Cedar

BlueStar Multi-Channel Solution Overview

Cedar Software

May, 2003

Table of Content

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

EAI Means Enterprise Application Integration

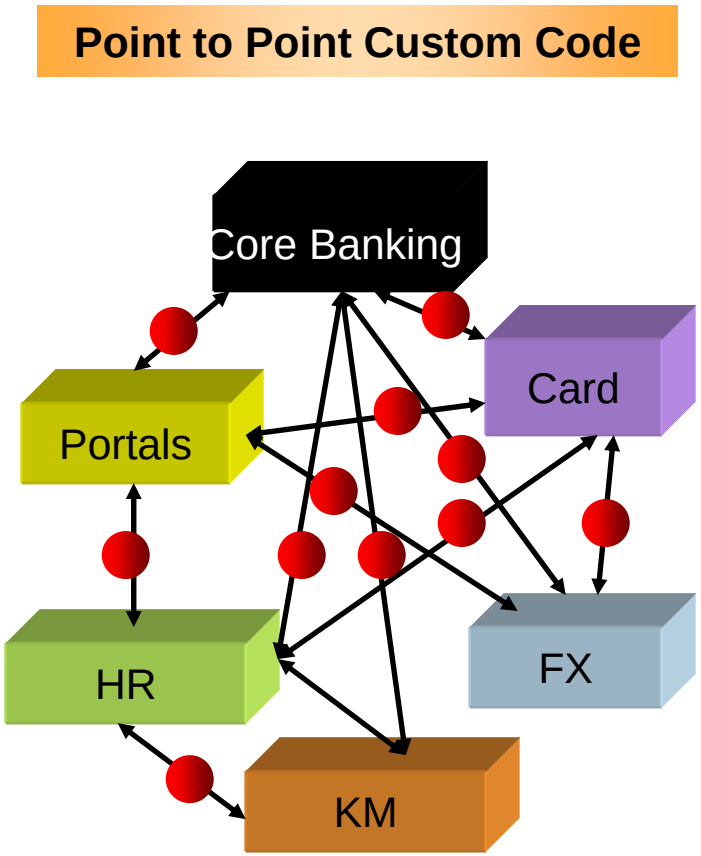
EAI deals with different enterprise integration relationships as follows:

- 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.)

