Topics

1. Middleware contnd
2. Application Servers
3. Message Brokers
4. Business Process Orchestrators
5. Integration Architecture issues: reading for the student
6. Class exercise: Discussions & presentations
1. Middleware - RECAP

What is middleware?

Middleware:

- **provides proven ways to connect the various components** in an application so that they can exchange information using relatively easy-to-use mechanisms.

- **can be used:**
  - in a wide range of different application domains
  - to ‘wire’ together numerous components in useful well understood topologies:
    - one-to-one  one-to-many  many-to-many

- is completely hidden from the application user.
1. Middleware - Technology classification: **RECAP**

| Business Process Orchestrators | e.g. BizTalk, StaffWare, ActiveBPEL |
| Message Brokers                | e.g. BizTalk, WebSphere Message Broker |
| Application Servers            | e.g. J2EE, CCM (CORBA COMPONENT MODEL), .NET |
| Transport                      | e.g. Distributed Objects systems (CORBA), Message-Oriented Middleware |

- **Transport layer**: provides the basic services for sending requests & moving data between components. **LAST LECTURE**

- **Application servers**: **THIS LECTURE**
  - typically **built on top of the basic transport services**.
  - **Provide additional capabilities** e.g. transactions, security, directory services.
  - **support a programming model** for building multi-threaded server-based applications that exploit the additional services
2. Application Servers (1)

- An application server:
  - is a **component-based** server technology
  - that **resides is the middle-tier** of an N-tier architecture
  - and provides:
    - distributed communications, security,
    - transactions, persistence
  - Application servers are **widely used to build Internet-facing applications**

<table>
<thead>
<tr>
<th>Business Process Orchestrators</th>
<th>e.g. BizTalk, StaffWare, ActiveBPEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Brokers</td>
<td>e.g. BizTalk, WebSphere Message Broker</td>
</tr>
<tr>
<td>Application Servers</td>
<td>e.g. J2EE, CCM (CORBA COMPONENT MODEL), .NET</td>
</tr>
<tr>
<td>Transport</td>
<td>e.g. Distributed Objects systems (CORBA), Message-Oriented Middleware</td>
</tr>
</tbody>
</table>
Client tier:
- web browser submits http request to web server
- commodity technology, not an element of application server

Web tier:
- handles client requests
- invokes web server-hosted components (e.g. servelets, Java Server Pages: JSPs, Active Server pages: ASPs) as specified by in-coming request
- uses request pmts to call the business logic tier to get required info
- formats & returns results as html to client
2. Application Servers (3)

Business component tier:
- the core business logic for the application
- business components are realised as e.g. Enterprise JavaBeans (EJB) in J2EE, .NET components or CORBA objects
- receive requests from web tier, access one or more dbs & return results to Web tier
- use a container to accommodate components & supply services: transaction, security, multi-threading, etc.

Enterprise Information Systems tier:
- one or more DB and back-end applications e.g. mainframes
- business components query & interact with these data stores
2. Application Servers (4)

Enterprise JavaBeans & J2EE

- **EJB architecture** defines a standard programming model for constructing server-side Java applications.

- A J2EE-compliant application server provides an **EJB container** to manage the execution of application components.

- The container provides an operating system process (Java virtual machine) to host **EJB components**.

When an EJB client invokes a server component:
1. Container allocates thread &
2. Invokes an instance of the EJB component.
2. Application Servers (5)

Enterprise JavaBeans & J2EE contd

- **Two types of EJB components**
  - **session beans**
  - **entity beans**

- **session beans:**
  - encapsulate the business logic of a 3-tier architecture
  - used for executing business logic &
  - to provide services for client calls

- **entity beans:**
  - used for representing business data objects
  - data members map directly to data items in associated database
  - **accessed by session bean** that provides business level service to client

EJB Component model

- EJB container
- EJB pool
- Lifecycle management
- Persistence
- Connection pool
- Thread pool

Application Server
- Transaction service
- Directory service
- Security service

manages all resources on behalf of component & all interactions between component & external systems
2. Application Servers (6)

- Two types of EJB components
  - session beans, entity beans

- **session beans:**
  - encapsulate the business logic of a 3-tier architecture

- **stateless & stateful session beans**

- **Stateful:**
  - Used for operations that require multiple requests to be completed
  - Maintain data between requests

- **Stateless:**
  - Used for operations that can be performed in a single request
  - Do not maintain persistent data between subsequent requests from a client
2. Application Servers (7)

- **EJB component model** is widely used for constructing server-side applications.

- Code construction is not difficult, however:
  - EJB model makes it possible to combine application components using many different architectural patterns.
  - => wide range of design options. So, which one to choose?
  - Bean interaction with the container is complex.
  - => can have significant impact on application performance.

- **Alternatives to EJBs** are: Web Services, CORBA Component Model (CCM)

---

### EJB Component model

- **EJB container**
  - Lifecycle management
  - Persistence
  - Connection pool
  - Thread pool

### Application Server

- Transaction service
- Directory service
- Security service

### E.g. of products for application servers:

- WebLogic (BEA)
- WebSphere (IBM)
- .NET applic. server (Microsoft)
- JBoss (Open source)
2. Application Servers (8)

READING FOR THE STUDENT:

- EJB programming
- Deployment descriptors
- Responsibilities of the EJB container.

