GCSS-AF Data Services

+ RIA Reporting Framework Overview

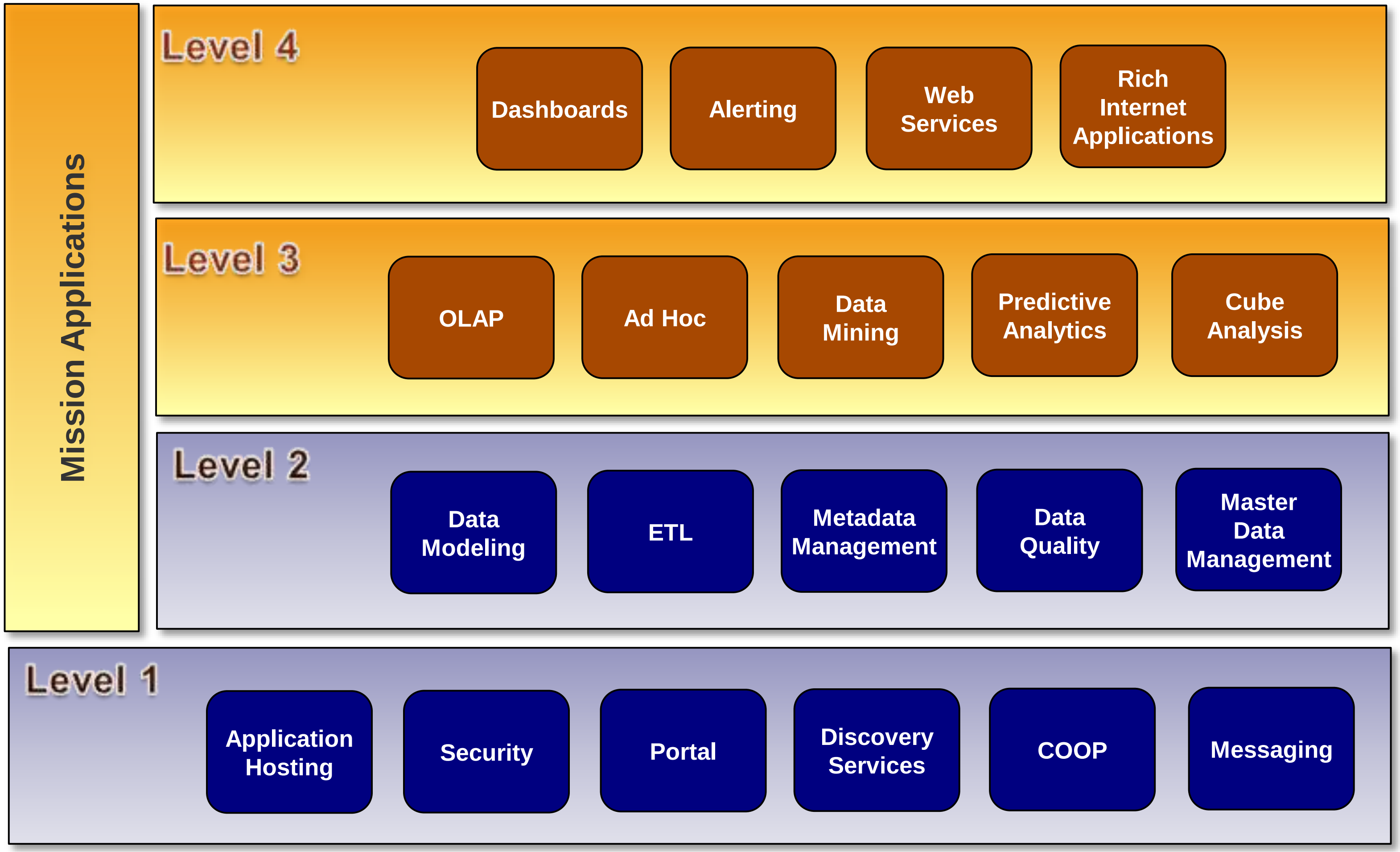
May 21, 2009

+ ROUNDARCH

Agenda

2:00	**GCSS-AF Data Services Introduction & A4 LIMS-EV Overview**
2:05	**LIMS-EV Demo**
2:15	**Architecture Discussion**
2:20	Dashboards on Demand (Mr. Jim Stogdill, Accenture)
2:40	Q&A

+ GCSS-AF Data Services



Mission Applications

Level 4

- Dashboards
- Alerting
- Web Services
- Rich Internet Applications

Level 3

- OLAP
- Ad Hoc
- Data Mining
- Predictive Analytics
- Cube Analysis

Level 2

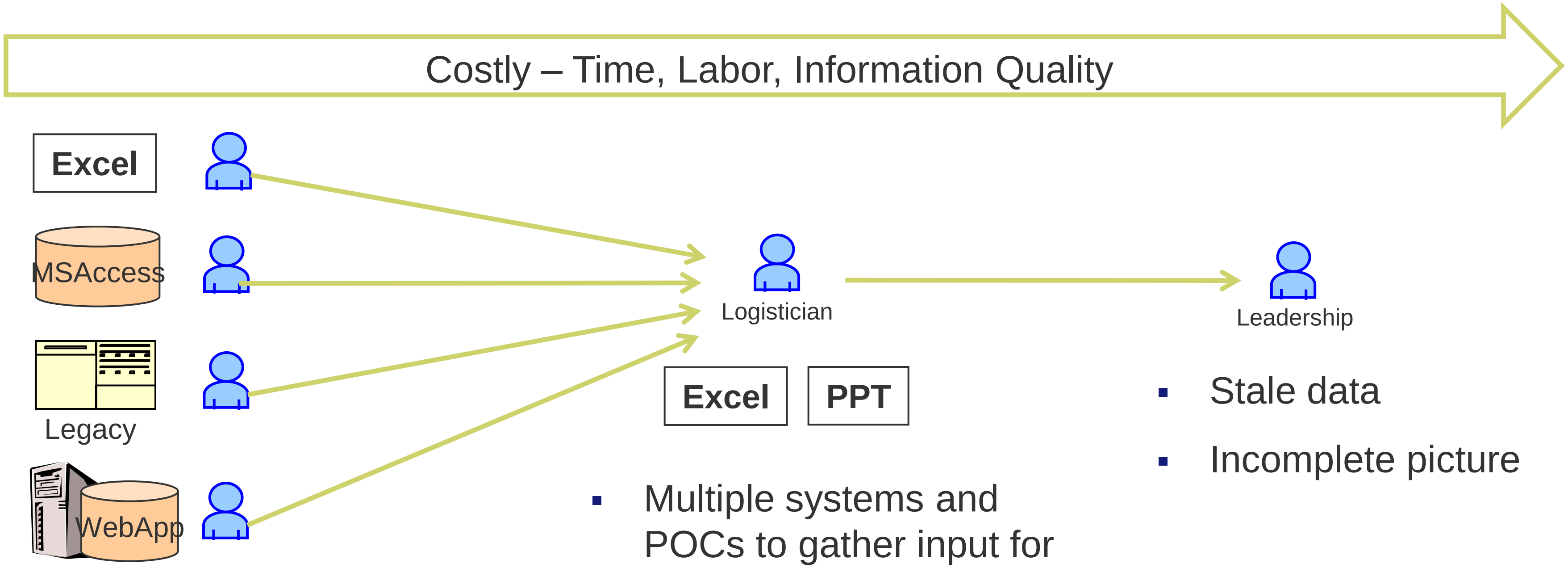
- Data Modeling
- ETL
- Metadata Management
- Data Quality
- Master Data Management

Level 1

- Application Hosting
- Security
- Portal
- Discovery Services
- COOP
- Messaging

