Lucas Jellema – AMIS (Nieuwegein, The Netherlands)

INSTANT AGILITY IN ORACLE FUSION MIDDLEWARE THROUGH DESIGN TIME @ RUN TIME

Oracle Open World 2011

COMMITTED TO ICT. INVOLVED IN PEOPLE.

IT DELIVERS ACCORDING TO BUSINESS SPECIFICATIONS

The Outside World
(Consumers, Partners, Customers, Government)

Emails, Alerts, Reports

Web Site

Web Application

Mobile Application

Web Service

Emails, Alerts, Reports

Enterprise Portal

Mobile App

Web Service

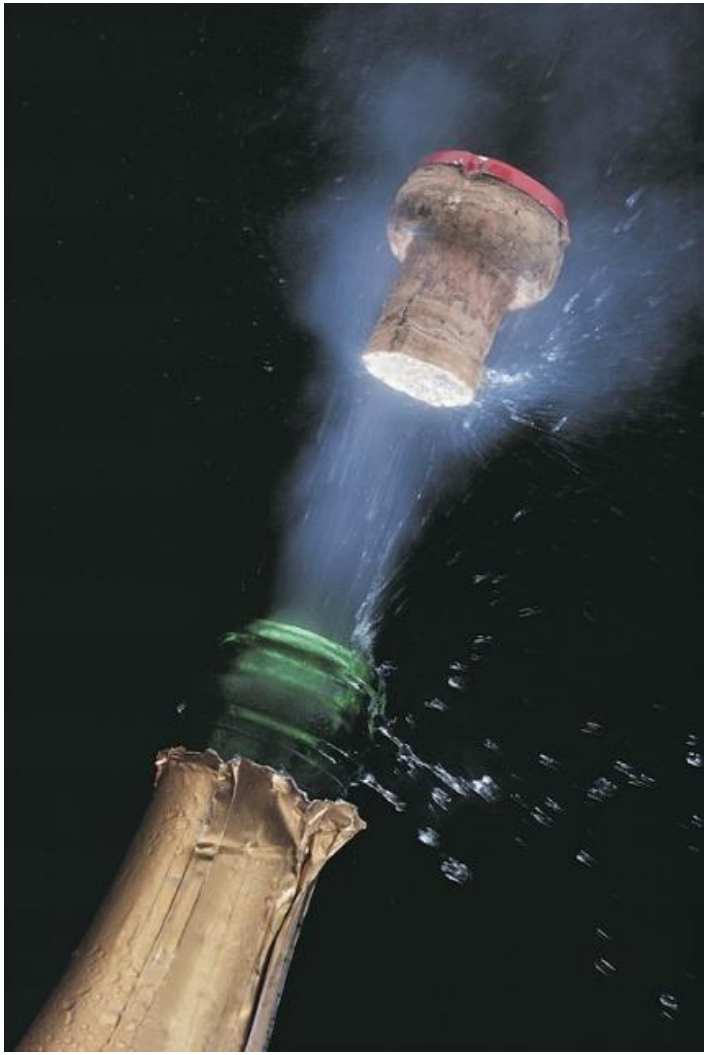
Dashboard

Process Coordinator & Todo List

Task UI

Enterprise

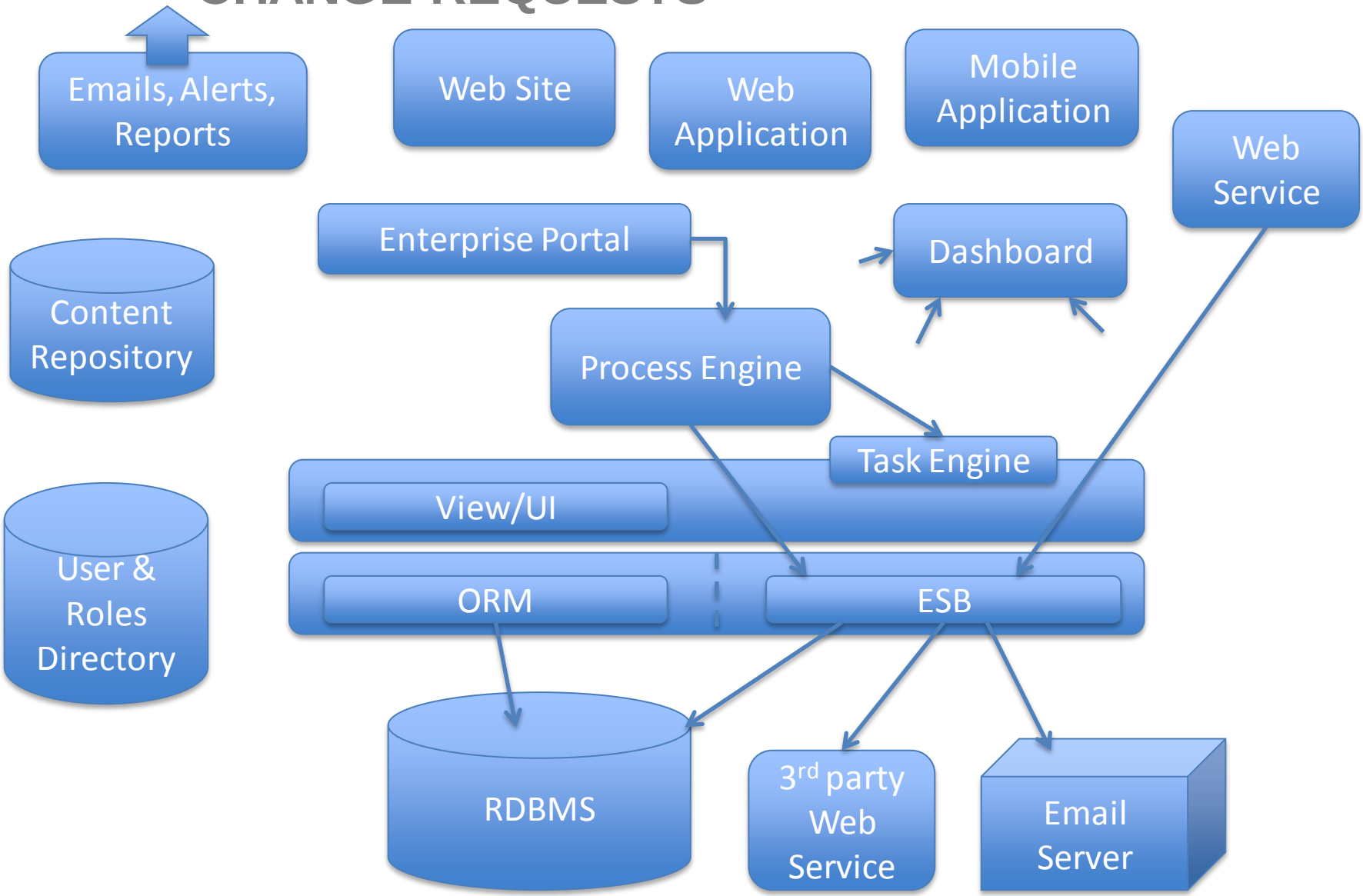
WHAT HAPPENS RIGHT AFTER DEPLOYMENT?



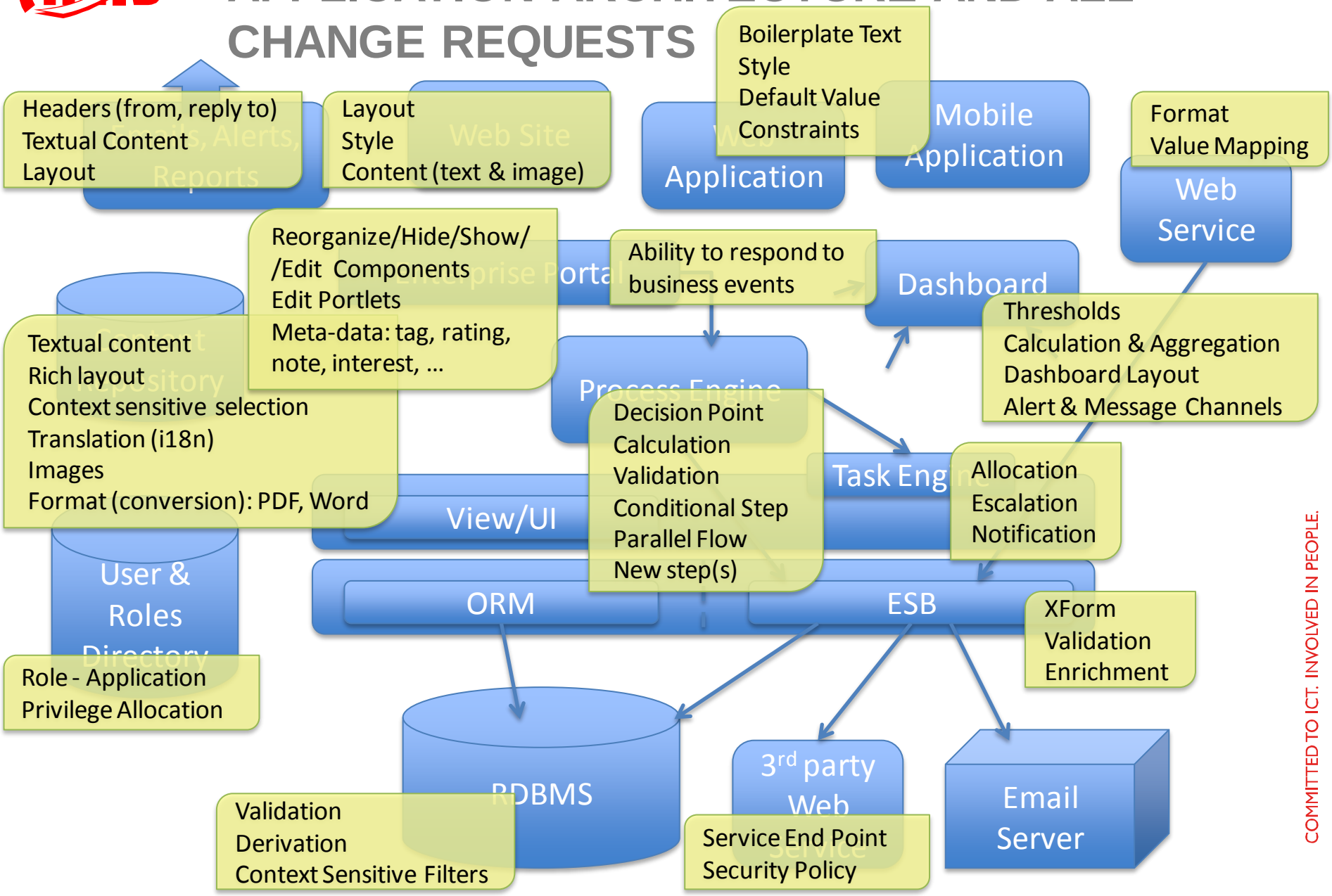
WHAT HAPPENS RIGHT AFTER DEPLOYMENT?

- **‘Sorry, this is not what we meant’**
- **‘Well, this was right when we asked for it – but that is no longer the case. Things have changed.’**
- **‘This is right for some user groups – but not as it turns out for all’**
- **‘Hi – we are from marketing and we want to launch a campaign to respond to our biggest competitor and we need immediate changes’**
- **‘It is perfect! Exactly what is needed for the current business situation.’…. ‘Well, the situation has slightly evolved. We need to adapt – asap please.’ (repeat many times)**

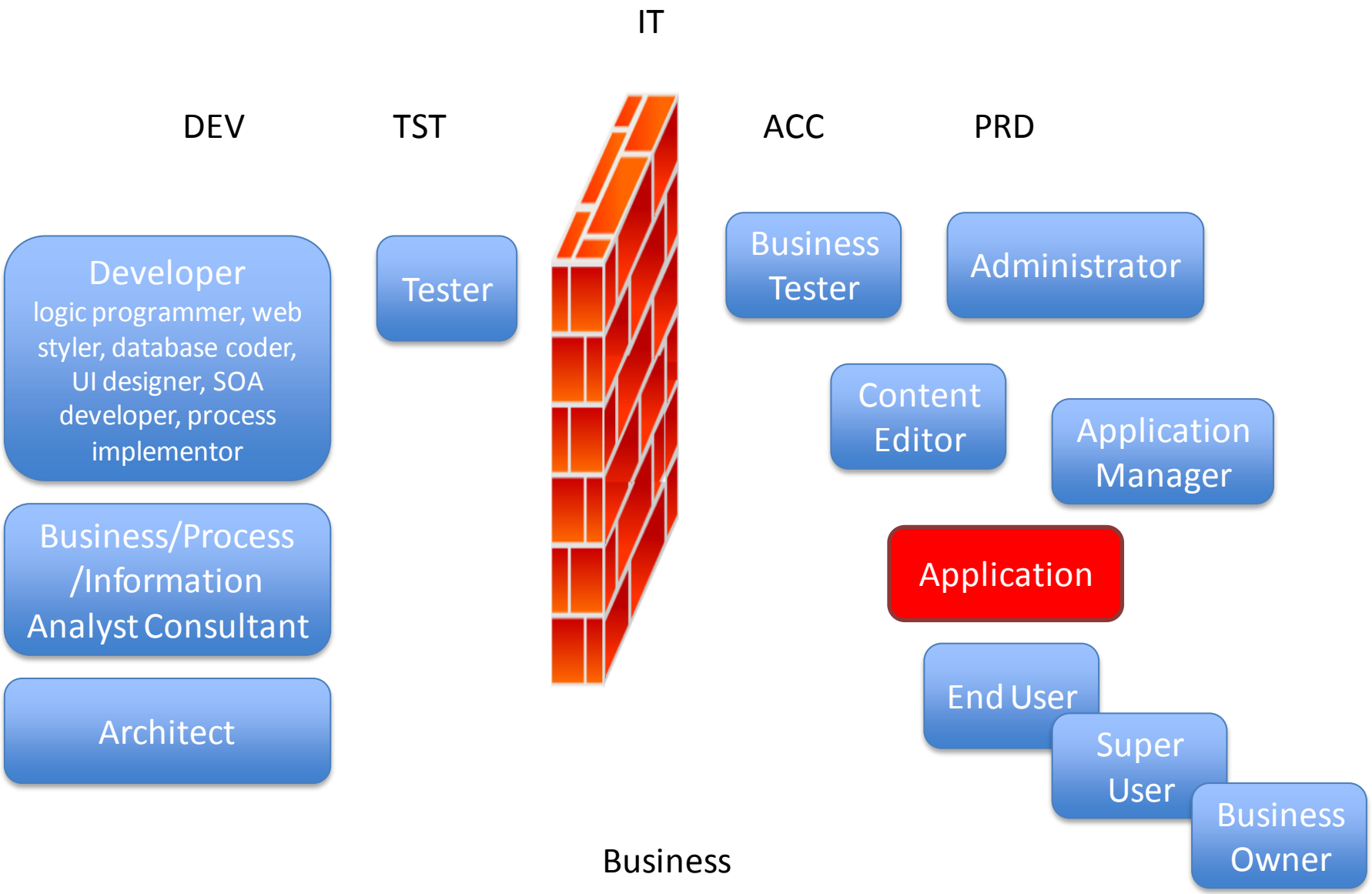
APPLICATION ARCHITECTURE AND ALL CHANGE REQUESTS