- Web reference for J2EE and EJB technology:
  http://java.sun.com/products.ejb/docs.html
  will re-direct to ORACLE website
- Current version is JEE 7 (Homework for the student)
3. Message Brokers (1)

<table>
<thead>
<tr>
<th>Business Process Orchestrators</th>
<th>e.g. BizTalk, StaffWare, ActiveBPEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Brokers</td>
<td>e.g. BizTalk, WebSphere Message Broker</td>
</tr>
<tr>
<td>Application Servers</td>
<td>e.g. J2EE, CCM, .NET</td>
</tr>
<tr>
<td>Transport</td>
<td>e.g. Distributed Objects systems (CORBA), Message-Oriented Middleware (MOM)</td>
</tr>
</tbody>
</table>

- **Message brokers:**
  - **use** either a **basic transport service (e.g. MOM)** or application servers or both & **add a message processing engine**.
  - **engine** provides high-level programming features for defining how to:
    - **exchange, manipulate, route messages** between application components
**3. Message Brokers (2)**

**RECAP:**

Message-oriented middleware MOM

**MOM basics:**

- Often **implemented as a server** that can handle messages from multiple concurrent clients.
- Senders & receivers use the same message format.
- No need to write code for message translation.
3. Message Brokers (3)

RECAP:

Publish-subscribe MOM

Publish-subscribe messaging:
- extends basic MOM mechanisms to support:
  - 1-to-many, many-to-many, many-to-one styles of communication
  - Topic is logical equivalent of queue in basic MOM
  - subscribers listen for messages sent to topics of interest
  - Again: senders & receivers use same message format

Publisher

Topic

Subscriber

Subscribe

register / subscribe

create / publish
3. Message Brokers (4)

<table>
<thead>
<tr>
<th>Business Process Orchestrators</th>
<th>e.g. BizTalk, StaffWare, ActiveBPEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Brokers</td>
<td>e.g. BizTalk, WebSphere Message Broker</td>
</tr>
<tr>
<td>Application Servers</td>
<td>e.g. J2EE, CCM, .NET</td>
</tr>
<tr>
<td>Transport</td>
<td>e.g. Distributed Objects systems (CORBA), Message-Oriented Middleware</td>
</tr>
</tbody>
</table>

- **So, why Message brokers?**
  - basic MOM & publish-and-subscribe MOM work well with most applications
  - **BUT:** become complex when message formats are not totally agreed upon.
  - Problem is common when creating applications from legacy systems that were never intended to be integrated.
  - Message brokers simplify this complexity
3. Message Brokers (5)

Application example:

- a web application that enables business customers to update personal info.
- the personal info is located in 3 different legacy systems
- message broker receives messages in one format ‘in-format’
- msg broker passes messages to each legacy system in unique format
- all code for message translation sits in message broker (loose coupling)
4. Business Process Orchestrators (1)

<table>
<thead>
<tr>
<th>Business Process Orchestrators</th>
<th>e.g. BizTalk, StaffWare, ActiveBPEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Brokers</td>
<td>e.g. BizTalk, WebSphere Message Broker</td>
</tr>
<tr>
<td>Application Servers</td>
<td>e.g. J2EE, CCM, .NET</td>
</tr>
<tr>
<td>Transport</td>
<td>e.g. Distributed Objects systems (CORBA), Message-Oriented Middleware</td>
</tr>
</tbody>
</table>

- **Business process orchestrators (BPOs):**
  - augment message broker features to support *workflow-style* applications
  - for *workflow-style* applications (e.g. ERP systems) business processes may take hours or days to complete due to the need for people to perform certain tasks.
  - BPOs provide tools to:
    - describe processes
    - execute the processes & manage intermediate steps
4. Business Process Orchestrators (2)

Example:

- Business systems involved in the process of a customer making an online purchase
- Business systems need to be accessed & updated
- Business processes are long running processes: involve human activities
- => long running transactions

A typical business process

- Customer purchasing via call centre
- Sales
- Accounts payable
- Credit validation
- Shipping
- Customer receiving
- Accounts receivable
- Oracle DB
4. Business Process Orchestrators (3)

BPO platforms:
- support the implementation of long-running, highly integrated business process activities
- Commonly build as a layer on top of a message broker
- extend message brokers by adding:
  - state management
  - development tools
  - deployment tools

Business process orchestration platform

Logic for:
state management, process description, transformations, data access
RECAP: Summary for Middleware

Middleware provides support for (some of):
- Naming, Location, Service discovery, Replication
- Protocol handling, Communication faults, QoS
- Synchronisation, Concurrency, Transactions, Storage
- Access control, Authentication

Middleware dimensions:
- Request/Reply vs. Asynchronous Messaging
- Language-specific vs. Language-independent
- Proprietary vs. Standards-based
- Small-scale vs. Large-scale
- Tightly-coupled vs. Loosely-coupled components
Assess the EJB component model in terms of the following middleware dimensions:

- Request/Reply vs. Asynchronous Messaging
- Language-specific vs. Language-independent
- Proprietary vs. (open) standards-based
- Tightly-coupled vs. Loosely-coupled components
References