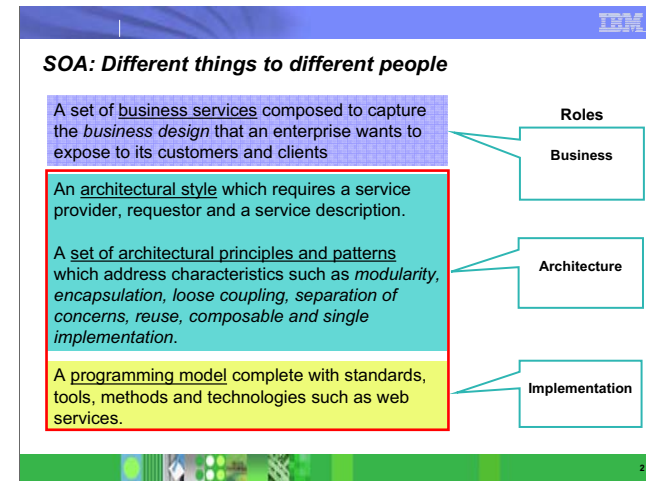
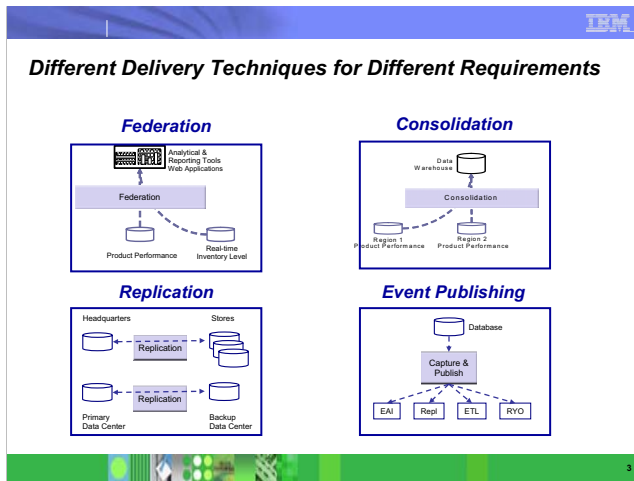




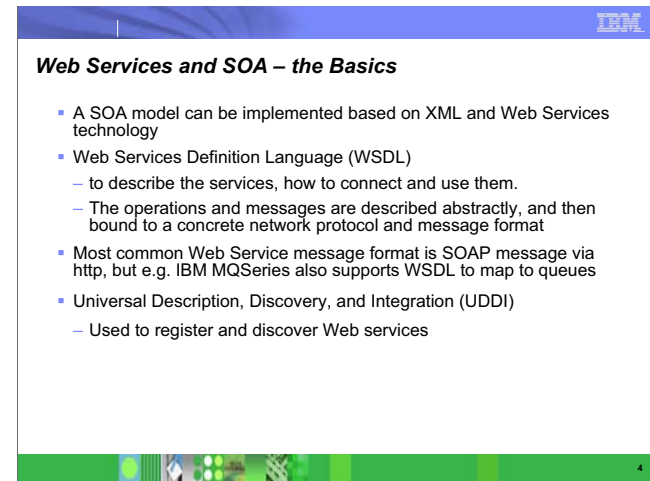
- DB2 data and stored procedures can be accessed via Web services, or via Java programs using JDBC or SQLJ. Come learn about the benefits of each of these technologies, along with tips on how to select the right method for your requirements, and best practices in the deployment of each.
- Agenda
 - ▶ Reminder / review of SOA and IOD
 - ▶ DB2 as a provider today
 - ▶ Tips to keep provider scenario of stored procedure as a service running smoothly
 - ▶ DB2 as a consumer today
 - ▶ Near term futures



- The Model-Driven Architecture (MDA), as proposed by the Object Management Group and implemented in a number of IBM Rational products, has a strong correlation to the concepts of SOA in a more abstract level. MDA is based upon the concept that any software process can be defined as a model and even a metamodel (a model of a model) that can then be translated into actual software components of an application. Thus a MDA creates a model that is compiled into a software application that in turn is compiled into an executable that can run on a platform. MDA does not distinguish between concepts of services and objects, but it does allow models to be composed of other subset models themselves, similar to the concept of process aggregation within Business Process Execution Language (BPEL), a core component of SOA.
- The remaining presentation sees SOA from the architectural role



- There are several different models for access to data. No one technique solves all problems. In some cases, the data needs to be consolidated in one place. In others, remote access is fine. Various styles of replication and event publishing also play a part.

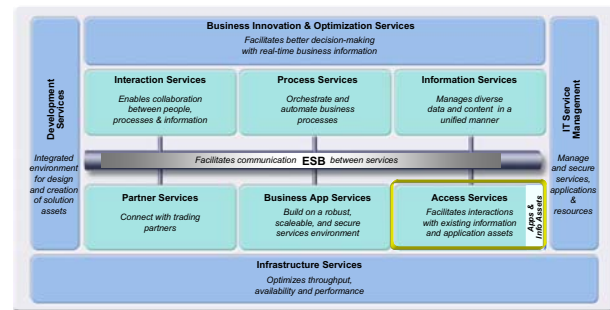


- A SOAP message is a XML document that contains the following building blocks:
 - A required Envelope element that identifies the XML document as a SOAP message
 - An optional Header element that contains header information
 - A required Body element that contains call and response information
 - An optional Fault element that provides information about errors that occurred while processing the message
- Here is an example of an SOAP message skeleton
 - `<?xml version="1.0"?>`
 - `<soap:Envelope`
 - `xmlns:soap="http://www.w3.org/2001/12/soap-envelope"`
 - `soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">`
 - `<soap:Header>`
 - `... ..`
 - `</soap:Header>`
 - `<soap:Body>`
 - `... ..`
 - `<soap:Fault>`
 - `... ..`
 - `</soap:Fault>`
 - `</soap:Body>`
 - `</soap:Envelope>`

