NGE Solutions

*Building the Next Generation Enterprises*

# PISA

## (Planning, Integration, Security and Administration)

**An Intelligent Decision Support Environment for IT Managers and Planners**

## Sample Architecture Document

**Note**

This is a sample report that has been generated by the PISA environment. PISA generates many documents as a result of short (15 to 20 minutes) interviews. These documents are produced as html documents that can be easily modified by using MS Word (just open these documents in MS Word and edit them). For display only purposes, this document has been converted to PDF Format.

***NGE Solutions, Inc. (www.ngesolutions.com)***

# Application Architecture Document For

# --Order Processing

# General Template

**How This Report is Generated**

Architecture development is a very time consuming and expensive undertaking. This architecture document is generated by IAA (Intelligent Architecture Advisor) – a PISA Advisor that attempts to significantly reduce this time. IAA uses the following algorithm:

1. Fetch a template that includes standard architecture templates and consists of several sections which contain fixed content that is always in an architecture document (e.g., project information), application specific information, and scenario (enterprise model) specific information.
2. Architecture specific information is represented in architecture patterns (APs) that consist of "knowledge chunks" (e.g., overview paragraphs and diagrams) about an application (e.g., inventory management). The design approach used in PISA heavily relies on a pattern repository (PR) which houses these patterns.
3. Many sections of the architecture template are populated by different components of the pattern repository; others are populated by creating scenario specific models based on a user interview.

The objective of IAA is to generate a template that has 70% of the information already populated. IAA is in its early stages of development.

# Definitions and Terms

== Modify as needed

```
ACL Authorized Control List
ACM Association of Computing Machinery
AI Artificial Intelligence
AIA Application Integration Architecture
API Application Programming Interface
ASP Application Service Provider
ASP Active Server Pages A Microsoft technology for
      building server side code
ATM Asynchronous Transfer Mode a packet switching
      Technology used typically in high data rate networks
ATM Automatic Teller Machine used in banking
B2B Business to Business
B2C Business to Consumer
B2E Business to Employee
B2G Business to Government
BREW Binary Runtime Environment for Wireless
BSP Business System Planning
CAD Computer Aided Design
CAM Computer Aided Manufacture
CBX Computerized Branch Exchange
CCITT Comit Consultatif Internationale de Tlgraphique et Tlphonique
            (The International Telegraph and Telephone Consultative
            Committee)
CDMA Code Division Multiple Access
CDPD Cellular Digital Packet Data
CGI Common Gateway Interface - A Web gateway technology
CIO Chief Information Officer
CORBA Common Object Request Broker Architecture
COTS Commercial Off-The-Shelf
CPU Central Processing Unit
CRM Customer Relationship Management
CSF Critical Success Factors
CSMA/CD Carrier Sense Multiple Access/Collision Detect
DBMS Database Management System
DCOM Distributed Component Object Model
DDBMS Distributed Database Management System
DDL Data Definition Language used in database
      management
DDTMS Distributed Data and Transaction Management System
DML Data Manipulation Language
DOD Department of Defense
DSL Digital Subscriber Loop
DTM Distributed Transaction Manager
DTMS Distributed Transaction Management System
EAI Enterprise Application Integration
EB Electronic Business
EC Electronic Commerce
EDI Electronic Data Interchange
EJB Enterprise Java Beans
```