Enterprise Reporting

The Challenge – Unifying Disparate Reporting Systems

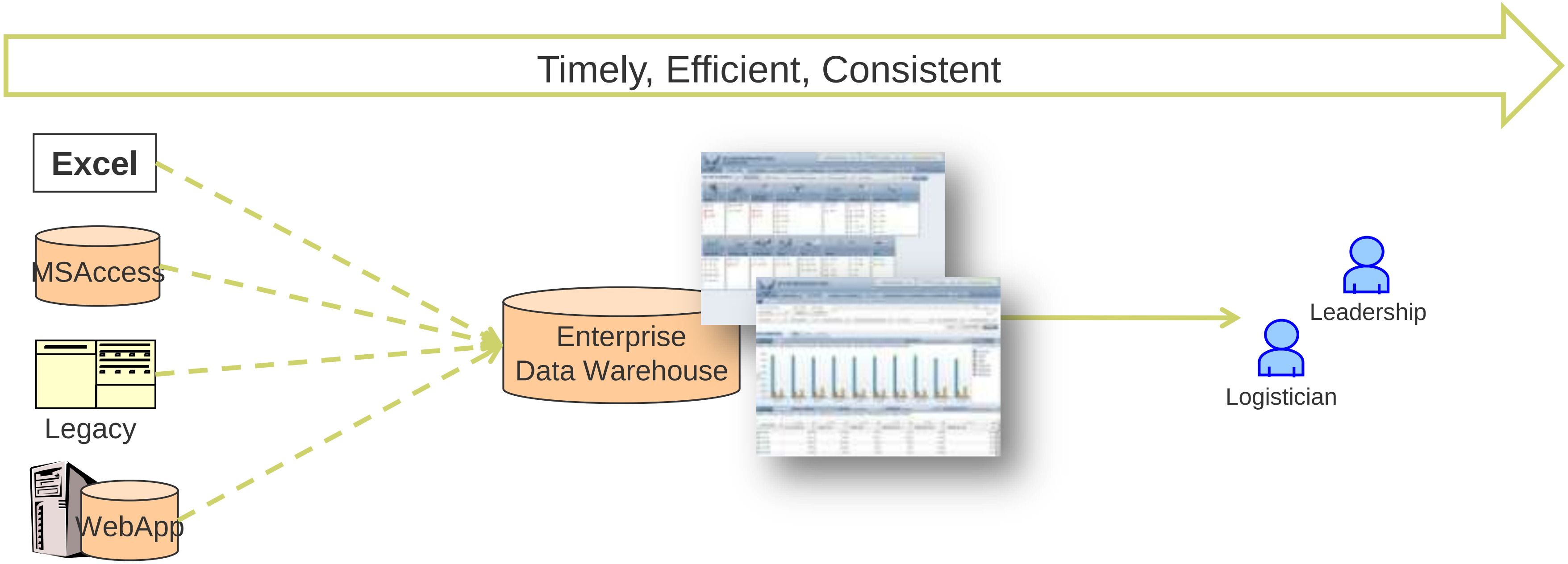


- Stale data
- Incomplete picture

- Local reporting systems
- Multiple formats
- Multiple metrics
- Multiple business rules
- Multiple processes

- Multiple systems and POCs to gather input for enterprise-level report
- Additional massaging of data
- Manual charting & report generation

Enterprise Reporting
The Solution – Data Services + SOA + RIA



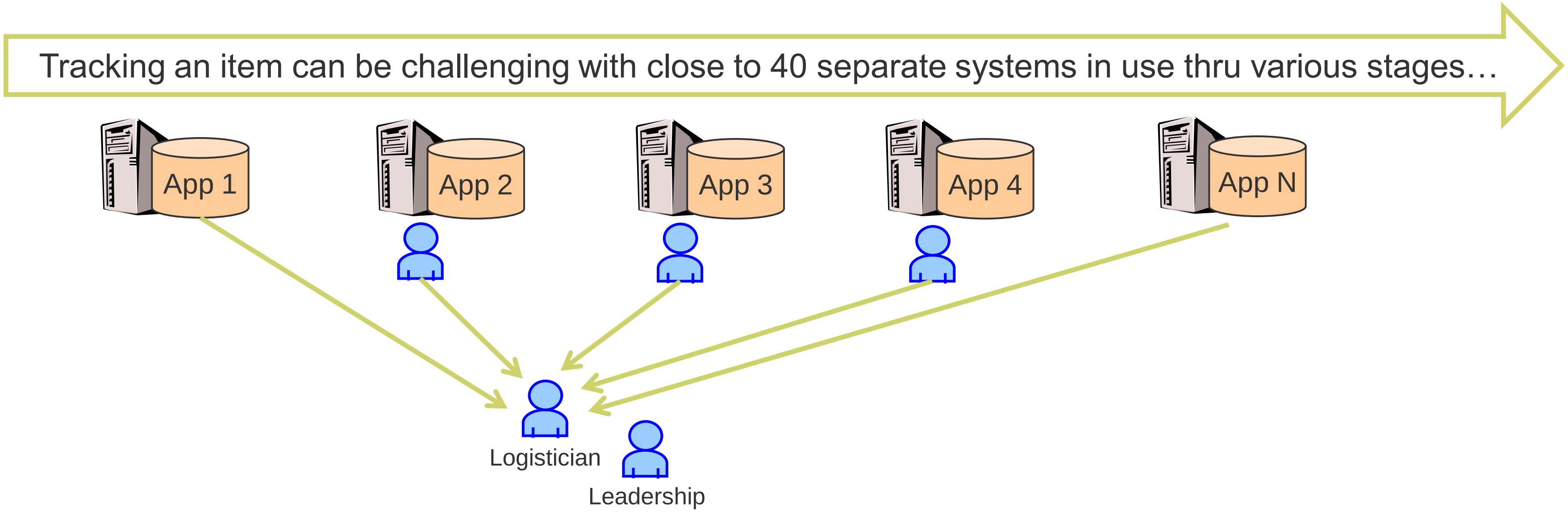
- Automated process to Extract, Transform, and Load data from local sources into an Enterprise Data Warehouse

- Consistent metrics
- Common business rules
- Single process
- One source for reporting

- Fresh data
- Enterprise view
- “Self Service”

