

Cedar

BlueStar Multi-Channel Solution Overview

Cedar Software

November, 2001

Table of Content

- EAI Framework
- Integrating Financial Channels
- BlueStar Overview
- BlueStar Architecture
- Summary

EAI Means Enterprise Application Integration

EAI deals with different enterprise integration relationships as follows:

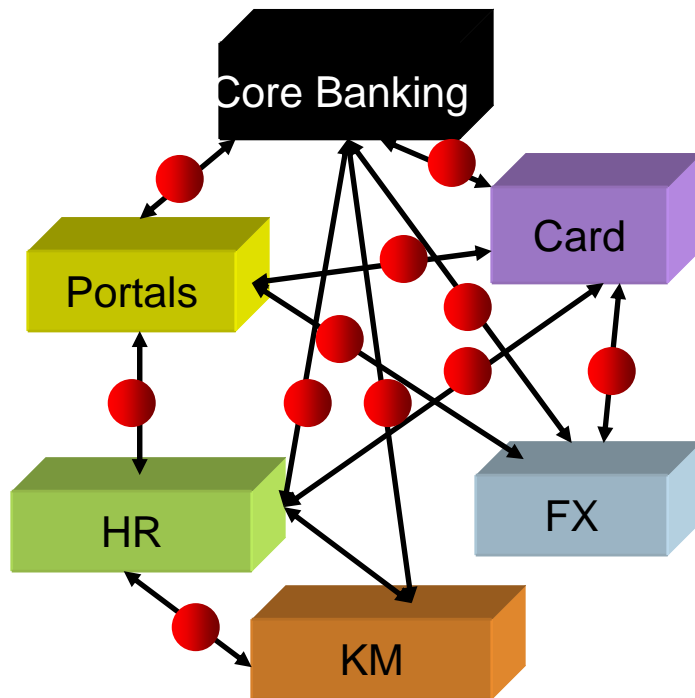
- a2a: Integration of **back-end** systems (Foreign Exchange-->Legacy, etc.)
- a2a: Integration of **front-end** to **back-end** systems (Branch<->ERP)
- b2c: Integration of **web applications** with **front** and **back-end** systems (self service, customer interactions, etc.)
- b2b: Integration of **applications across corporate or organizational boundaries** (integration of business processes between trading partners, suppliers, distributors, etc.)



Traditional Integration Approach

The traditional integration between enterprise systems focuses on *custom* point-to-point interfaces. Such integration will run into following challenges.

Point to Point Custom Code



- **Limited reusability** due to tight coupling of integrated applications via point-to-point interfaces
- **Invasive integration approach** requiring modifications to source applications
- **Change** to one application **can affect all interfaces** to/from that application
- **Limited scalability** across the enterprise
- **No centralized management** or visibility of information flows and business rules

A set of state-of-art technologies that enables the integration of end-to-end business processes and information across disparate applications can increase the organization's ability to respond and adapt to change by providing the following services:

- Business process management
- Application connectivity
- Translation and transformation
- Communications middleware and message routing



BlueStar Accomplished via EAI

EAI creates a standard environment in which information is communicated in a consistent format, which all applications understand, thus enabling “real-time” “end-to-end” “hub-based” business process integration – the BlueStar.

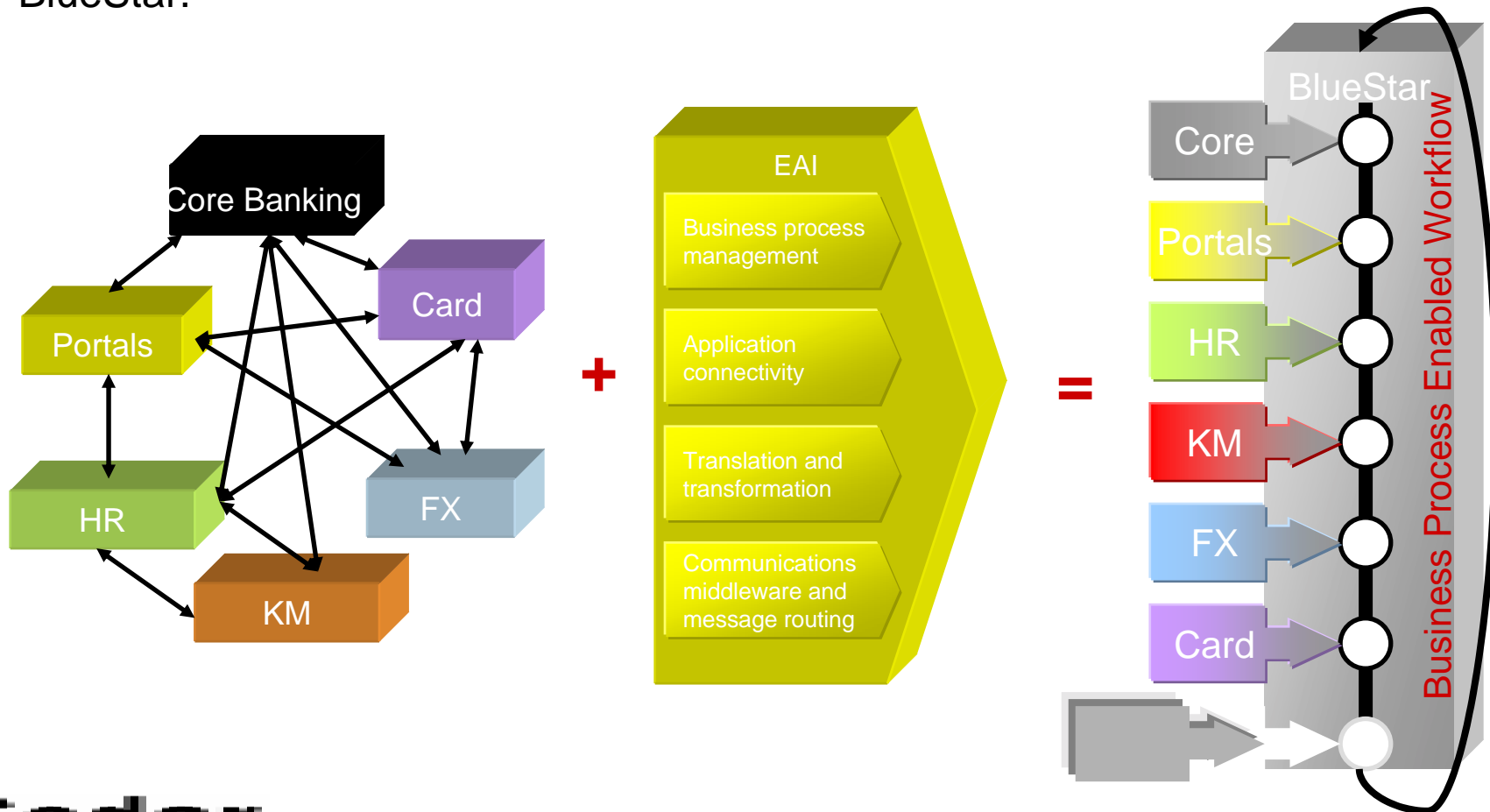


Table of Content

- EAI Framework

- Integrating Financial Channels

- BlueStar Overview

- BlueStar Architecture

- Summary

What benefits do you expect to gain from channel integration?

- Better customer service
- Lower costs
- Consolidated view of client
- Better customer self-service
- Ability to remain competitive
- Better marketing
- Single platform capability

Today's disparate and decentralized efforts won't scale and will keep firms on the integration treadmill --supporting proprietary systems and custom legacy code. As a result, their efforts are doomed to fail. Why?

- Mass customization
- Mismatched skills
- Uncoordinated projects

Developers should use a combination of standards-based integration server and application server as a standard app deployment platform. It will provide:

- A common platform as apps
- Centralized integration of traffic control
- Hooks into any financial app or legacy database
- Support for financial-specific protocols

Process tool give developers a common understanding of how information must flow between people and system, and they will enable integration teams to:

- Simplify process design
- Eliminate process bottlenecks
- Make process hand-free
- fill gaps with best-of-breed technologies (for example, automatic data mapping to XML standards like IFX)

- **Performance**
 - Load balancing
 - Distributed session management
 - Fault tolerance
- **Integration**
 - Pre-built adapters and gateways to enterprise apps and databases
- **Development**
 - Component modeling
 - Object-relational mapping tools
 - Object repositories
- **Momentum**
 - Multilanguage support

- **Performance**
 - Bus architecture
 - Clustering for components
- **Development**
 - Connector kits to build link to apps with nonstandard APIs
 - Parameter-driven for application development
- **Process management**
 - Graphical tools for modeling and modifying process
- **Momentum**
 - A proven based of customers in financial services

Table of Content

- EAI Framework
- Integrating Financial Channels
- BlueStar Overview
- BlueStar Architecture
- Summary

Who feels it the most?

- The Large Enterprise
 - Thousands of users and servers
 - Complex applications and processes
- The Application Service Provider
 - Hundreds to thousands of servers, acres of racks
 - Bring the user online = making the b2b trade through internet

What is BlueStar?

- BlueStar is a system for automating complex EAI hub for financial services.
- BlueStar is not:
 - Base Machine Installs
 - gateway middleware
 - Message middleware
 - Process Integration
- Can BlueStar “play” in the above?
 - Yes.