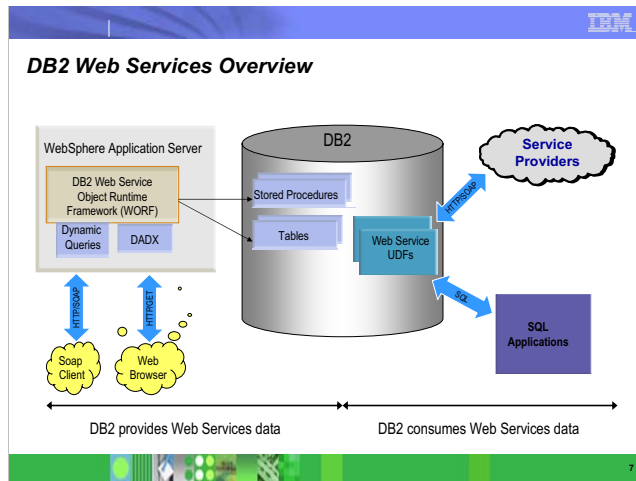
Why is Data Important in SOA

- In developing SOA solutions, ignoring the state of an organization's data risks exposing data that could be misinterpreted, incomplete, or in some cases, wrong.
 - When exposing SOA services, *data* is often exposed to users, either explicitly or implicitly
 - Data has a "*semantic context*" which often is limited in scope. Using the data out of its context may lead to erroneous results
 - Data and their inter-relationships are *complex*, and require understanding and interpretation to expose what is needed
 - In many organizations, the *quality, accuracy, and precision* of its data is, at best, highly inconsistent.
- The goal of SOA is to build flexible, dynamic *solutions*.
All aspects of *design must be addressed*

SOA Reference Architecture



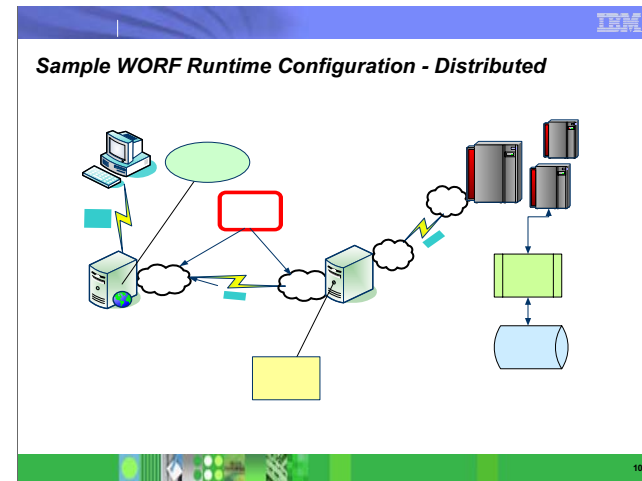
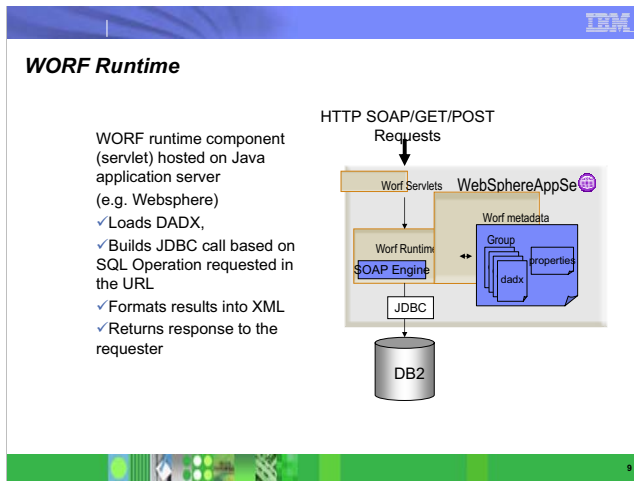
- Focusing just on the inner, light-blue boxes in the upper row – *Interaction Services, Process Services and Information Services* addresses the traditional delineation between presentation logic, business logic and data logic introduced by three-tier distributed application architectures and the Model-View-Controller pattern. When *Business Application Services* are considered as a decomposition of *Process Services* then an n-tier distributed system architecture is emerging from *Access services* are dedicated to integrating legacy applications and functions into the service-oriented architecture. This includes simple wrapping of those functions and rendering them as services (in the case where the existing function is a good match with the semantic requirements of the business model in which it will be used), or in more complex cases augmenting the logic of the existing function to better meet the needs of the business design. In the latter case, the access service may in fact invoke multiple legacy functions to achieve the semantic requirements of the service.
- The *Access Services* make the assets of legacy application available in the new architecture, therefore providing a low entry level into SOA.



- DB2 provides Web Services consumer as well as producer support. Details are discussed in the following charts.

DB2 Web Services Provider . . .

- PQ91315 adds Web Services Object Runtime Framework (WORF) support to DB2 z/OS V7 and V8 for WAS on z/OS
- Runtime framework for wrapping SQL functionality within Web Services
- Can be generated via WSAD/RAD tooling
- Utilizes Document Access Definition Extensions (DADX)
 - XML file that defines SQL operations that exist within a Web Service
- Utilizes group.properties file
 - Provides DB2 JDBC connection information for the target database
- URL to invoke Web Service identifies the DADX file as well as the SQL Operation (Method) within it that is to be invoked
- Supports the following web services operations
 - Any DB2 stored procedures (SPs)**
 - SQL requests
 - The XML Extender DAD*