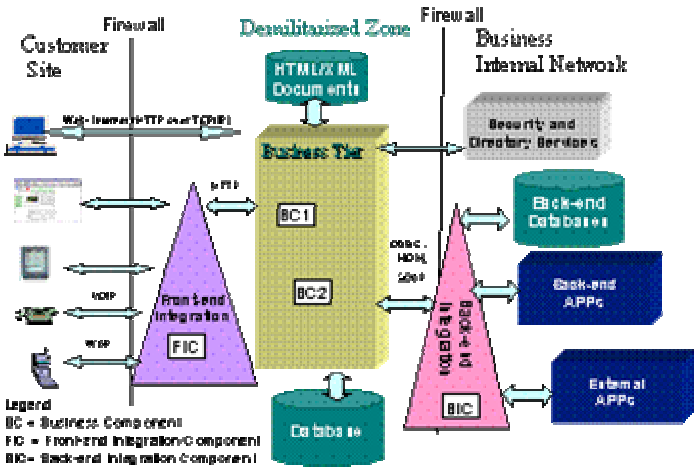
ERP Enterprise Resource Planning
ETSI European Telecommunication Standards Institute
FCC Federal Communications Commission
FDDI Fiber Distributed Data Interface
FDM Frequency Division Multiplexing
FSO Free Space Optics
FTP File Transfer Protocol
GUI Graphical User Interface
I/O Input/Output
IDL Interface Definition Language used in CORBA and other distributed object middleware services
IEEE Institute for Electrical and Electronic Engineers
IMS Information Management System - IBM DB/DC system on mainframes
IP Internet Protocol
IPC Interprocess Communication
IRM Information Resource Management a management methodology
ISDN Integrated Services Digital Network
ISO International Organization for Standardization
ISP Internet Service Provider
IT Information Technology
ITU International Telecommunications Union
ITUT International Telecommunications Union Telecommunications Services Sector
J2EE Java Version 2 Enterprise Edition
J2ME Java Version 2 Mobile Edition
JDBC Java Database Connectivity
LAN Local Area Network
LDBMS Local Database Management System
LLC Logical Link Control
LMDS Local Multipoint Distribution Service
LU Logical Unit - an endpoint in the IBM SNA environment
MAC Medium Access Control
MAN Metropolitan Area Network
Mbps Million bits per second
MMIT Microsoft Mobile Internet Toolkit
MOM Message Oriented Middleware
MVS Multiple Virtual System - operating system on IBM's mainframes
NBS National Bureau of Standards
NFS Network File Services - SUN Microsystem's File System for Networks
NIST National Institute of Standards and Technology
OAG Open Application Group a standards organization
ODBC Open Database Connectivity a de-facto standard for remote SQL
OMA Open Mobility Alliance
OMG Object Management Group the group that developed CORBA
OODBMS Object-Oriented Database Management System
OOPL Object-Oriented Programming Language
OS Operating System
OSF Open Software Foundation
OSF-DCE OSF Distributed Computing Environment
OSF-DME OSF Distributed Management Environment
OSI Open System Interconnection

PBX Private Branch Exchange
PGP Pretty Good Privacy
PKI Public Key Infrastructure
QoS Quality of Service
QPSK Quadrature Phase Shift Keying
RDA Remote Database Access
RFID Radio Frequency Identification
RPC Remote Procedure Call
SCM Supply Chain Management
SET Secure Electronic Transaction a security standard
SNMP Simple Network Management Protocol - TCP/IP Network management Protocol
SOAP Simple Object Access Protocol part of Web Services
SONET Synchronous Optical Network
SQL Structured Query Language
SSL Secure Socket Layer
TCP Transmission Control Protocol
TCP/IP Transmission Control Protocol/Internet Protocol
UDDI Universal Description, Discovery and Integration - a registry for Web Services
UDP User Datagram Protocol - a protocol that runs on IP
UMTS Universal Mobile Telecommunication System (Mainly 3G Cellular Technology)
UWB Ultra Wideband
VAN Value-added Network
VPN Virtual Private Network
VXML Voice eXtensible Markup Language
WAN Wide Area Network
WAP Wireless Application Protocol
WLL Wireless Local Loop
WML Wireless Markup Language
WS Web Services
WSN Wireless Sensor Network

# 1. OVERVIEW -- ORDER PROCESSING

This architecture document is based on an interview with AIM. It uses a service oriented architecture (SOA) based on components that provide these services. The componnts consist of the following (see the diagram)

- **BCs** (Business Components) that imbed the business logic of the application and provide business services. At present, we are assuming one BC per application (you can modify it, if you wish)
- **FICs** (Front-end Integration Components), also known as user integration components, that allow different types of user devices (e.g., mobile, handheld) to invoke the BCs.
- **BICs** (Back-end Integration Components) that BCs to interact with different back-end and external applications.