- A single platform for integrating enterprise application.
- Channel integration.
- Improved speed of service – one-stop shopping. Human productivity.
- Delegated administration – pull the LOB admin or end-user into an enterprise process.
- Solution designed for growth and scaling.
- Potential for headcount savings.

BlueStar Key Features

- Parameter driven (SQL)
- XML enabling
- Role based security
- BlueStar “engines” do the work.
 - BizTalk for complex long duration workflow.
 - Compensating operations.
- Scripting philosophy
- Programmatic Interface
- Completely auditable
- 100% COM+ component technology
- Session management
- Multilanguage support
- Data mapping between flat-file, EDI and XML
- Dynamic load-balancing
- Automatic recovery
- Transport neutral
- Support synchronous, asynchronous and transactional connections
- Automatic code page conversion
- Build on top of Microsoft .NET Servers (W2K, SQL2K, BTS and HIS)

Key Features – Standards-Based support

- File format
 - XML and XML Schema
 - Flat file
 - EDI
 - Binary
- Protocol support
 - FTP, SMTP, POP3
 - HTTP, HTTPS, Web Services
 - SNA (LU0, LU2, APPC/LU6.2, TN3270, TN5250, AS/400 Data Queue)
 - TCP/IP socket
 - MSMQ
 - MQSeries
 - DCOM
 - X.25
- DES, RSA, OCSP, PKCS#7 and W3C DSIG (Smart Card, HSM)
- Code page support
 - Unicode (UTF8, UTF16)
 - IBM EBCDIC
 - Unisys mainframe Traditional Chinese
 - ...any code page support from Windows 2000 and HIS 2000

Table of Content

- EAI Framework
- Integrating Financial Channels
- BlueStar Overview
- BlueStar Architecture
- Summary

Business Schema – BizTalk WebDAV

The screenshot displays the BizTalk Editor interface for the file TWFX11.xdr.xml. The left-hand tree view shows a hierarchy starting with 'IFX', containing various request and response elements. The 'PntAddRq' element is expanded, and 'Tw_ORG_BA_MsgId' is selected. The right-hand pane shows the 'Declaration' tab for this element, displaying a table of properties.

Property	Value
Name	Tw_ORG_BA_MsgId
Description	
Type	Element
Model	Closed
Content	Text Only
Data Type	String
Data Type Values	
Minimum Length	6
Maximum Length	6

Document Transformation - BizTalk Mapper

The screenshot displays the BizTalk Mapper interface for a document transformation project. The main window is titled "BizTalk Mapper - CHTING_Rq_to_CTCBML_CHTBillingRq_Mapper.xml".

Source Specification: FNS_MC_CHTING_Rq

- MsgId
- RqUID
- DateTime
- ChannelId
- TerminalId
- OfficeId
- TELNo
- AccountingType

Destination Specification: CTCBML

- version
- SignonRq
- SignonPwd
- SignonRole
- CustId
- CustPwd
- ClientDt
- AccountingType
- CHTSvcRq
- version
- RqUID
- SPName
- CHTBillingRq
- MsgId

Source Attributes:

Property	Value
Name	FNS_MC_CHTING_Rq
Description	CHT Bill Inquiry Request
Type	Element
Model	Closed
Content	Element Only
Specification Name	FNS_MC_CHTING_Rq
Standard	CUSTOM
Standards Version	
Document Type	
Version	1.0
Default Record Delimiter	CR (0xd)
Default Field Delimiter	
Default Subfield Delimiter	
Default Escape Character	
Code Page	Traditional-Chinese-Big5 (950)
Receipt	
Envelope	No
Structure	Positional
Source Tag Identifier	
Source Tag Position	

Destination Attributes:

Property	Value
Name	CTCBML
Description	中國信託商業銀行 MultiChannel Markup Language
Type	Element
Model	Closed
Content	Element Only
Order	Sequence
Specification Name	CTCBML
Standard	XML
Standards Version	
Document Type	
Version	0.9
Receipt	
Envelope	
Target Namespace	urn:schema:chinatrust-com:multichannel

Deletes page and all objects placed on it.

Configurable UI – Parameter Driven for message design

The screenshot shows a 'Message Schema' dialog box with the following configuration:

- Message Name: SOCKMSG
- Message Type: ASCII
- Message Header: (empty)
- Transport PorgID: PoolSocket.MySocket
- Receive Multiple:
- Length: 88

Request Message Schema:

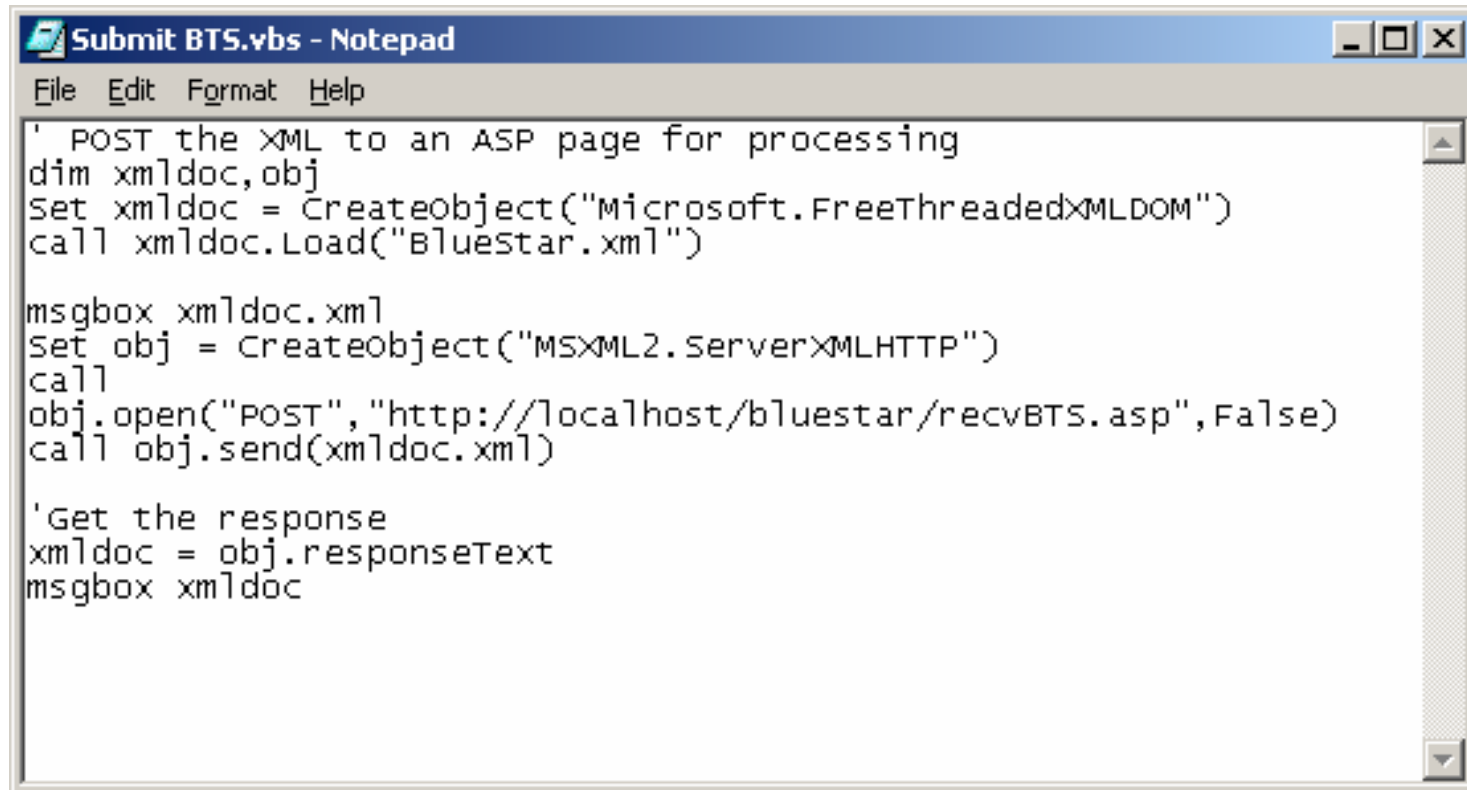
- Len=11:Del=46
- Len=16:Del=46
- Len=16:Del=46
- Len=9:Fill=0:Align=Right:Del=46
- Len=1:Del=46
- Len=1:Del=46
- Len=3:Del=46
- Len=6:Del=46
- Len=16:Del=46

Response Message Schema:

- Len=11:Del=1
- Len=16:Del=1
- Len=16:Del=1
- Len=9:Del=1
- Len=1:Del=1
- Len=1:Del=1
- Len=3:Del=1
- Len=6:Del=1
- Len=16:Del=1

Buttons: New, Edit, Delete (for both schemas), Save

- Script Host components for application integration
 - Host the execution of VB/PERL scripts.