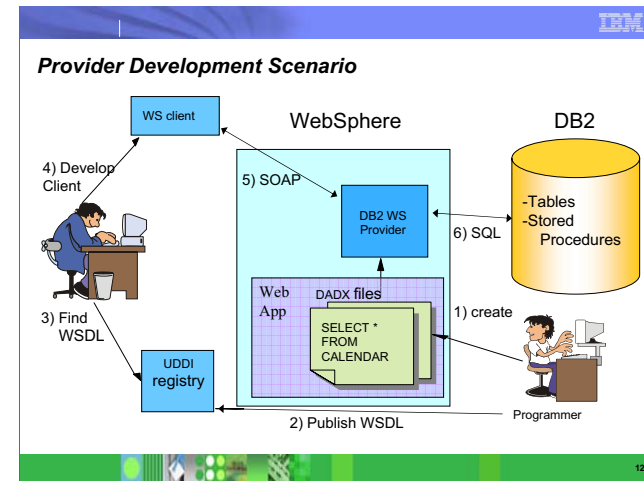
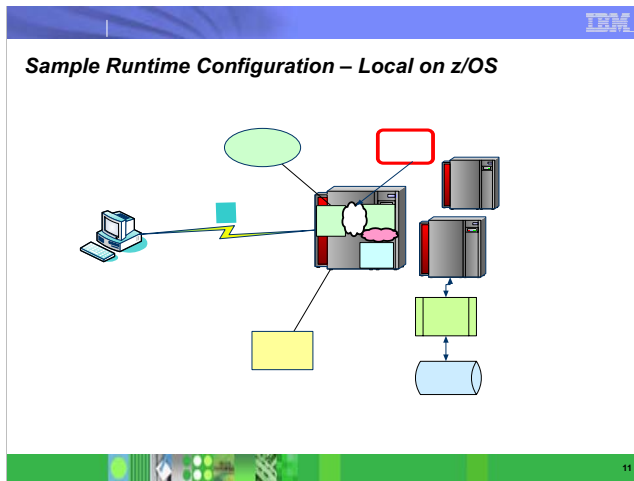


- This page shows the distributed configuration as a sample runtime environment. The client request is format as SOAP message to send to the WebSphere environment where the WORF runtime resides.WORF uses a type 4 driver to call the remote DB2 database server.

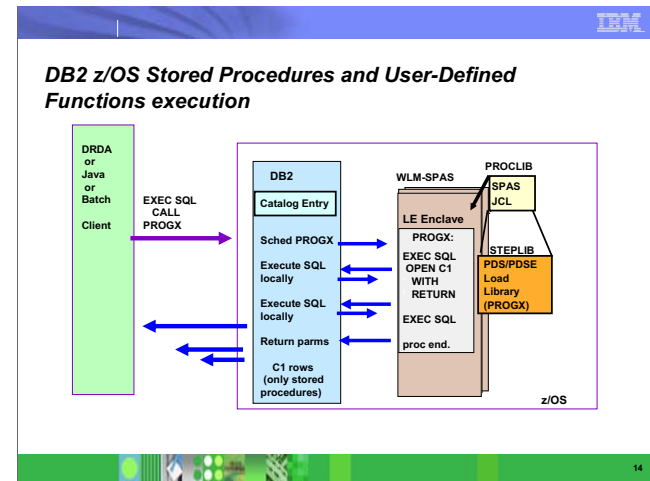
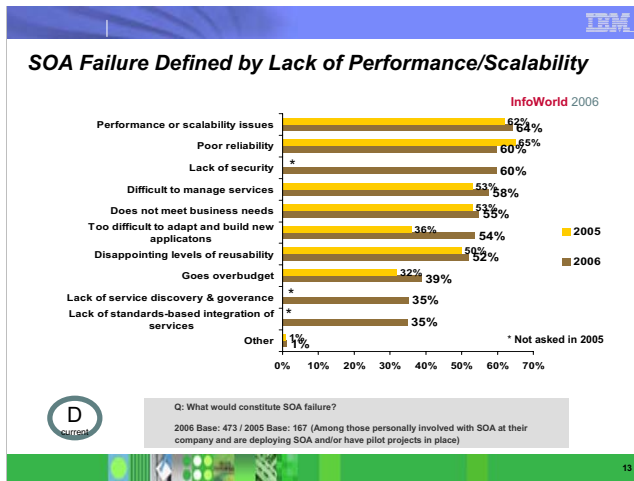
HTTP
Via
TCP/IP

SO
Enve
Via F

Web Server



- This assumes that an existing stored procedure or SQL statement is to be exposed as a web service.
- 1) The DB programmer identifies the SP or SQL statement to expose, and creates the WSDL for access, using tools. The DADX file is installed to the web server, also using the tools.
- 2) The WSDL is published in the UDDI registry.
- 3) The application programmer finds the WSDL, and 4) writes a client to invoke the web service.
- 5) When the client is executed, a SOAP request is sent to web sphere, and the DB2 web services provider uses the DADX to 6) issue the SQL statement to DB2.



- This shows the environment. When the SP executes, the SQL access is via the local RRS attach, which is a very efficient API. If result sets are left open by the SP program, they are sent to the distributed user without any further network requests.
- User-defined functions (UDFs) execute the same way, however they do not have the ability to return result set cursors.

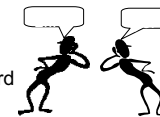
Using V8 current package path

- SP program made up of multiple parts
- Each could be in development | test | production
- Prior to V8:
 - Each program must try SET CURRENT PACKAGESET and handle -805
- In V8, using current package path:
 - Invoker 1 : development,test,production
 - Invoker 2: test,production
 - Invoker 3: production


- The current package path special register provides a list of collection id's to be searched. This allows complex stored procedure applications to be bound in different collections, and the invoker to specify the search order.

