WellSTAR (RBDMS 3.0) Architecture General Design Overview

11/15/2016
Agenda

• WellSTAR (RBDMS 3.0) Solution Guiding Principals
• Architecture General Design Overview
• Architecture General Design Details
Guiding Principals

• STAY TRUE TO THE RBDMS FUNCTIONALITY AND UNDERLYING DATA MODEL

• MODERNIZE THE SYSTEM TO SUPPORT USER ACCESS FROM CROSS PLATFORM, WEB-BROWSER ENABLED DEVICES, AND SYSTEM ACCESS/INTEGRATION VIA WEB SERVICES

• SUPPORT OFFLINE/DISCONNECTED SYSTEM FUNCTIONALITY AND DATA SYNCHRONIZATION

• MAKE THE SYSTEM HIGHLY EXTENSIBLE THROUGH THE IMPLEMENTATION OF DYNAMIC/CONFIGURABLE USER INTERFACE COMPONENTS, BUSINESS RULES, AND BUSINESS PROCESSES

• IMPLEMENT FEATURES AND CAPABILITIES SPECIFIED BY DOC AND DOGGR WHILE AVOIDING UNNECESSARY COST DRIVERS AND TECHNOLOGY OBSOLESCENCE
Guiding Principals

• STAY TRUE TO THE RBDMS FUNCTIONALITY AND UNDERLYING DATA MODE
Guiding Principals

- MODERNIZE THE SYSTEM TO SUPPORT USER ACCESS FROM CROSS PlattFORM, WEB-BROWSER ENABLED DEVICES, AND SYSTEM ACCESS/INTEGRATION VIA WEB SERVICES
Guiding Principals

- SUPPORT OFFLINE/DISCONNECTED SYSTEM FUNCTIONALITY AND DATA SYNCHRONIZATION

1. Download data from online RBDMS repository for offline work.
2. Do work offline using client version of RBDMS.
3. Synchronize offline data with online RBDMS repository.
Guiding Principals

• MAKE THE SYSTEM HIGHLY EXTENSIBLE THROUGH THE IMPLEMENTATION OF DYNAMIC/CONFIGURABLE USER INTERFACE COMPONENTS, BUSINESS RULES, AND BUSINESS PROCESSES
Guiding Principals

- IMPLEMENT FEATURES AND CAPABILITIES SPECIFIED BY DOC AND DOGGR WHILE AVOIDING UNNECESSARY COST DRIVERS AND TECHNOLOGY OBSOLESCENCE
General Design Overview

Key Considerations

- Proven n-Tier Design Pattern
- Service Oriented Architecture
- Flexibility and Configurability
- Security, Reliability, and Performance
Proven n-Tier Design Pattern - Presentation Tier

Presentation Tier

Online Forms Portal

Agency Portal

G2G/B2G

Service API

Business Tier

Business Components & Batch

REST WebAPI Controller(s)

Services & Integration

Data Tier

Data Access Components

Repositories (SQL Server)

Utilities

Return Data Flow
Proven n-Tier Design Pattern - Business Tier
Proven n-Tier Design Pattern - Data Tier

Presentation Tier
- Online Forms Portal
- Agency Portal
- Service API

Business Tier
- REST WebAPI Controller(s)
- Business Components & Batch
- Services & Integration
- Utilities

Data Tier
- Entity Framework
- Data Transfer Object (DTO)
- WellSTAR SQL DB
- Repositories (SQL Server)

Return Data Flow
Service Oriented Architecture (SOA)
SOA – Workflow Service

- Business Components
- Other Services
- Utilities
- Batch Processes

Invoke Workflow

Workflow Service

REST Web API
(local data API support)

Invoke Task

- Business Components
- Other Services

Retrieve workflow definition, workflow instance, task, and status information

Save workflow instance, task, and status information

Workflow Service Repository
SOA – Business Rules Service

- Business Components
- Other Services
- Utilities
- Batch Processes

Business Rules Service

- Retrieve Rule Definition
- Save Rule Metrics

Business Rules Service Repository

Invoke Rule

Return Result
SOA – Notification Service

- Business Components
- Other Services
- Utilities
- Batch Processes

Submit request(s)

Notification Service

Send to recipient(s)

Retrive notification templates, user/system preferences, etc.

Save notifications & correspondence for manual processing

Notification Service Repository
SOA – Integration Service (a.k.a. Service Bus)

Source
WellSTAR business components, services, utilities & batch processes

Post Message
(sync or async)

Return Result
(synchronous)

Integration Service
(a.k.a. Service Bus)

Retrieve Message Profile for Processing Instructions

Integrate Service Repository

Save Asynchronous Message Posts & Update Status/Results

Destination
WellSTAR or other systems in or out of DOC/DOGGR network

Transmit Message
<table>
<thead>
<tr>
<th>Presentation</th>
<th>Business</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Page Labels, Text, Pre-Defined Content</td>
<td>Rules</td>
<td>Repository Type</td>
</tr>
<tr>
<td>Web Page Logo, Fonts, Colors, &amp; other Style Properties</td>
<td>Workflow</td>
<td>Configuration Settings</td>
</tr>
<tr>
<td>Form Validation &amp; Display Formatting</td>
<td>PDF Form Rendering</td>
<td></td>
</tr>
<tr>
<td>Form Sections &amp; Display Order</td>
<td>Configuration Settings</td>
<td></td>
</tr>
<tr>
<td>Configuration Settings</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Security, Reliability, and Performance

<table>
<thead>
<tr>
<th>Security</th>
<th>Reliability</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Directory Authentication</td>
<td>Microsoft Framework &amp; Technology Stack</td>
<td>Application Caching</td>
</tr>
<tr>
<td>Claims Based Authorization</td>
<td>Modular, Loosely Coupled Application Tier &amp; SOA Design</td>
<td>Database Caching</td>
</tr>
<tr>
<td>Hardware Security – Firewalls &amp; Routers</td>
<td>RBDMS Core Framework Abstraction</td>
<td>SQL Query Optimization</td>
</tr>
<tr>
<td>Platform Security – Active Directory Groups, Policies, Service Accounts, &amp; Audit Profiles</td>
<td>Automated Unit Tests</td>
<td>Dedicated Reporting Database</td>
</tr>
<tr>
<td>Logging &amp; Monitoring</td>
<td>Exception Handling</td>
<td>Microsoft Framework &amp; Technology Stack</td>
</tr>
<tr>
<td></td>
<td>Logging &amp; Monitoring</td>
<td>Logging &amp; Monitoring</td>
</tr>
</tbody>
</table>