APPLICATION ARCHITECTURE AND ALL CHANGE REQUESTS



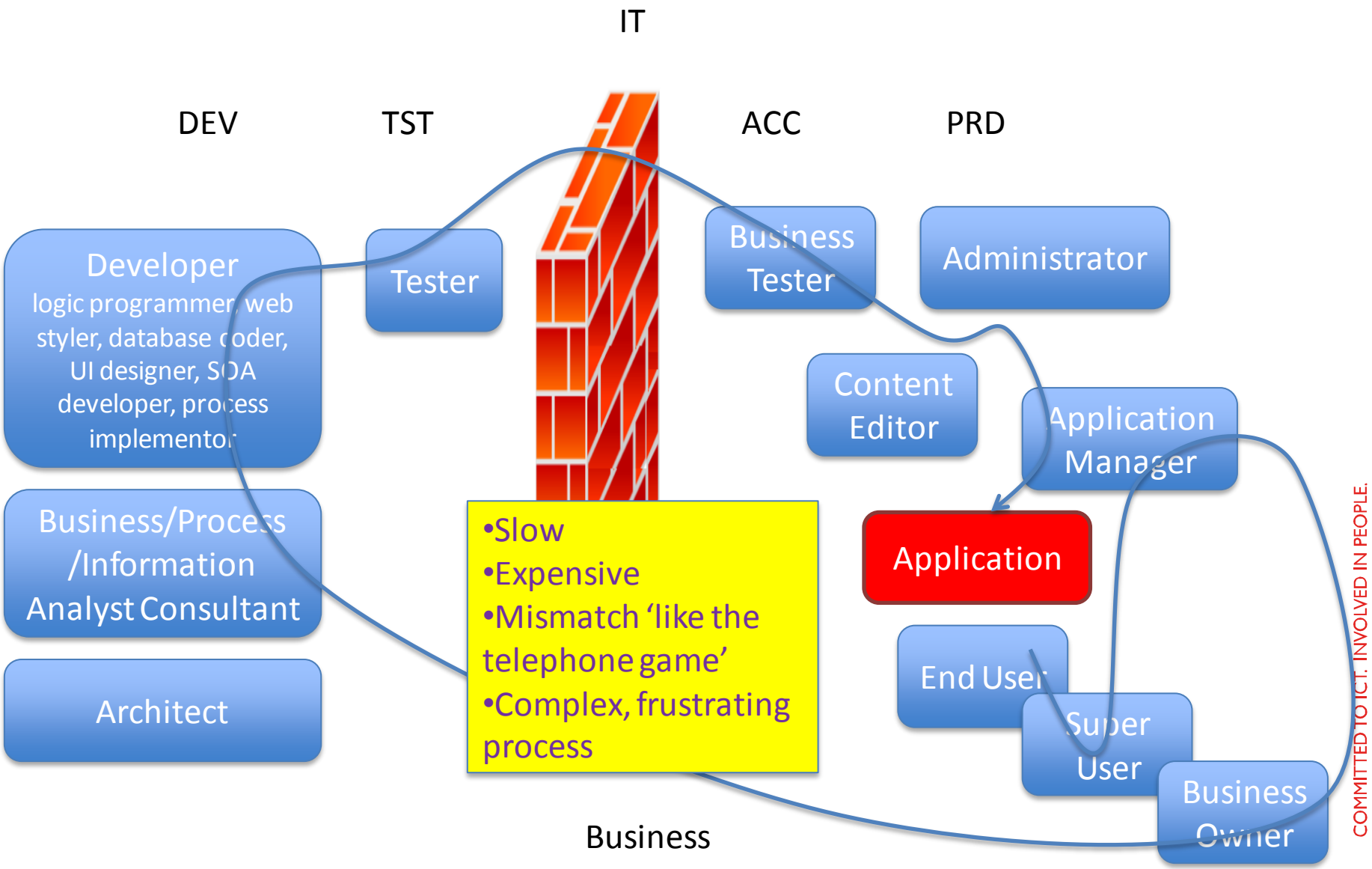
ORGANIZATION & ENVIRONMENT



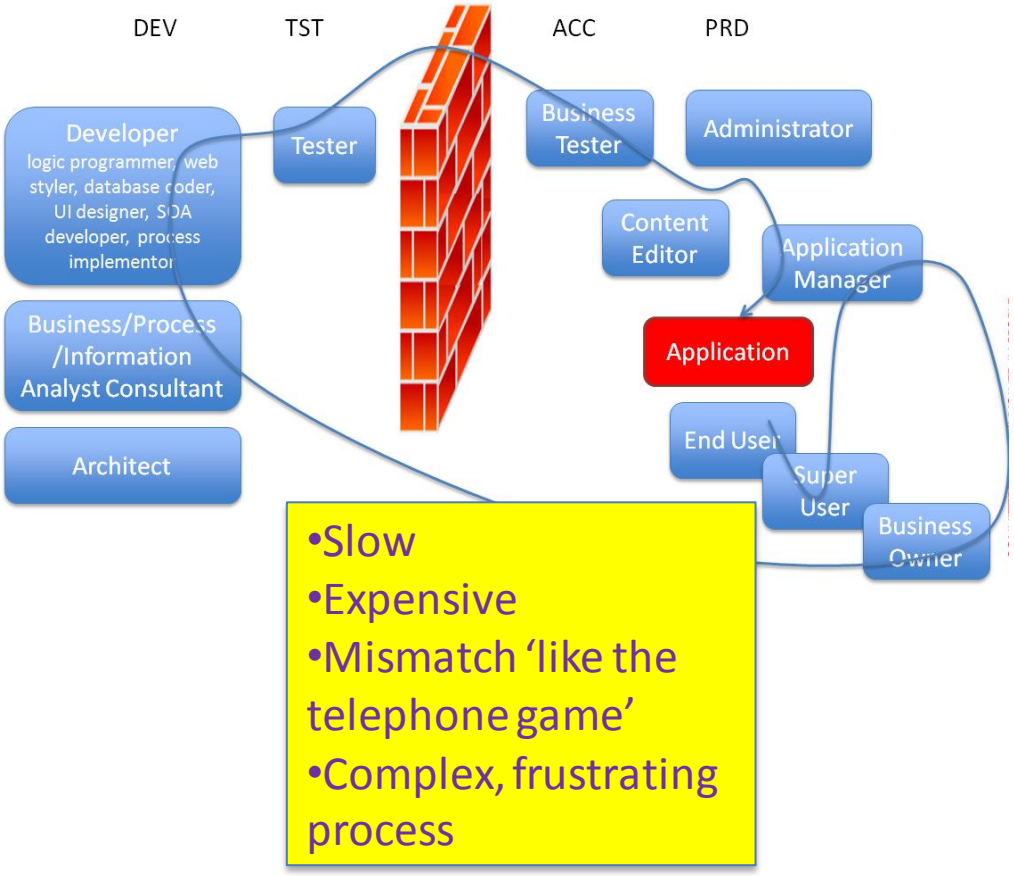
BUSINESS AGILITY

- ***Adapt* to changing business requirements**
  - Rapid
  - Efficient (cost wise)
  - Low risk
  - GWYRN: Get what you *really* need

ORGANIZATION & ENVIRONMENT



WHAT IS VS WHAT SHOULD BE

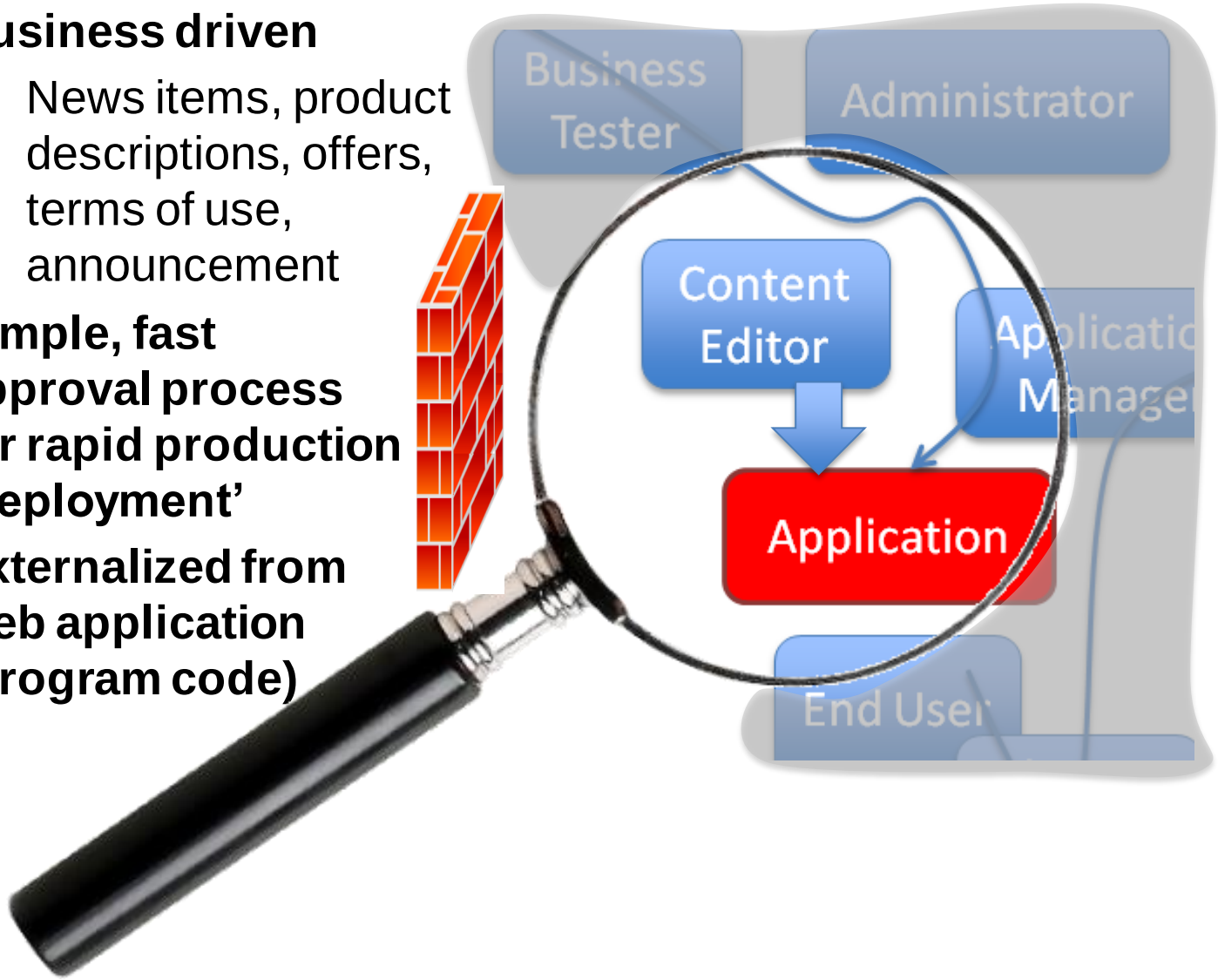


?

- Rapid
- Efficient (cost wise)
- Low risk
- GWYRN: Get what you *really* need

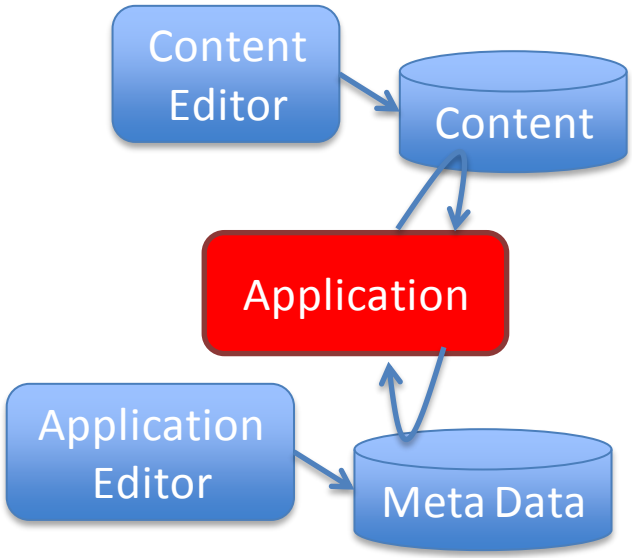
AGILE CONTENT MANAGEMENT

- **Business driven**
  - News items, product descriptions, offers, terms of use, announcement
- **Simple, fast approval process for rapid production 'deployment'**
- **Externalized from web application (program code)**



THE ESSENCE OF THE SOLUTION TO THIS TRADITIONAL PROBLEM

- **All aspects of applications, processes, services that may require run time change**
  - should be meta-data driven
  - defined outside of program code
  - “externalize changeable aspects outside of applications, just like content is externalized”



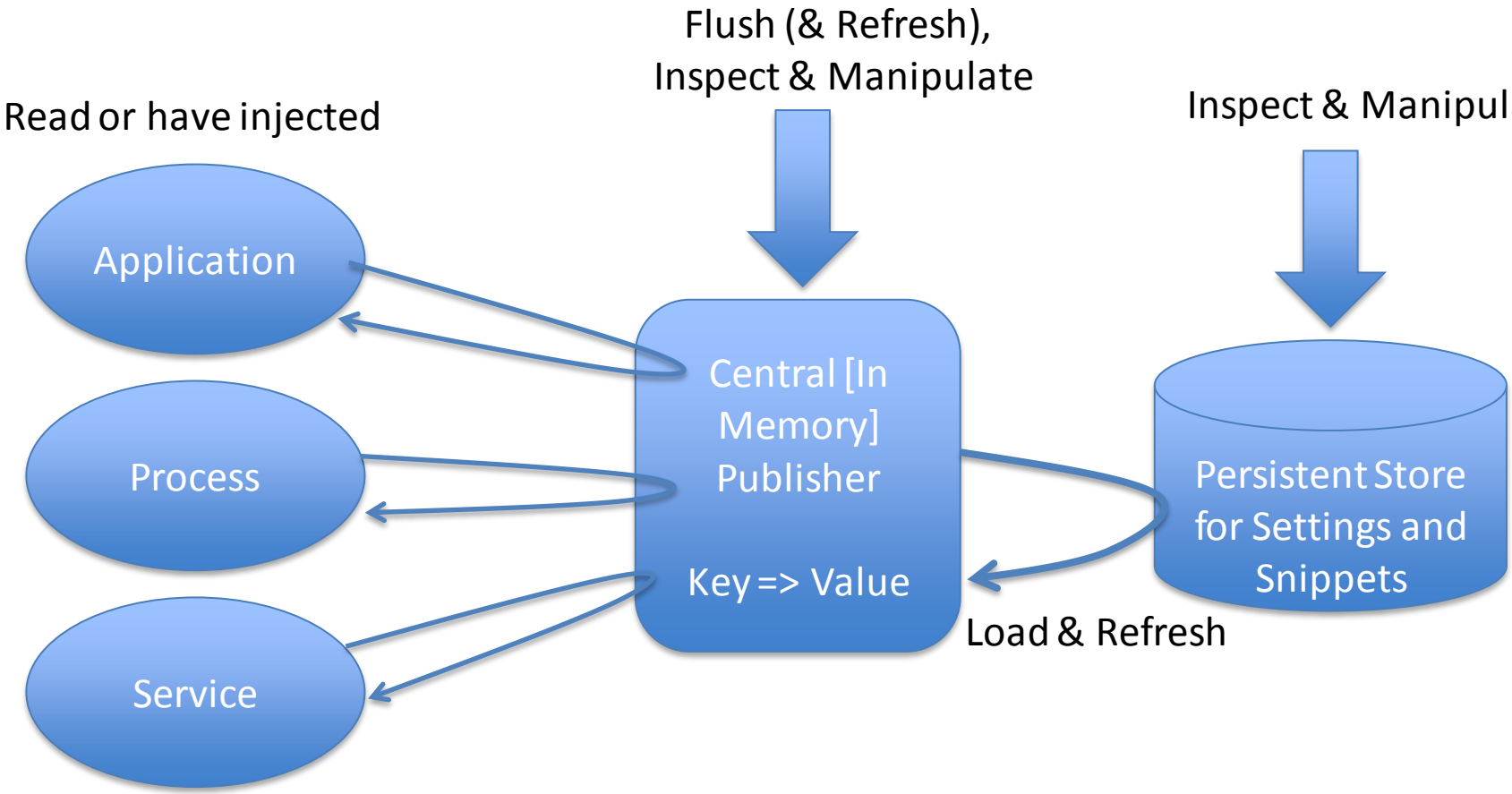
THE ESSENCE OF THE SOLUTION TO THIS TRADITIONAL PROBLEM

- **Applications, UI components, services, processes and other elements need to acquire or have injected the meta data that governs part of their behavior**
- **Run time infrastructure is required**
  - to support change of meta-data – WYSIWYG style -
  - And to absorb/apply meta-data changes at run time without restarting application components
- **Ideally: cater for multiple, co-existing, context sensitive sets of meta-data changes**
- **A 'run time application editor' role should be introduced**

CUSTOM IMPLEMENTATION OF (FUNCTIONAL) DESIGN TIME @ RUN TIME

- Externalize meta-data with normal, non FMW means
- Centralize parameters in MBeans that can be manipulated with JMX from outside JVM
- Resource Bundles (multi-dimensional) based on database table, cached but dynamically refreshable
- User preferences to drive behavior of UI
- JSF PhaseListener to post-process UI component tree based on meta data
- Custom JSF components that dynamically create UI components – driven by meta data
- Servlet filter to manipulate CSS resources based on current context and meta data settings
- Singleton BPEL process to expose (and manipulate) parameters for use in other BPEL processes
- Set up event infrastructure and publish relevant business events wherever they originate

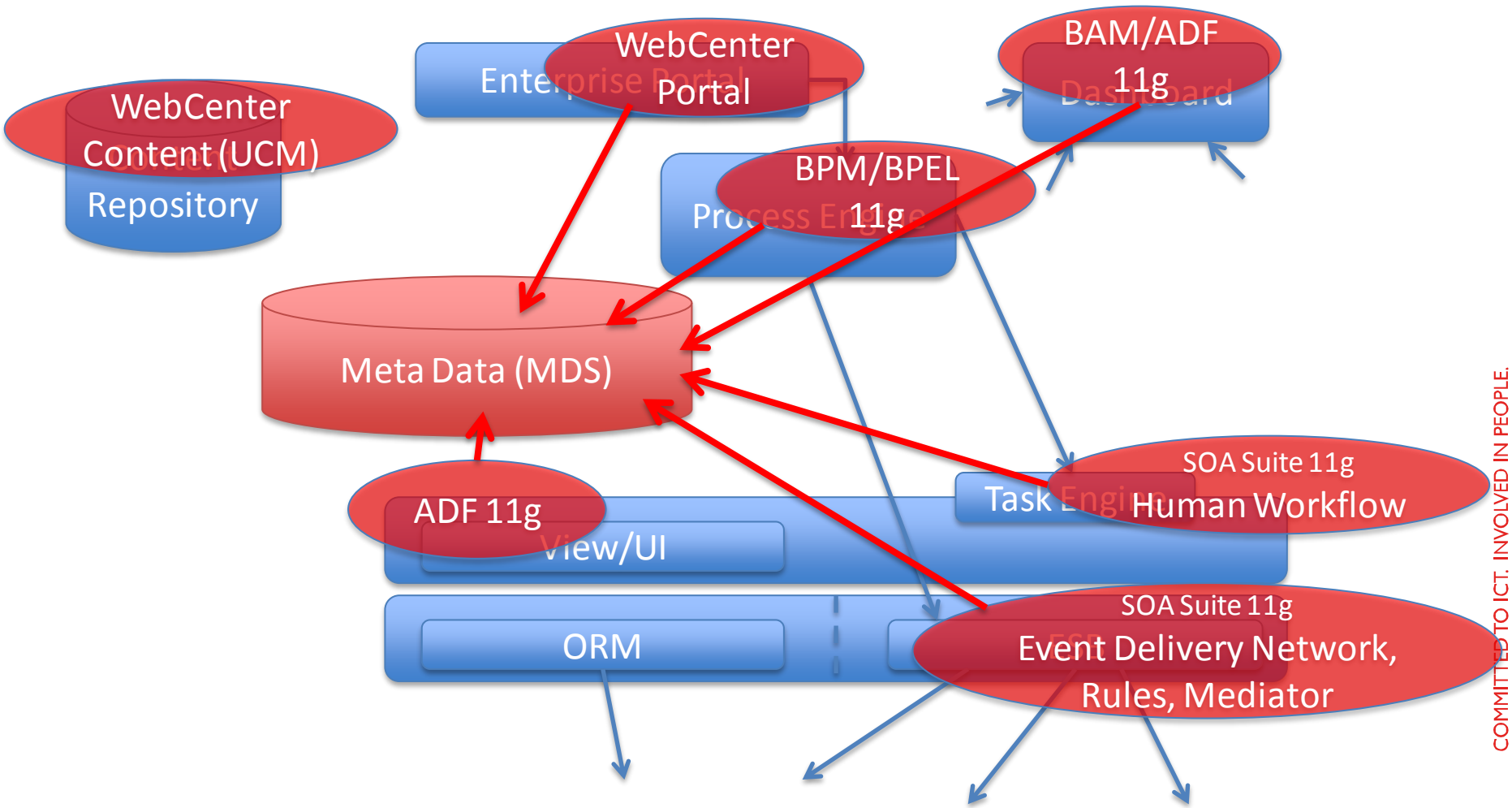
CENTRALIZE CONSTANTS, PARAMETERS, FRAGMENTS, CALCULATIONS, ..



