



Deploying Instant JChem on an Enterprise Scale

Brett Hiemenz

SDI Portfolio Delivery

September 9, 2011

The Challenge



- Migrate 400 ISIS Base projects to Instant JChem (IJC) in 12 months
- Secure access to the projects
- Establish single home for shared projects
- Deliver search performance that meets or exceeds the legacy system

Migrations are time-consuming

- How many projects do we *really* need?
 - Only about 200 of the 400 projects are frequently viewed. Migrate those and archive the rest.
- How can we reduce the time per migration?
 - The answer:

Automate at every opportunity

Migrating ISIS HViews to Instant JChem

1. Convert ISIS views to standard Oracle views

Custom script parses ISIS HView file (.hvd) and produces Oracle SQL

2. Promote Oracle views as IJC objects and create a 'standard' data tree

Java application drives the Instant JChem *Discovery Informatics Framework* API to create objects needed in an IJC form.

3. Create IJC forms from ISIS forms

Another Java application translates controls in ISIS Forms (.frm files) to IJC widgets. Again, the app creates the widgets using IJC's *Discovery Informatics Framework*.

- Each step requires some manual intervention, and actual IJC view layout still needed to be done by hand.
- **Automation cut average migration time from 2-3 days down to 1 day.**

Security Authentication and Authorization

- User **authentication** integrates with corporate LDAP/Active Directory system
- Also needed to address multiple **authorization** scenarios, including:
 1. Grant all discovery scientists view-only access to most projects
 2. Grant administrative access for all projects to expert users and support personnel
 3. Restrict access for certain projects to project-team members only
- ChemAxon introduced new capabilities to satisfy all these scenarios in **IJC 5.5**
- But IJC 5.5 security mechanism only governs project-level security, not views.
 - There is no central control of which views are visible.
 - There is no ability to restrict who can create and share views.
 - View ownership cannot be transferred within the IJC application.

Sharing Instant JChem Projects via URLs

- Instant JChem URLs provide client benefits and administrator benefits
 - IJC launches directly from a web page link
 - IJC automatically loads a specific project
 - Clients are not required to enter database connection details, and database access details are secured
 - Some IJC application settings can be standardized at launch time
- So how do we manage 200 project URLs? Our solution is Microsoft Sharepoint.
 - Advantages:
 - Requires minimal HTML knowledge
 - Can quickly grow a scientific community around the IJC projects
 - Projects URLs can be sorted and searched by name, owner, status, and modification date
 - Disadvantages:
 - Sharepoint limits URLs to 255 characters, and IJC URLs can be quite long
 - Building new URLs is not straightforward. Java Web Start configuration files must be created for each new URL and hosted on a separate Java (Apache Tomcat) server.

Architectural Considerations

■ Current GSK constraints

- Chemical structure and biological databases are hosted at one UK site, not replicated. This is a high-latency connection for users outside UK/Europe.
- These databases are not in the same Oracle instance
- The ISIS project database has thousands of tables and views feeding the forms

■ Current Instant JChem constraints

- IJC cannot join across Oracle schema instances.
- To use the JChem cartridge, the IJC schema must be located in the same DB instance
- Search results are obtained individually rather than in batches, slowing record-to-record navigation.

Problems to solve

1. Co-location of data needed by any single IJC form
2. Performance impact due to latency

Possible Remedies to Architectural Gaps

- Simulate co-location of source databases
 1. Use database links to make all tables/views appear in a single IJC database instance
 2. Or host multiple physical databases under a single virtual database using Composite

Recommendation: use virtualization or physical co-location of all IJC data.

- Choose one of the two solutions for latency
 1. Reduce the number of queries between IJC client and its databases.
 2. Reduce the geographic distance between IJC client and its databases.

Recommendation: strike a balance between both solutions. Minimize unnecessary queries and use geographical data replication where possible. Another option is virtual clients (Citrix).

Thoughts on Future IJC Architecture

Thick or Thin Client?