## *Overall Integration Strategy*

**The Suggested Straetgy is:**
ACCESS IN PLACE because you require access to misture of resources with varying capabilities
**Advantages:**
-Does not modify the existing applications thus reducing the risk of failures
-Leverages the current investment in applications
-The new applications can access the most recent data that is created/stewarded by the existing apps.

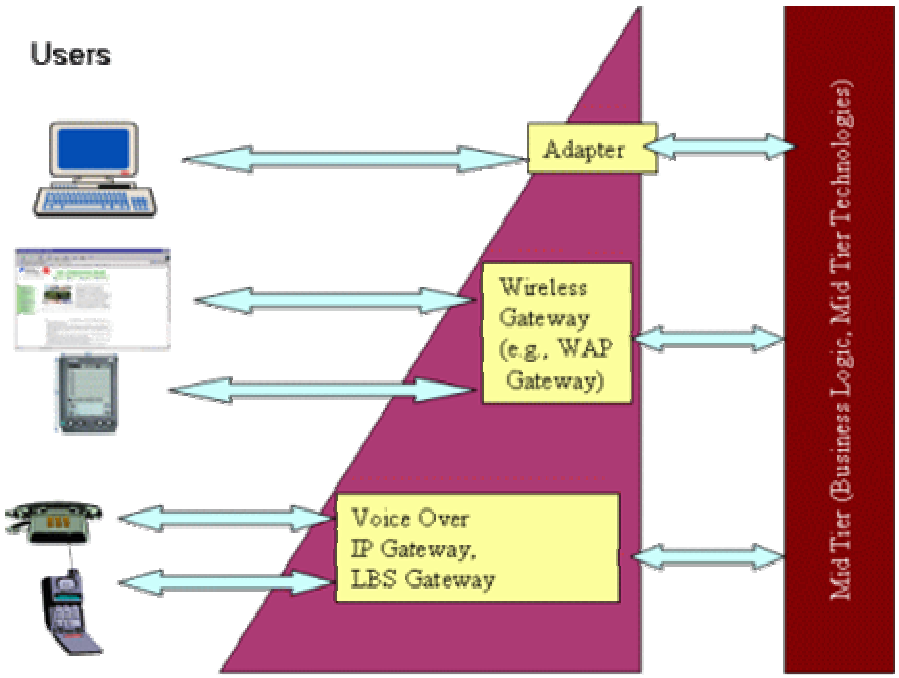
**Disadvantages:**
-Tends to overload the existing applications and could thus cause performance problems
-Only surrounds the old systems, does not replace them. Thus does not change the target applications that are too inflexible and expensive to maintain
-

# 2. FRONT-END INTEGRATION

Different type of user devices require different level of front-end integration

- Regular Web browsers do not need any integration because they directly invoke a web server
- Wireless devices need a wireless gateway for front-end integration
- Voice Over IP (VOIP) devices need VOIP gateways and location based services (LBS) require LBS support such as GPS

Note: The thickness of integration layer indicates the complexity and cost of integration technologies




**Suggestions:**
-Web Adapter Suggested:Use Web access over http - Public Internet Access
-Wireless Adapter Suggested:Use WAP gateway and develop WML applications for Web