DB2 9: Native SQL Procedure Language

- Eliminates generated C code and compilation
- Fully integrated into the DB2 engine
- Extensive support for versioning:
 - VERSION keyword on CREATE PROCEDURE
 - CURRENT ROUTINE VERSION special register
 - ALTER ADD VERSION
 - ALTER REPLACE VERSION
 - ALTER ACTIVATE VERSION
- BIND PACKAGE with new DEPLOY keyword




- **Native SQL stored procedures**
- Stored procedures written in SQL procedure language enhance portability and ease of use when using DB2 for z/OS as your enterprise information source. This language is an ANSI standard language. It is similar to the proprietary stored procedure languages of several competitive databases, which assists in migrating and porting to DB2 for z/OS. This is very similar to the changes in DB2 for LUW V8.2.
- SQL stored procedures are supported by the DB2 Development Center tooling, providing an environment to code, test, and debug modules from your connected workstation. This language is currently converted to C when the CREATE PROCEDURE statement is executed. The C program is then automatically prepared, compiled, linked, and bound. The developer does not need to work with the C code.
- SQL stored procedures code will be natively integrated into the DB2 engine, eliminating the conversion to C. Additionally, extensions to the bind command will allow for the promotion of the program and access paths between environments without needing to recreate the stored procedure.
- Native SQL stored procedures are eligible for zIIP processors when they are invoked from a remote client.




Debugging Stored Procedures

- Add (DISPLAY, println, etc.) debugging statements
- "I got here"
- Take these out in production – it's expensive
- To allow multiple Stored Procedures to write output messages to the same sysout file without abend 02A
 - Change runopts to include **'MSGFILE(SYSOUT,*,*, ENQ)'**
 - Suggestion: put timestamps on messages




- ENQ of MSGFILE may have been added in JES in recent z/OS release



Interactive Debugging

- SQL procedures debugging in Version 8
- IBM Debug Tool
 - Debug interactively from the workstation
 - How-to in the DB2 **"Application Programming and SQL Guide"**
 - See www.software.ibm.com/awdtools/debugtool
 - Works with **COBOL, C/C++, PL/I**
- **Unified debugger support in DB2 9**



- The interactive debugger for SQL procedures is a huge improvement, available in V8.
- The IBM Debug Tool has been proven by many customers to be a good debugging solution for other languages.

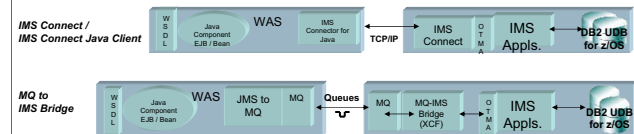
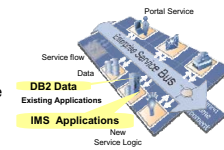
Java SP Hot Topics

- Persistent Reusable JVM topics
 - Gone from JVM 5!!
 - Never used in DB2 9
 - Optional with PK09213
- Java Shared Classes
- Build for Universal Driver and DB2 z/OS V7
- Deployment development/production
- DB2 9: Multiple/common Jars for an application
- DB2 9: Debugging

- The most questions come in from these topics.

Access DB2 data via IMS™ as Web Services

- Enabled as Web services, the unchanged IMS Transactions, accessing DB2 data, can today support a Service Oriented Architecture (SOA)
- IMS C, Cobol and MFS-based Transactions can be enabled as Web services via WebSphere/Rational Servers/Tooling
- IMS Web Services Connectivity Configuration Examples:



- Existing legacy IMS application can be brought as asset into SOA. IMS provides Webservices support via several configurations, either via JCA or SOAP Gateway connecting to IMS Connect.

IBM

Access DB2 data via CICS as Web Services

- Enabled as Web services, the unchanged CICS Transaction accessing DB2 data, can today support a Service Oriented Architecture (SOA)
- CICS supports Web Services using the WebSphere Application Server Connector Architecture (JCA) and native Web Services.

21

- Similar is true for CICS application. CICS has an excellent Web Services provider and consumer support either via JCA or via the native support.

IBM

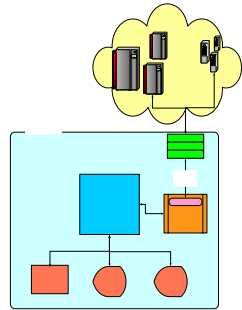
DB2 Web Services Consumer

- PQ91316 adds function as a Web Service consumer to DB2 z/OS V7 and V8
 - Allow mainframe-based enterprise applications to invoke Web Services regardless of their location
- Four SQL User-Defined Functions (UDFs)
 - Three input parameters
 - ENDPOINT_URL : The URL of the web service. Defined as VARCHAR(256)
 - SOAP_ACTION : Used as part of the SOAP request. Depending on the Web server, this parameter can be optional. If it is required, the required value can be found in the WSDL of the Web service. Defined as VARCHAR(256)
 - SOAP_BODY : The operation name and the parameters to the Web service. The data in this parameter should be well-formed XML. Defined as VARCHAR(32672) or CLOB(1M)
 - One output parameter
 - Defined as VARCHAR(32672) or CLOB(1M)

22

DB2 Web Services Consumer . . .