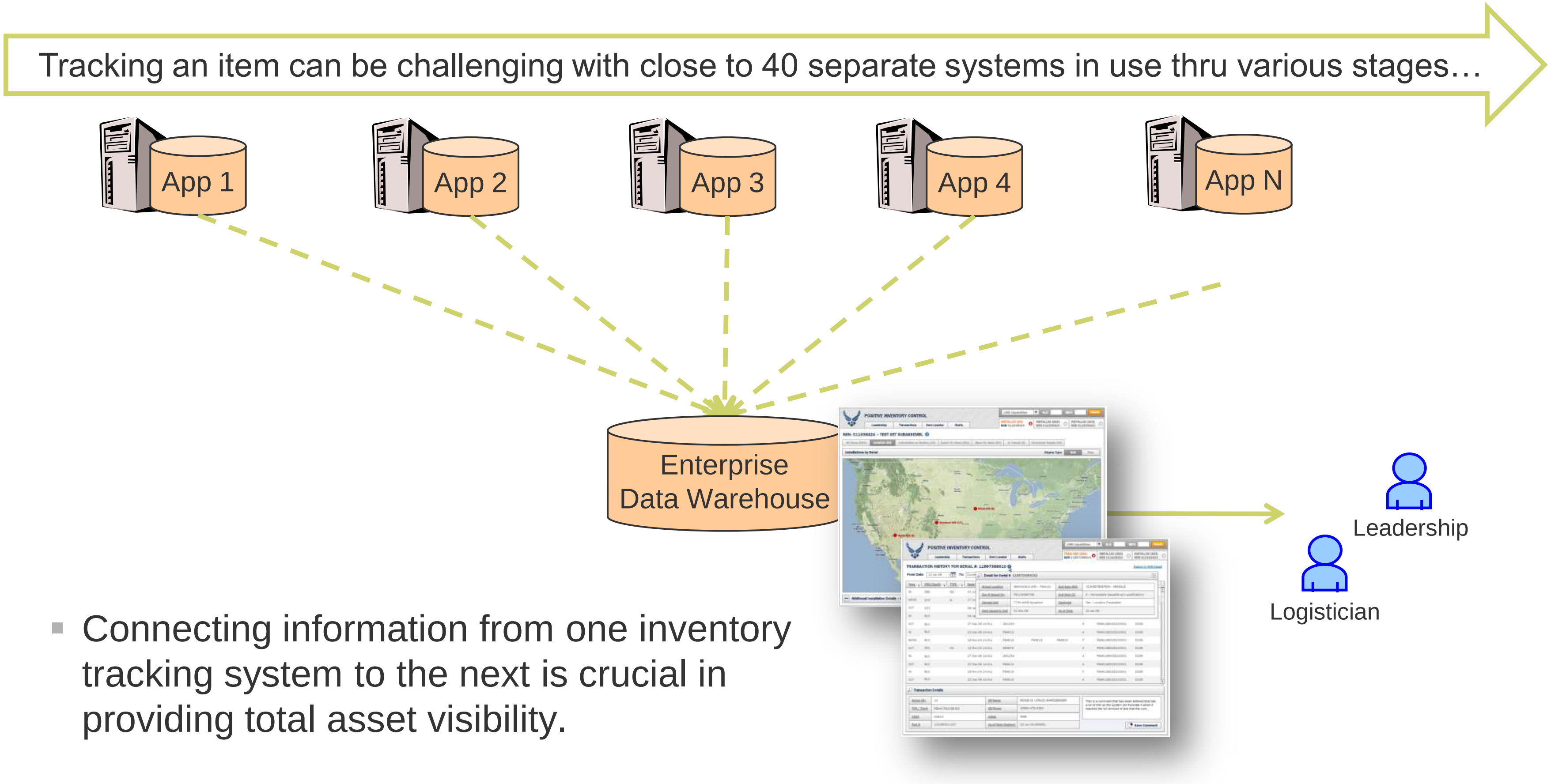
Initial release focused on one process (maintenance) and one asset (aircraft) and then expanded to other processes (supply, transportation, etc.) and other assets (vehicles, equipment, munitions, etc.)

Enterprise Inventory Tracking
The Challenge – Connecting Tracking Data End-to-End


Tracking an item can be challenging with close to 40 separate systems in use thru various stages…
App 1
App 2
App 3
App 4
App N
Logistician
Leadership

- In order to track an item, an individual may have to access an application, make a phone call, and/or send an email depending on the stage and status of that item.
- In order to get a history of the movement of that part, an individual would have to check multiple systems, make multiple calls, and send multiple emails in order to get a complete history.

Enterprise Inventory Tracking
The Solution - Data Services + SOA + RIA



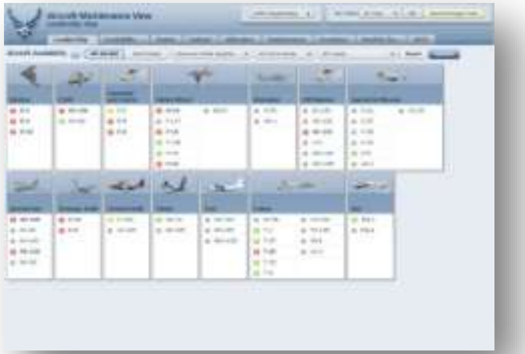
Tracking an item can be challenging with close to 40 separate systems in use thru various stages…

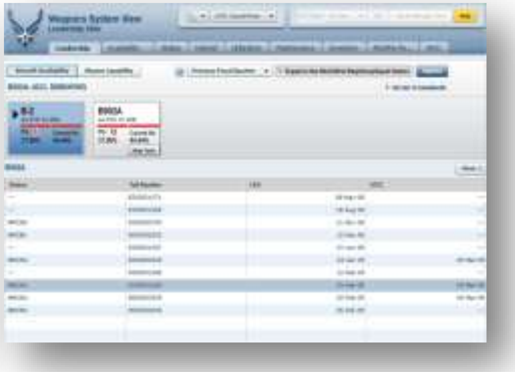
- Connecting information from one inventory tracking system to the next is crucial in providing total asset visibility.

DEMO – LIMS-EV

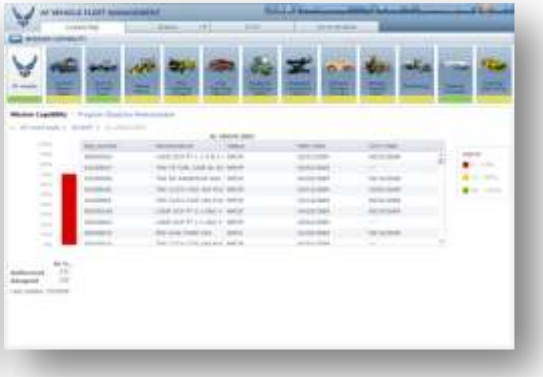
Logistics, Installations and Mission Support – Enterprise View

Weapons System View: 3-Clicks from Tail Number detail…





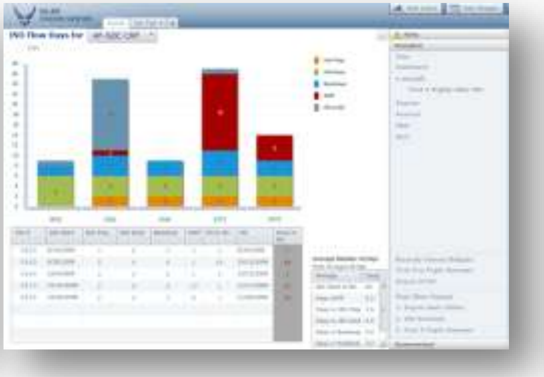
Vehicle View: 3-Clicks from Reg Number detail…



DEMO – LIMS-EV (cont.)

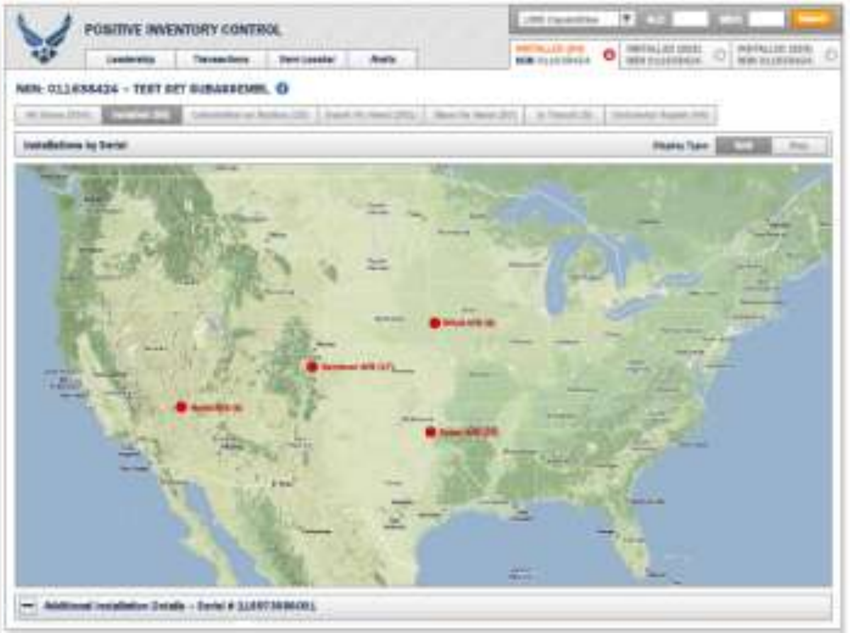
Logistics, Installations and Mission Support – Enterprise View