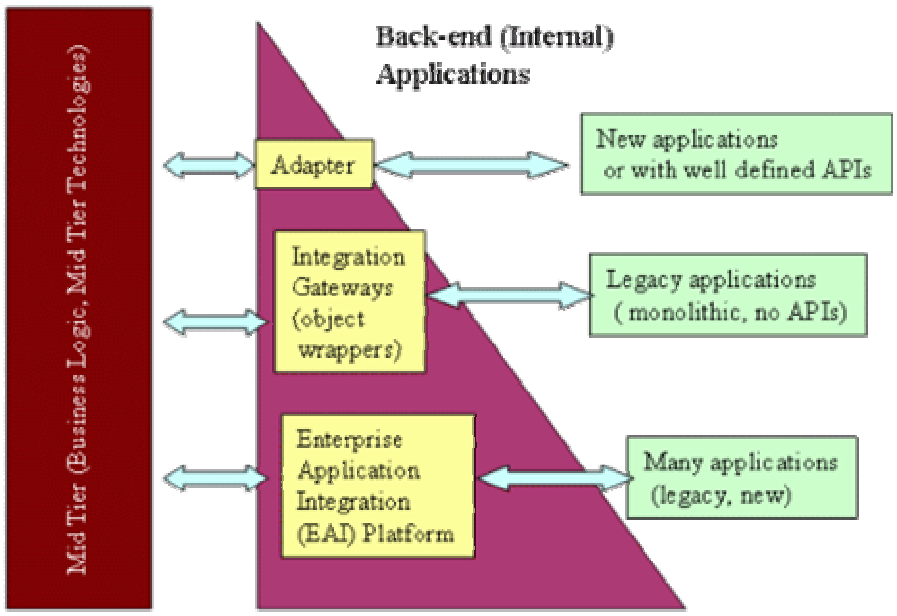
surfing over cellular networks
-LBS Adapter Suggested:Nothing needed
-Program Access Adapter Suggested:Use XML access over HTTP
-Other Considerations:Many individual adapters can be componed into one large FIC

# 3. BACK-END INTEGRATION WITH INTERNAL APPLICATIONS

Different type of applications require different level of back-end integration

- Simple adapters can be used for new applications or with well defined APIs
- Wrappers/gateways are commonly used for legacy applications
- EAI platforms are typically used to integrate many applications

(API = Application Programming Interface)
Note: The thickness of integration layer indicates the complexity and cost of integration technologies.




Backend Application name = **app1**

**Suggestions**
- Suggested Backend Integration Adapter : ODBC/JDBC
-Other: Use data warehouse that contains all the needed data, especially if you need read only data
-Other: No data translation needed for back-end applications

Backend Application name = **app2**
**Suggestions**
- Suggested Backend Integration Adapter : MOM/SOAP
-Other: Use data warehouse that contains all the needed data, especially if you need read only data
-Other: No data translation needed for back-end applications

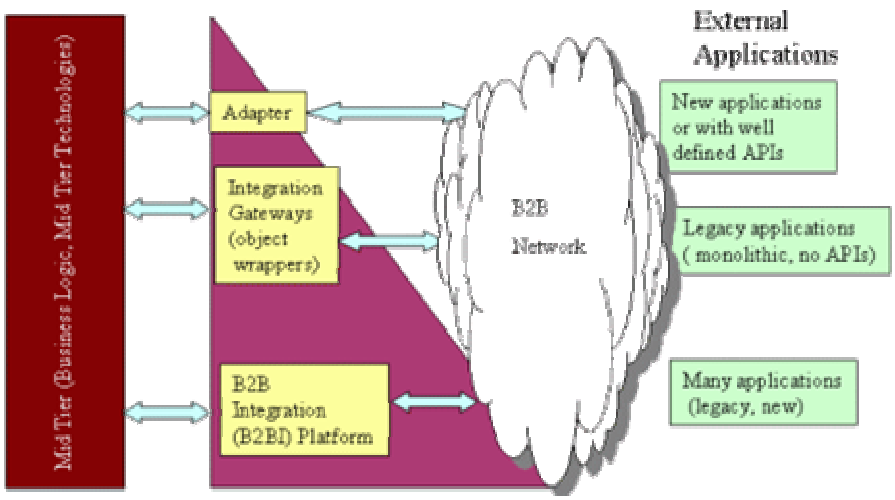
Backend Application name = **app3**
**Suggestions**
- Suggested Backend Integration Adapter : screen scraper
-Other: Use data warehouse that contains all the needed data, especially if you need read only data
-Other: No data translation needed for back-end applications

# 4. B2B INTEGRATION

Different type of applications require different level of external integration over a B2B network

- Simple adapters can be used for new applications or with well defined APIs Wrappers/gateways are commonly used for legacy applications
- B2B platforms are typically used to integrate many applications
- More security and privacy is needed for B2B trade

API = Application Programming Interface
Note: The thickness of integration layer indicates the complexity and cost of integration technologies




