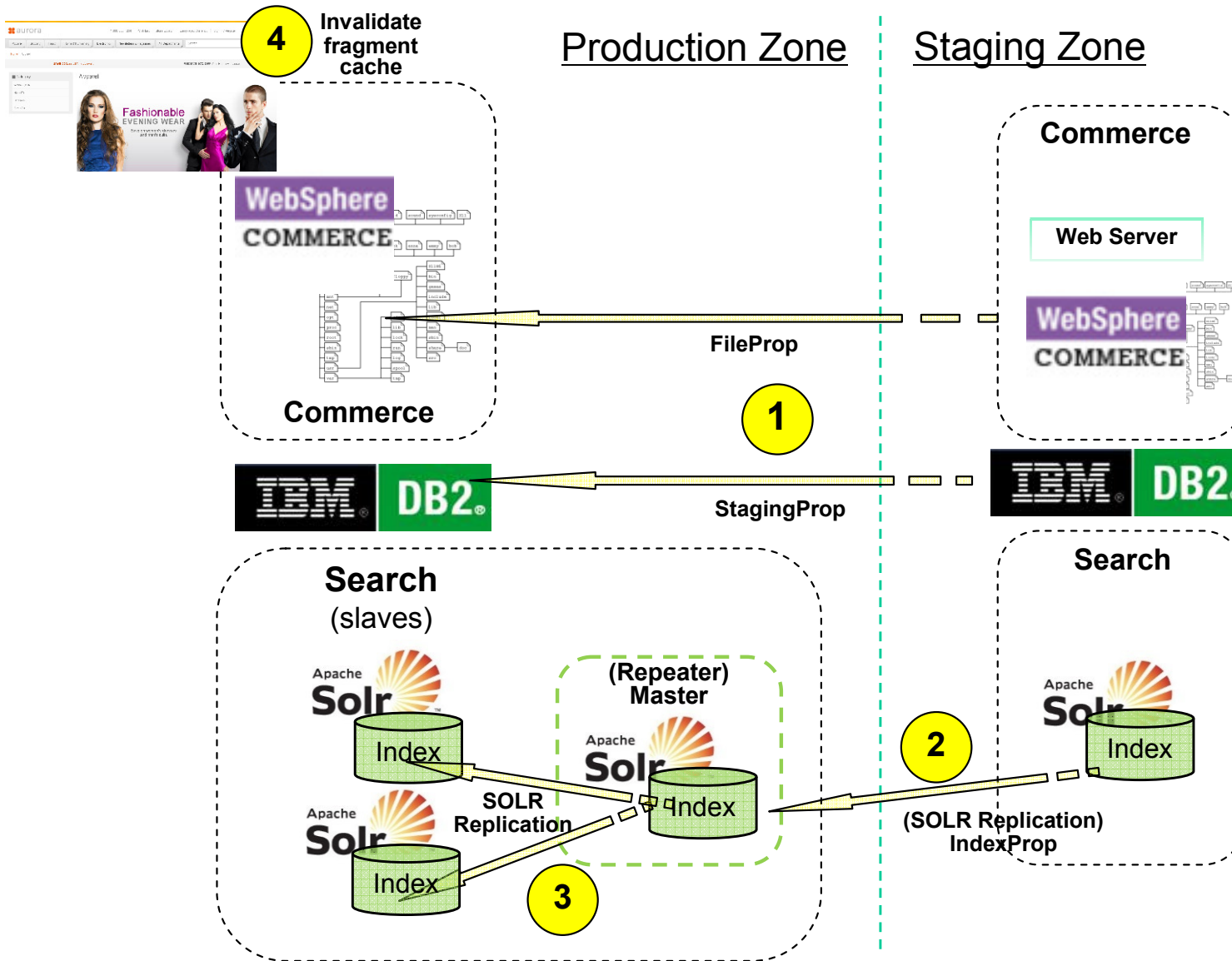
Advanced deployment configuration



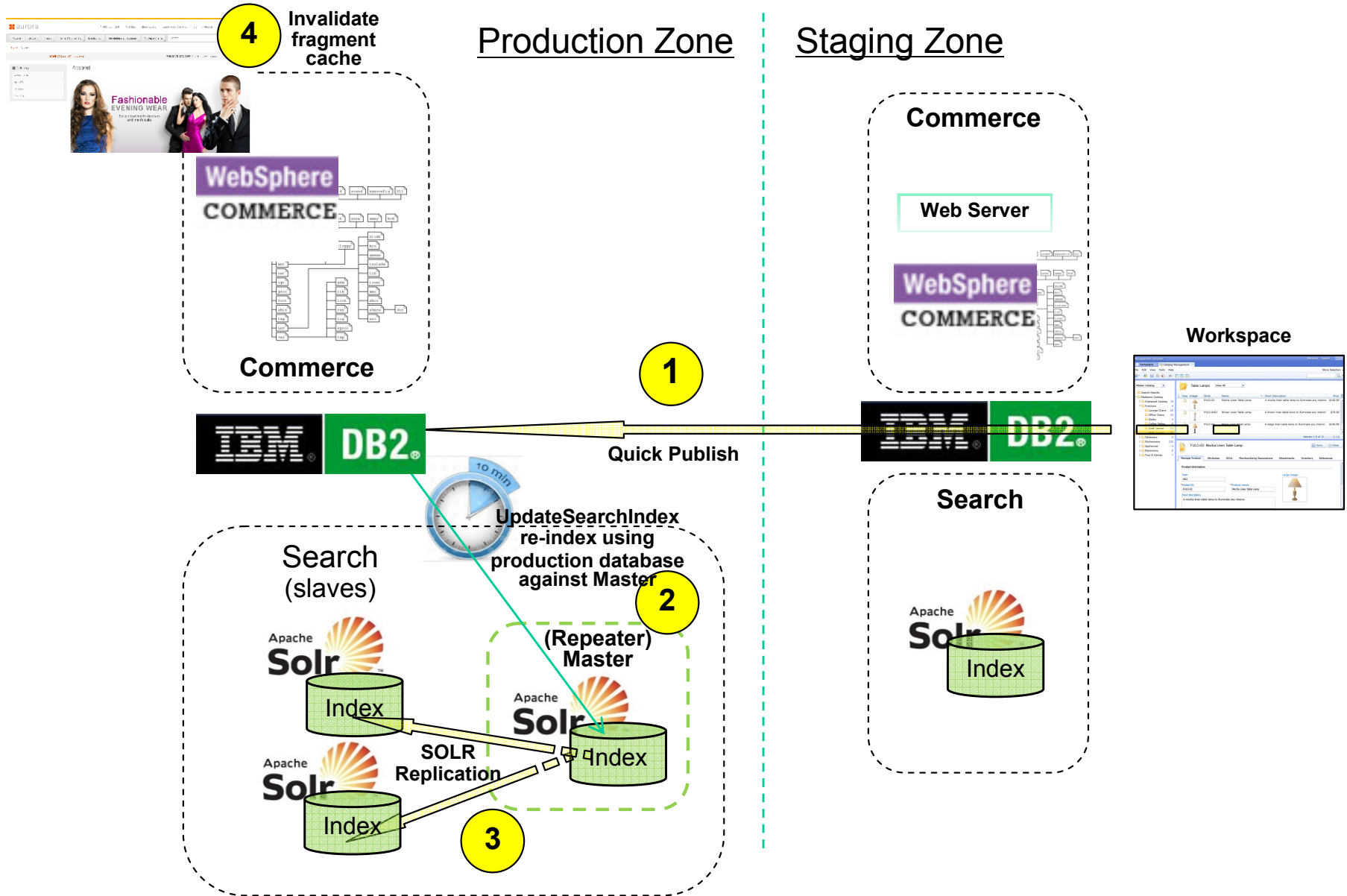
Advanced deployment configuration



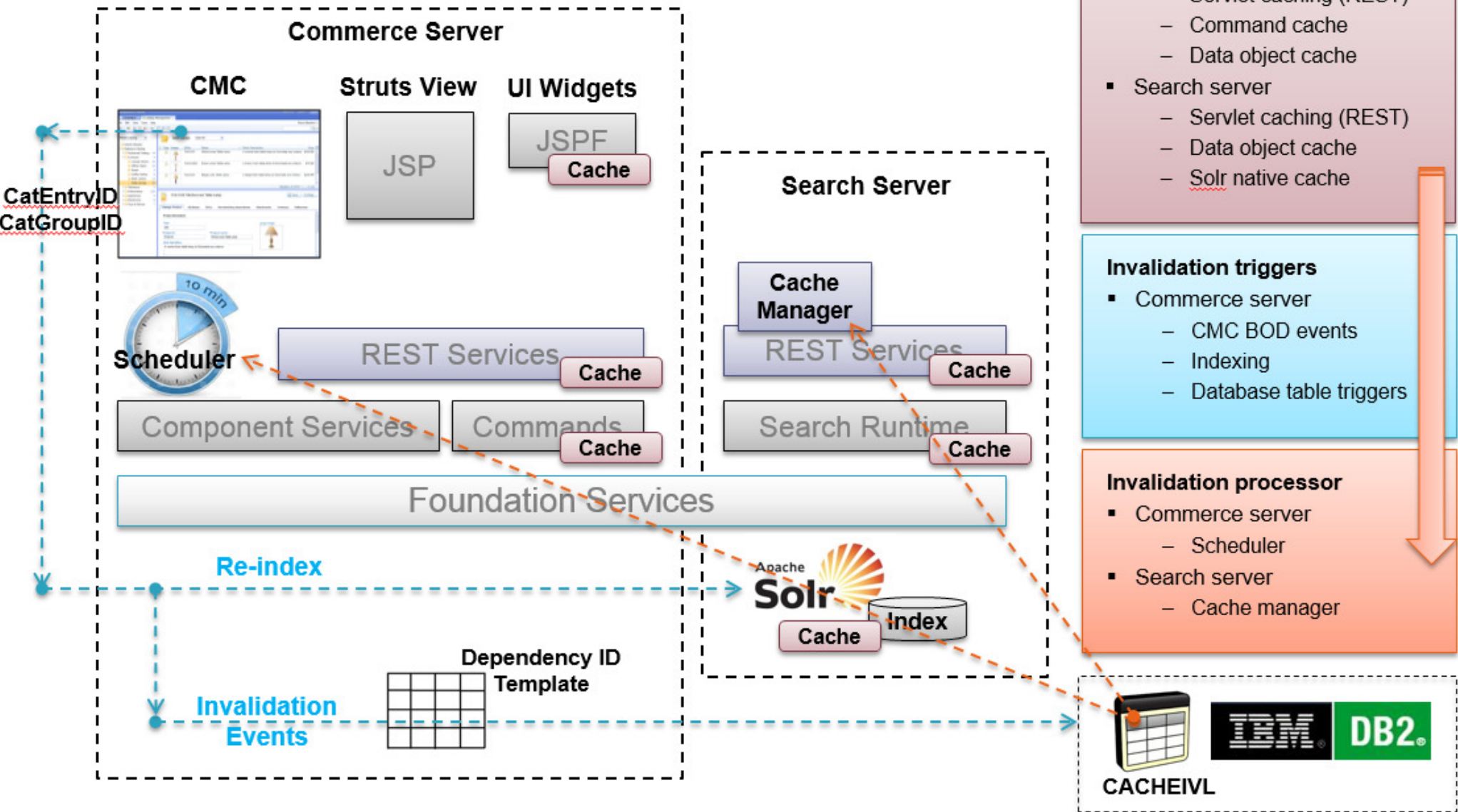
Index lifecycle – Staging propagation