Mashups & Widgets: Providing & Consuming Data/Content Modules



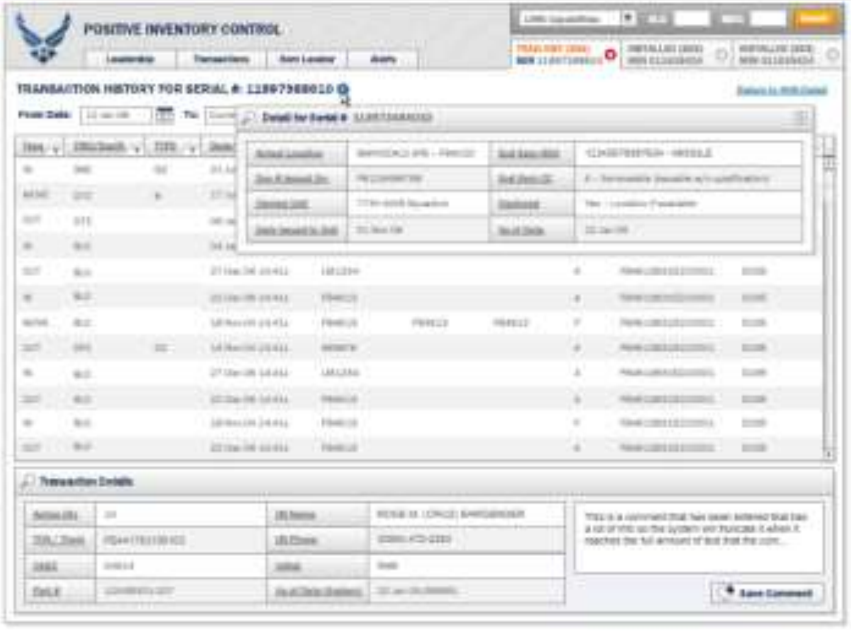
Geospatial Views: Map-based Views of Data…

DEMO – PIC
Positive Inventory Control



POSITIVE INVENTORY CONTROL

Installations by Serial



POSITIVE INVENTORY CONTROL

TRANSACTION HISTORY FOR SERIAL #: 11897368010

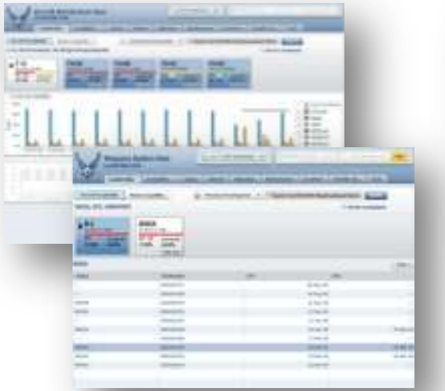
Transaction Details

- Initial release focused on the data with subsequent releases adding more visualization including map displays.

Phased Delivery by Reusing Design Patterns & Components

Analyst View
Initial Release focused on 1 asset (aircraft) & 1 process (maintanence)

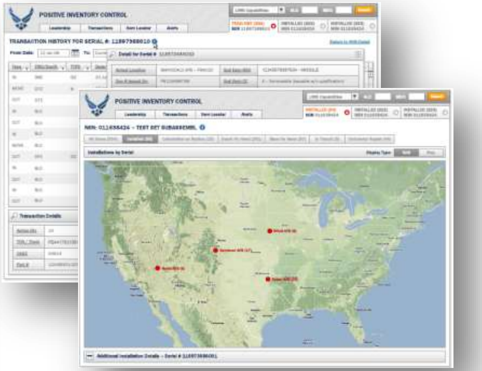
Scorecard Views
Top-level summary dashboards for leadership



Detailed Views
Drill-thru to underlying drivers behind metrics.

Additional Assets
Expand solution to Vehicles, Equipment & Munnitions

Utilities
Online slidshow presentations with real-time data



Additional Processes
Expand to Inventory Tracking, Supply, & Transportation

Other Applications
Framework leveraged for CSAF Dashboard, Data Quality, & Widgets/Mash-ups

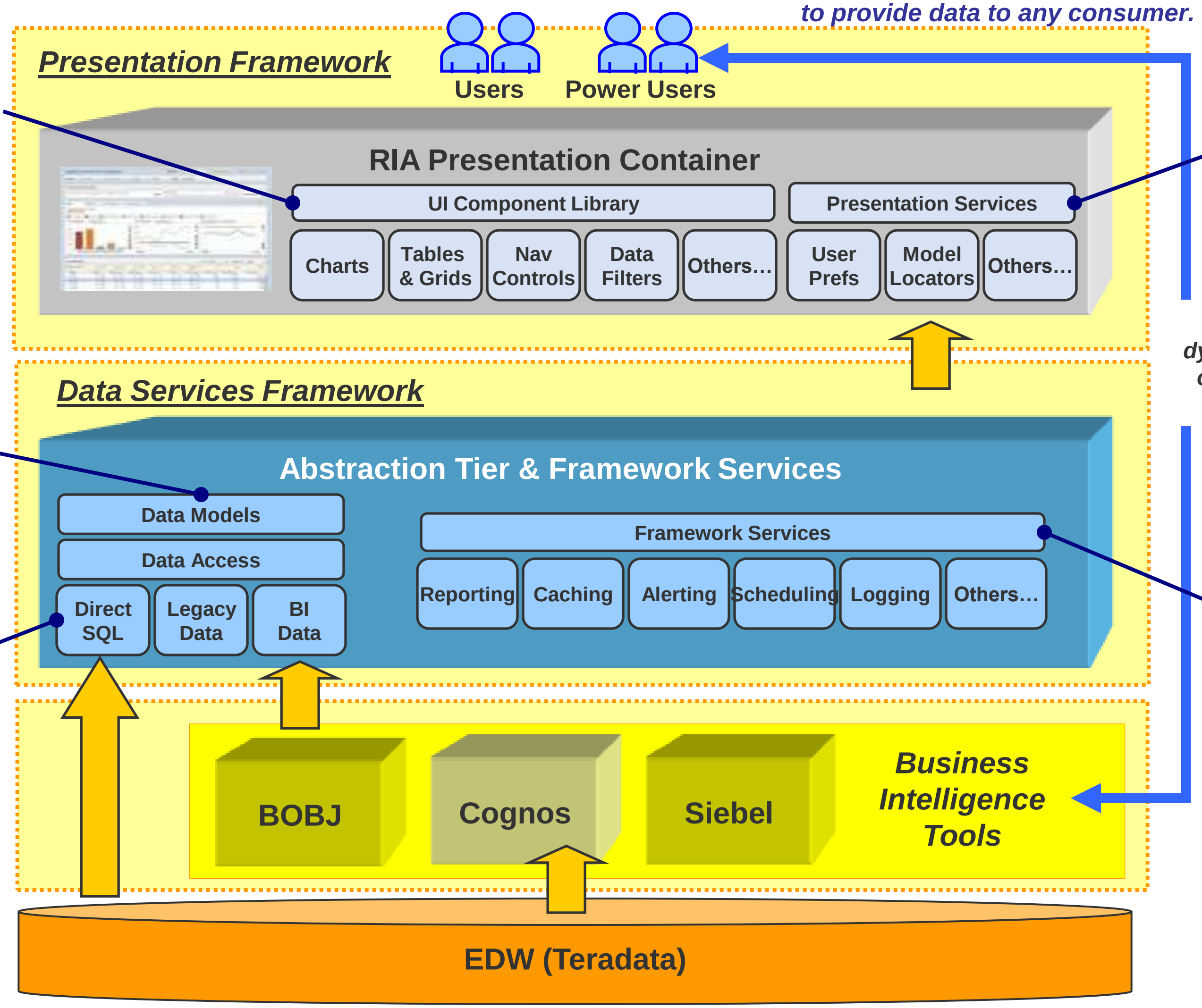
The Data Services RIA Reporting Framework

Used to Implement LIMS-EV

- *Loosely coupled architecture allows Presentation Layer to consume data from any web service and Data Services Layer to provide data to any consumer.*

- *Library of reusable components that can be leveraged to build rich internet applications*

- *Common presentation services to help manage the overall user experience*



Presentation Framework

Users

Power Users

RIA Presentation Container

UI Component Library

Charts

Tables & Grids

Nav Controls

Data Filters

Others…

Presentation Services

User Prefs

Model Locators

Others…

Data Services Framework

Abstraction Tier & Framework Services

Data Models

Data Access

Direct SQL

Legacy Data

BI Data

Framework Services

Reporting

Caching

Alerting

Scheduling

Logging

Others…

BOBJ

Cognos

Siebel

Business Intelligence Tools

EDW (Teradata)

Access to dynamic tools for only those that need it.