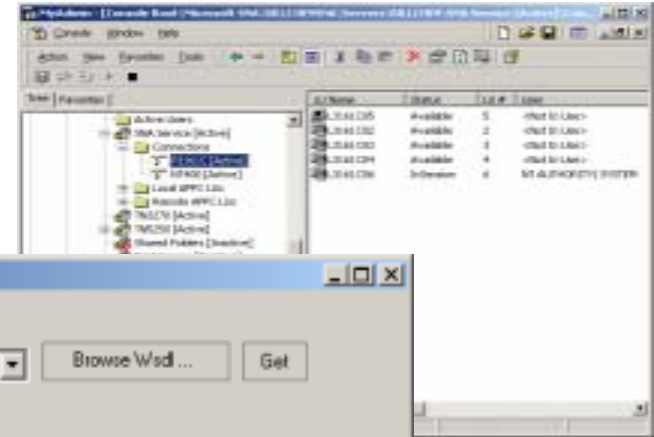
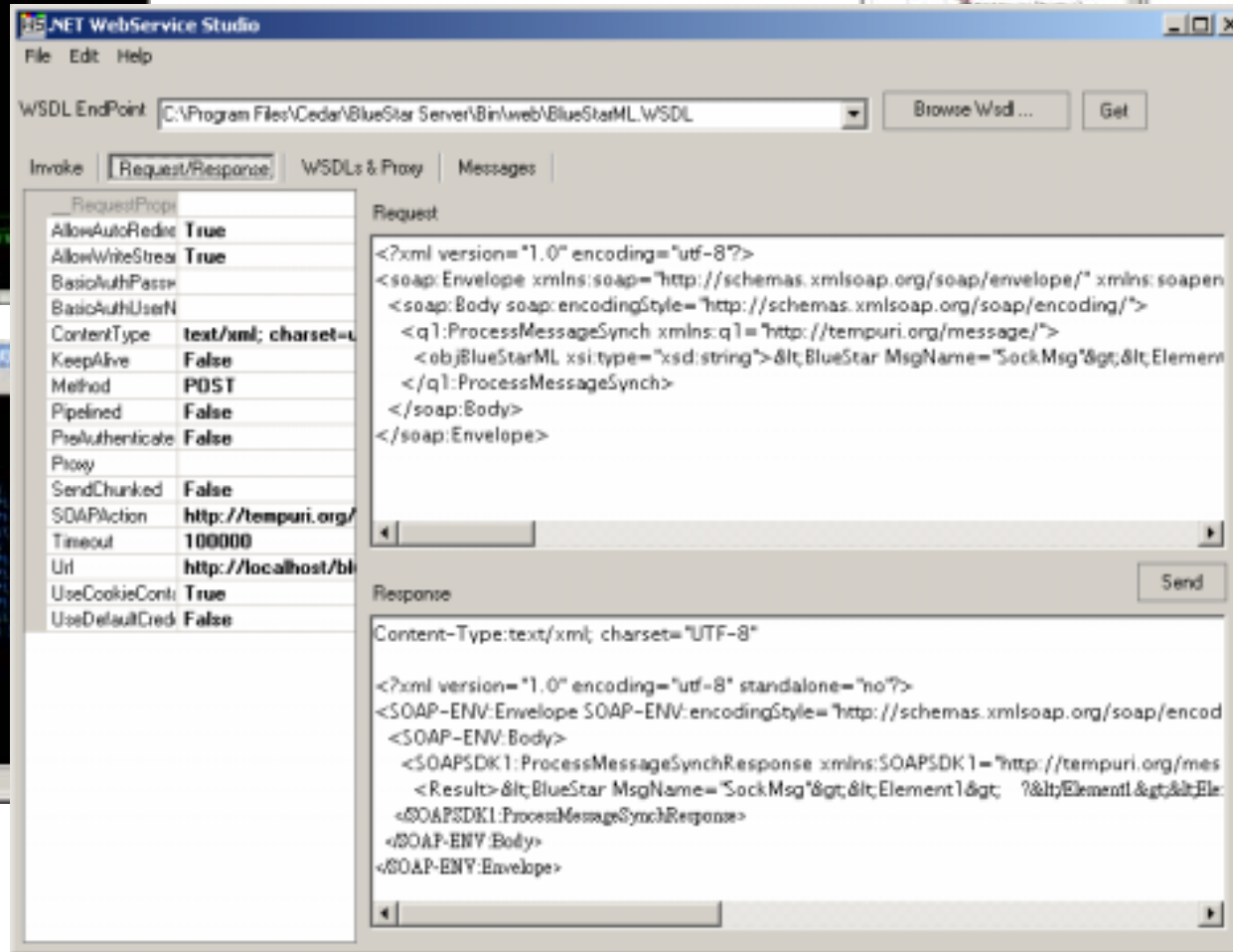
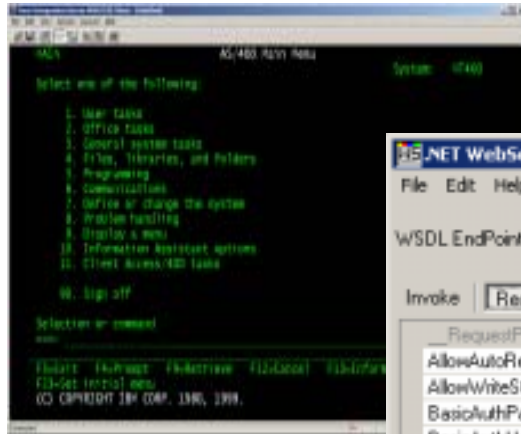
```
Submit BTS.vbs - Notepad
File Edit Format Help
' POST the XML to an ASP page for processing
dim xmlDoc,obj
Set xmlDoc = CreateObject("Microsoft.FreeThreadingXMLDOM")
call xmlDoc.Load("Bluestar.xml")

msgbox xmlDoc.xml
Set obj = CreateObject("MSXML2.ServerXMLHTTP")
call
obj.open("POST","http://localhost/bluestar/recvBTS.asp",False)
call obj.send(xmlDoc.xml)

'Get the response
xmlDoc = obj.responseText
msgbox xmlDoc
```

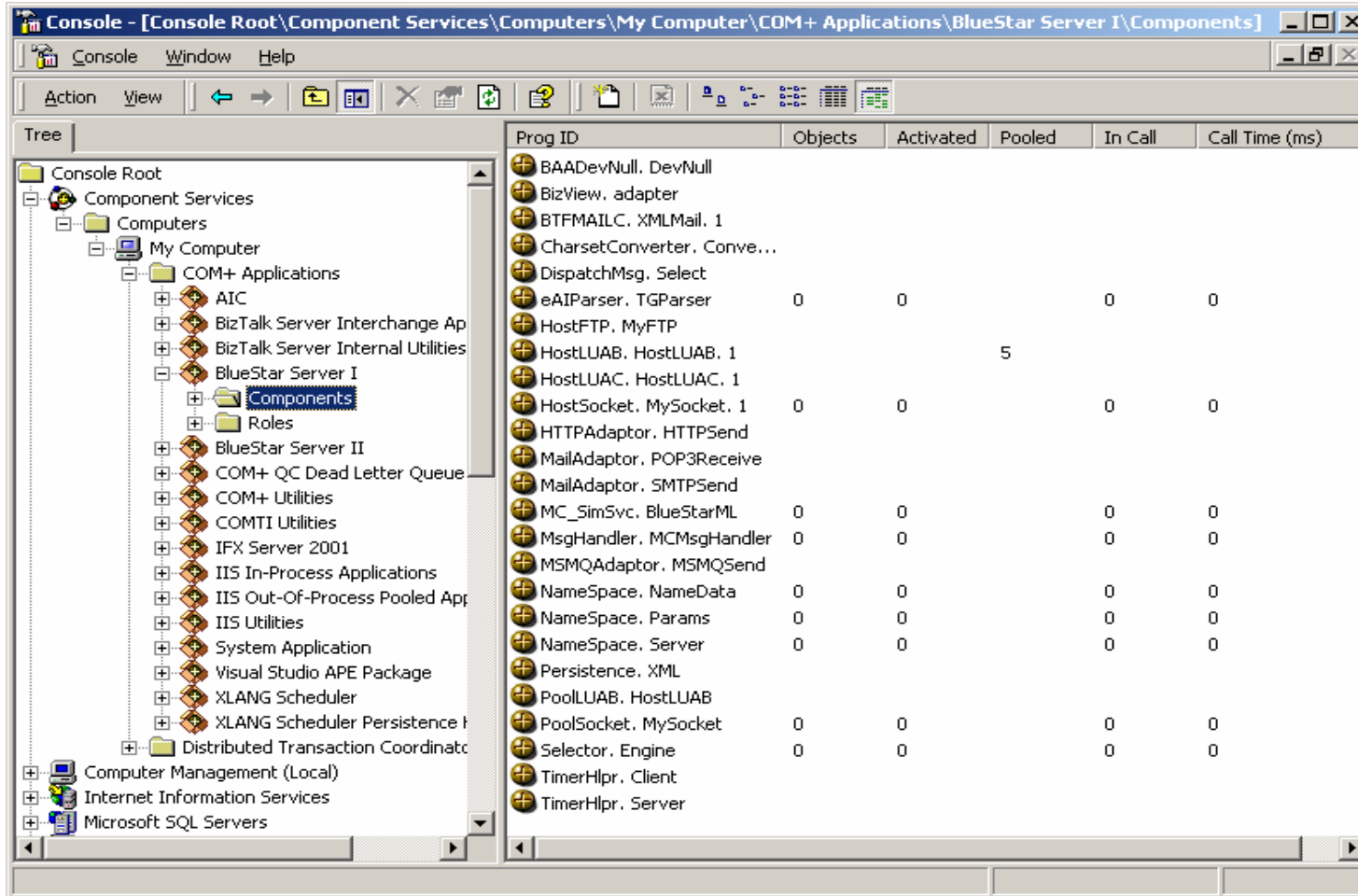
Web Services

- Web Services for application integration
 - Host Integration through Web Services.