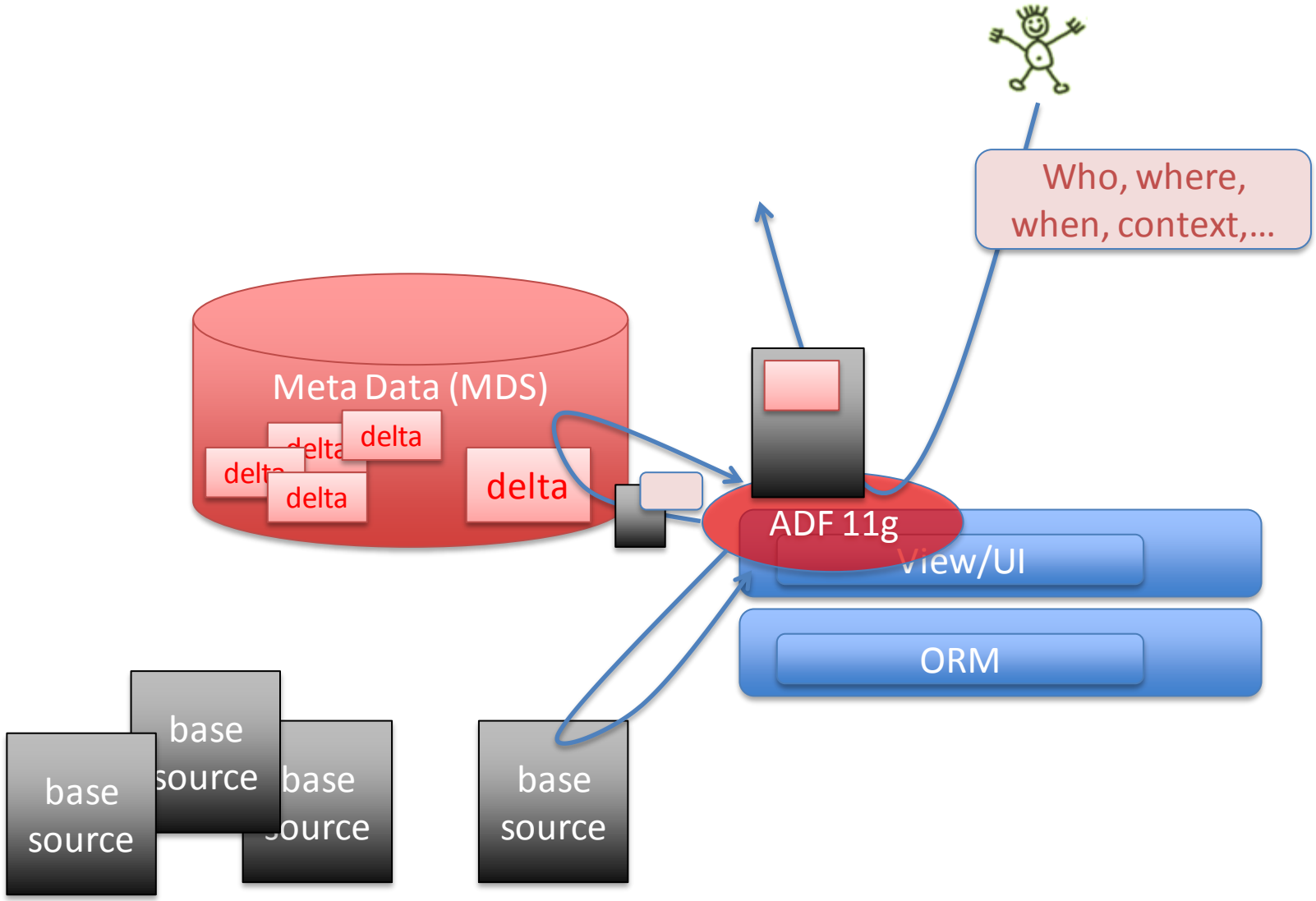
CHALLENGES

- Define process for making, testing, deploying and distributing run time changes
- Have changes applied in staging or sandbox – test and approve before go-live
- Deploy changes with(out) impacting running sessions and process instances
- Train staff to apply run time changes
- Enforce authorization on run time editing
- Have changes survive Application Upgrade
- Feed changes back to developers and ‘regular’ application life cycle
- WYSIWYG tooling for making run-time changes
- Prepare application components for run-time editing

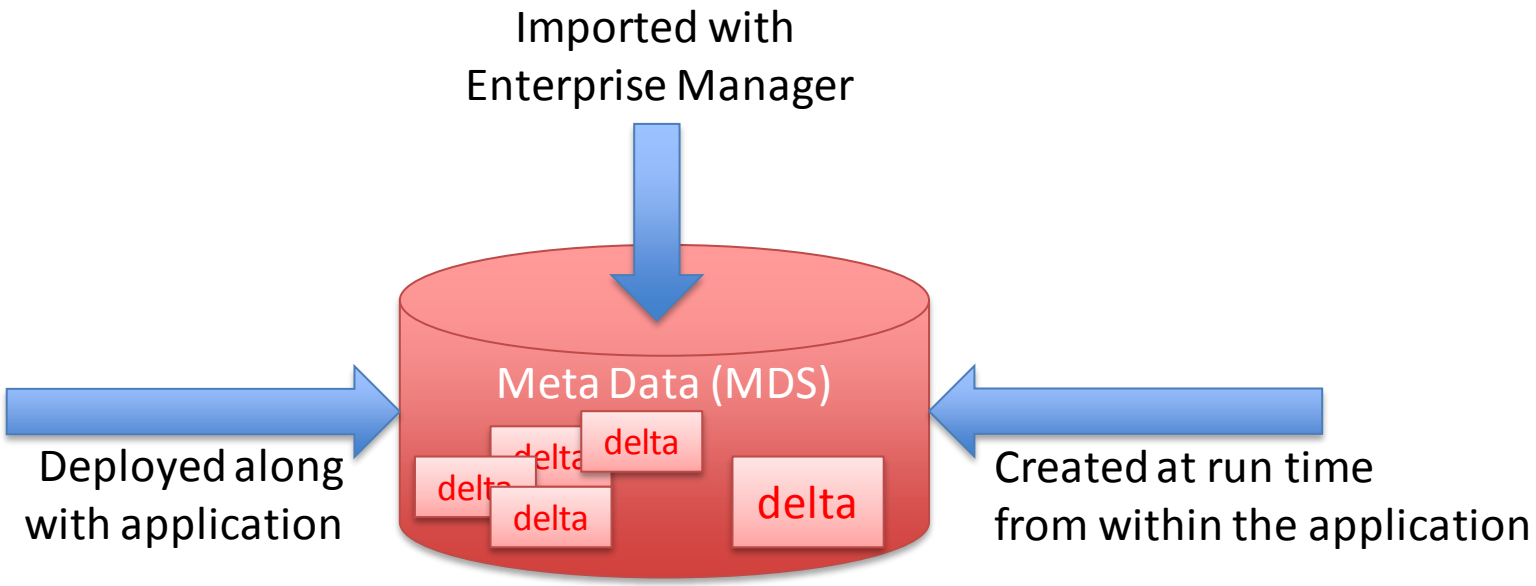
RUN TIME META DATA IN ORACLE FUSION MIDDLEWARE



CONTEXT SENSITIVE SOURCE CUSTOMIZATION AT RUN TIME



MDS CUSTOMIZATION MANAGEMENT



ADF CHANGE PERSISTENCE

- **Built-in infrastructure to record and persist configuration of UI components by the user**
  - Persistence for the duration of the session (in memory) or cross-sessions (as a delta in MDS)
- **Example of built-in change persistence**
  - Column configuration in tables (hide/show, reorder, resize, sort)
  - Expand/collapse of panel boxes, accordion panels and panel headers
  - Expand/collapse state and divider position in panel splitter
- **Change persistence has to be explicitly enabled in the ADF application – at the desired level**
  - When enabled: it is entirely declarative and transparent to developer and user

EXPLICIT PROGRAMMATIC ADF CHANGE PERSISTENCE

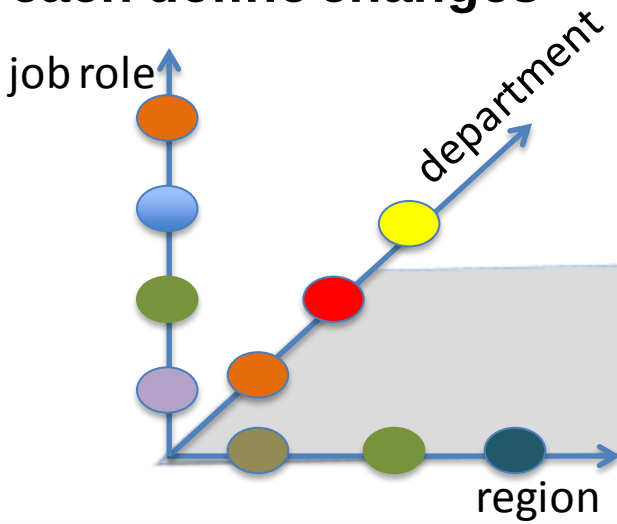
- **In addition to declarative, implicit change persistence, ADF applications can do explicit change persistence**
- **The application at run time can use the ChangeManager object to persist changes**
  - Those changes are stored in the session or MDS
- **When a new view is constructed, changes are applied by the ChangeManager – prior to rendering**
  - From MDS and session
  - Both implicit and explicit changes (no distinction made)
  - (Ordered by customization layer)
- **Example:**
  - User determines the order of navigation tabs
  - Application saves the new order to ChangeManager
  - The change is stored in session or MDS
  - Next time when the page is accessed, the change is retrieved (from session or MDS) and applied to page

INTRODUCING CUSTOMIZATION LAYERS

- **Customize**
  - One (new) size fits all
  - Multiple sizes to fit all
  - Personal(s)ize to fit individual

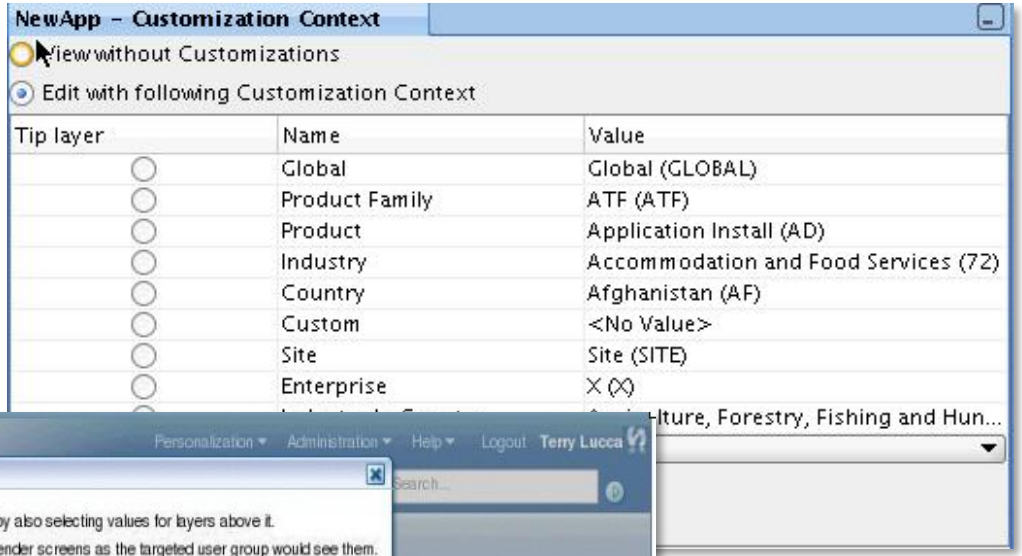
CUSTOMIZATION LAYERS

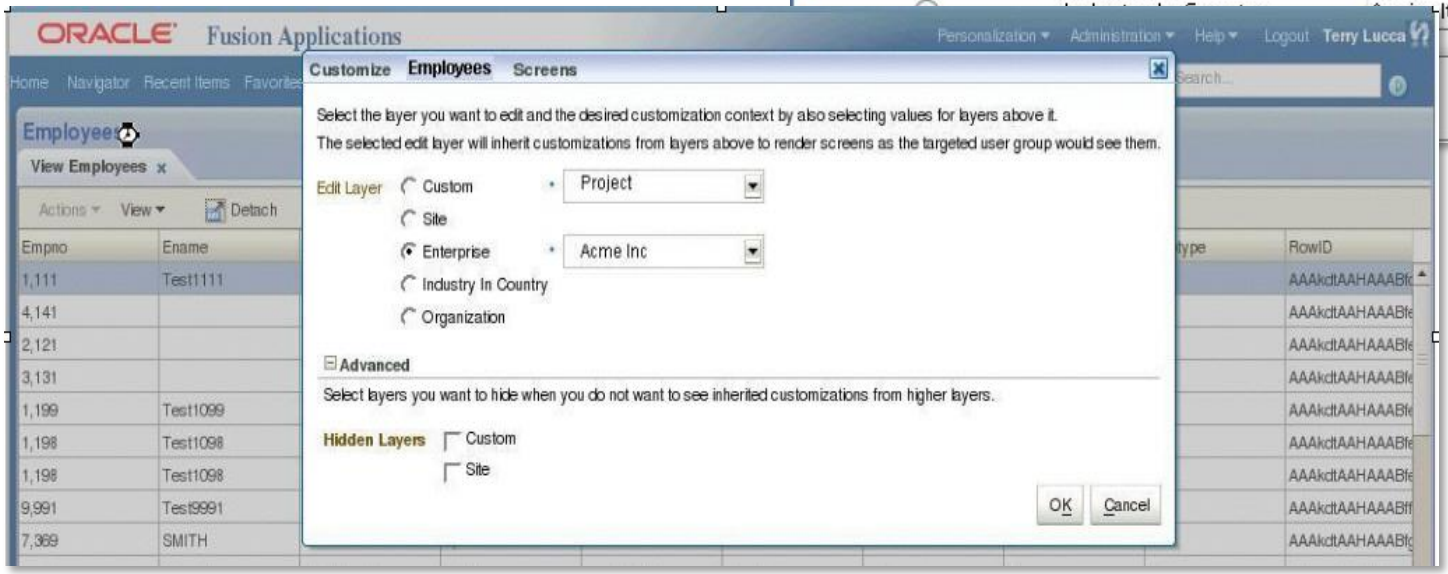
- **Customizations may be required for various, unrelated reasons – that may apply at the same time**
  - Additional fields because of a more senior role
  - Fewer fields because non applicability in a certain region
  - Fields with different constraints, default values and display characteristics because of departmental rules
- **ADF can work with many customization layers – unrelated dimensions that may each define changes**
- **For each layer, the current value is determined and the customization in that layer for that value is applied**
- **The order of layers is crucial!**