- *A common and consistent method for accessing data regardless of the source*

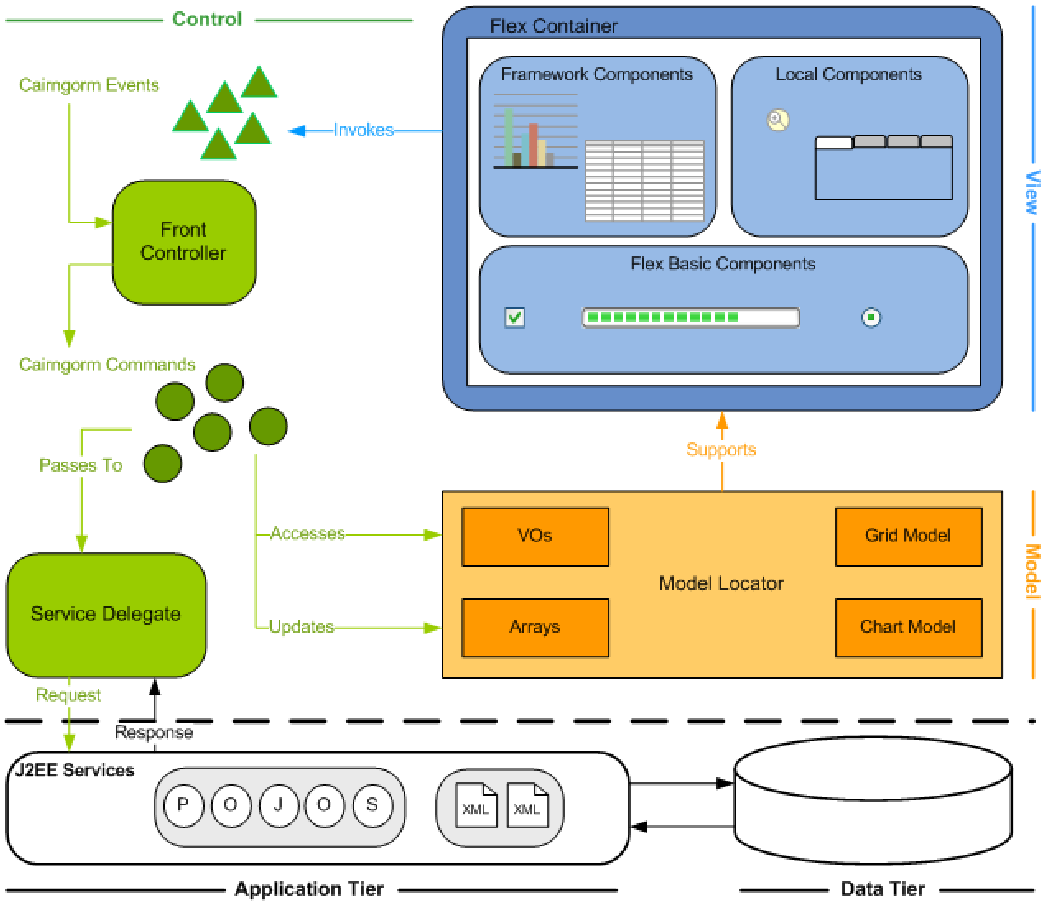
- *SOA-based services that can easily be extended and/or exploited*
- *J2EE components & web services*

- *Ability to access data directly in the warehouse w/o the overhead of BI tools*

+ Presentation Framework Architecture

The Presentation Framework builds on top of the web services middle tier to apply the UX design patterns

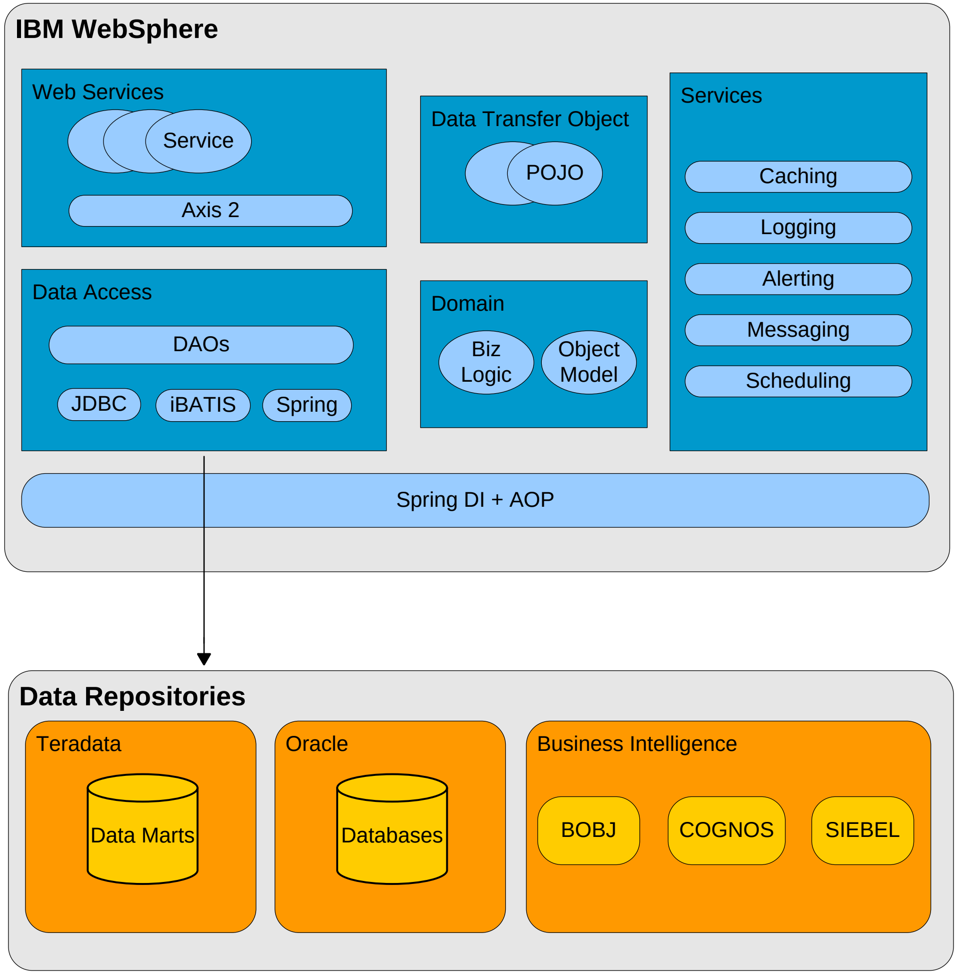
- Adobe Flex allows for visual representation of charting, navigation, and other components for user interaction
- Cairngorm is a framework built with ActionScript to allow for a centralized event based notification system within frontend Flex code
- ActionScript is the object oriented language that provides the glue between Flex components and Service Layer Middle Tier code