Cedar

Traditional Integration Approach

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



- 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

Cedar

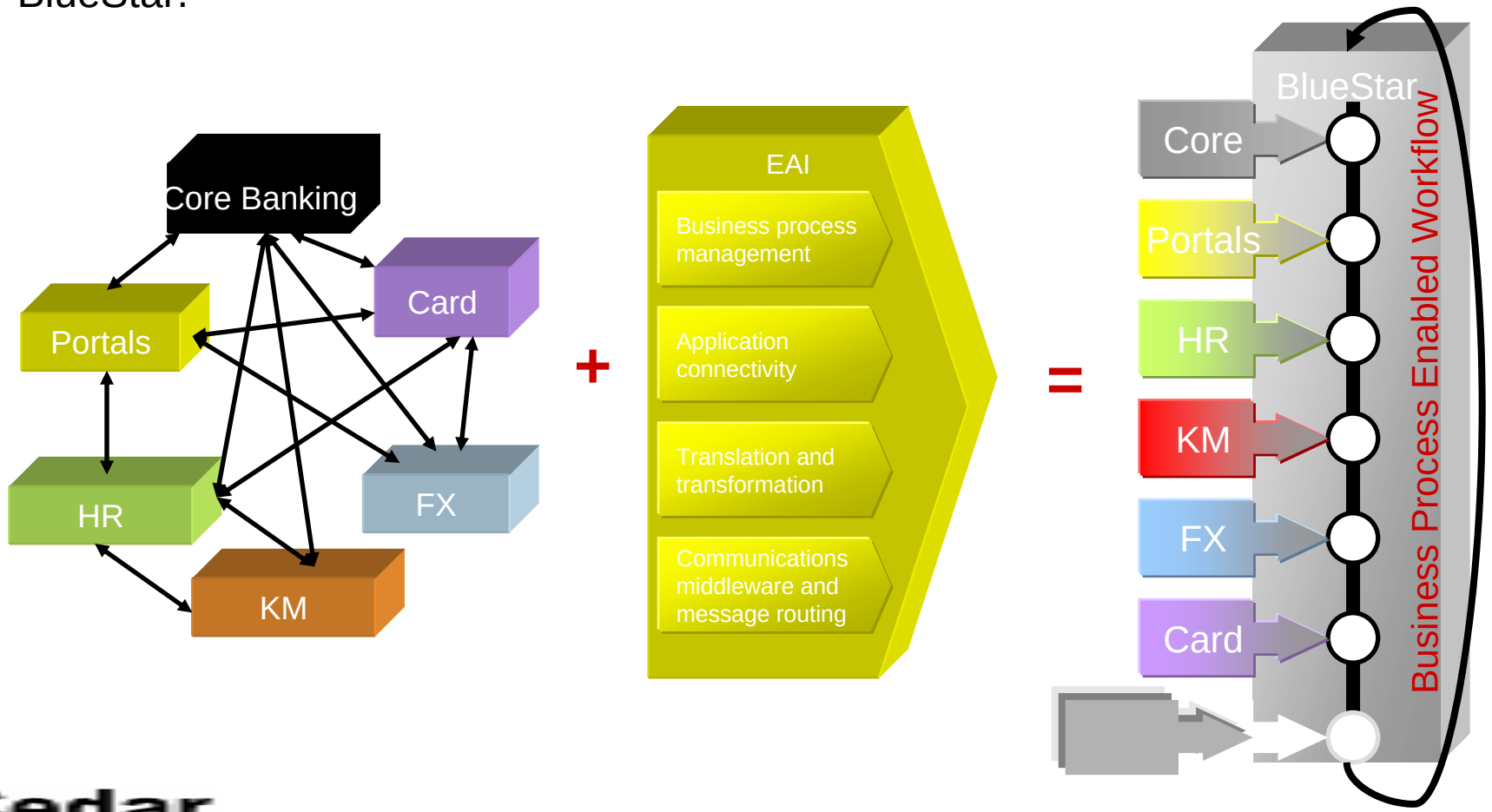
EAI Framework

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.



Cedar

Table of Content

Benefits of channel integration

What benefits do you except 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 integration efforts won’t success

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

Firms need Standards-Based App Server and Integration Server

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

Embrace process tools to design and manage financial process

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)

Application Server

- **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

Integration Server

- **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

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.