FUSION APPLICATIONS
11 CUSTOMIZATION LAYERS

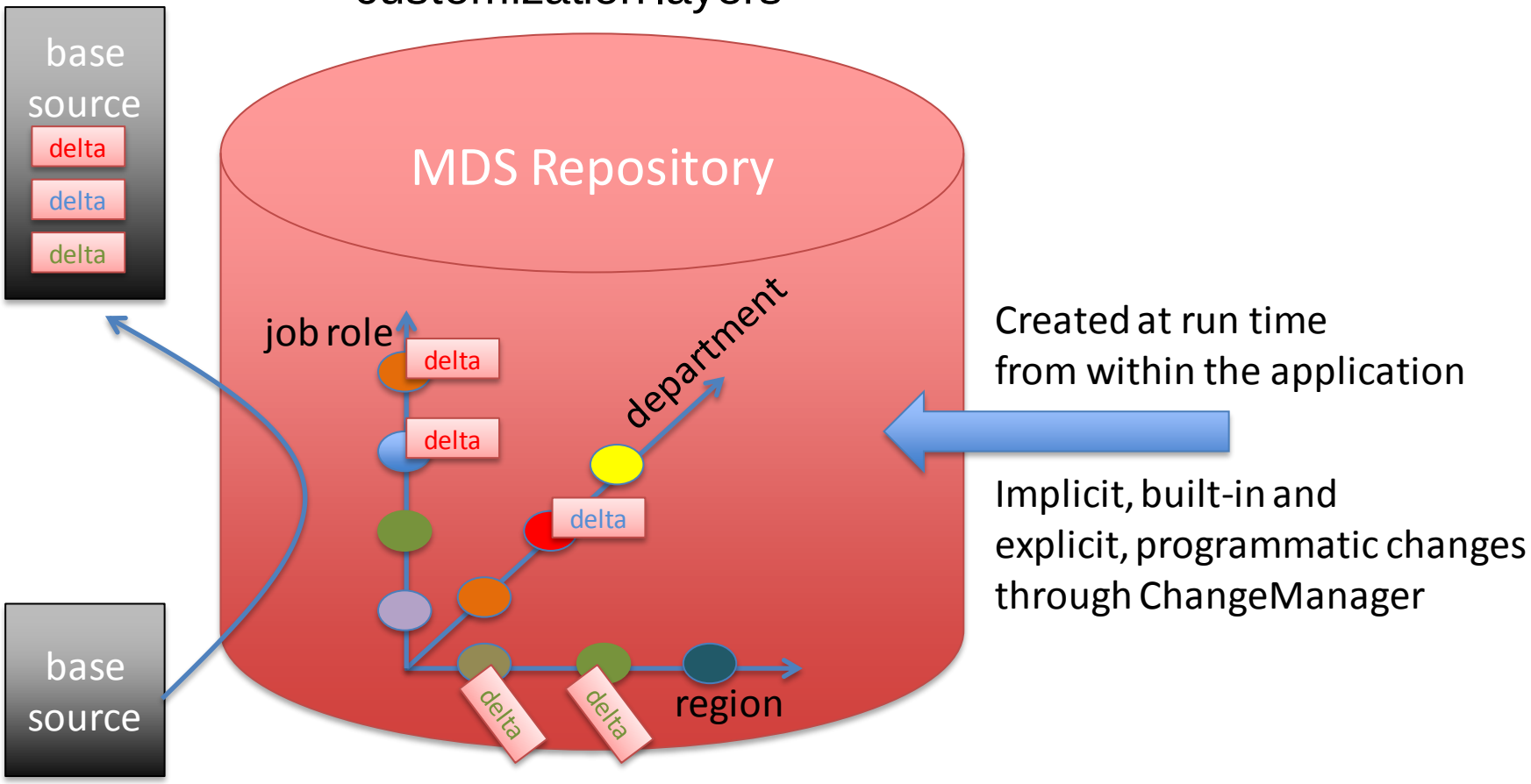
- **Fusion Applications leverages Customization to fine-tune *one size fits no one* base product**
- **11 customization layers are used to tune along different dimensions**
- **Also at run time**





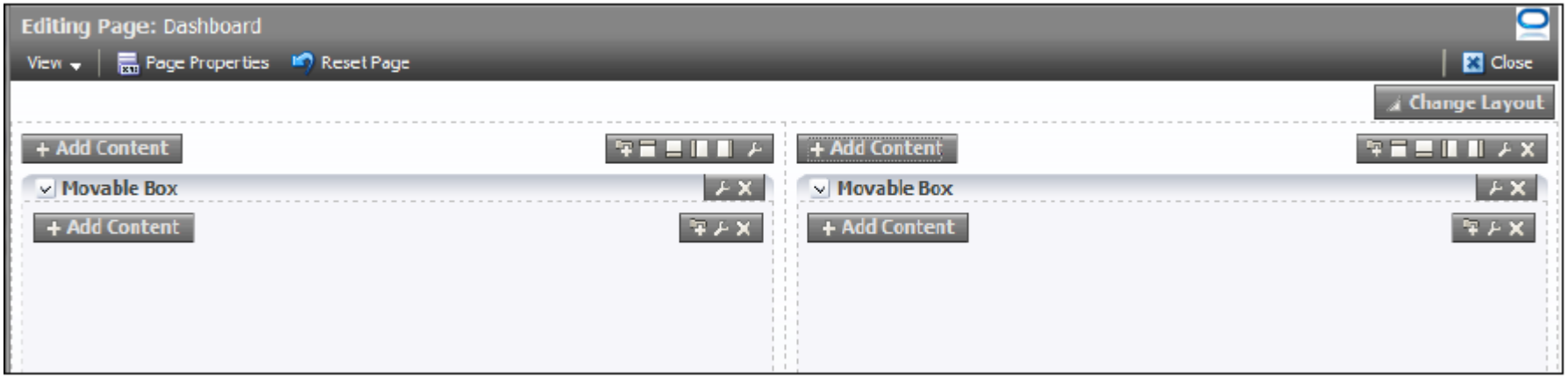
ADF CUSTOMIZATION

- **Changes per artifact (page, task flow, template, …)**
  - associated with predefined and dynamically derived customization layers



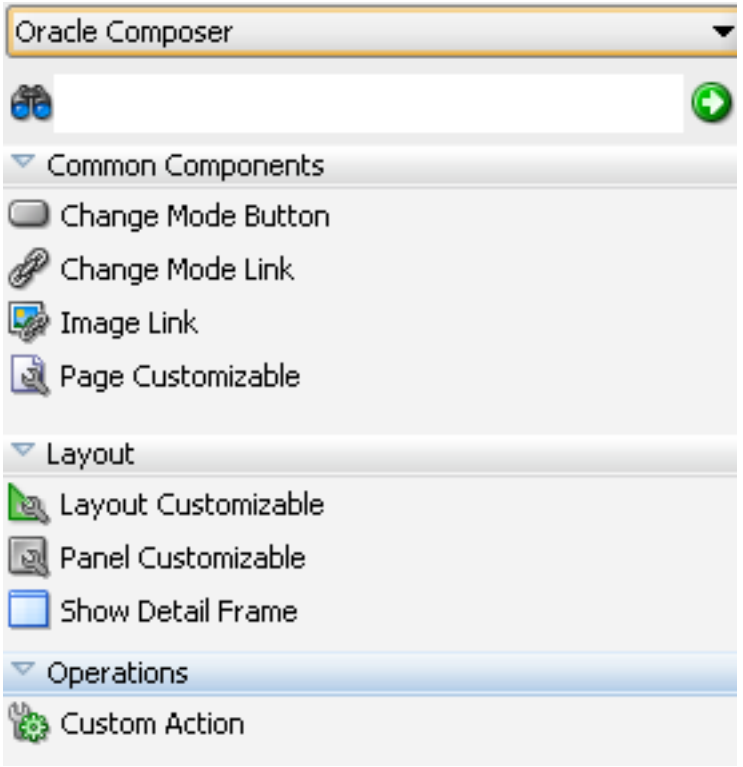
WEBCENTER COMPOSER

- **ADF Change Persistence has declarative support for only a limited set of UI characteristics**
- **For other 'customizable' aspects a lot of run time infrastructure needs to be developed**
- **Enter: WebCenter Composer**
  - Pre-built, fully integrated
    Run Time WYSIWYG Customization Editor
  - Turn on Edit Mode for page at run time – start making changes that are persisted in MDS



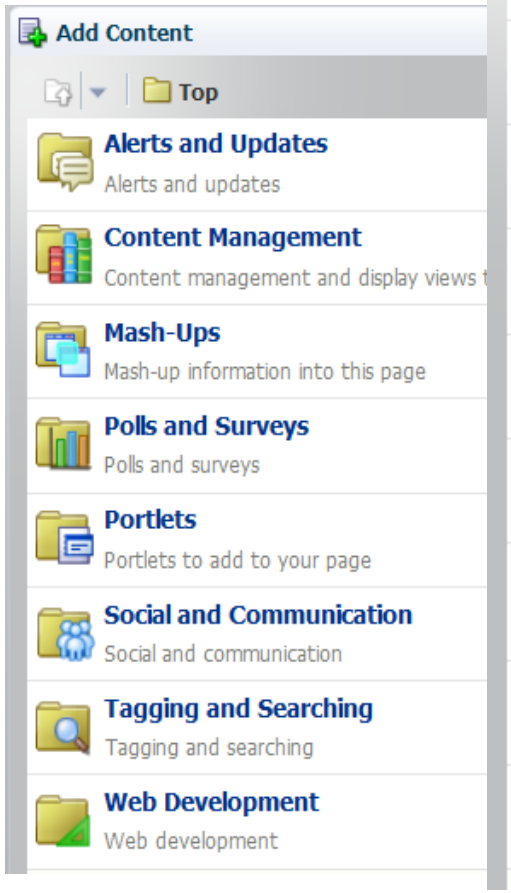
PAGE EDIT MODE

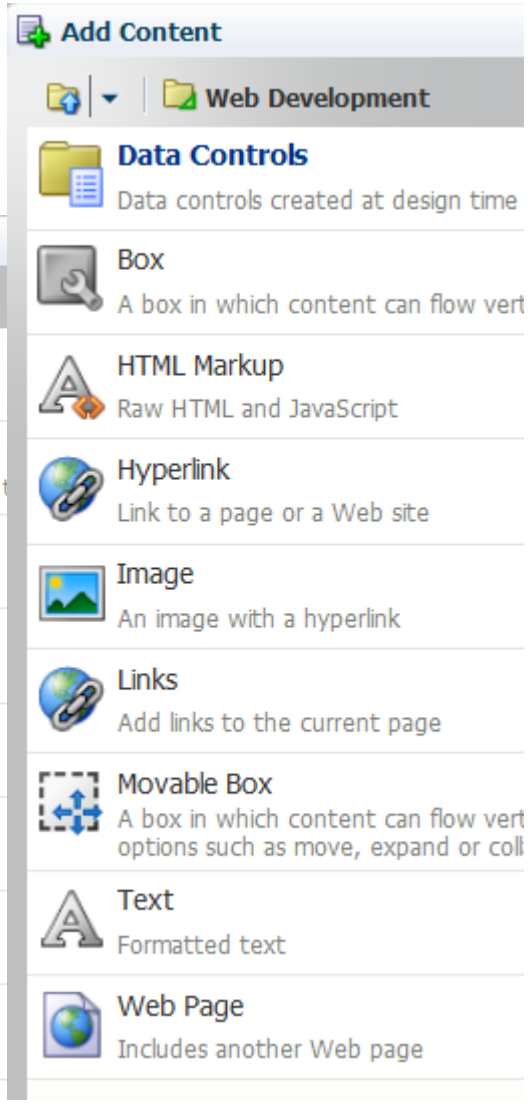
- **Pages created at design time can be made (partially) editable by including customizable components**
- **These components are used to turn on edit mode**
  - Only for authorized users
- **In edit mode:**
  - Customizable area can be rearranged
  - Components can be reconfigured (properties modified)
  - Components can be added
  - Page (Area) Layout can be changed
  - Resource Strings can be customized



WEBCENTER COMPOSER – RUN TIME RESOURCE CATALOG

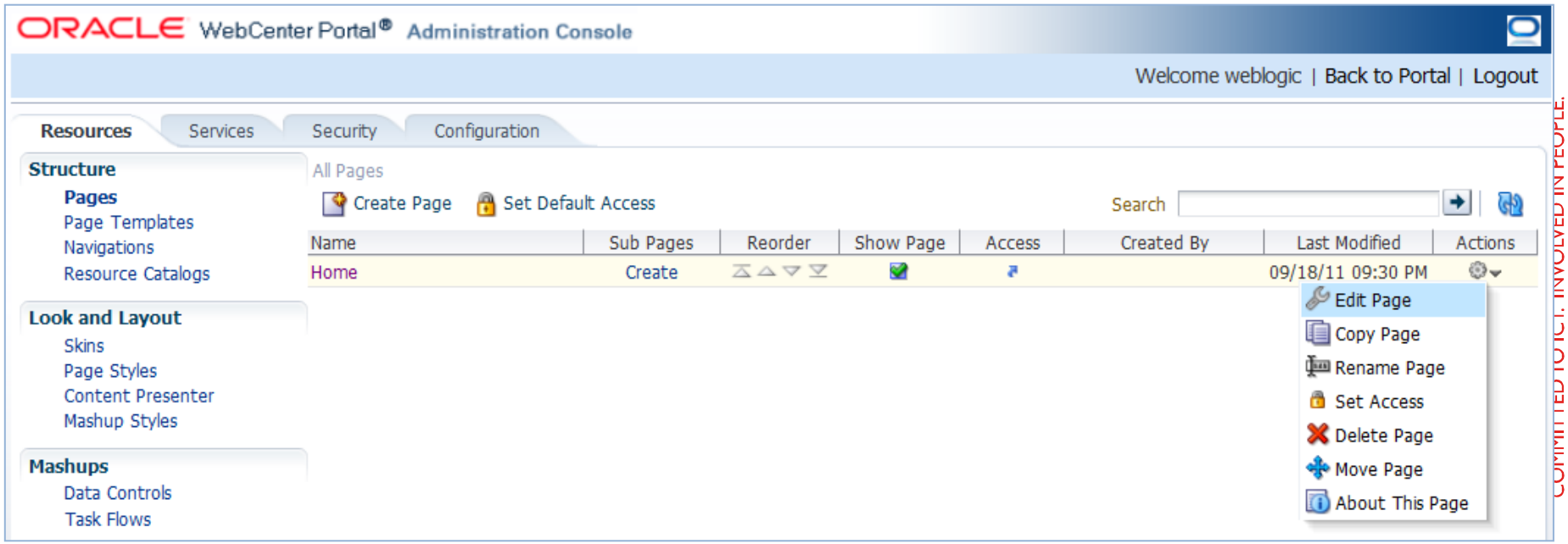
- **Resource Catalog contains components that can be added to pages at run time**
  - Standard UI widgets (box, image, rich text)
  - Portlets from Portlet Providers
  - Custom Task Flows
  - WebCenter Services
  - Content Presenter
  - Custom Components