External Application name = **app5**
**Suggestions**
-- Suggested Backend Integration Adapter : ODBC/JDBC
--Other: Use a B2B Integration Pltaform that can provide different type of adapters/warppers )
--Other: No data translation needed for external applications

External Application name = **app6**
**Suggestions**
-- Suggested Backend Integration Adapter : MOM/SOAP
--Other: Use a B2B Integration Pltaform that can provide different type of adapters/warppers )
--Other: No data translation needed for external applications

External Application name = **app7**
**Suggestions**
-- Suggested Backend Integration Adapter : screen scraper
--Other: Use a B2B Integration Pltaform that can provide different type of adapters/warppers )
--Other: No data translation needed for external applications

# 5. BUSINESS COMPONENT DESIGN AND APPLICATION PATTERNS

## Information Retrieval

### Sample Service Definitions

Service definitions are at the foundation of modern system architectures such as Service Oriented Architectures (SOAs) and component-based architectures.

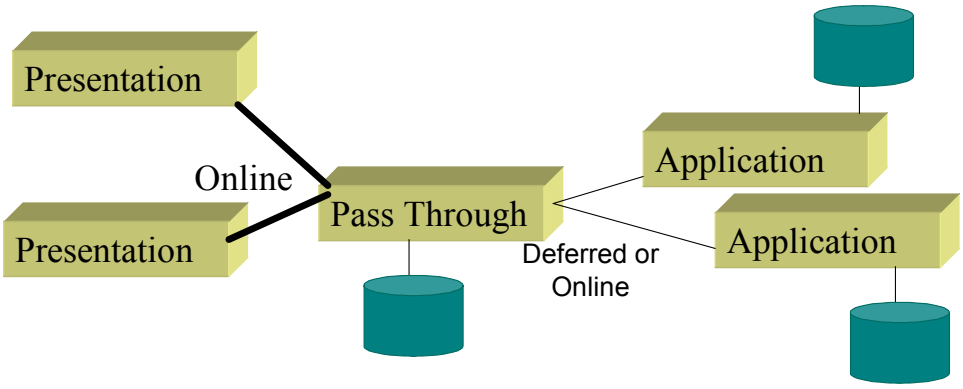
A service definition basically defines a business service such as catalog service and the operations such as read-catalog supported by the service. These service definitions can be defined in plain English first and then translated into formal definition languages such as WSDL (Web Services Definition Language) and IDL (Interface Definition Language).

The following is an informal service definition to get you started. You should refine and modify it as needed.

Service: Information Retrieval

1. Operation: support static content in HTML and XML documents
2. Operation: support dynamic content through gateways that can dynamically generate Web content from non-Web resources (e.g., relational databases).
3. Operation: support dynamic content
4. Operation: provide appropriate sharing level (dedicated versus shared machine)
5. Operation: support backup/recovery, site security, site administration, hotline, etc.

### Application Pattern 3: Links multiple presentation tiers to any back-end client, but the backend is not hidden from the user




Key Idea: Links multiple presentations to multiple back-end systems through a middle tier (middle tier does not hide back-end)

**Key Features**

This architecture supports one-to-one-to-one relationship between the front-end user and back-end llegacy applications in a pass-throgh manner. The business driver for this design is fast, highly scalable, highly available Web-enablement of existing business transactions. Each application stands on its own, with no need to combine information or features from multiple applications into a single response to the user. For example, a business could provide access to an order entry application from customer server representatives in a call center. At the same time, it could provide access to inventory data to suppliers using an intranet connection.

**Considerations**

A presentation node is linked to a particular application node through the application router at any given time. The middle tier serves as a router and provides a common interface to multiple back-end applications, facilitating the connection but containing no business logic to combine the application data. You can use local read/only data to facilitate the routing decisions. In a slight variation, the router node can do more with the local data, using it as a temporary cache for frequently requested data, work in progress, or staged data.