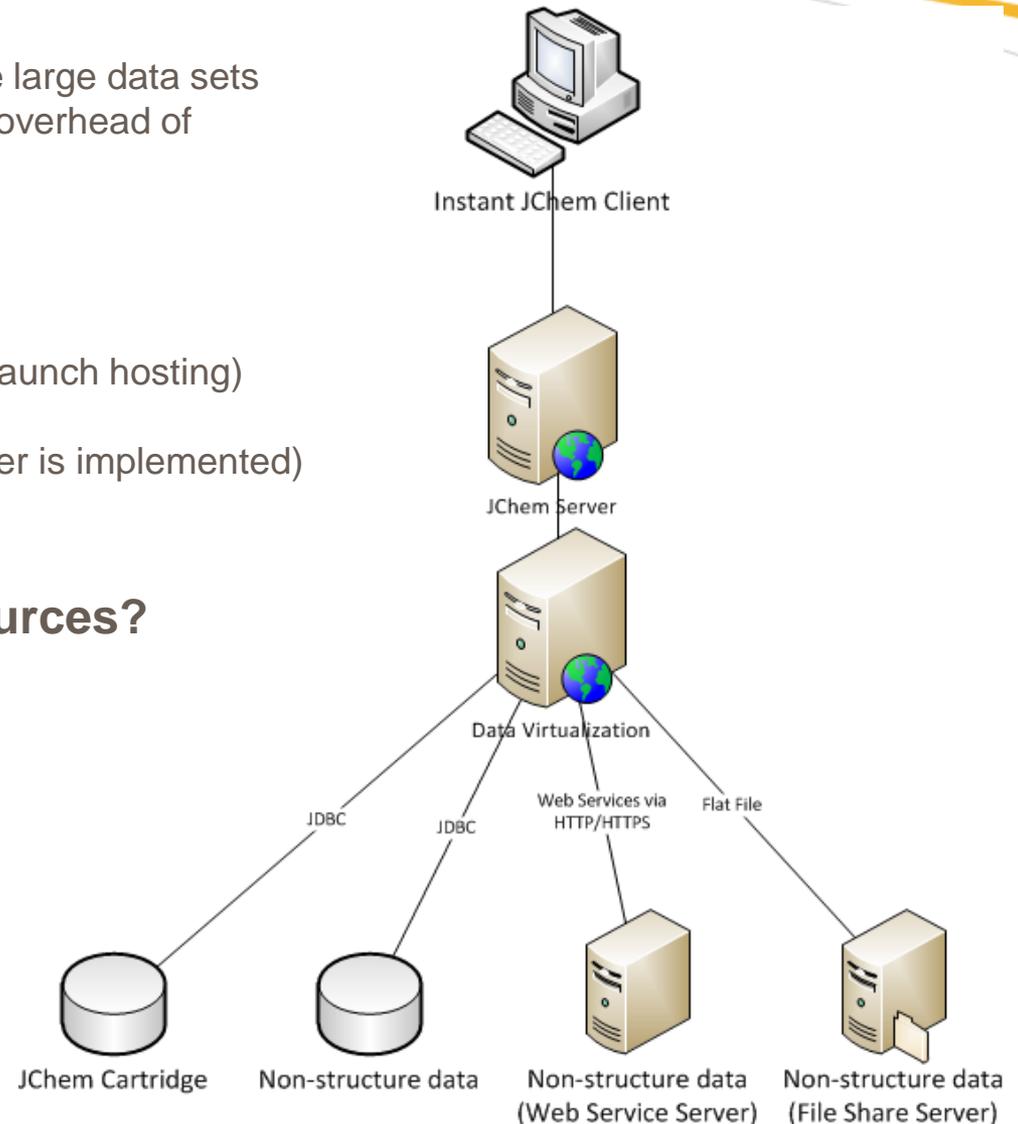
- Thick: more powerful UI; easier to manipulate large data sets
- Thin: no installation footprint; server handles overhead of database joins

Role of the JChem Server?

- IJC client server (project web pages or JWS launch hosting)
- Control project and form security
- Database client (if no data consolidation server is implemented)

How best to join multiple data sources?

- Virtualization (Composite)
- Warehousing (Netezza, Teradata)





GlaxoSmithKline