- Process flow
 - Receive input parms from SQL statement
 - Compose HTTP/SOAP request based on the input to the UDF
 - Invoke TCPIP socket call via zOS USS APIs and send HTTP/POST request
 - Receive reply from Web Service Provider
 - Validate HTTP headers
 - Strip SOAP envelope and return SOAP Body (include namespace referenced in SOAP envelope) to DB2 client application
- Tooling support for DB2 z/OS in RAD V7
 - Create a SQL UDF specifically for a WS operation
 - WSAD/RAD create those from WSDL



Web Services Consumer Example

- Local SQL can be replaced by SQL call that resolves into WebServices call

```
SELECT rate FROM Ratetable
WHERE country1=c1 AND country2= c2;
```

```
SELECT rate FROM TABLE (getRate(?,?));
getRate() is a wrapper UDF that calls SOAP UDF
and reads XML reply and returns results as SQL
result,
```

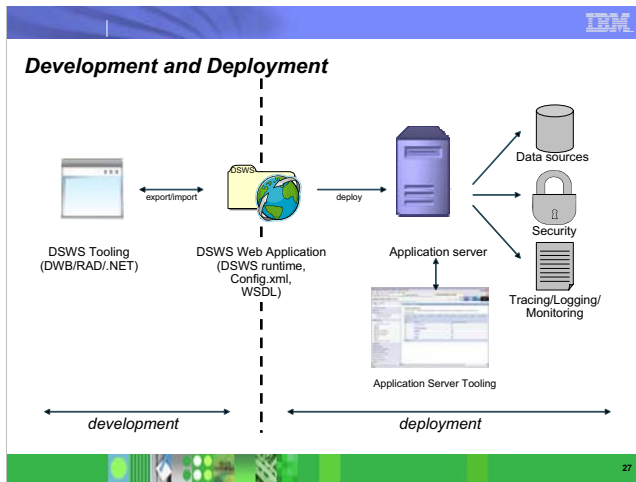
- CREATE FUNCTION getRate (
 - country1 VARCHAR(100),
 - country2 VARCHAR(100))
 - RETURNS DOUBLE
 - LANGUAGE SQL
 - EXTERNAL ACTION
 - NOT DETERMINISTIC
- -----
- -- SQL Scalar UDF that wraps the web service, "getRate", described by
- -- <http://www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl>
- -----
- RETURN db2xml.extractDouble(
 - db2xml.xmlclob(
 - db2xml.soaphttpv('http://services.xmethods.net:80/soap',
 - "
 - varchar('<m:getRate xmlns:m="urn:xmethods-CurrencyExchange" ' ||
 - 'SOAP-
 - ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">' ||
 - '<country1 xsi:type="xsd:string">' || country1 || '</country1>' ||
 - '<country2 xsi:type="xsd:string">' || country2 || '</country2>' ||
 - '</m:getRate>'))), '/*');

SQL/XML and DB2 9

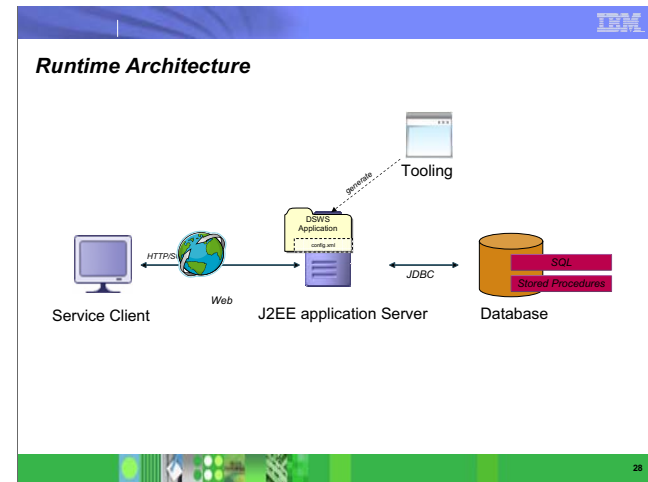
- XML is a key 'data interchange' format
 - Business critical information exchanged between B2B/B2C, inter- and intra-enterprise environments
 - input and output data of WebServices calls are in XML
- DBMS provides
 - Large scale storage / Management
 - Transformation / Aggregation
 - Interrogation / Analysis of data
- Combine strengths of both worlds by additions to SQL that enable
 - XML view of relational data
 - SQL data viewed and updated as XML
 - Done via document shredding and composition
 - DTD and Schema validation
 - XML as a rich data type
 - Atomic storage and retrieval
 - Powerful query capabilities, indexing, ...

Technology Future for DB2 Web Services Provider

- Data Server Web Services (DSWS) (planned for 2007)
 - Next generation of the DB2 Web Service provider
 - Allows simple and fast SOA-enablement of database operations
 - Successor of WORF
 - Re-architected
 - Makes more use of standard Java libraries (like Transformer API)
 - Full XML integration
 - Better tooling integration (DWB/RAD)
 - More focus on logging, tracing and monitoring
 - XSLT processing
 - Better Web 2.0 integration
 - Simpler in regards to tooling, configuration and runtime

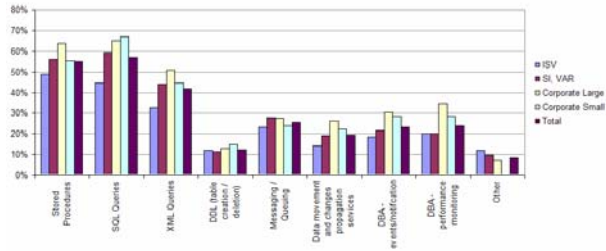


- Here is a picture of the Data Server Web Services life cycle, with development and deployment.



Common DB Tasks thru Web Services

- Stored Procedures, SQL Queries and XML Queries are the dominating operations



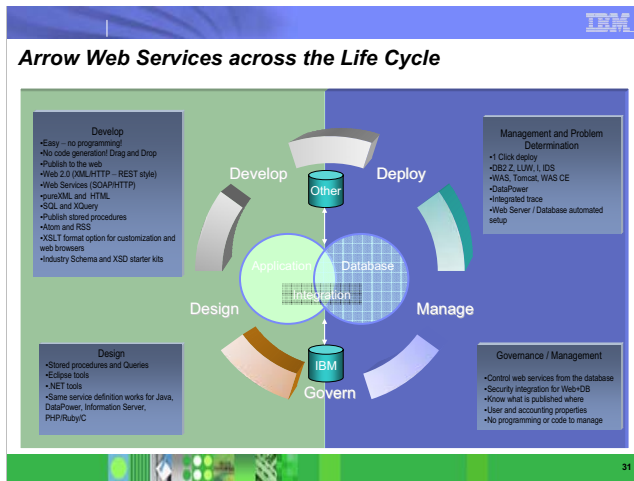
- Strong evidence of requirements to expose data assets as Web Services
- Need first class tools to meet these requirements