BlueStar Value Proposition

- 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

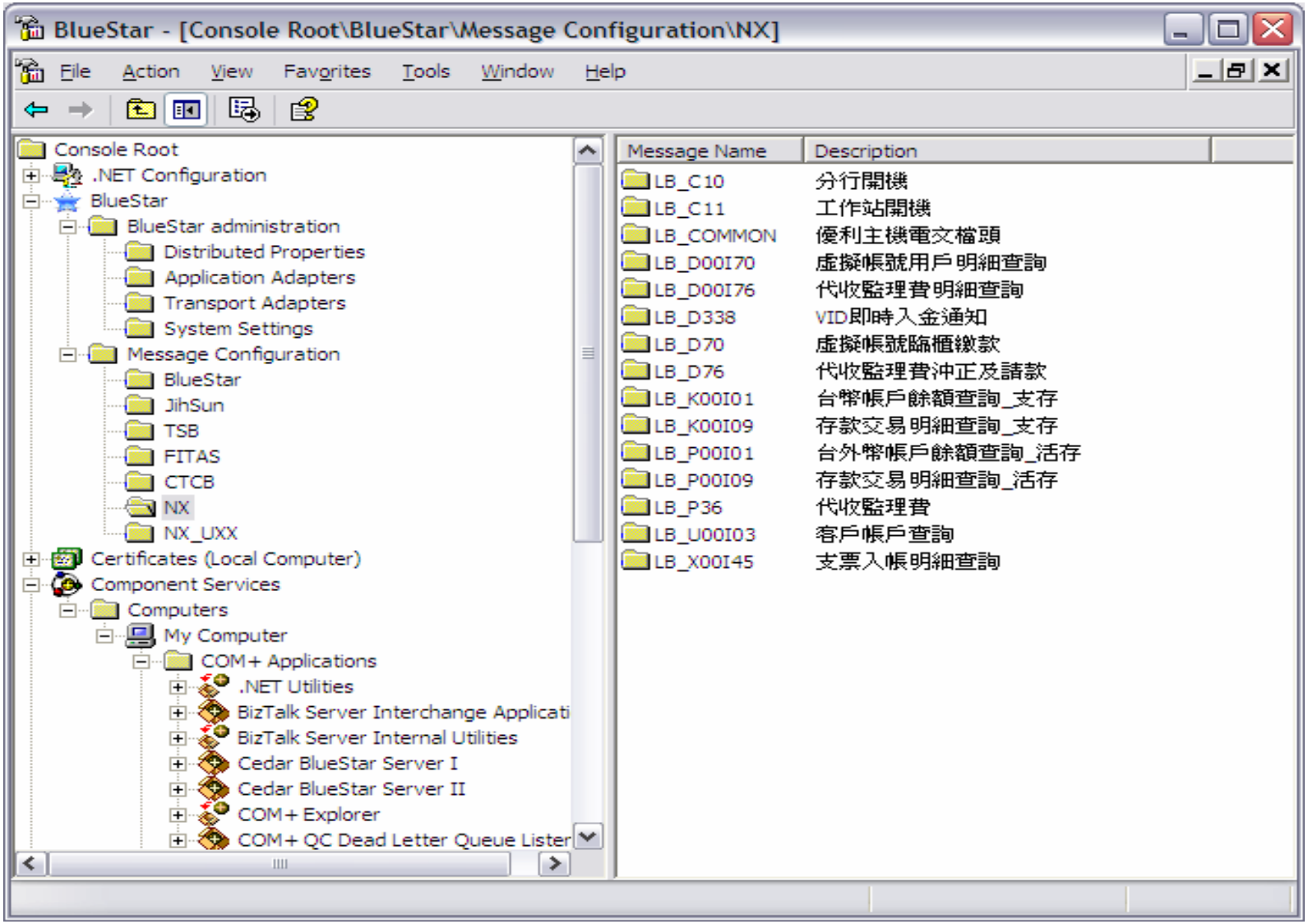
- Parameter driven (SQL)
- XML enabling
- Role based security
- BlueStar “engines” do the work.
  - BPM for complex long duration workflow.
  - Compensating operations.
- Scripting philosophy
- Programmatic Interface
- Completely auditable
- 100% COM+ component technology
- Session management
- Multilanguage 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 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