Index lifecycle – Quick publish



FEP 7 Search caching and invalidation



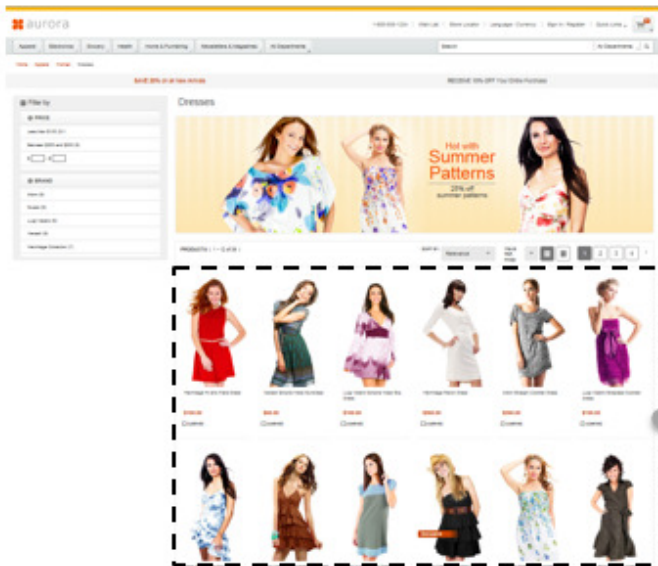
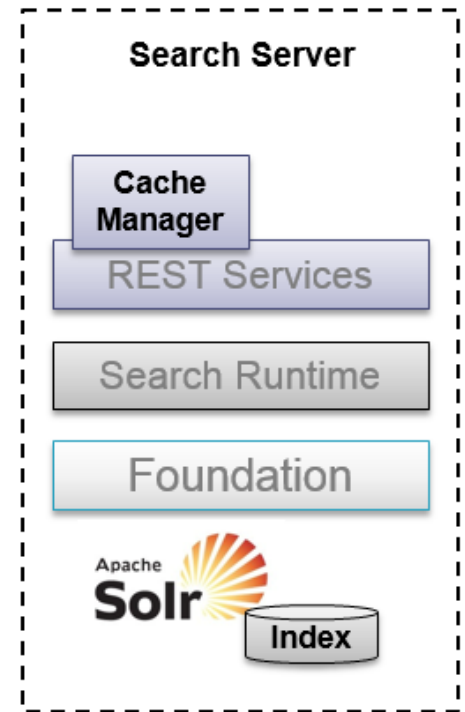
Logging and tracing

- Commerce Server

- Serviceability trace
- Transaction
- Command trace
- Component trace
 - Catalog
 - Foundation
 - Scheduler
 - Indexer
 - Cache invalidation

- Search Server

- Wink
- Serviceability trace
- Component trace
 - REST
 - Foundation
 - Cache invalidation
- Solr



References

- **Architecture overview**

http://pic.dhe.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=%2Fcom.ibm.commerce.developer.doc%2Fconcepts%2Fcsdsearchinteractionov_fep7.htm