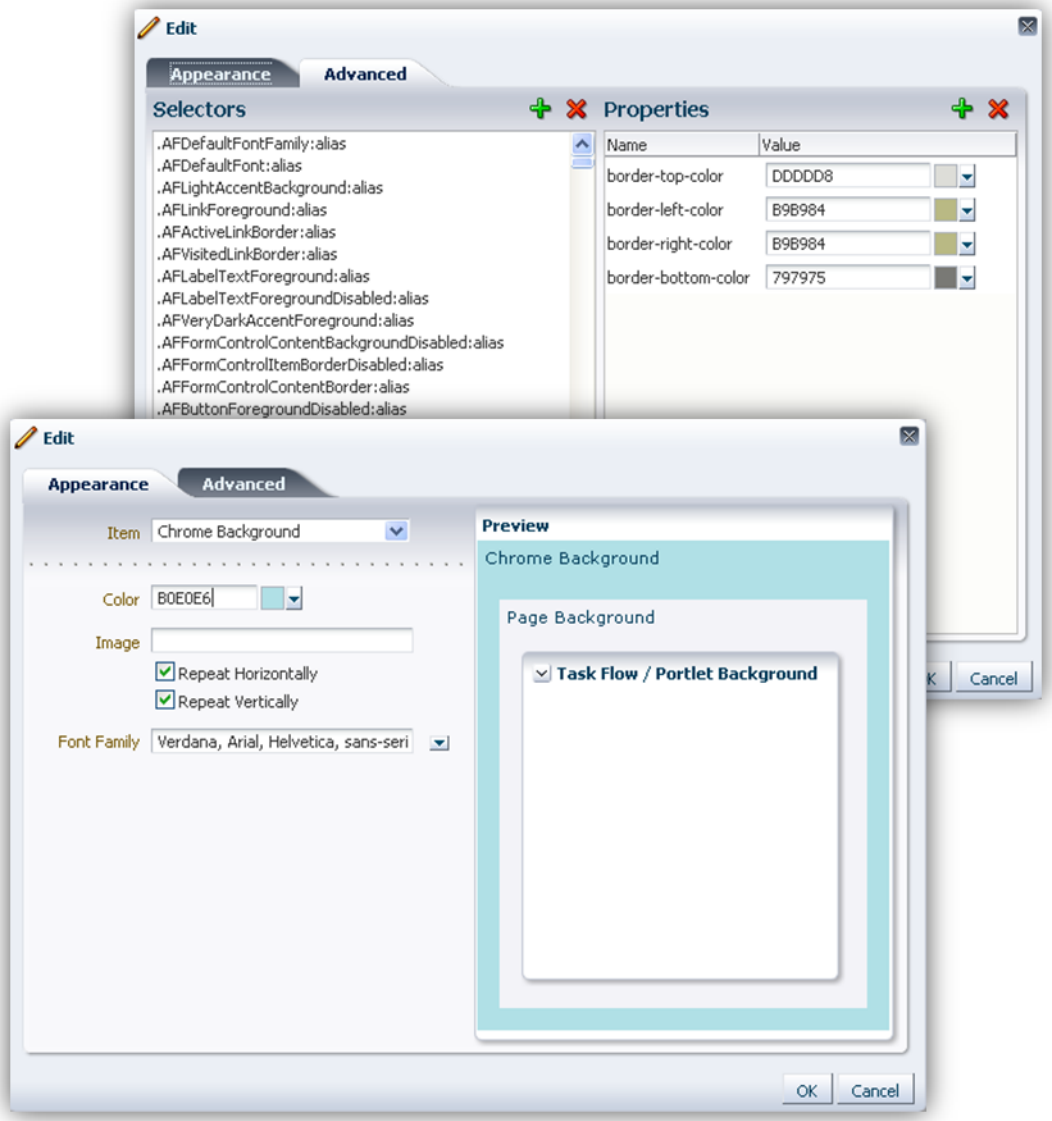


WEBCENTER COMPOSER
RUN TIME APPLICATION EDITING

- **WebCenter Composer provides run time support for**
  - Page creation, hide/show, removal
  - Page Access management
  - Navigation & Menu editing
  - Skin and Page Style & Template administration
  - Wire Portlets & Task Flows – parameters & events



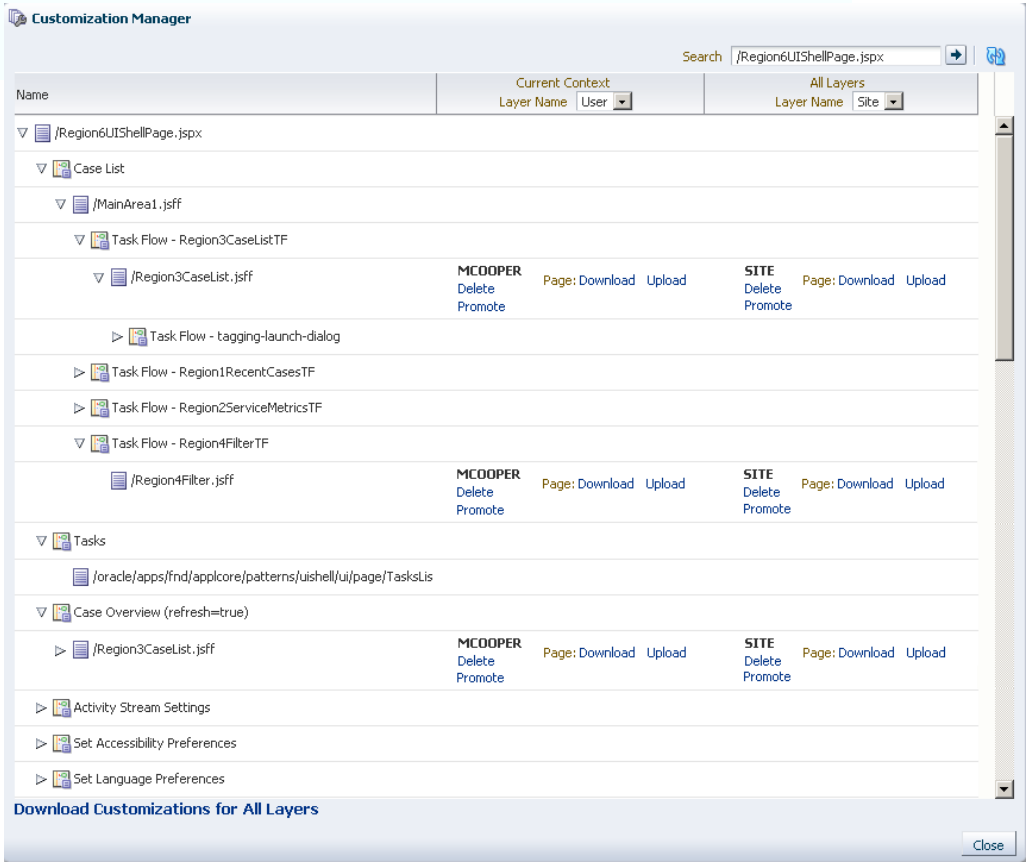
WEBCENTER RUN TIME SKIN EDITING



WEBCENTER CUSTOMIZATION MANAGER

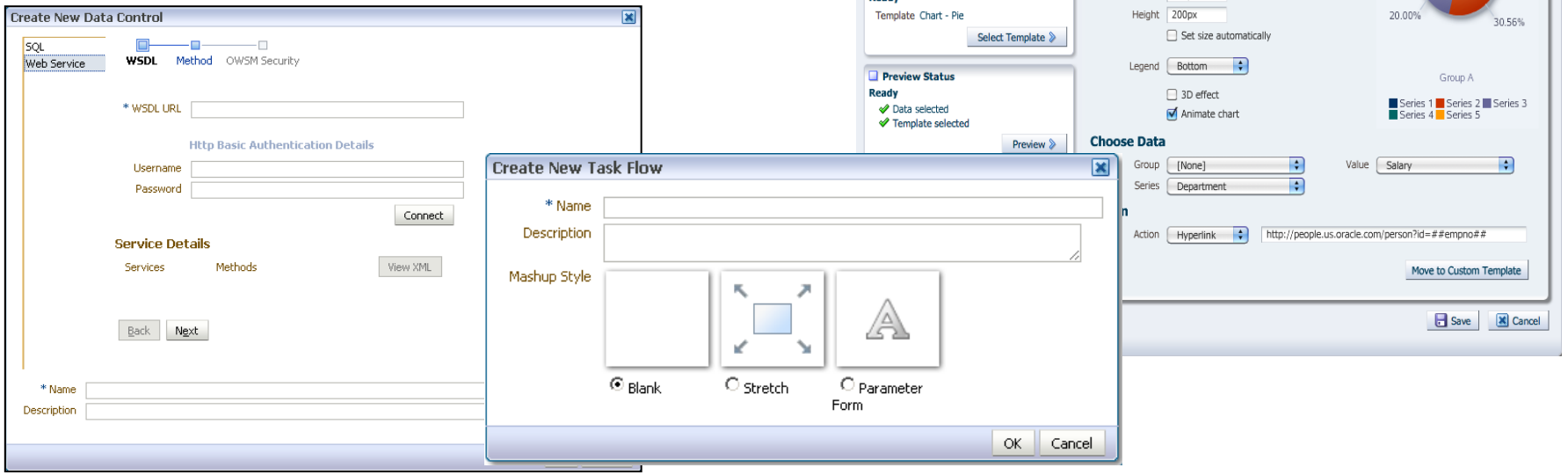
- **Customization Manager for run time administration of page, task flow and fragment customizations**

- **Support for**
  - Download
  - Upload
  - Promote
  - Remove



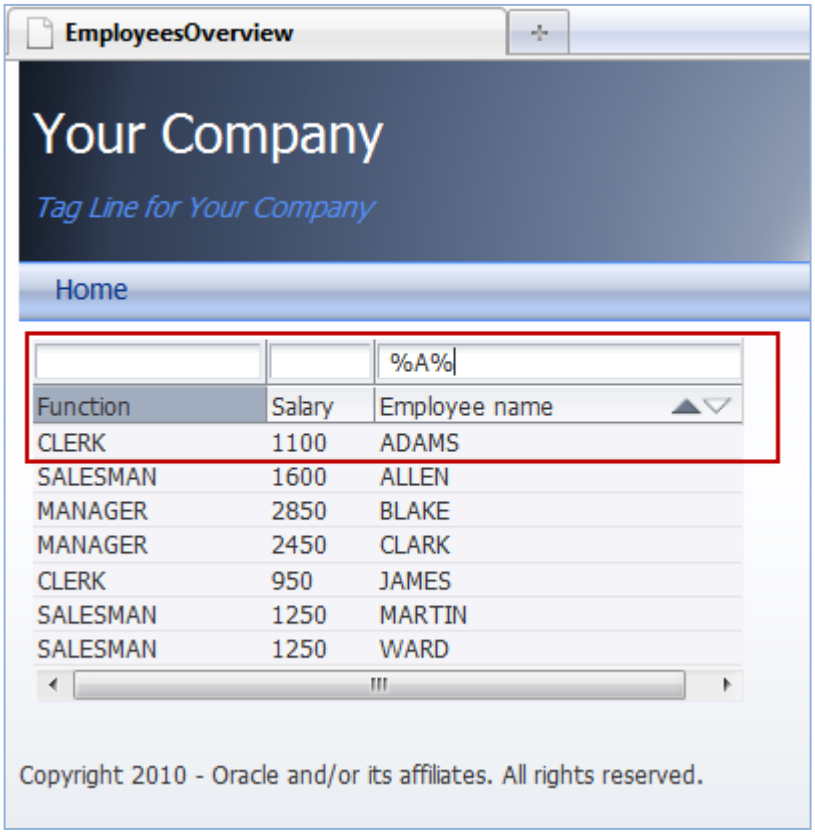
ORACLE BUSINESS MASHUP - RUN TIME APPLICATION EDITING WITH A VENGEANCE

- **DT@RT tools empowering Business Users to quickly create Enterprise Mashups / visualizations connecting data in ADF pages**
  - Runtime Create Data Controls
  - Runtime Create / Build Reusable Mashups / Task flows using Data Controls
  - Use them in pages and wire



ZERO CODING PAGE CREATED AT RUN TIME USING WEBCENTER COMPOSER

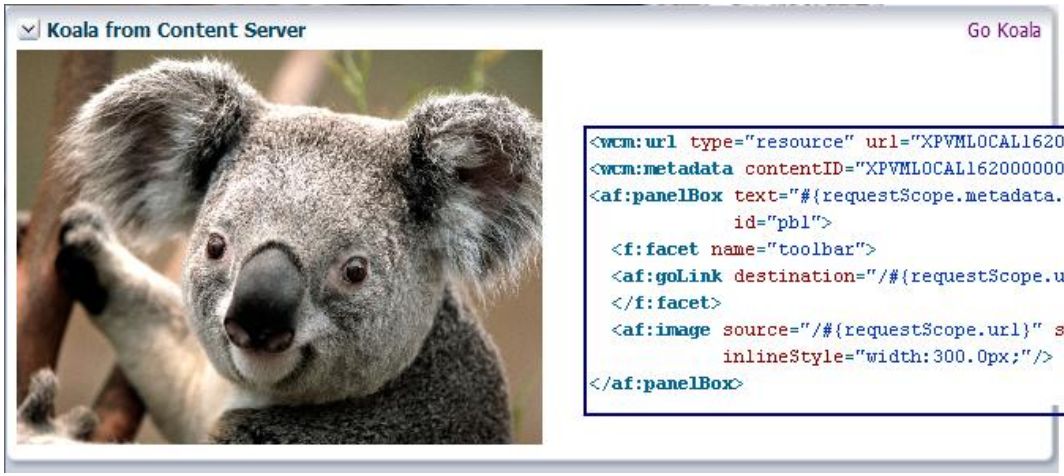
- **Create Data Control**
  - Using SQL query against JDBC Data Source
  - Or based on SOAP WebService
- **Set Data Control to *show***
- **Create new page**
- **Edit newly created page**
- **Add Data Control**
- **Configure Data Visualization for Data Control (table, graph)**



CONTENT INTEGRATION AND EDITING

- **Content can be integrated in in ADF applications**
  - Using custom mechanisms
  - Using WebCenter Content Services on top of an Content Repository (such as UCM)
  - Using OWCM tags selecting from UCM – and providing edit-content-in-place capabilities inside the web page
- **Content can be changed (at run time)**
  - Change content item - no change in application
  - (dynamically) select different content
- **Note: content is wide range**
  - From image and downloadable PDF to prompt and block headers

INTEGRATING UCM CONTENT ITEMS USING OWCM TAGS



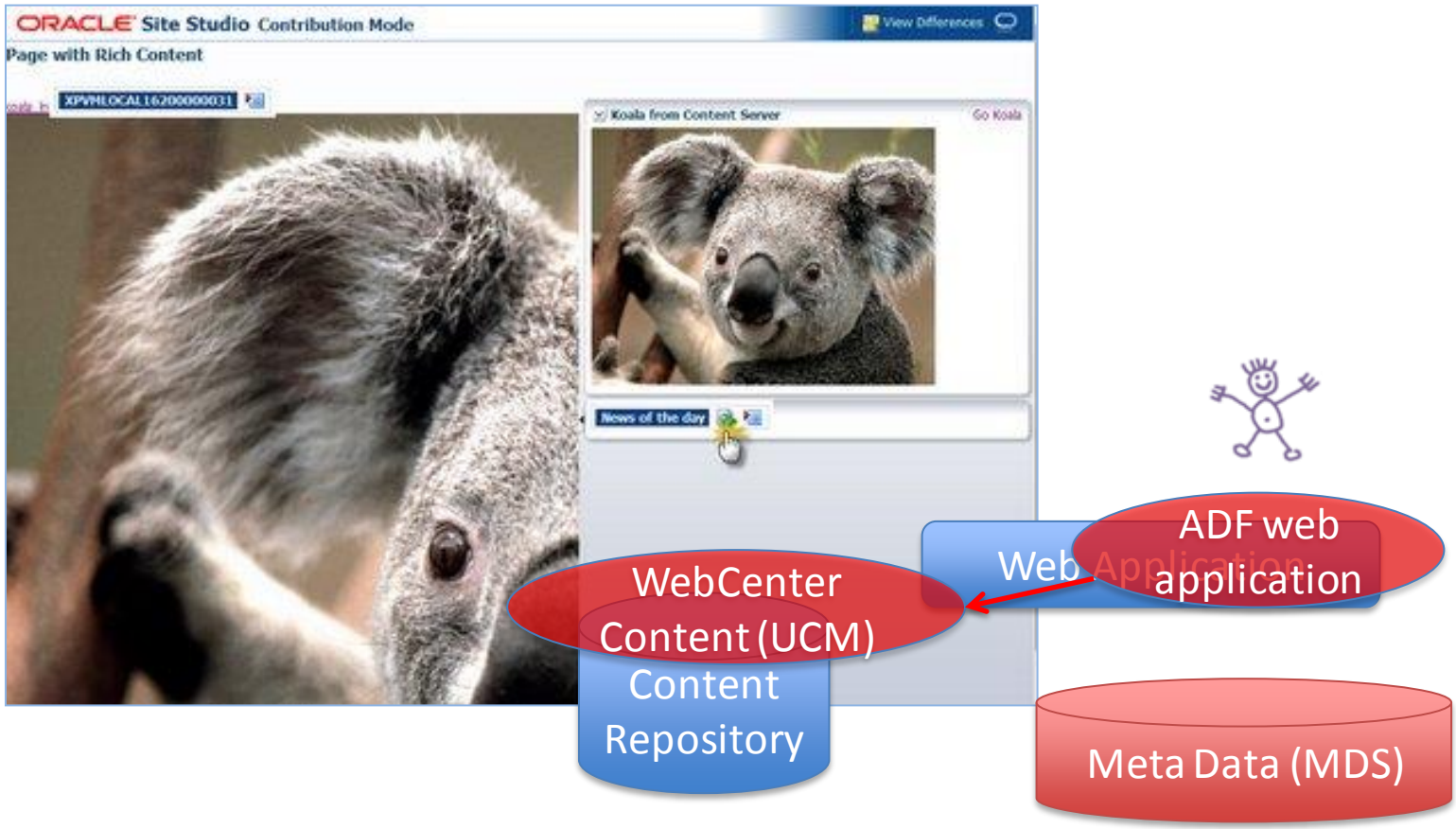

```
<wcm:url type="resource" url="XPVMLOCAL16200000031" var="url"/>
<wcm:metadata contentID="XPVMLOCAL16200000031" var="metadata"/>
<af:panelBox text="#{requestScope.metadata.resultSets.DOC_INFO.rows[0].dDocTitle} from Content Server"
             id="pbl">
  <f:facet name="toolbar">
  <af:goLink destination="/#{requestScope.url}" text="Go Koala" id="cll"/>
  </f:facet>
  <af:image source="/#{requestScope.url}" shortDesc="#{requestScope.metadata.resultSets.DOC_INFO.rows[0].dDo
           inlineStyle="width:300.0px;"/>
</af:panelBox>
```