29

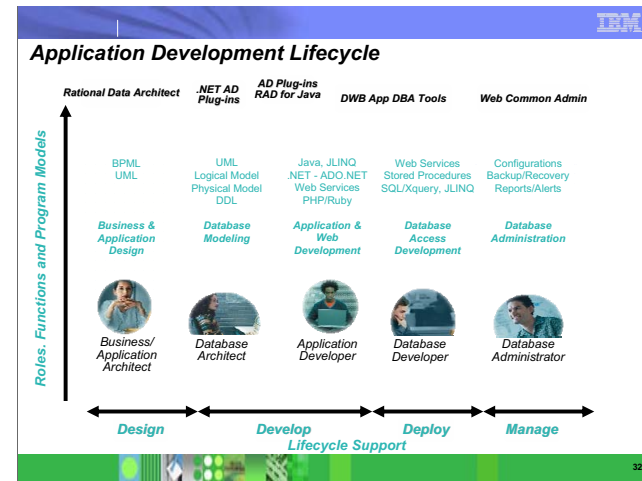
What problems does Arrow solve

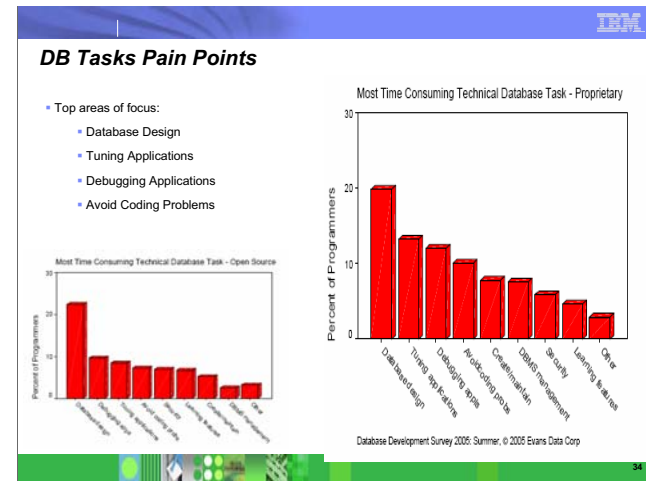
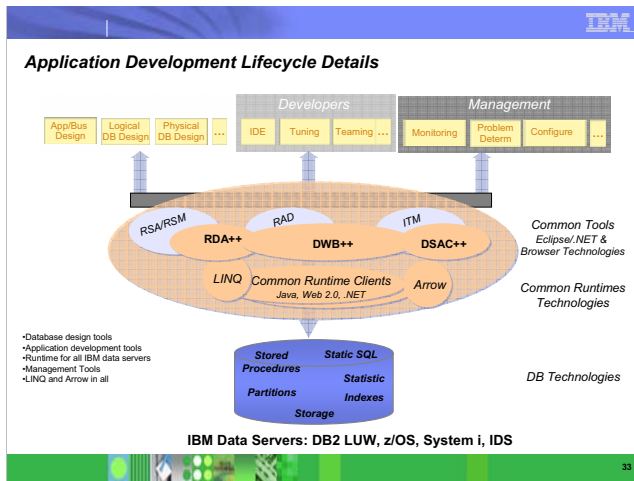
1. Maximizes the value of existing data assets for new classes of applications by exposing to the web.
2. Provides simple web access to information and integrates the data server into Service Oriented Architecture.
3. Simplifies the development of pureXML applications through URI access to integrated XML and Relational information.
4. Enabling App Developers and Web 2.0 developers through URI access from modern programming languages including PHP, Ruby, and Python.
5. DB Administrators need end-to-end monitoring, problem determination, and manageability across WebSphere and DB2 due to application and database disconnects.
6. DB Administrators need to quickly publish controllable data assets.

30



Very popular customer request (Ford, Schwab, Principal Financial, New York Life, Verizon, Nexaweb, Univar, ...)
 Most features are 2007, some features are 2008





- Top 4 areas of focus:
 - DB Design
 - Tuning
 - Debugging
 - Coding

Value of DataPower for z and DB2 for z/OS

- Simply, the Offloading and Acceleration of high overhead Web Services processing
 - Addresses a similar theme to z specialty engines (e.g. zAAPs, zIIPs)
- Compute cycle intensive XML and security-related processing (especially Crypto, xslt processing)
- Other general functional capabilities: application level firewalling, routing, filtering, ...
- Appliance model: improved consumability



... Continued ...