Object Pooling - Dynamic host session management

- Using COM+ Object Pooling for application connection
 - Provide the dynamic session management feature
 - Cache the host connection in memory

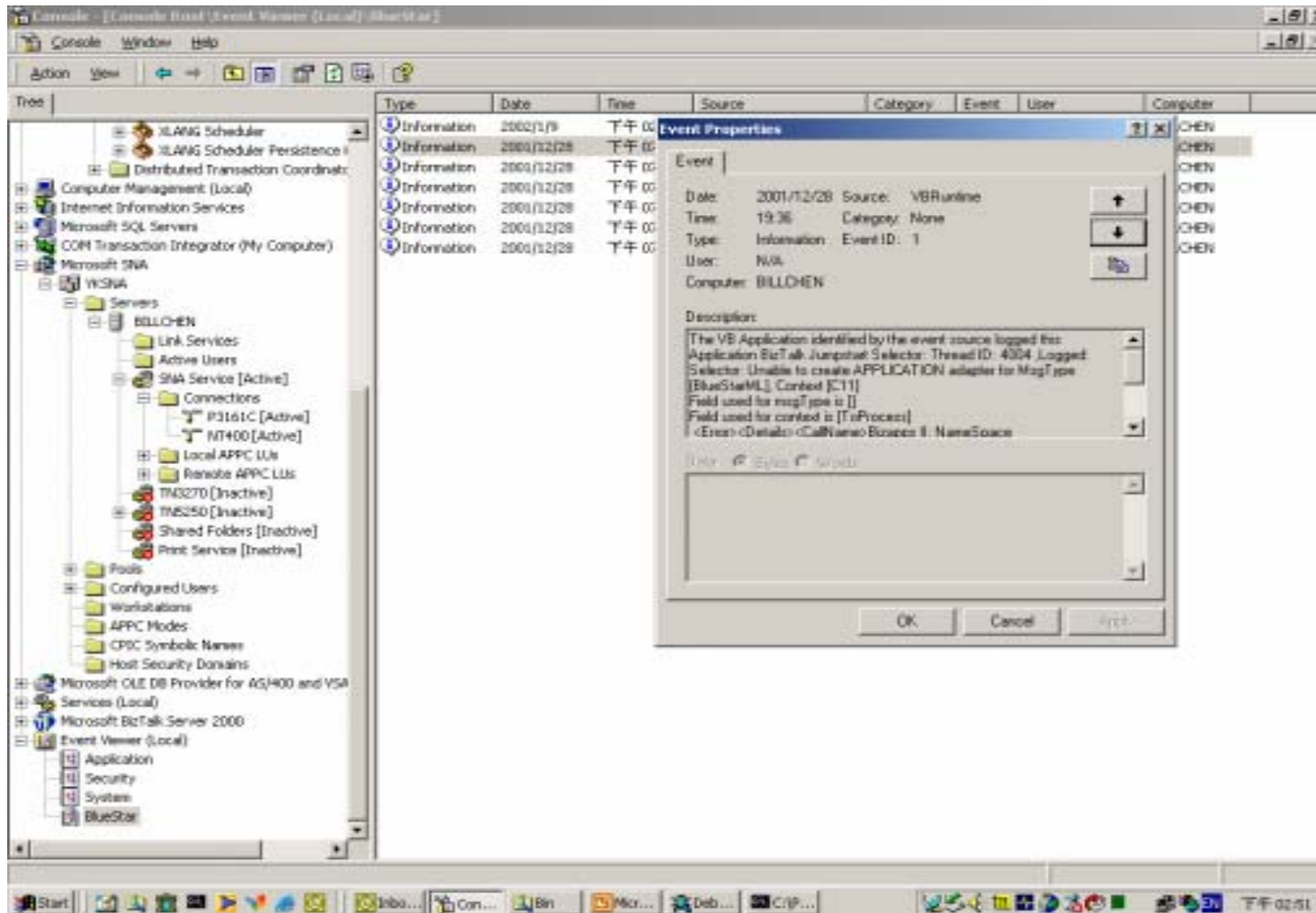


The screenshot shows the Windows Console window with the following path: Console Root > Component Services > Computers > My Computer > COM+ Applications > BlueStar Server I > Components. The 'Components' folder is expanded, and a table displays the following data:

Prog ID	Objects	Activated	Pooled	In Call	Call Time (ms)
BAADevNull, DevNull					
BizView, adapter					
BTFMAILC, XMLMail, 1					
CharsetConverter, Conve...					
DispatchMsg, Select					
eAIParser, TGParser	0	0		0	0
HostFTP, MyFTP					
HostLUAB, HostLUAB, 1			5		
HostLUAC, HostLUAC, 1					
HostSocket, MySocket, 1	0	0		0	0
HTTPAdaptor, HTTPSend					
MailAdaptor, POP3Receive					
MailAdaptor, SMTPSend					
MC_SimSvc, BlueStarML	0	0		0	0
MsgHandler, MCMsgHandler	0	0		0	0
MSMQAdaptor, MSMQSend					
NameSpace, NameData	0	0		0	0
NameSpace, Params	0	0		0	0
NameSpace, Server	0	0		0	0
Persistence, XML					
PoolLUAB, HostLUAB					
PoolSocket, MySocket	0	0		0	0
Selector, Engine	0	0		0	0
TimerHlpr, Client					
TimerHlpr, Server					

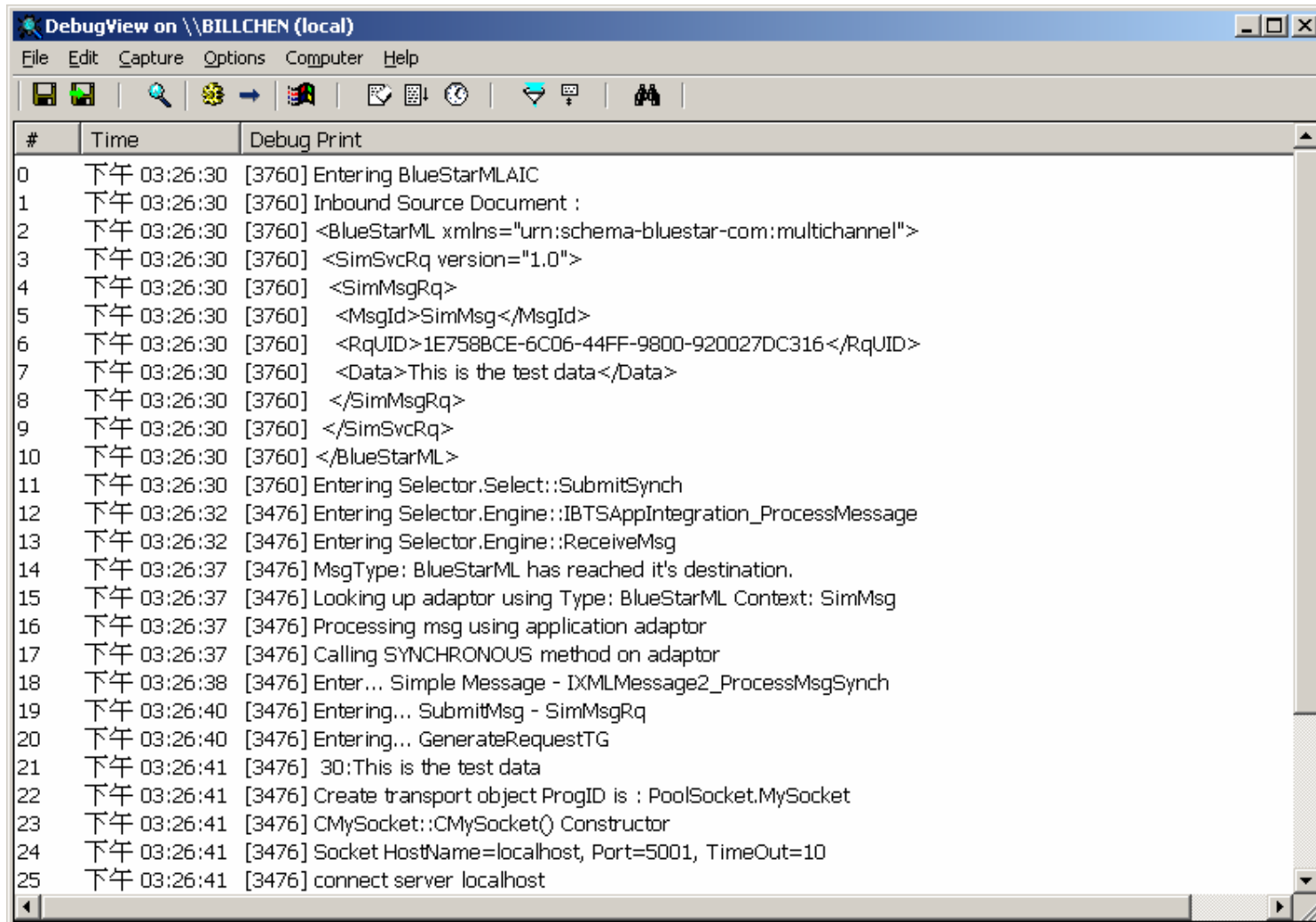
Event Monitoring

- Integrated with Windows Event Viewer for BlueStar monitoring
 - Completely auditable with BlueStar system and business objects



Debugging Tool

- Debugging Tool for application debugging



The screenshot shows the DebugView application window titled "DebugView on \\BILLCHEN (local)". The window has a menu bar with "File", "Edit", "Capture", "Options", "Computer", and "Help". Below the menu bar is a toolbar with various icons. The main area contains a table with three columns: "#", "Time", and "Debug Print". The table lists 26 entries (0 to 25) showing the execution of a BlueStarML application. The debug prints include XML tags, timestamps in Chinese (e.g., "下午 03:26:30"), and process IDs (e.g., [3760], [3476]).