+ Data Services Framework Architecture

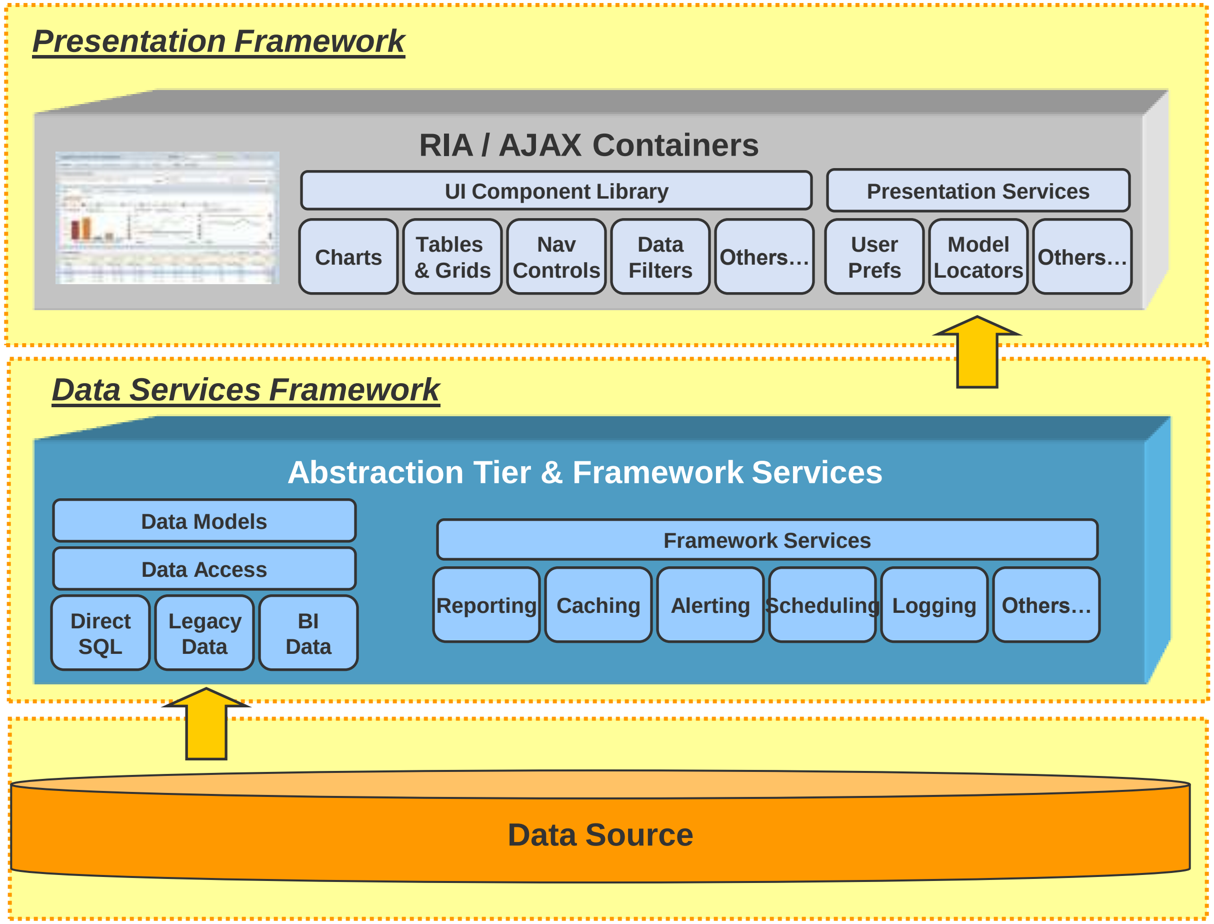
The Data Services framework focuses on providing an architectural solution to easily and reproducibly extract data to be exposed to frontend applications such as Flex

- Axis2 Exposes data via a SOAP transport layer
- iBatis is wired to the Service Layer via Spring to expose data via SQL generation
- Any dialect of RDBMS is supported due to the flexibility of iBatis
- Caching is a deployed service; other services are notional roadmap items



+ Exploiting The Data Services RIA Reporting Framework

- *Dev Framework is not a product but rather a developer toolkit. Requires a level of development & configuration to create an application.*
- *A continually evolving framework that matures with tech refreshes and new components/services deployed with new applications.*



Presentation Framework

RIA / AJAX Containers

UI Component Library

Charts | Tables & Grids | Nav Controls | Data Filters | Others…

Presentation Services

User Prefs | Model Locators | Others…

Data Services Framework

Abstraction Tier & Framework Services

Data Models

Data Access

Direct SQL | Legacy Data | BI Data

Framework Services

Reporting | Caching | Alerting | Scheduling | Logging | Others…

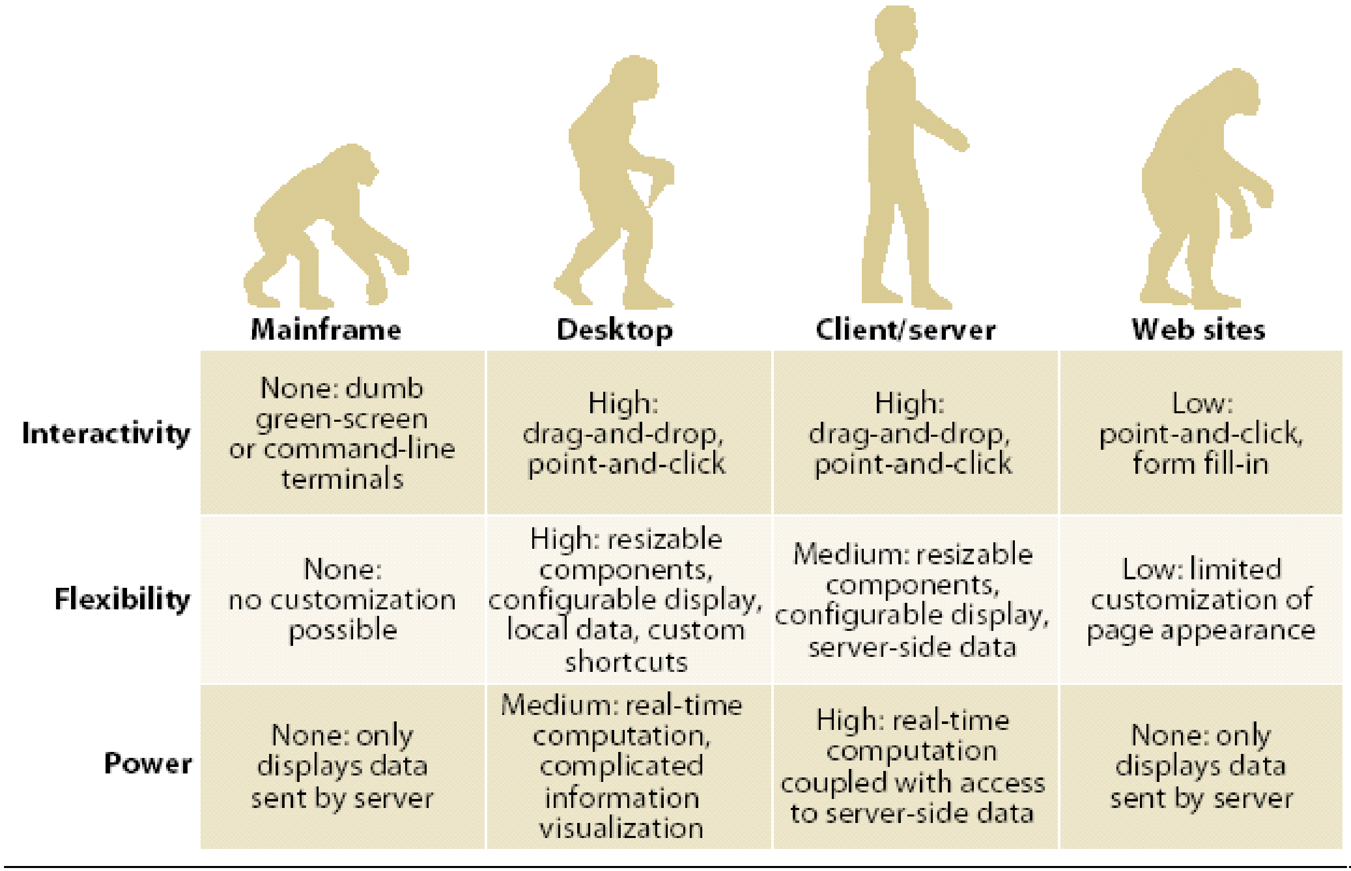
Data Source

- *Requires knowledge of Adobe Flex*
- *Developers with Object-Oriented & Java programming backgrounds can easily pick up Flex*
- *Requires knowledge of Java and J2EE frameworks such as Spring, iBatis, Axis2*