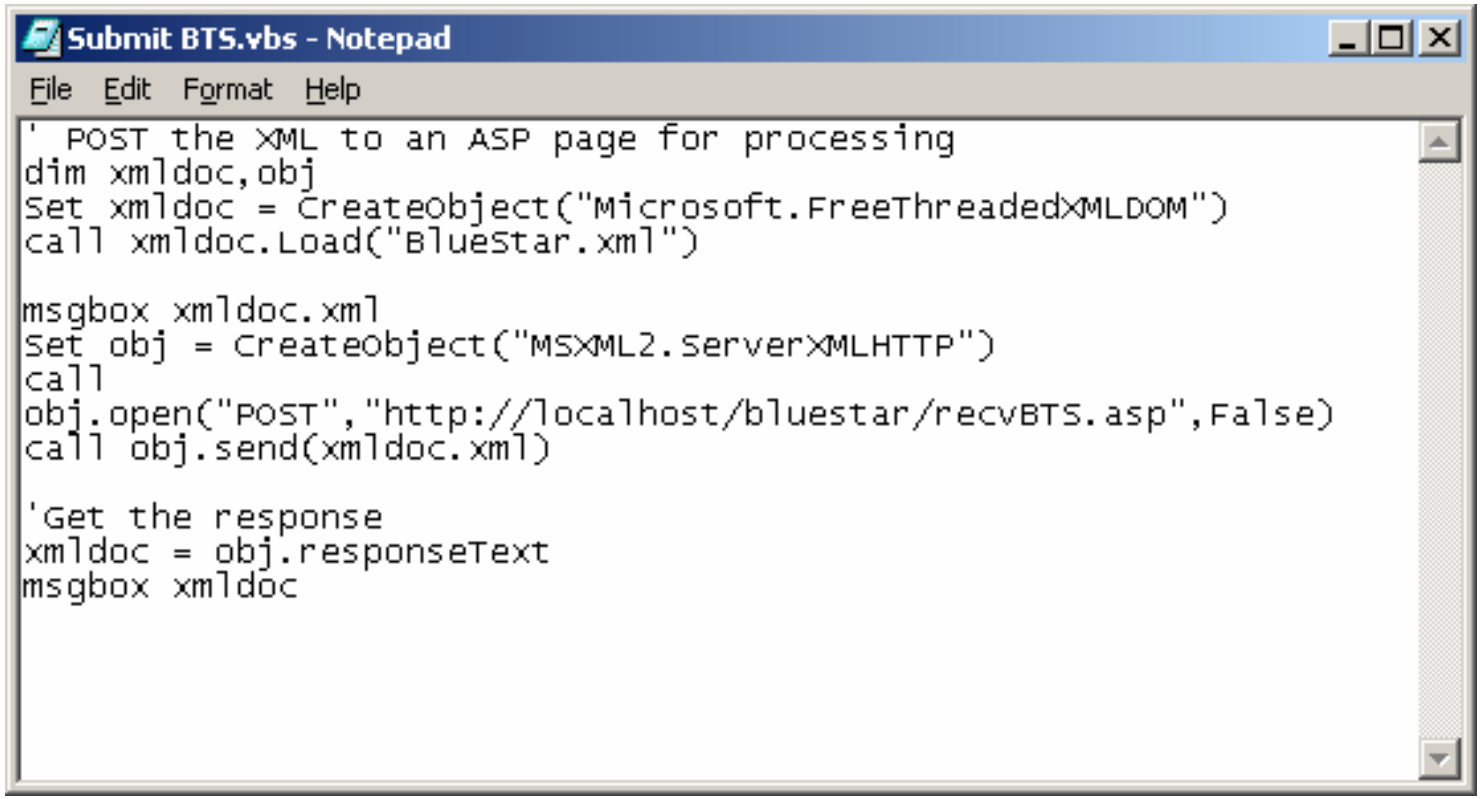
Table of Content

Configurable UI – Parameter Driven for message design



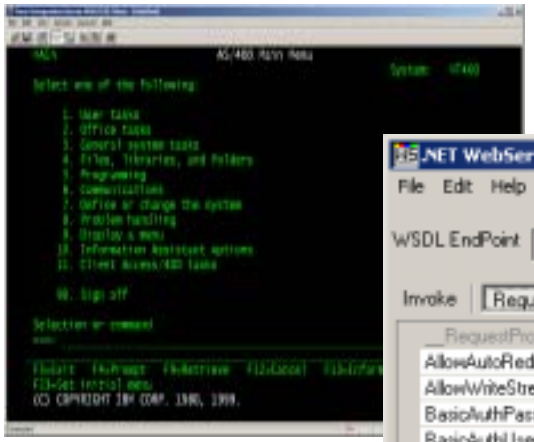
Scripting

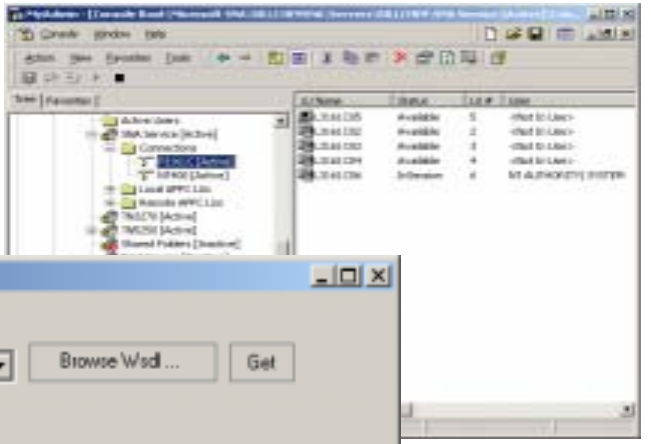
- Script Host components for application integration
  - Host the execution of VB/PERL scripts.