35

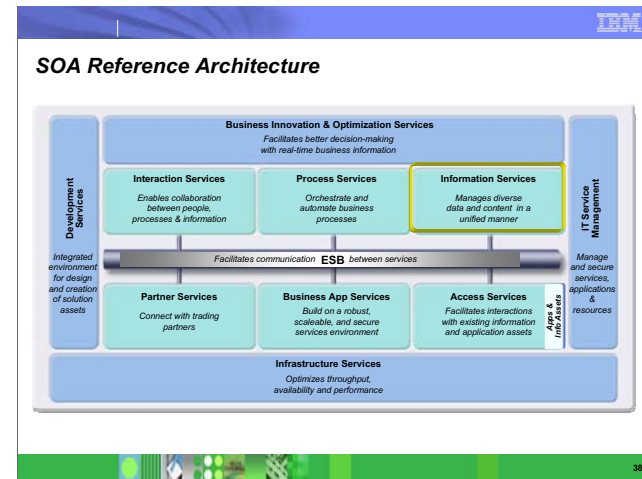
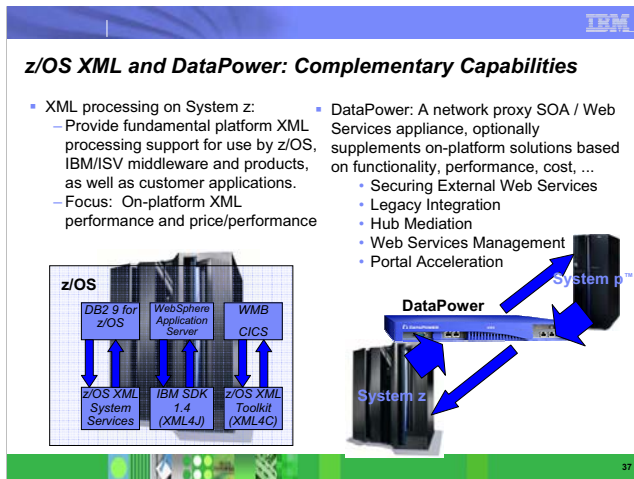
- <http://www.ibm.com/software/integration/datapower/>

DataPower DB2 Web Services Feature (planned end of 2007)

- High performance Web Services processing and mapping to SQL statement
- DataPower communicates with DB2 via DRDA
- DB2 and DataPower tooling support



36



• The

▶ Currently, XML capabilities on System z (on z/OS) are intended to provide fundamental high-performance XML parsing that can be leveraged by the operating system itself, IBM middleware and other products, as well as customer and ISV applications. z/OS XML System Services is only one of several XML capabilities on z/OS. z/OS XML System Services is integrated in the base of z/OS and is accessible via an Assembler interface today with C/C++ interface planned for z/OS V1.9.

▶ As XML increasingly becomes a dominant data format in I/T processing it is increasingly important to deal with the larger processing requirements of XML (such as parsing and validation). The SOD for z/OS XML System Services future exploitation of zAAP and zIIP specialty engines is intended to improve the cost competitiveness of processing XML data on z/OS.

▶ XML parsing and processing on z/OS is complementary to that found in the IBM Websphere DataPower SOA appliance – they really don't overlap.

▶ z/OS XML and related support operate at the level of XML parsing, that is, interpreting textual XML data, verifying the data adheres to the W3C XML standard rules describing well-formed XML content, optionally verifying that the data adheres to a specific schema or grammar, and returning extracted information to the application in a structured form for further processing. DataPower primarily serves to provide functionality in the higher level area of web services processing in support of SOA. While the processing of XML formatted messages is certainly a part of web services processing, there are many other aspects to it, such as processing in support of the various WS-* standards, such as WS-SEC, transformations that can be made to the messages, application level firewalling functions, hub mediation, etc. z/OS does not provide this type of functionality, but DataPower does play a role here, supplementing support available in on-platform solutions such as WebSphere Application Server, CICS, WESB, WMB, etc.

Information Services in IBM Products

- Business Information Services
 - WebSphere Customer Center
 - WebSphere Product Center
 - IBM Entity Analytics
- Information Integration Services
 - IBM Information Server
- Content & Discovery Services
 - WebSphere Information Integrator Content Edition
- Data Services
 - DB2 9

39

- We've just finished discussing IBM Data Services. We expect to see many of these delivered with the next DB2 for Linux, UNIX and Windows version, which is now in open beta. Data Server functions can deliver with IDS, DB2 on Systems p, i, x and z.
- Next, we'll extend some of the function with the IBM Information Server.

The IBM Solution: IBM Information Server

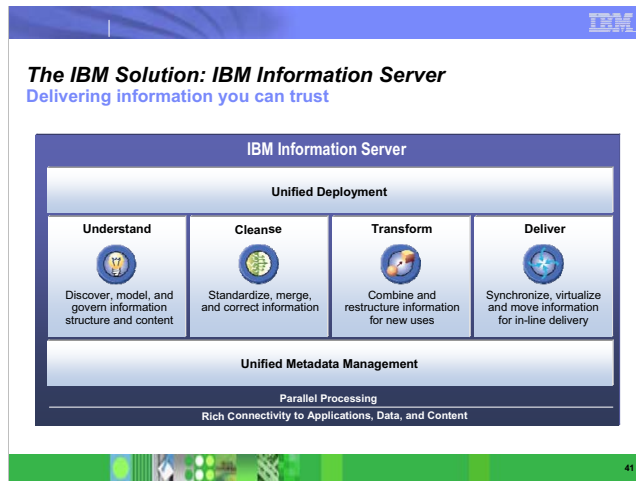
Delivering information you can trust

40

The culmination of our integration work has led us to our latest platform offering – the IBM Information Server. IBM Information Server is a revolutionary new software platform from IBM that helps organizations derive more value from the complex, heterogeneous information spread across their systems. It enables organizations to integrate disparate data and deliver trusted information wherever and whenever needed, in line and in context, to specific people, applications, and processes. IBM Information Server helps business and IT personnel to collaborate to understand the meaning, structure, and content of any type of information across any sources. It provides breakthrough productivity and performance for cleansing, transforming, and moving this information consistently and securely throughout the enterprise, so it can be accessed and used in new ways to drive innovation, increase operational efficiency, and lower risk.