+ Why RIA (time permitting)

+ Evolution of the User Interface

Figure 1 User Interface Evolution



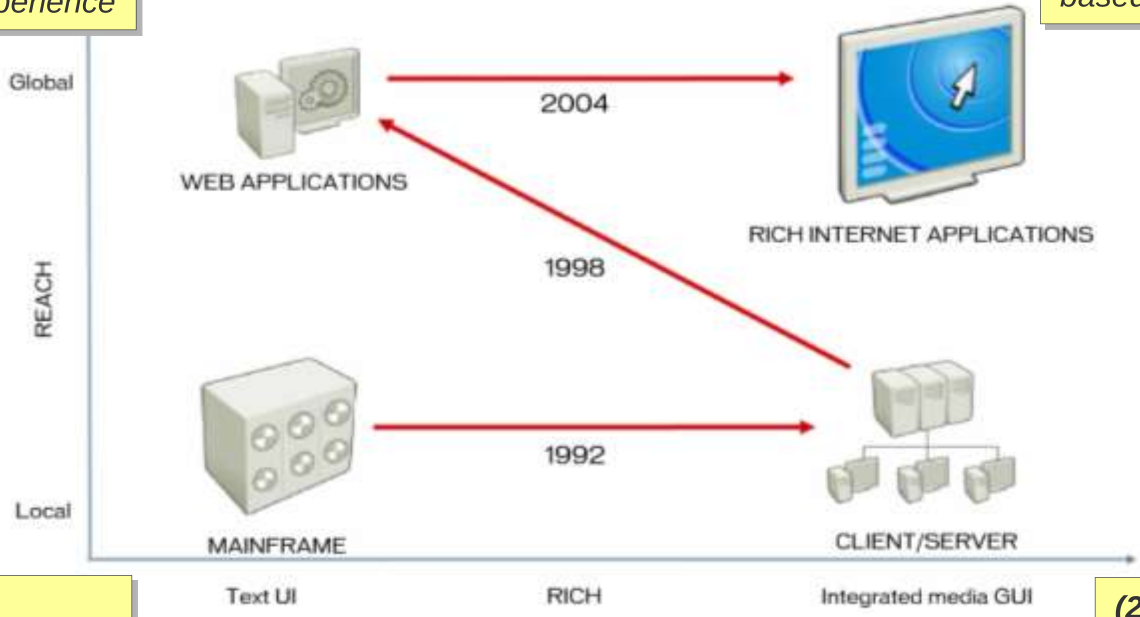
	Mainframe	Desktop	Client/server	Web sites
Interactivity	None: dumb green-screen or command-line terminals	High: drag-and-drop, point-and-click	High: drag-and-drop, point-and-click	Low: point-and-click, form fill-in
Flexibility	None: no customization possible	High: resizable components, configurable display, local data, custom shortcuts	Medium: resizable components, configurable display, server-side data	Low: limited customization of page appearance
Power	None: only displays data sent by server	Medium: real-time computation, complicated information visualization	High: real-time computation coupled with access to server-side data	None: only displays data sent by server

Source: Forrester Research, Inc.

UI Technology Evolution & the Resultant Experience Design

(3) Web Applications
Page load interaction model but advancements in visual design and style; a new focus on experience

(4) Rich Internet Applications
Intuitive and highly functional interactions; cinematic experience replaces page based model



(1) Mainframe
Effectively no UI Design - You were just glad to get information to a terminal

<u>Source – Adobe, Inc</u>

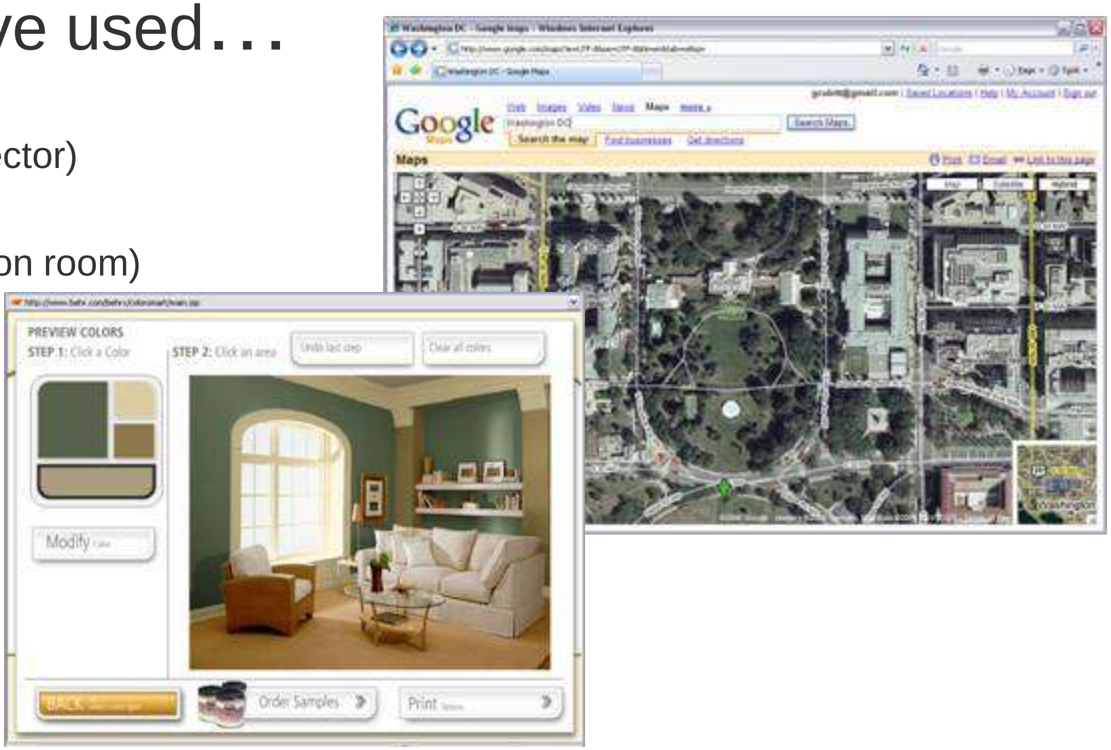
(2) Client/Server
UI design for heavy platform dependent clients; focus on functionality

+ What is a Rich Internet Application (RIA)?