- **Image in UCM Content Server – embedded in web page**
- **Placeholder Tag can be added to provide a spot for dynamic content to be added to the page:**

```
<af:panelBox id="pb3" text="News Flash" >
<wcm:placeholder actions="EPRISUTMN" name="News of the day"/>
</af:panelBox>
```

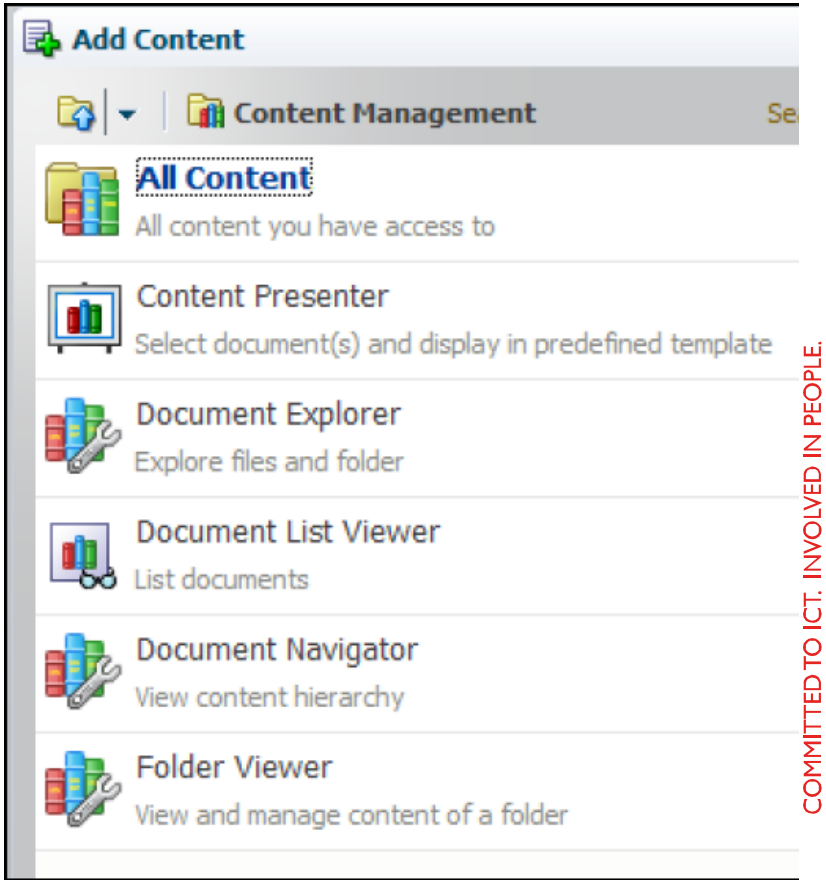
INTEGRATING UCM CONTENT USING OWCM TAGS

- **CTRL+F5 brings page in Content Contributor mode – allowing all Content Items to be edited**
  - Any changes are recorded in the Content Server

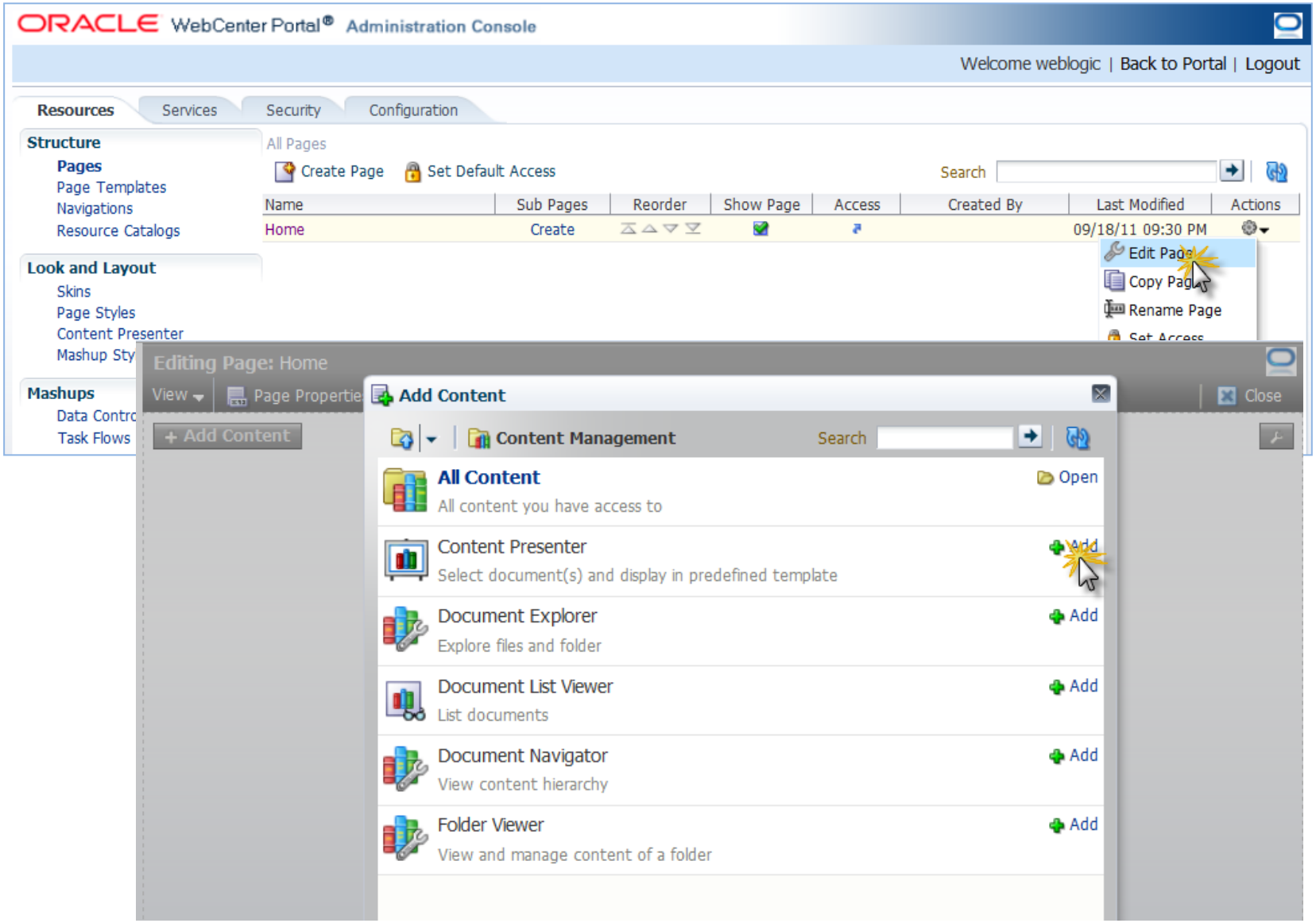


INTEGRATING CONTENT USING WEBCENTER CONTENT SERVICES

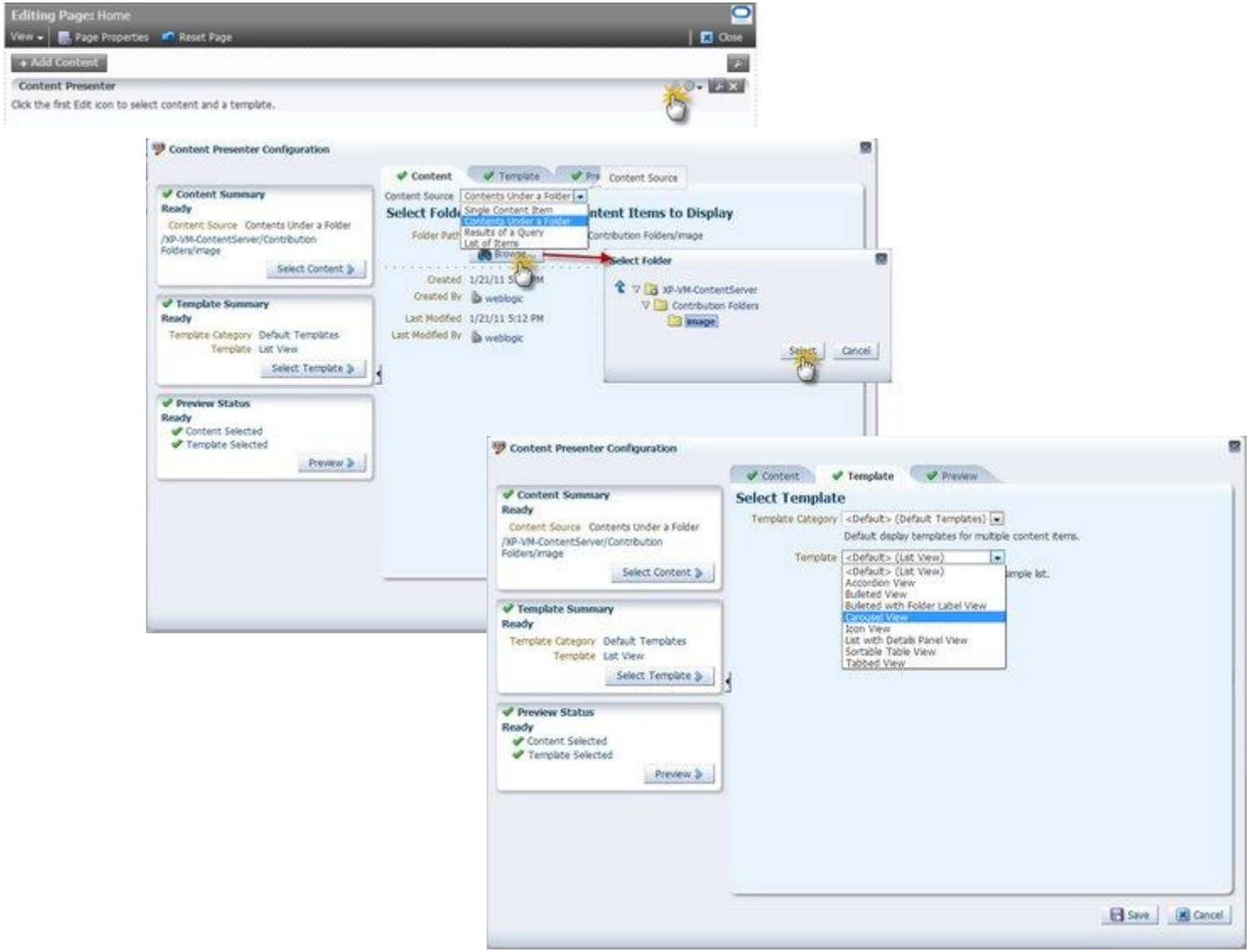
- **When a connection with a Content Repository has been configured**
  - Content Server (UCM), Oracle Portal, File System
- **WC Content components can added to a page**
  - At design time or run time (composer)
- **to present content items**
  - In various display styles
  - Using static content item selection or dynamically evaluated content query



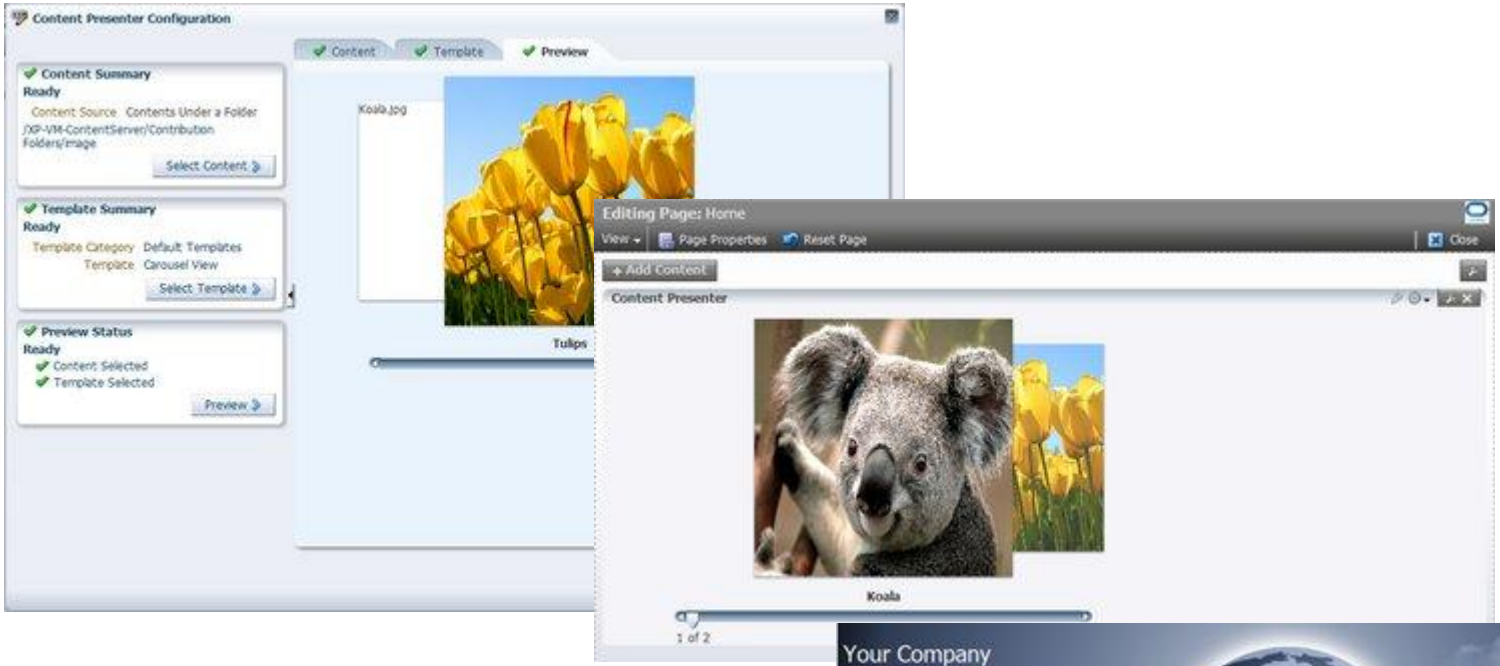
RUN TIME ADDING CONTENT TO WEB APPLICATION

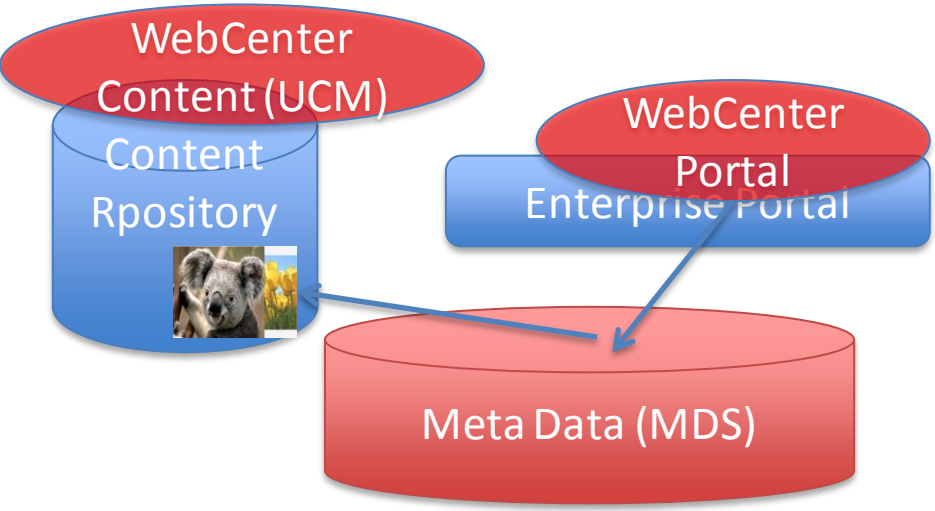


RUN TIME ADDING CONTENT TO WEB APPLICATION (2)



RUN TIME ADDING CONTENT TO WEB APPLICATION (3)





RUN TIME CONTENT EDITING – THROUGH UCM – IMPACTS WEB APPLICATION

WebCenter Content (UCM)