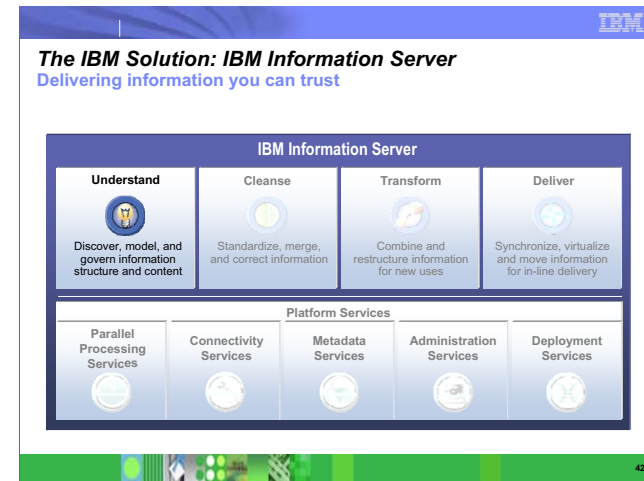
IBM Information Server is designed to help companies leverage their information across all its sources. IBM Information Server delivers all of the functions required to integrate, enrich and deliver information you can trust for your key business initiatives. IBM Information Server allows you to:

- **Understand** all sources of information within the business, analyzing its usage, quality, and relationships
- **Cleanse** it to assure its quality and consistency
- **Transform** it to provide enriched and tailored information, and;
- **Federate** it to make it accessible to people, processes, and applications



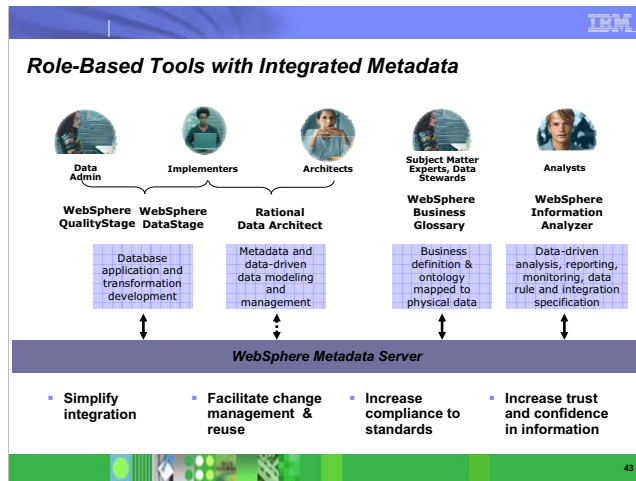
All of these functions are based on a parallel processing infrastructure that provides leverage and automation across the platform. The Information Server also provides connectivity to nearly any data or content source, and the ability to deliver information through a variety of mechanisms. Underlying these functions is a unified metadata management foundation that provides seamless sharing of knowledge throughout a project lifecycle, along with a detailed understanding of what information means, where it came from, and how it is related to information in other systems. Integration logic built within IBM Information Server can easily be deployed and managed as a shared service within a SOA.

For more information on SOA and Information Server, see the new redbook, SOA Solutions Using IBM Information Server, SG24-7402, <http://www.redbooks.ibm.com/abstracts/sg247402.html?Open>



IBM Information Server provides:

- access to the broadest range of information sources
- the broadest range of integration functionality, including federation, ETL, in-line transformation, replication, and event publishing
- the most flexibility in how these functions are used, including support for service-oriented architectures, event-driven processing, scheduled batch processing, and even standard APIs like SQL and Java.
- The breadth and flexibility of the platform enable it to address many types of business problems and meet the requirements of many types of projects. This optimizes the opportunities for reuse, leading to faster project cycles, better information consistency, and stronger information governance.
- Regarding Service-Oriented Architectures, **information integration enables information to be made available as a service**, publishing consistent, reusable services for information that make it easier for processes to get the information they need from across a heterogeneous landscape.

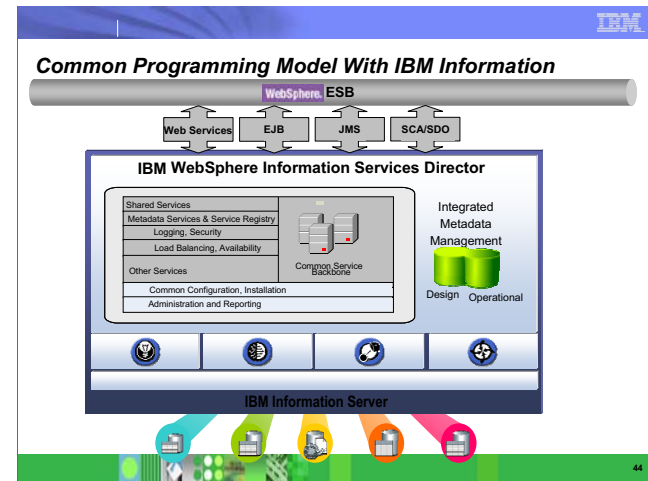


So the key is that there are multiple different roles involved in the typical development project, each creating and consuming different types of metadata. These roles typically work in vacuums, sharing information offline or through documents and spreadsheets. However, with the IBM Information Server, metadata is managed across these roles and functions. As each role creates new metadata, that metadata is immediately available to others working on the project. This is all managed through the IBM Metadata Server module within IBM Information Server.

This means that as a data analyst is profiling a source system, he can link the columns he finds to specific business definitions entered in Business Glossary by subject matter experts. As soon as that linkage is created, business users can see the link and understand how and where their business terms are implemented in various source systems. The key is allowing each role to actively share metadata with other roles, without forcing those other roles to use the same technologies – each role uses technologies with which they are comfortable, but each can see the whole universe of shared understanding that is pertinent to them.

This provides an enormous amount of downstream leverage within projects, dramatically reducing the time between specification and build, and reducing project cycle times. The metadata can be used to shorten the time it takes to design integration logic, and sometimes even to automate the creation of code.

It also creates an ongoing record of shared understanding that carries forward to future projects. The metadata itself becomes an asset that improves the overall understanding of the business, and allows projects to be executed more efficiently on an ongoing basis.



Summary

- Companies look into implementing SOA
- DB2 for z/OS is an important component in SOA
- Legacy DB2 for z/OS applications and Stored Procedures can be published as Web Services and can consume Web Services results
- XML is used to describe message format and to communicate between services