Web Services

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





.NET WebService Studio

File Edit Help

WSDL EndPoint: C:\Program Files\Cedar\BlueStar Server\Bin\web\BlueStarML.WSDL | Browse Wsdl ... | Get

Invoke | Request/Response | WSDLs & Proxy | Messages

AllowAutoRedir	True
AllowWriteStrea	True
BasicAuthPass	
BasicAuthUserN	
ContentType	text/xml; charset=u
KeepAlive	False
Method	POST
Pipelined	False
PreAuthenticate	False
Proxy	
SendChunked	False
SOAPAction	http://tempuri.org/
Timeout	100000
Url	http://localhost/bl
UseCookieCont	True
UseDefaultCred	False

Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soapen
  <soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <q1:ProcessMessageSynch xmlns:q1="http://tempuri.org/message/">
      <objBlueStarML xsi:type="xsd:string">&lt;BlueStar MsgName="SockMsg"&gt;&lt;Elemen
    </q1:ProcessMessageSynch>
  </soap:Body>
</soap:Envelope>
```

Send

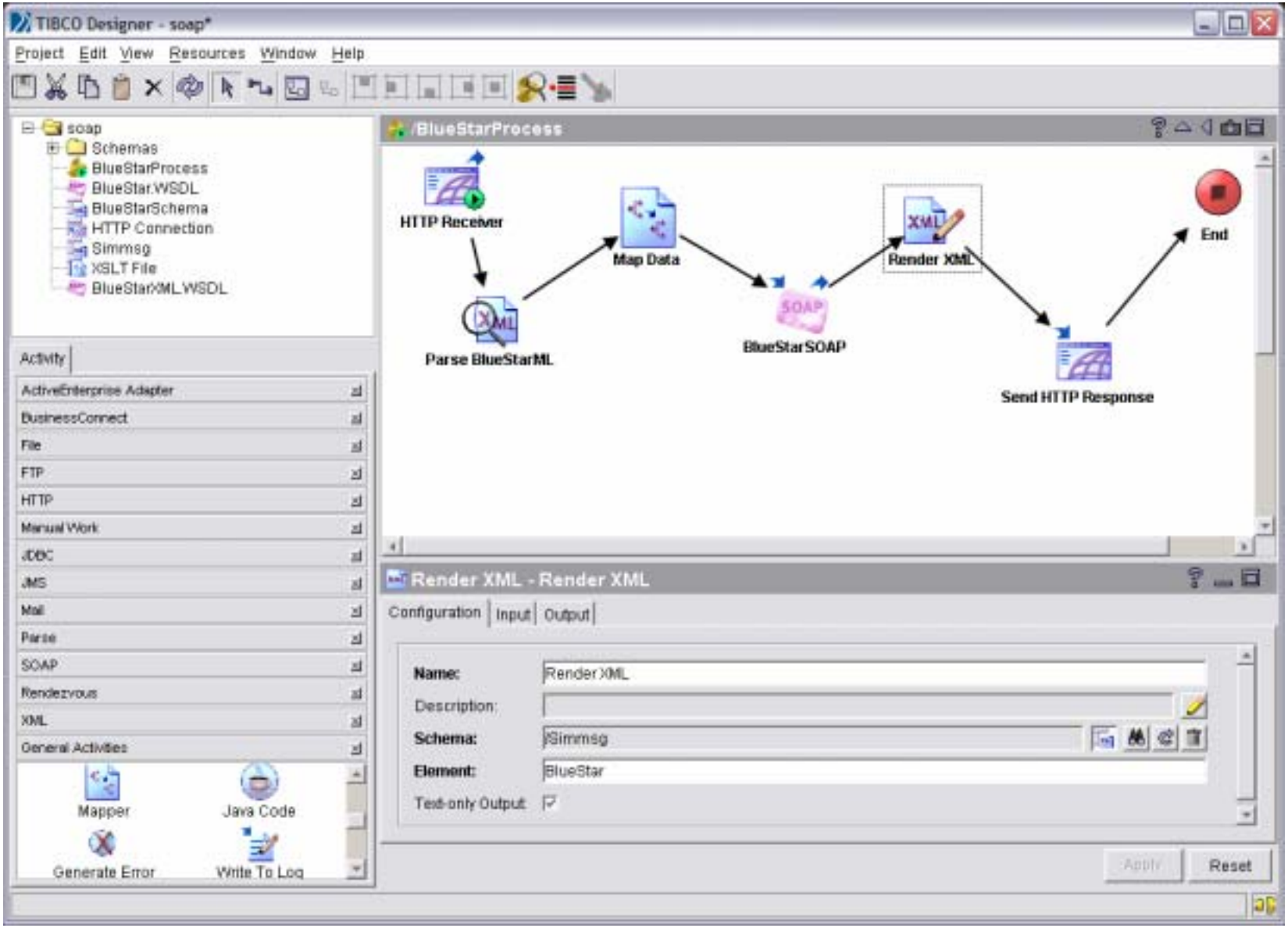
Response

