Scout at the RID
by Bertin Kiekebosch

RID = Regionaal Instituut Dyslexie.
A little personal historie
The new job (challenge) at the RID

- Website Maintenance
- Apps development
- Integration problems
- High costs
- Performance problems With 3rd party applications
- No flexibility
The technologies at the RID

Joomla, HTML CSS, Javascript

IPad & Android

Excel, XML, Webservices

Everything done by external contractors

Hard coded reporting only

MS SQL Server databases
How it felt
The search for tools in 2011

• Multitier
• Rich functional GUI
• Multiple databases
• Integration with other systems
• Productive
• Java
• Functionality of existing software
Nominees

- Spring
- Swing GUI
- RCP/ RAP
- Riena
- Scout

And the winner is ? But in 2011 ...
The existing applications

App 1
- DB1

App 2
- DB2

App 3
- DB3

App 4
- DB4
The new architecture
Dynamic database selection

- Most users only work with one database.
- Some users work with multiple databases (staff reporting/ account management).
- Most services work with one old database
- Some services work with the new database
Dynamic database selection

- Dedicated services for the new database: no brainer.

- Dynamic selection of an old database: store selection criteria in the Session Data and dynamically select a Service. See ClientProcessService.load and MpxSqlServiceProvider.getSqlServiceClass

- Loop over multiple databases for reports. See StafReportService.getVestigingenForDatabase(). Only works when database selection loop is at the client side.
Lookup services

• Works with the SQL class (convience wrapper)  
  See AbstractSqlLookupService. 
  execLoadLookupRows().

• Override it in your concrete LookupService  
  See class GemeenteLookupService
Verbose SQL Statements

• No additional frameworks (hibernate)
• Write SQL directly. See class AccountProcessService
• With many columns ?? See class ClientProcessService
• Field mapping to the rescue. See class SqlFieldMapping.
Custom components

- If Scout does not meet your requirements.
- It's 'easy' to create your own components.
- It's 'easy' to enhance components.
Custom components

- Lets forget the ‘easy’
- It is hard to re-engineer, to find out what the original idea was, how it should work.

- Lets say it is do-able.
How it feels now!

Questions?