- Cross between web applications and traditional desktop applications
- They transfer some of the processing to the client computer
- They combine the best of the desktop model with the best of the web model
- They create web applications with highly robust user interfaces that are not bound to the traditional request/reply model
- They run in a web browser and are typically executed using AJAX or Flash

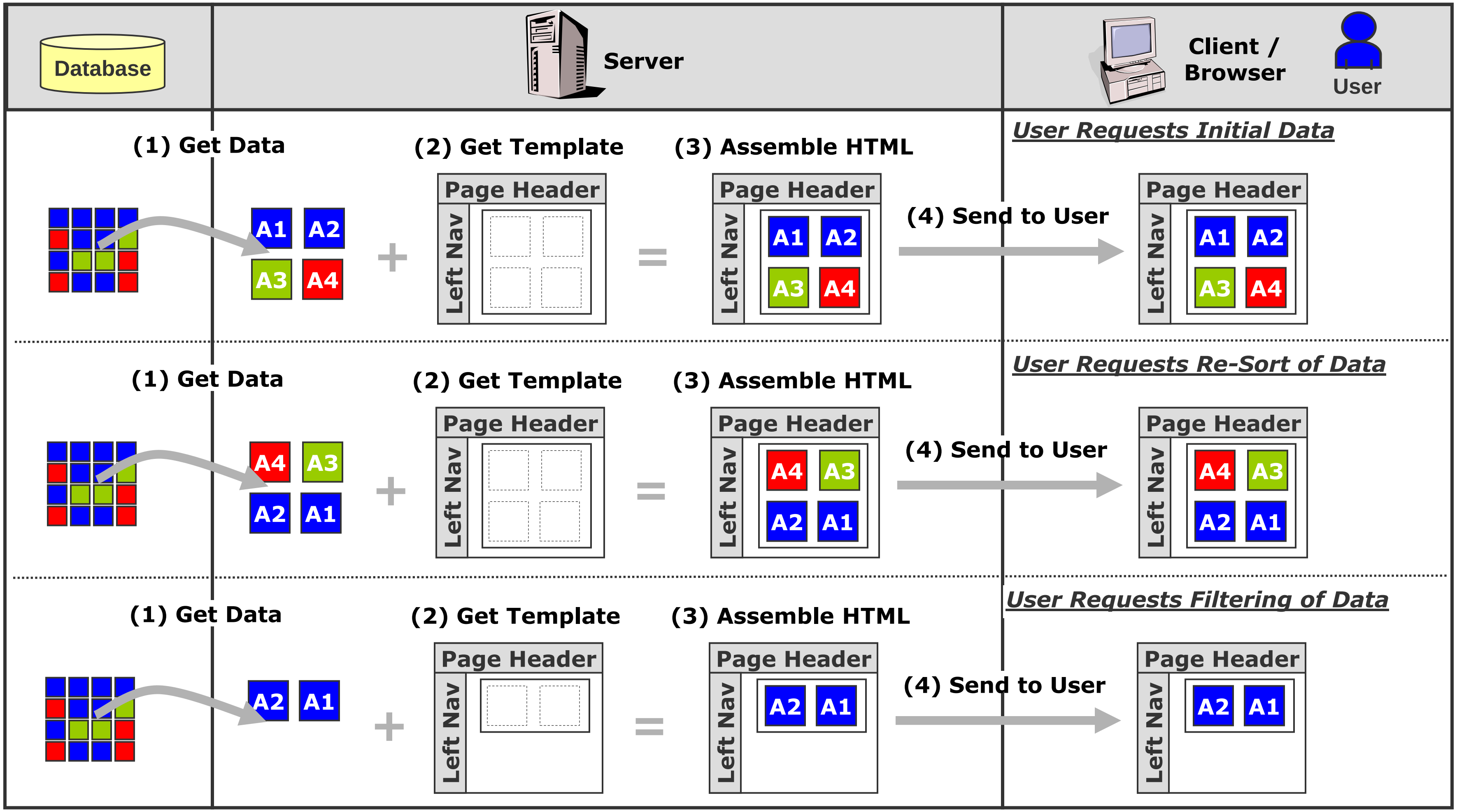
Examples you may have used…

- Google Maps
- Ford Vehicle Showroom (vehicle selector)
- Nike Store
- Behr Paints Color Smart (test colors on room)



Design Pattern Comparison

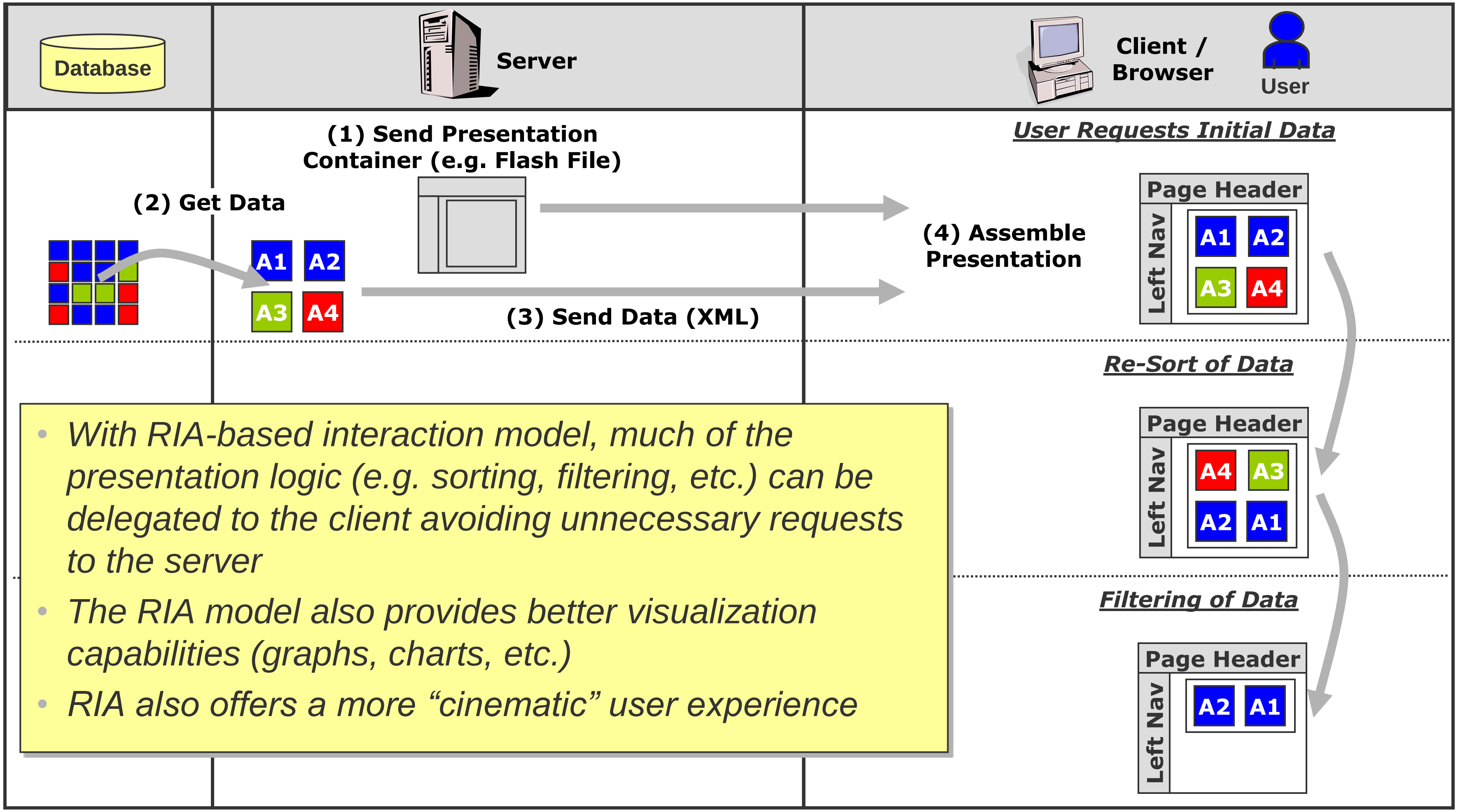
Traditional Web Page-Based Interaction



- Presentation is handled by the server resulting in full page loads, heavy server requests, more bytes to transfer, and more wait time for the user.

Design Pattern Comparison

RIA-Based Interaction



- *With RIA-based interaction model, much of the presentation logic (e.g. sorting, filtering, etc.) can be delegated to the client avoiding unnecessary requests to the server*
- *The RIA model also provides better visualization capabilities (graphs, charts, etc.)*
- *RIA also offers a more “cinematic” user experience*

Questions?