- **Search programming model, extension points, and customization tasks**

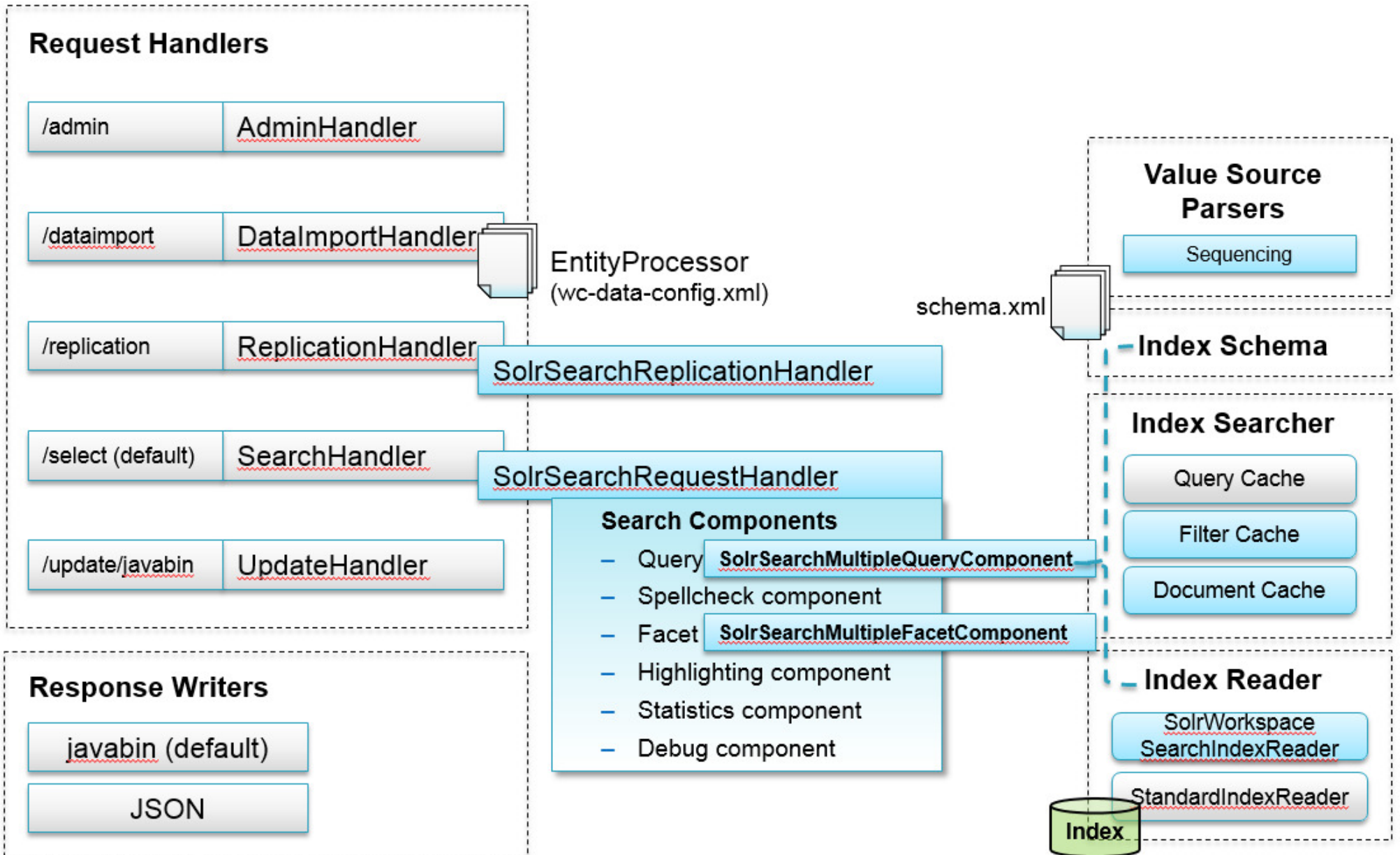
http://pic.dhe.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=%2Fcom.ibm.commerce.developer.doc%2Fconcepts%2Fcsdsearchguide_fep7.htm