#	Time	Debug Print
0	下午 03:26:30	[3760] Entering BlueStarMLAIC
1	下午 03:26:30	[3760] Inbound Source Document :
2	下午 03:26:30	[3760] <BlueStarML xmlns="urn:schema-bluestar-com:multichannel">
3	下午 03:26:30	[3760] <SimSvcRq version="1.0">
4	下午 03:26:30	[3760] <SimMsgRq>
5	下午 03:26:30	[3760] <MsgId>SimMsg</MsgId>
6	下午 03:26:30	[3760] <RqUID>1E758BCE-6C06-44FF-9800-920027DC316</RqUID>
7	下午 03:26:30	[3760] <Data>This is the test data</Data>
8	下午 03:26:30	[3760] </SimMsgRq>
9	下午 03:26:30	[3760] </SimSvcRq>
10	下午 03:26:30	[3760] </BlueStarML>
11	下午 03:26:30	[3760] Entering Selector.Select::SubmitSynch
12	下午 03:26:32	[3476] Entering Selector.Engine::IBTAppIntegration_ProcessMessage
13	下午 03:26:32	[3476] Entering Selector.Engine::ReceiveMsg
14	下午 03:26:37	[3476] MsgType: BlueStarML has reached it's destination.
15	下午 03:26:37	[3476] Looking up adaptor using Type: BlueStarML Context: SimMsg
16	下午 03:26:37	[3476] Processing msg using application adaptor
17	下午 03:26:37	[3476] Calling SYNCHRONOUS method on adaptor
18	下午 03:26:38	[3476] Enter... Simple Message - IXMLMessage2_ProcessMsgSynch
19	下午 03:26:40	[3476] Entering... SubmitMsg - SimMsgRq
20	下午 03:26:40	[3476] Entering... GenerateRequestTG
21	下午 03:26:41	[3476] 30:This is the test data
22	下午 03:26:41	[3476] Create transport object ProgID is : PoolSocket.MySocket
23	下午 03:26:41	[3476] CMySocket::CMySocket() Constructor
24	下午 03:26:41	[3476] Socket HostName=localhost, Port=5001, TimeOut=10
25	下午 03:26:41	[3476] connect server localhost

The screenshot shows a web browser window titled "BlueStar Server 2001". The browser's address bar and menu bar are visible. The main content area is divided into two panes. The left pane is a navigation tree with the following structure:

- Getting Started with Cedar BlueStar Ser
 - Release Notes
 - Readme
 - Overview
 - Understanding BlueStar
- Administration
 - Installation Guide**
 - Installing Selector Trigger
 - Administration
- Development
 - Envelopes and Plug-Ins Explained
 - The Envelope Plug-In Code Genera
 - How to Use the Plug-In Generator
 - Application Adapters
 - Mail Transport Adapters
 - MSMQ Transport Adapter
 - Building Database Wrappers
 - Debug Viewer
- SDK Samples

The right pane displays the "Cedar BlueStar Server 2001 - Installation Guide" page. The page content is as follows:

Cedar BlueStar Server 2001 - Installation Guide

Cedar Software Corporation
July 2001

Summary: This article describes how to install the Cedar BlueStar Server 2001.

Contents

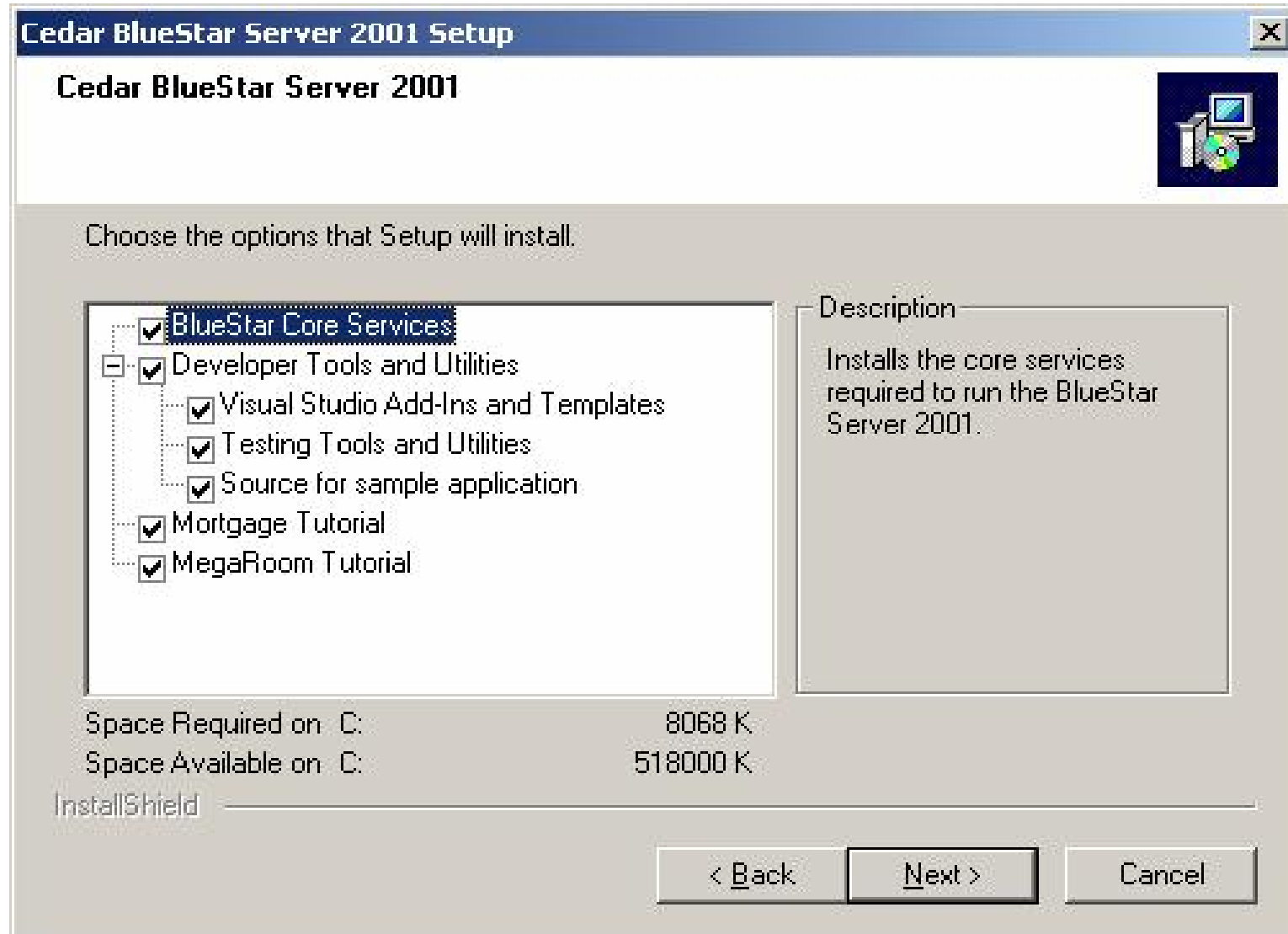
- [Introduction](#)
- [Removing Earlier Releases](#)
- [Platform Requirements and Installation Order](#)
- [Installing the BlueStar Server 2001](#)
- [Un-Installing](#)
- [Modifying a BlueStar Server 2001 Installation](#)
- [Some Installation Details](#)
- [Verifying Installation](#)

Introduction

The BlueStar Server includes an automated setup application. This document provides instructions on how to install the prerequisite applications, and the BlueStar Server, as well as some simple tests to verify that the installation worked properly.

Removing Earlier Releases

The setup program provided with the Cedar BlueStar Server 2001 does not attempt to remove previous versions. However, the setup will check if earlier versions of the BlueStar are still installed on the server. The install is aborted if previous versions are detected. In order for the installation to complete successfully, all MTS packages, components, and DLLs must be removed first.