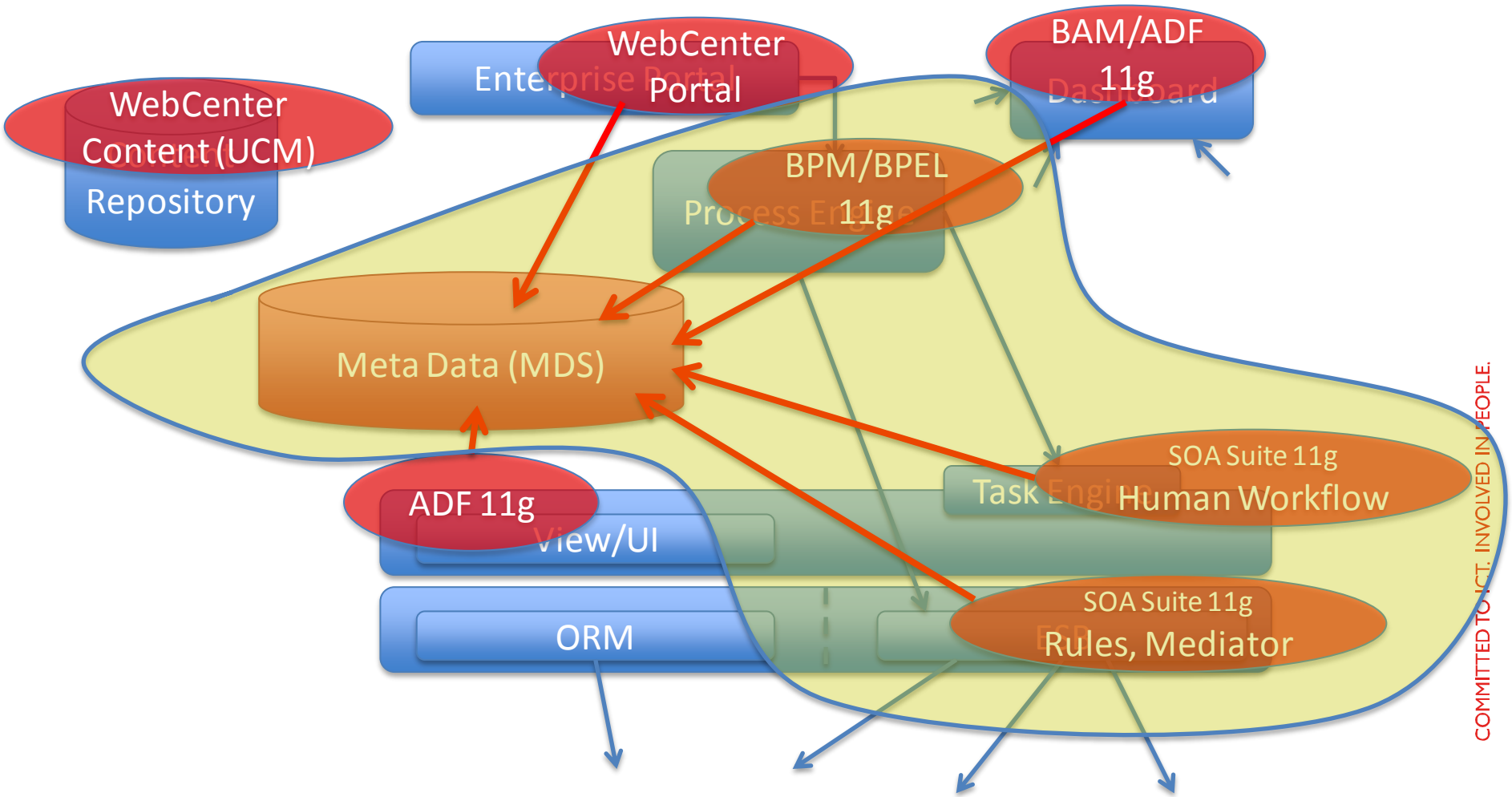
Content Rpository

WebCenter Portal

Enterprise Portal

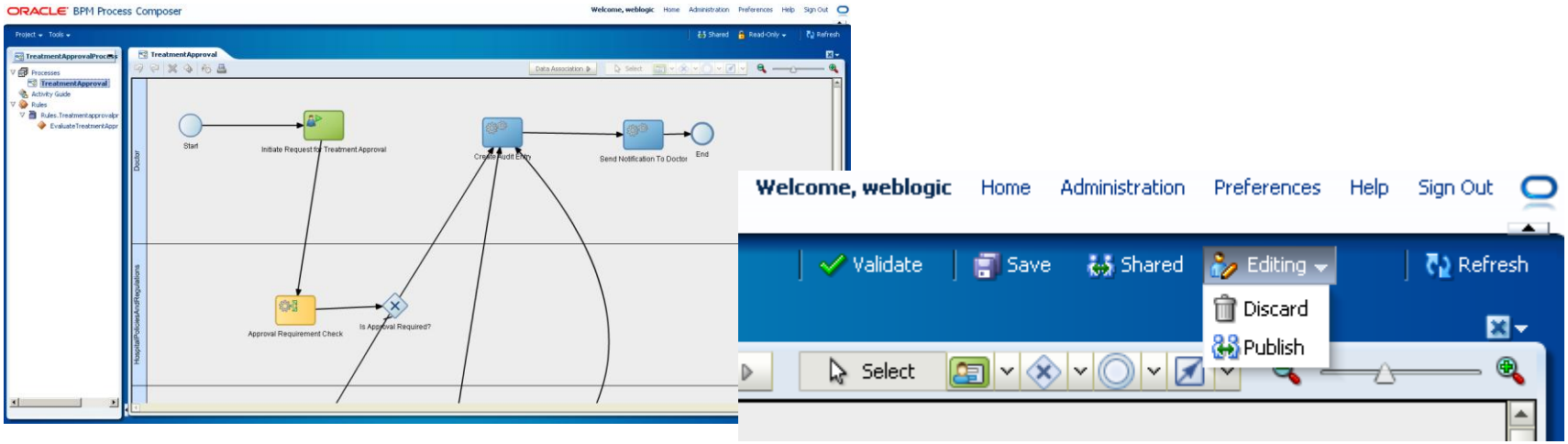
Meta Data (MDS)

RUN TIME META DATA IN ORACLE FUSION MIDDLEWARE

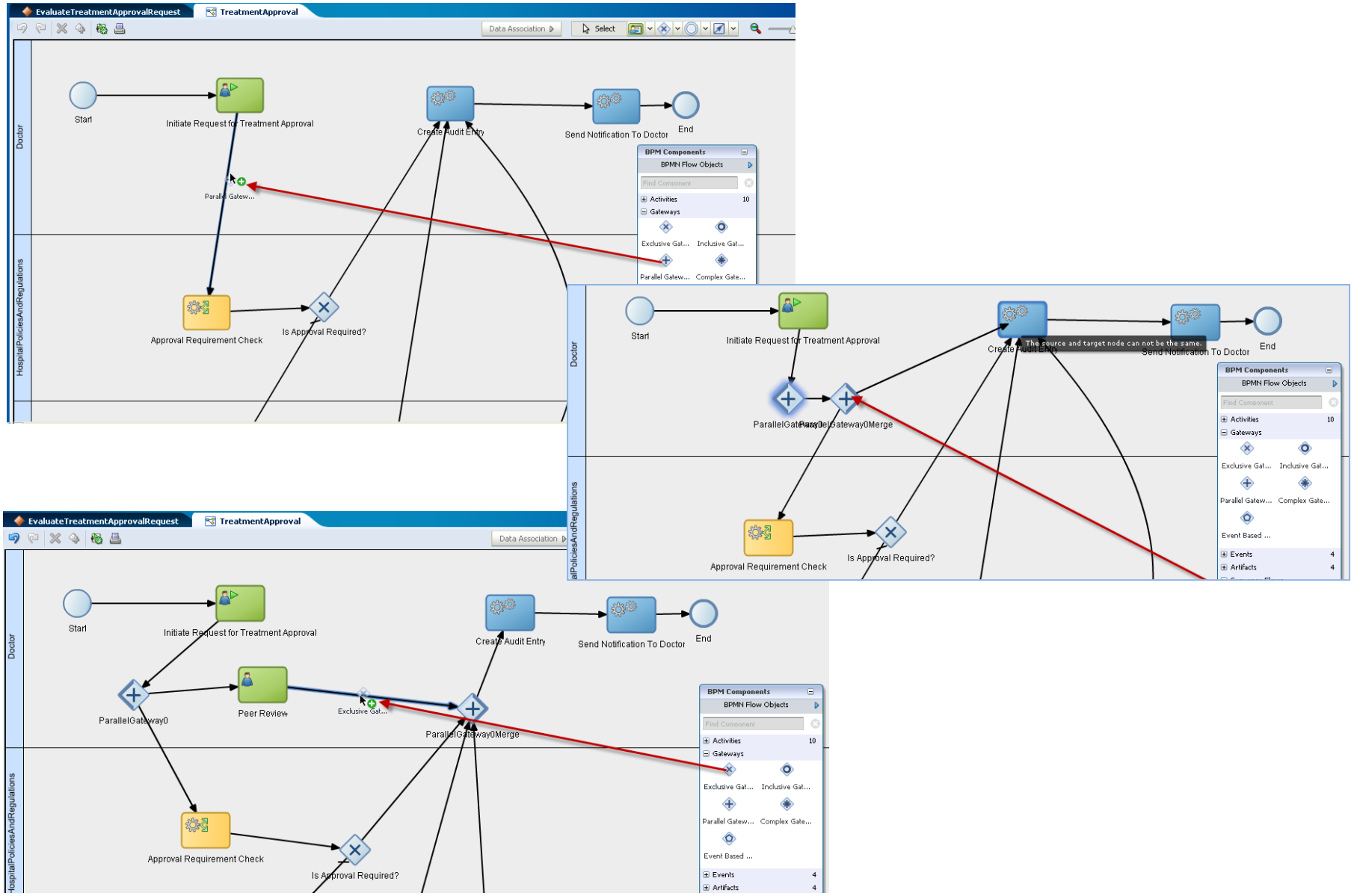


DESIGN TIME @ RUN TIME
BPM PROCESS COMPOSER

- **Browser based User Interface that exposes process models for reviewing as well as editing**
  - Interacts with BPM Studio (JDeveloper) via MDS
  - For example analysts and business users creating the abstract process model and developers implementing it
  - Can also deploy revised BPM models to the run-time
  - Similar to SOA Composer, yet targeted at analysts
- **http://host:port/bpm/composer**

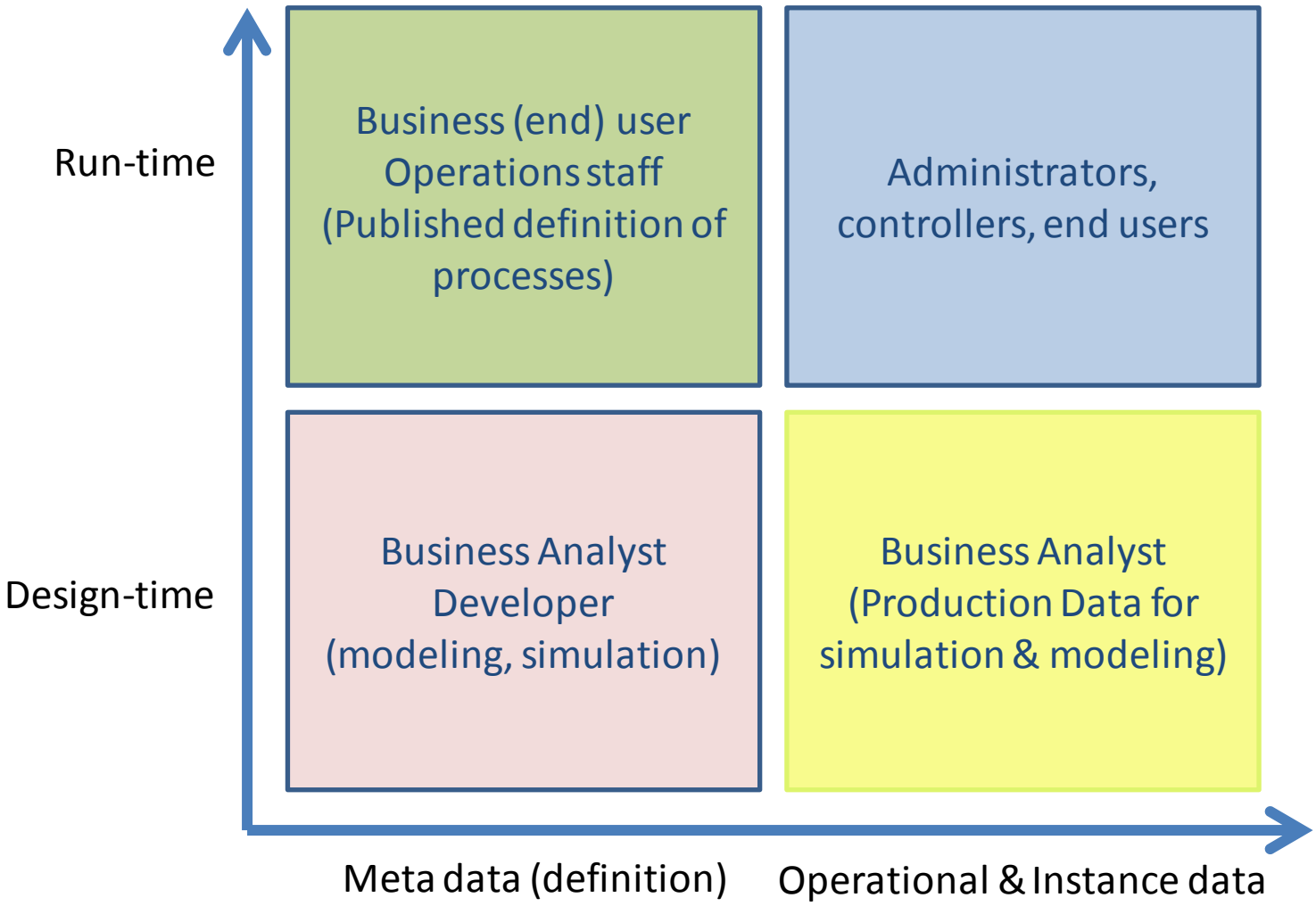


ON LINE PROCESS REVISION

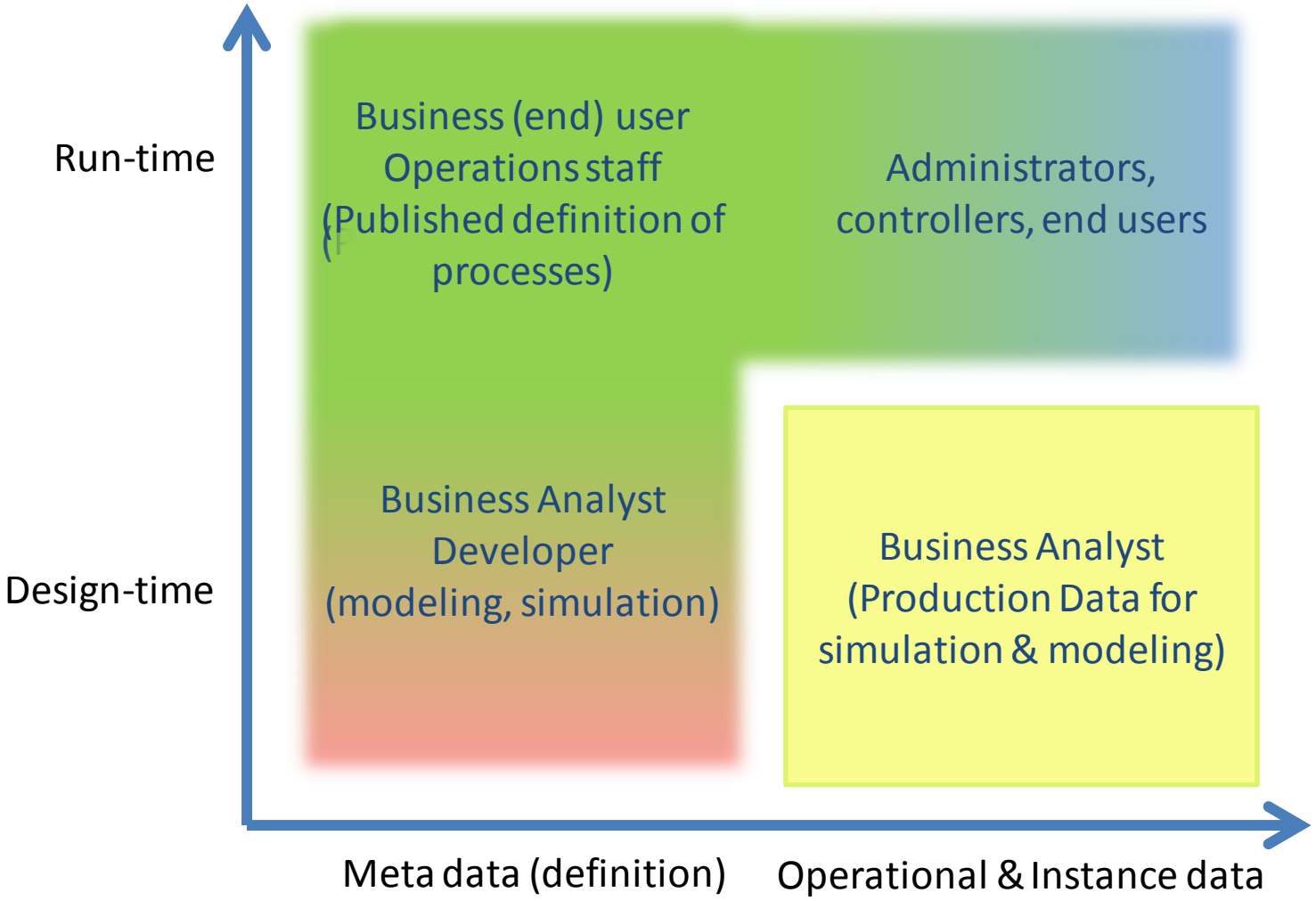


COMMITTED TO ICT. INVOLVED IN PEOPLE.