The communication between the presentation logic and business logic layer is synchronous, meaning that any request coming from the user interface invokes business logic on the application node using the router. After the executing the business logic, control is passed back to the presentation node that uses the results to update the user interface. The connection between the router node and the application nodes can be a fast asynchronous or synchronous connection. The communication type depends on the communication characteristics and capabilities of the back-end system.

Traditional transaction monitors have been typically used to support the application router on the middle tier. These may still be appropriate unless you have a requirement to move rapidly to application topology 6 and beyond. For this level of functionality more advanced middleware is required, for example, WebSphere Enterprise Edition or decomposition/recomposition support anticipated in future releases of MQSI.

**Example (IBM)**

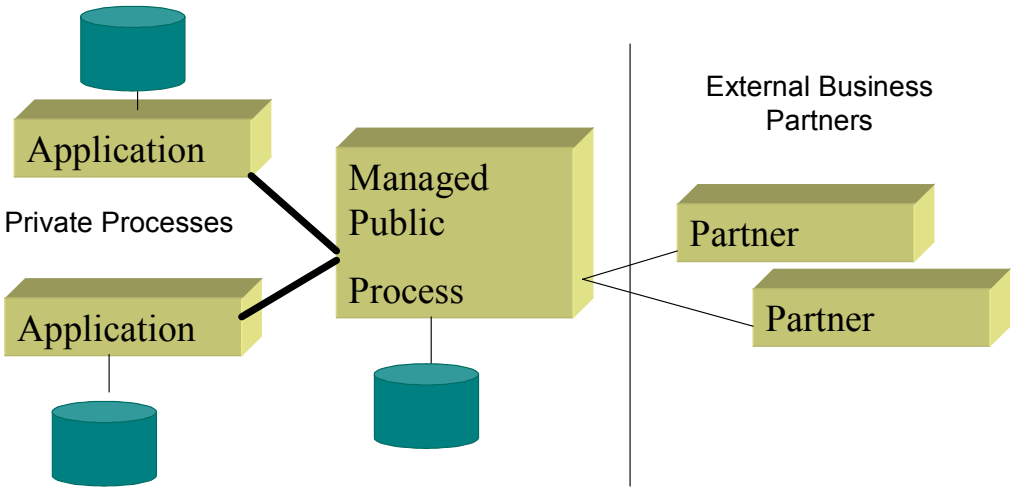
A banking company purchases a mortgage company. Currently, the bank's customer representatives use

Web transactions to access back-end financial data using a topology 2 solution. The mortgage company

has a separate Web site its representatives use for submitting and reviewing mortgage applications.

They also use a topology 2 solution. To reduce expenses and allow easy access for the customer representatives to both types of data, the company wants to merge the two Web sites. A topology 5 solution is chosen initially, because it would allow the two sites to merge with little or no changes required for the existing applications.

## Application Pattern 4: business logic in the intermediate tier




Key Idea: In this case, a business process is jointly shared by partners through a public process that implements B2B rules such as "if out of stock, initiate back-order".

**Key Features**

This architecture supports fast, highly scalable, highly available Web-enablement of existing business transactions. This architecture is similar to architecture 3, however, some customiztaion and business logic is included in the middle tier. This allows you to hide the back-end application interfaces from the users to facilitate ease of use and to provide a single point of entry into the applications. Information or features from multiple applications need to be combined into a single response to the user.

**NEXT STEP: For a more technical view of this application architecture, with infrastructure components, and continue** click here

**Considerations**

There are a number of possible approaches to doing business request decomposition into multiple back-end transactions and recombining the responses into a single business response. These include:

- RYO (roll-your-own) programming that issues multiple asynchronous requests to back-end systems and combines the responses
- Using a message broker plus a rules engine, possibly with a two-phase commit (2PC) or compensations mechanism
- Using a component broker with 2PC protocols

**Example**

The banking company mentioned in previous architecture that purchased a mortgage company now wants to offer its customers the ability to make mortgage payments directly from their checking or savings account. A topology 6 solution can let users take money from their accounts and make mortage payments. The application was designed to ensure that there is enough money in the account to cover the mortgage payment. It also was designed to ensure that if something goes wrong during the transaction, any changes made to either database are backed out.