- **Administering WebSphere Commerce Search**

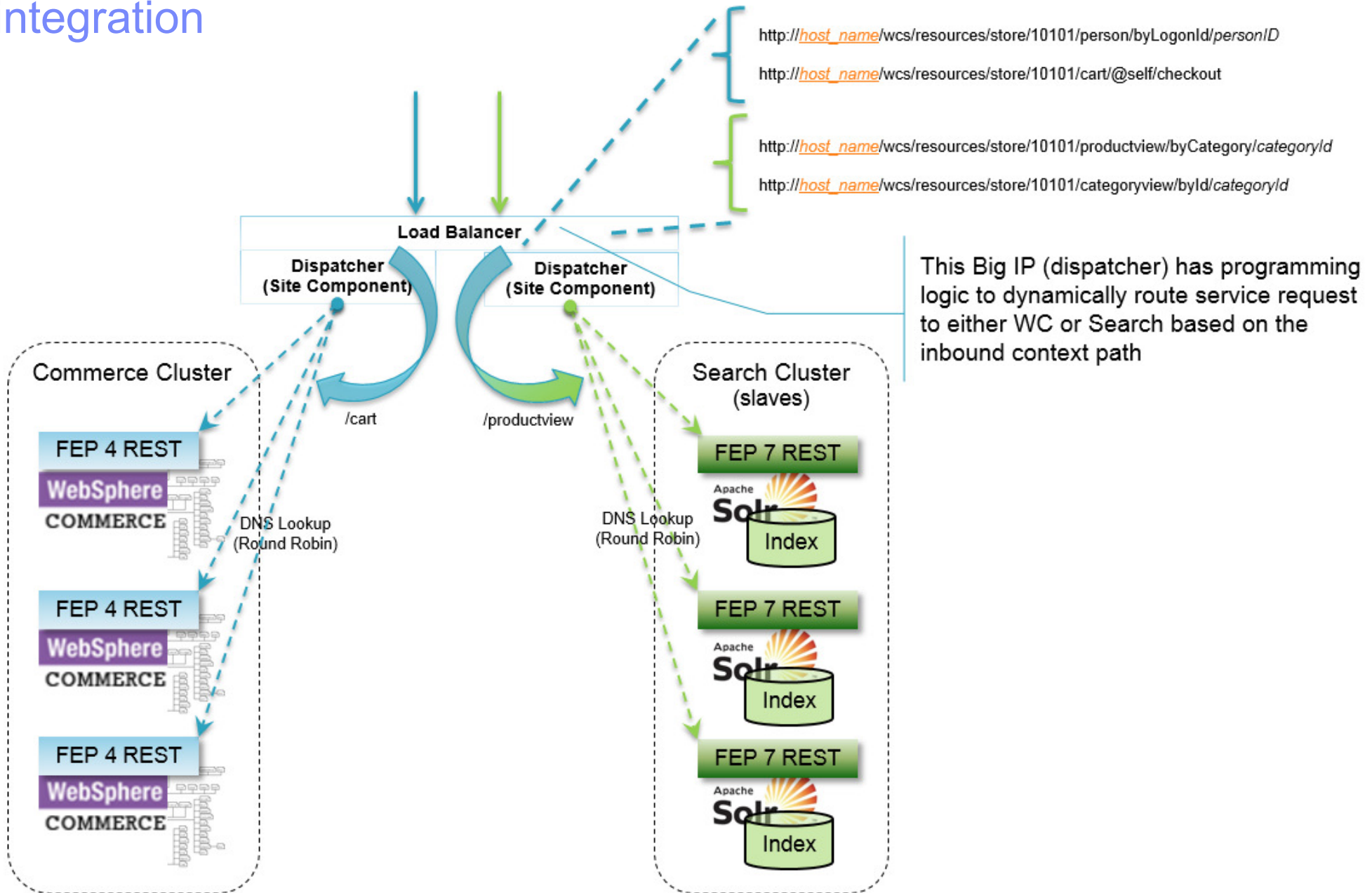
<http://pic.dhe.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=%2Fcom.ibm.commerce.developer.doc%2Fconcepts%2Fcsdmanagesearch.htm>

Backup slides

Search extensions in Solr 4.3



Advanced deployment – REST service routing for 3rd party integration



Thank You!

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, Coremetrics, DB2, PowerVM, Rational, WebSphere, and z/VM are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2014. All rights reserved.