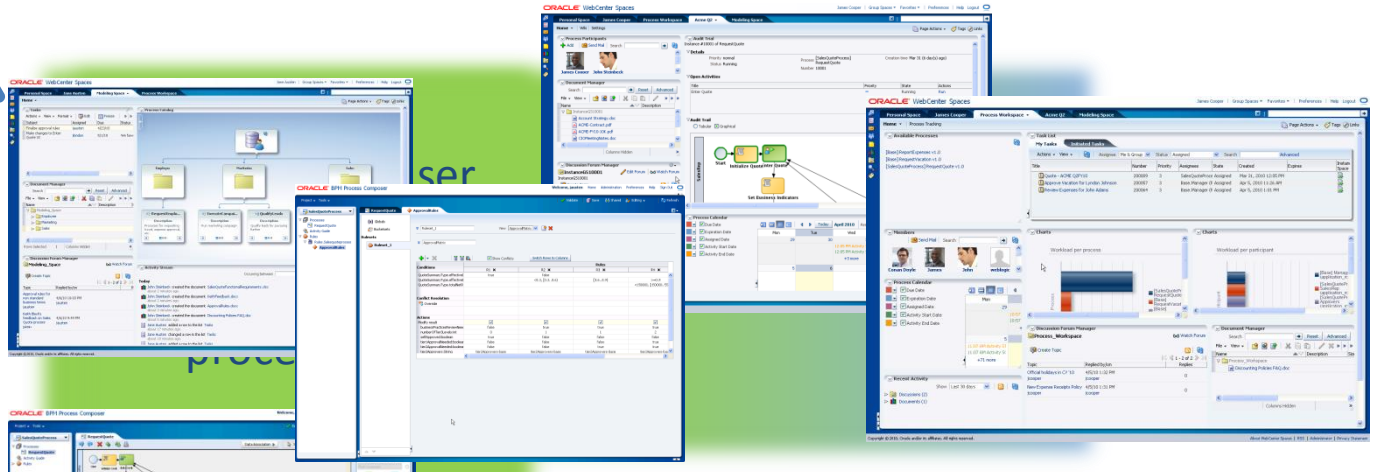
TIME, LEVEL AND STAKEHOLDERS



ORACLE BPM 11G BLURS THE LINES



ORACLE BPM 11G BLURS THE LINES



Run-time

Design-time

(modeling, simulation)

Business Analyst
(Production Data for simulation & modeling)

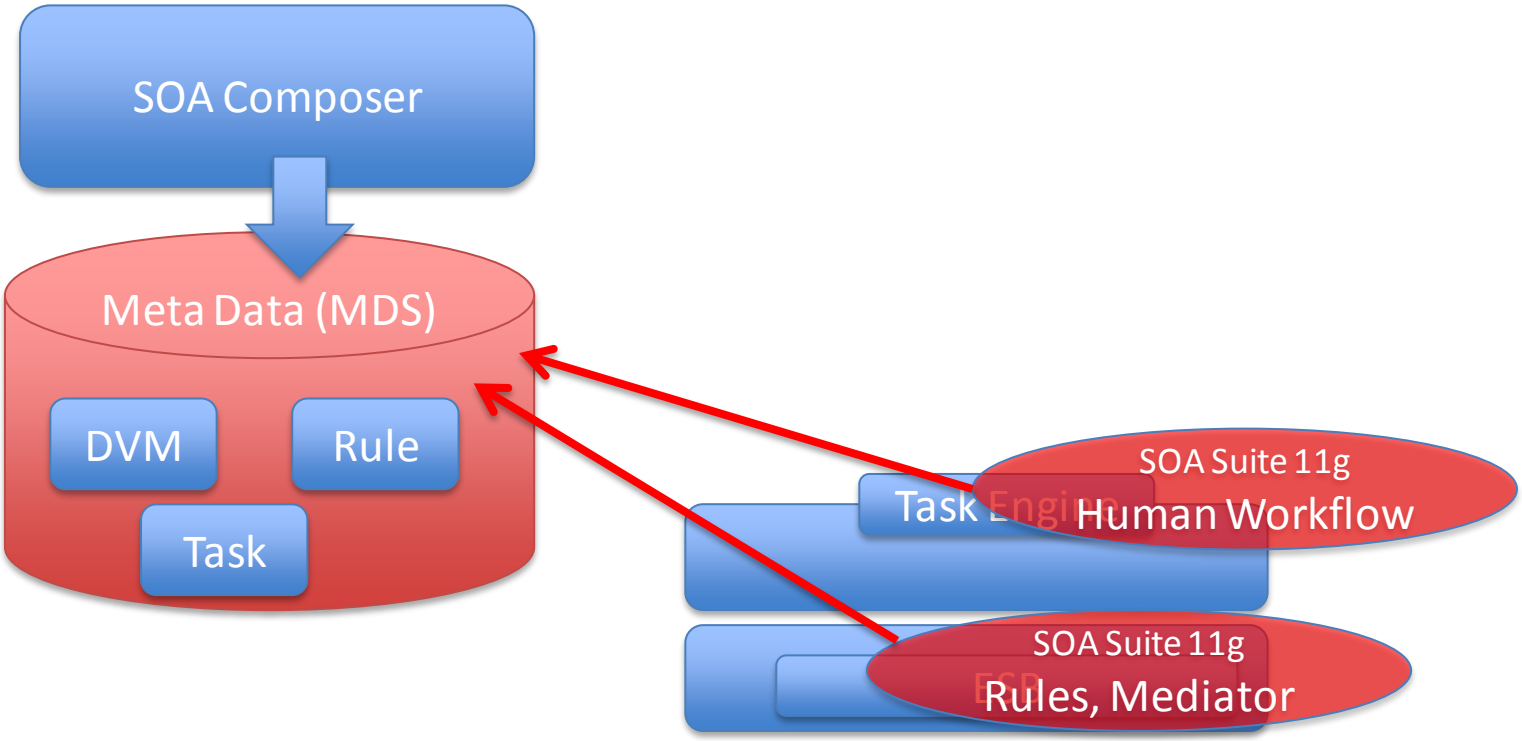
Meta data (definition)

Operational & Instance data

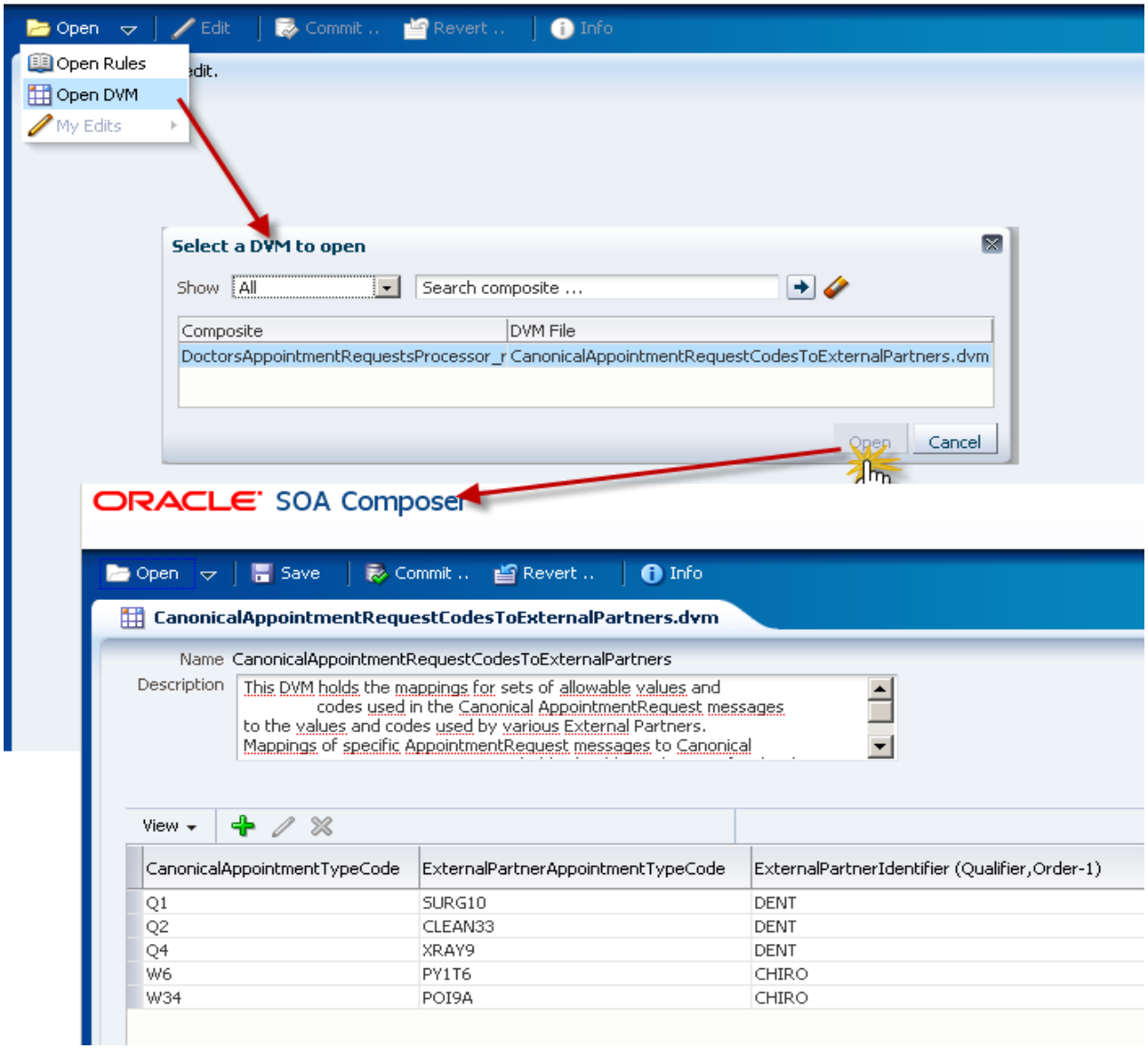
COMMITTED TO ICT. INVOLVED IN PEOPLE.

SOA COMPOSER RUN TIME META DATA IN ORACLE FUSION MIDDLEWARE

- **SOA Suite 11g stores Business Rule, Domain Value Map and Task Definition in MDS**
- **Live editing (runtime environment) through SOA Composer**

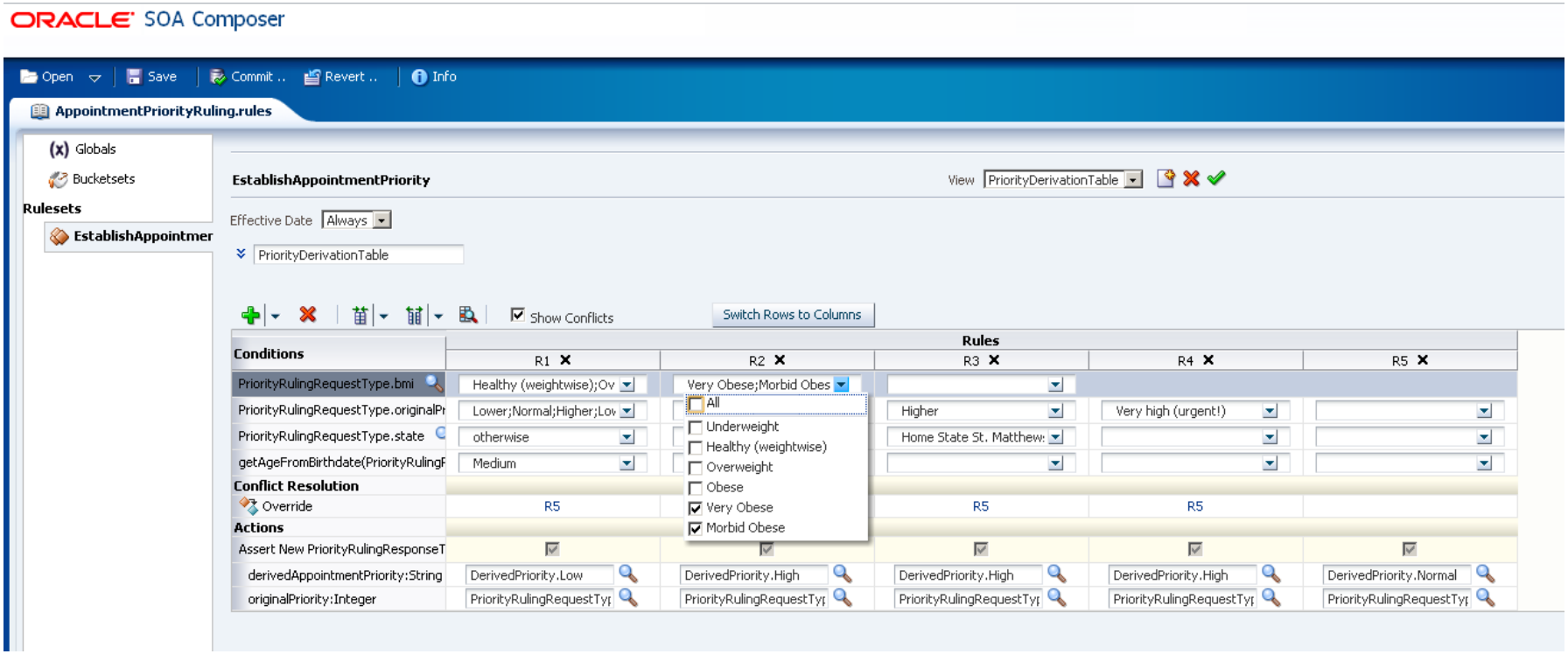


SOA COMPOSER AND DOMAIN VALUE MAP



EDITING DECISION TABLE BUSINESS RULE IN SOA COMPOSER

- **The Decision Table can be reviewed and edited at run time in SOA Composer**
  - Note: bucketsets can be managed too
- **Note: Commit means 'publish to live environment'**



INSTRUCTIONS FOR DEVELOPER TO ENABLE DESIGN @ RUN

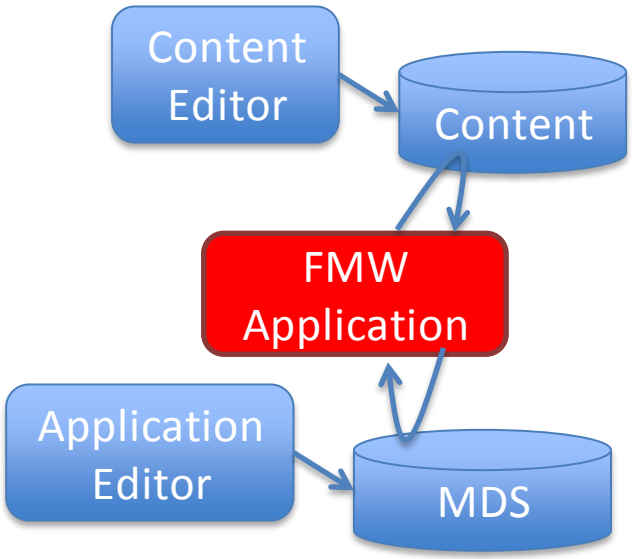
- **Configure ADF Customization in applications**
  - Including appropriate customization layers
- **Add Panel Customizable to page**
- **Configure Edit (Page) privilege**
- **Use Business Rules for calculations, validations, decisions**
- **Use Domain Value Maps for conversions and system parameters**
- **Leverage WebCenter Services**
- **Publish Events**
  - ADF contextual events from View
  - EDN Business Events from BPM, BPEL, ADF BC, …
- **Externalize constants, content and dependencies**
  - to central parameter store, CMS and injection mechanisms

NEW ROLE/PROCESS

- **Who does run time editing? What is the new role called?**
  - Application Editor? Run-time Composer?
  - (what is an *application* in today's IT landscape?)
- **Approval process (from editing to live)**
- **Administration of MDS Repository**
- **Test procedures for live application changes**
- **Distribute customizations**
- **Have customizations survive application upgrades**
- **Feed back run time customizations to development?**
- **How are 'running instances' and 'on line users' impacted by new customizations?**

SUMMARY

- **A higher degree of agility – and an application that more accurately and rapidly follows business needs**
- **Can be achieved through 'design time at run time':**
  - application editing in (or near) the run time environment
  - Very similar to editing of content items
- **Oracle Fusion Middleware provides an infrastructure for design time at run time**
  - MDS Repository
  - Content Server (UCM)
  - ADF Change Persistence and Customization
  - WebCenter Composer
  - SOA Composer
  - BPM Composer



SUMMARY (2)

- **For design time at run time to be effectively used, the applications and processes need to be prepared**
- **The actual 'live application editing' is a very important task – the pivotal action for instant agility**
  - Application editing requires very good understanding of the business requirements
  - Application editing is still a fairly *technical* operation that requires deep insight in application and tools
- **The process for making changes, testing, authorizing and publishing them needs to be defined**
- **The feedback loop to development needs to be established**
  - To perhaps add a more advanced implementation of the new requirement in the base product
  - To at least not interfere with the run time additions

APPLICATION SETTINGS FOR CUSTOMIZING APPLICATION BEHAVIOR

- **Some Application Behavior is developed as *dynamically configurable***
  - Through Preferences or Application Settings
  - *Display*: Locale, Format, Units for Height and Weight (converter parameters), feature on/off
  - *Business logic*: Default Values for new records, Validation parameters, Field & Button Authorization
  - *Styling*: Images, Fonts, Colors, Skins/Themes
- **Settings can be configured at Factory (default), Organization and User level**
  - At run-time through self service pages

INTEGRATING APPLICATION SETTINGS DURING DEVELOPMENT

APPLICATION SETTINGS DATA MODEL