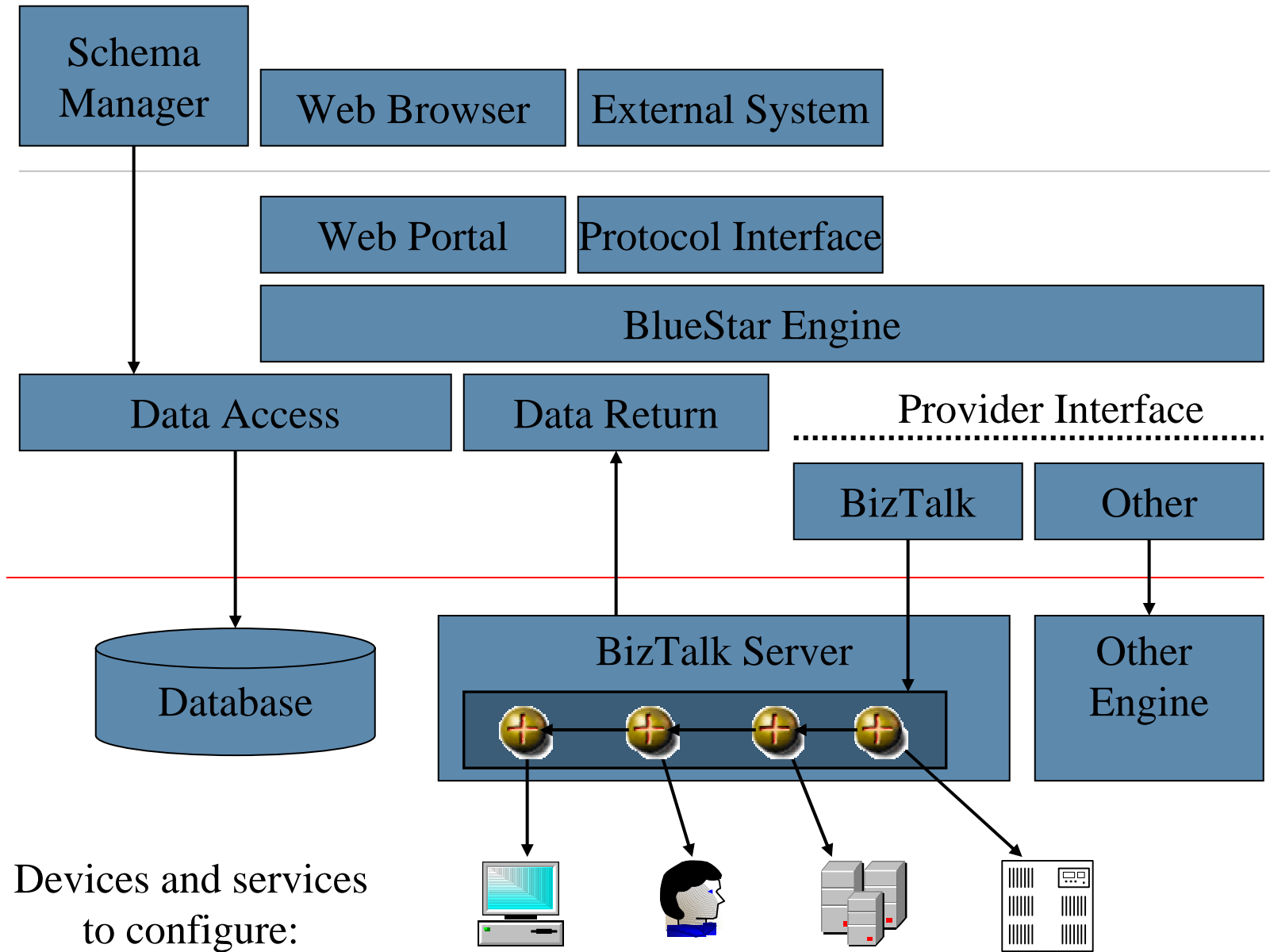


BlueStar's Key Design Characteristics

- XML for data and XSLT for presentation/transformation
 - Internal data structures are XML
 - XML messages between sub-systems
 - XSLT Transforms to convert provider messages
- Provider Interface and moniker
 - COM interface to abstract BlueStar engine
 - XML data structure passed to provider
- BizTalk Extensions for BlueStar
 - BlueStarML AIC
 - State Management, Transaction context, logging for BizTalk Transactions, Auditing/Event management
 - COM+ interface for hosting scripts
 - Simplified object model for script writers
 - Parser
 - Parameter driven for legacy application integration

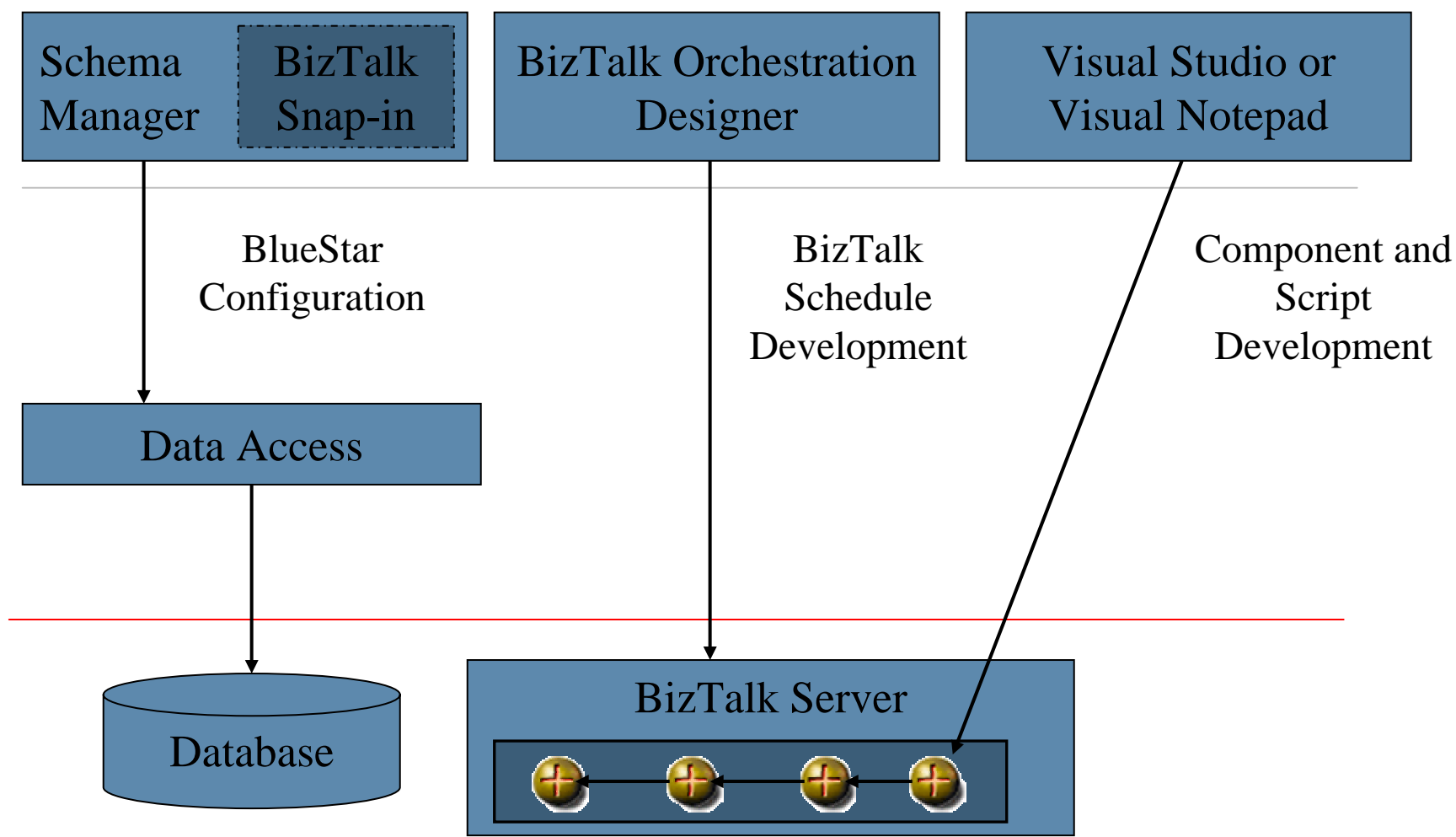
- BlueStar provides a solution framework to support channel integration
- The catalog and interfaces are standard components
- “Engines” actually perform tasks
- Engines receive a standard XML document and may need to transform it
- Engines can support synchronous or asynchronous behavior

BlueStar Architecture



Devices and services
to configure:

Design a BlueStar Scenario



Devices and services to configure:

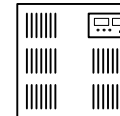
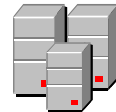
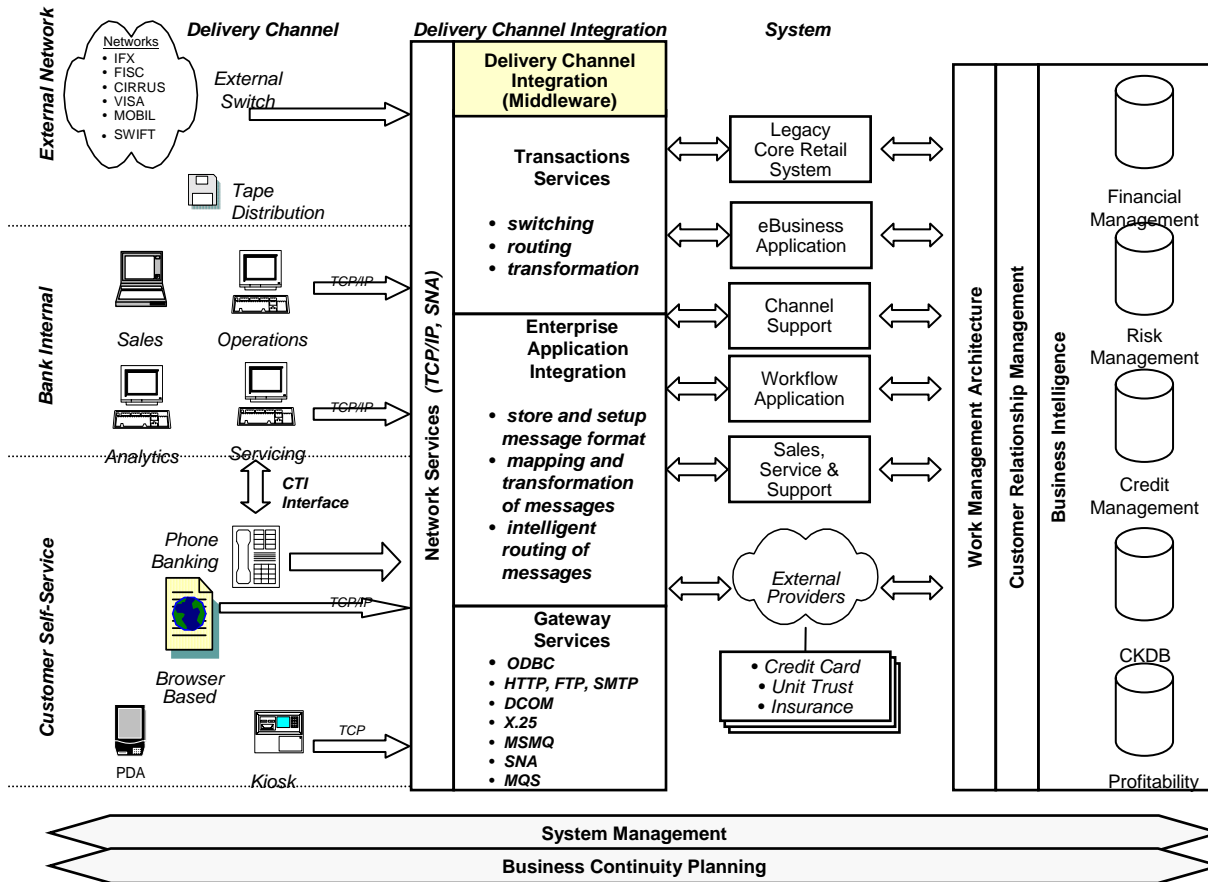


Table of Content

- EAI Framework
- Integrating Financial Channels
- BlueStar Overview
- BlueStar Architecture
- Summery

Target Core Retail Architecture

- The IT Architecture Project has developed the target architecture blueprint for Consumer Financial Systems.

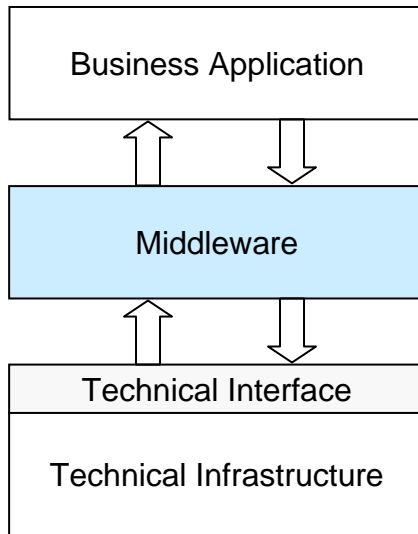


Addressing these business imperatives :

- Improved customer relationship management, credit and exposure analysis, profitability monitoring and marketing
- Product factory capabilities for rapid rollout of new retail segment products and services
- Consistent capabilities across all geographic markets
- Support for 3rd party products and integration

Will require these changes :

- Upgrade and implementation of new delivery channel components
- Standardized interfaces and middleware across all components



Definition

Middleware defines software services which enable business applications to interact seamlessly with each other. It consists of products which integrate applications regardless of the complexities of systems, networks, protocols, and operating systems.

Business Benefits

- Shields business functionality from technical infrastructure
- Enables flexibility and improves time-to-market for new capabilities
- Reduces development time and overall solution cost
- Provides enhanced functionality and scalability
- Reduces risk during system conversions and transitions
- Provides platform transparency
- Enforces architecture and application standards

Middleware Categories

- **Three middleware categories have been defined. These categories will guide product selection and standards development.**

	<u>Middleware Categories</u>	<u>Description</u>
1	Channel Transaction Services Middleware	The main purpose of platform middleware is to handle transaction management and resource management, switching between delivery channels and the back end processes.
2	Enterprise Application Integration Middleware	Application Integration Middleware allows storing and setting up of different message format, mapping and transformation of messages, and intelligent (rule-based) routing of messages between applications. It is responsible for the exchange of messages between applications that reside on different platforms. Communication is done either synchronously or asynchronously. The messaging architecture supports rule-based message routing, message queuing, transactional integrity and guaranteed delivery.
3	Gateway Services Middleware	Gateway Services Middleware enables user applications to interact seamlessly using industry standard protocols with different databases, application systems, and external networks

- General Strategy for EAI Implementation.

- Implementation of EAI within the bank will adopt phase implementation approach
- Priority will be given to applications that could gain the immediate and most benefits of using an EAI system.
- Standards and Guidelines on the usage of EAI need to be in place before begin of any EAI project.
- For migration of existing point-to-point applications to the EAI system. A migration plan will have to be in place before migration begin.
- For each EAI implementation project, a list of issues and recommendations on usage of the EAI system should be delivered together with other project deliverables.

Investment Requirement - BizTalk

- Investment requirement is examined in 3 areas - hardware, software licensing and resources.

<u>Investment</u>	<u>Details</u>
1 Hardware	- 2 Production Servers' requirement as Intel Server with 1GHz CPUs and 1GB Memory
2 Software	- SQL Server 2000 - Eicon X25 Software (Optional) - BizTalk Server 2000 - Host Integration Server 2000 (Optional) - Microsoft Visual Studio 6.0 - Windows 2000 Advanced Server
3 Resources and Others	- Manpower cost for services estimated to be 250 mandays - Maintenance / technical support cost

Risk Identification and Mitigation Approaches

- The primary risk for the EAI implementation have been identified and mitigation approaches are developed.

<u>Risk</u>	<u>Description</u>	<u>Mitigation</u>
1 Schedule	Roll-outs of EAI interfaces depend on various roll-outs of external systems and task may not start on time	<ul style="list-style-type: none">- Co-ordinate the roll-out of each systems and to minimise the risk of having the EAI implementation as the bottleneck- Apply deadline to tasks- Ensure that migration procedure is published on time in order to minimise migration risk due to schedule impact
2 Performance	The ability to fulfil bank's requirement	<ul style="list-style-type: none">- Performance benchmark at early stage- Performance testing to be done as part of the implementation plan
3 Scalability	The ability to scale to bank's future requirement	<ul style="list-style-type: none">- Acquire hardware with scale up capabilities and to use hot-standby solution to support additional loads- Target platform should be on Windows 2000
4 Skills	Skills required to develop and maintain the interfaces	<ul style="list-style-type: none">- Vendor to conduct formal training- Knowledge transfer during development with the vendor
5 Operation	High availability	<ul style="list-style-type: none">- Thorough testing of the proposed system recovery solution and high availability solution

Impact on Current Applications

- Two major impacts are identified and will be taken into consideration for the implementation planning of EAI solution.

Impact

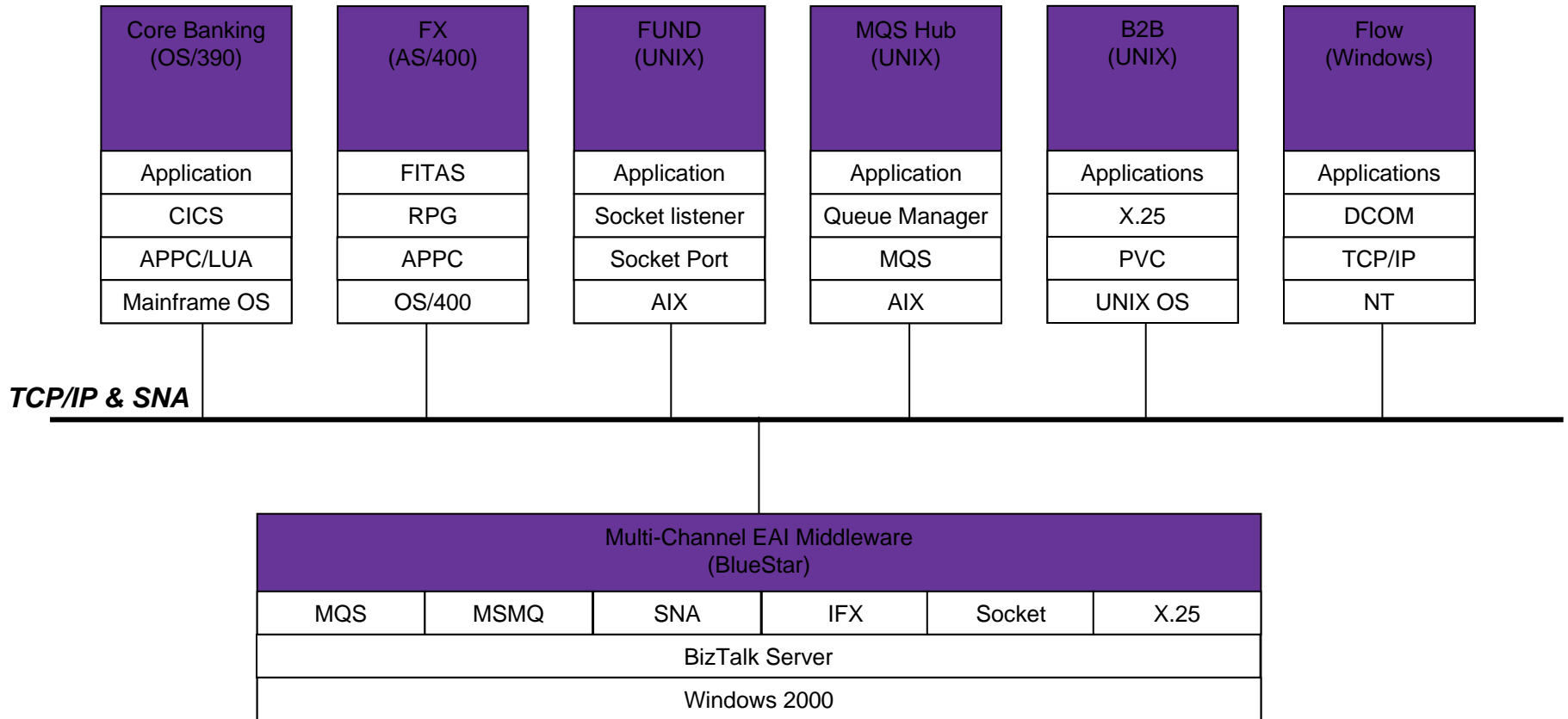
- 1 Alight to Common Interface Approach
- 2 Standardised Messaging Formats

Implication

- Some exiting applications rely on synchronous communication
- Applying asynchronous approach may require a re-design of existing process
- Performance concern has to be addressed
- Messaging format should comply to a set of unified standards
- Existing format will have to be modified
- Extra information such as the source of the application may be included within the message

Architecture Component Diagram - BlueStar

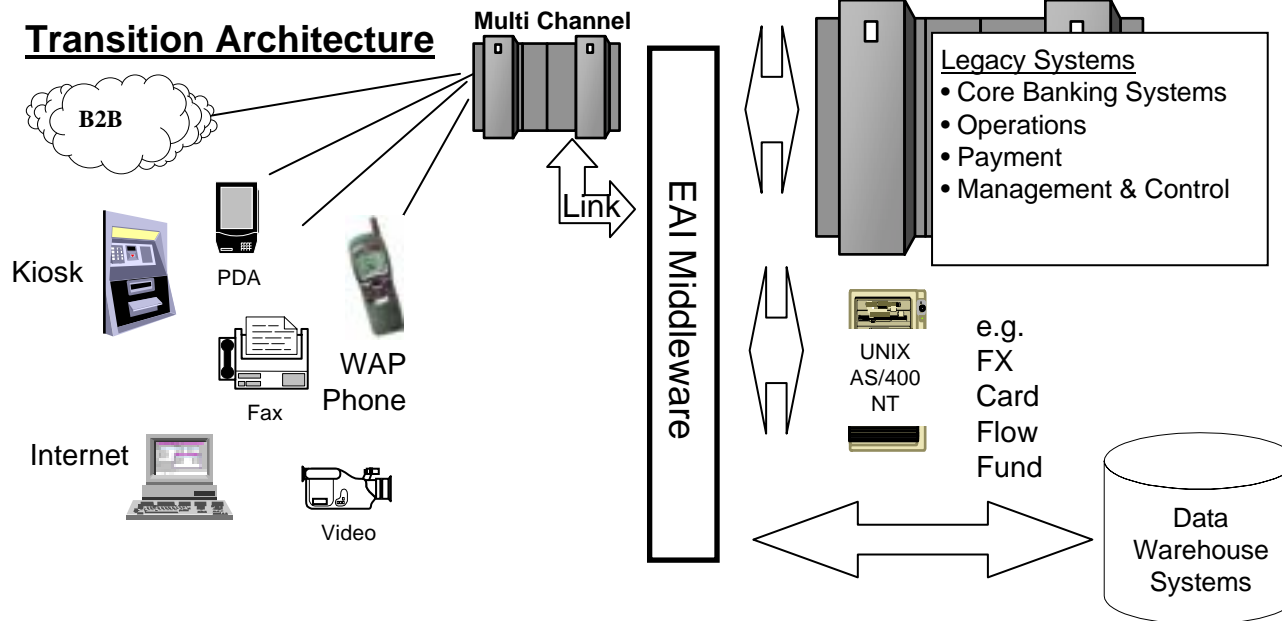
- The high level architecture component diagram for the key strategic applications that come out from the implementation phase is illustrated.



** Physical location to be determined.*

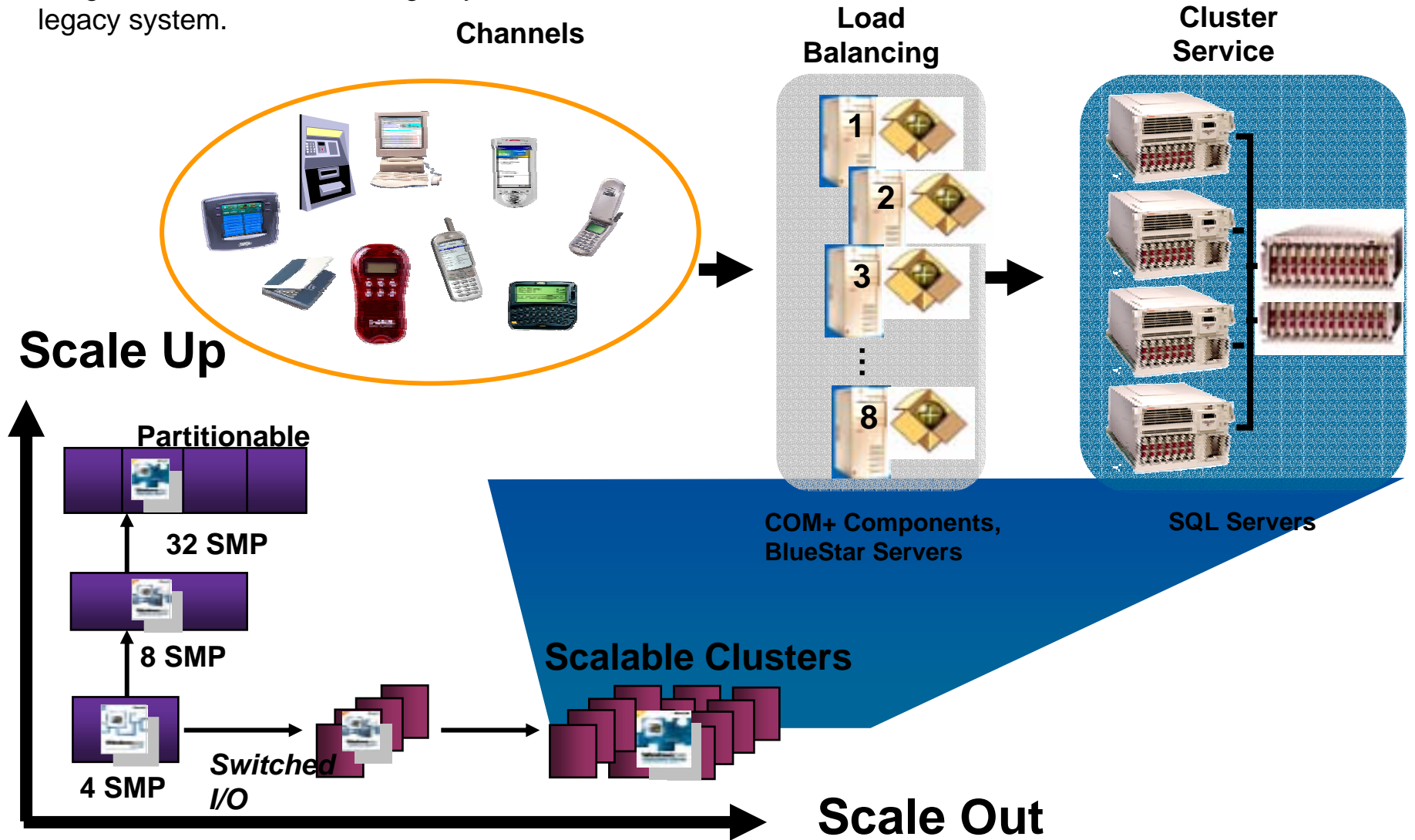
Delivery Channel Integration Architecture

- It was recommended to the management that Multi-Channel will be used as the channel transaction middleware for the transition architecture.



High Availability Solution Proposed by BlueStar

- A high availability solution proposed for the integration of the three strategic systems and the legacy system.



Security Architecture

- Security issues are addressed in 3 levels : communication security, application security and hub security.