```
Content-Type:text/xml; charset="UTF-8"

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encod
  <SOAP-ENV:Body>
    <SOAPSDK1:ProcessMessageSynchResponse xmlns:SOAPSDK1="http://tempuri.org/mes
      <Result>&lt;BlueStar MsgName="SockMsg"&gt;&lt;Element1&gt;  ?&lt;/Element1&gt;&lt;/Ele
  </SOAPSDK1:ProcessMessageSynchResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Cedar

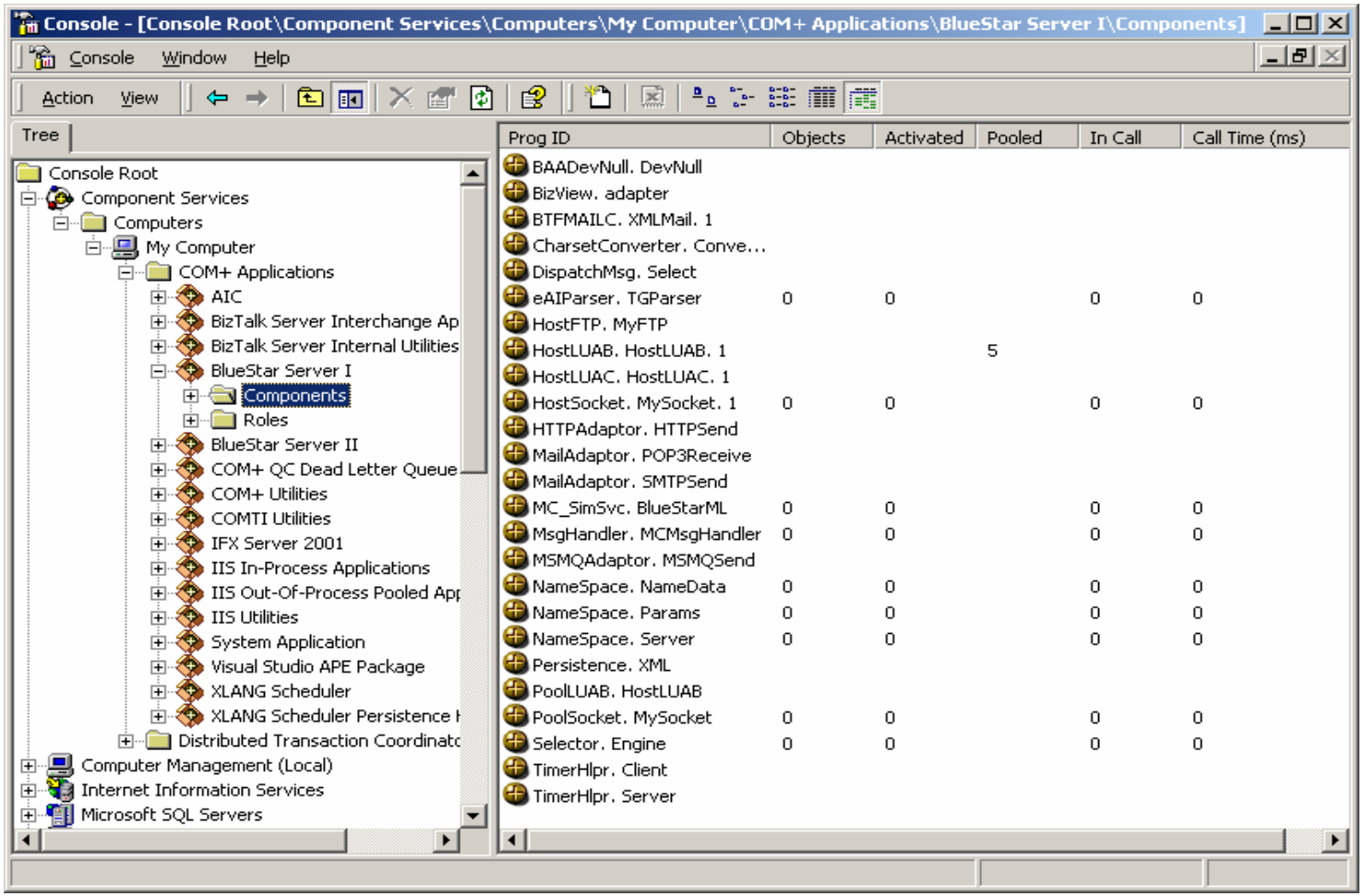
TIBCO BusinessWorks Integration



Cedar

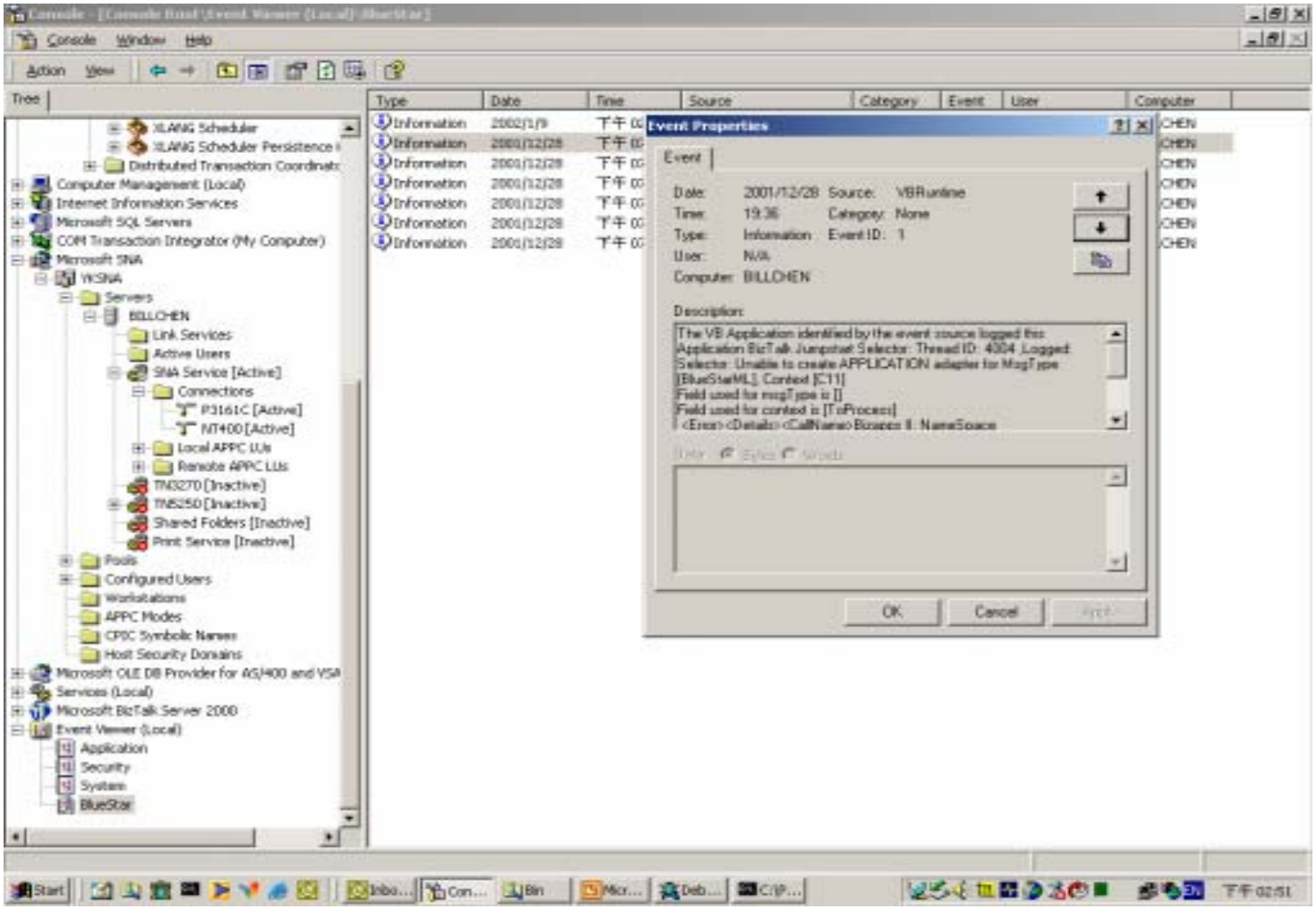
Object Pooling - Dynamic host session management

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



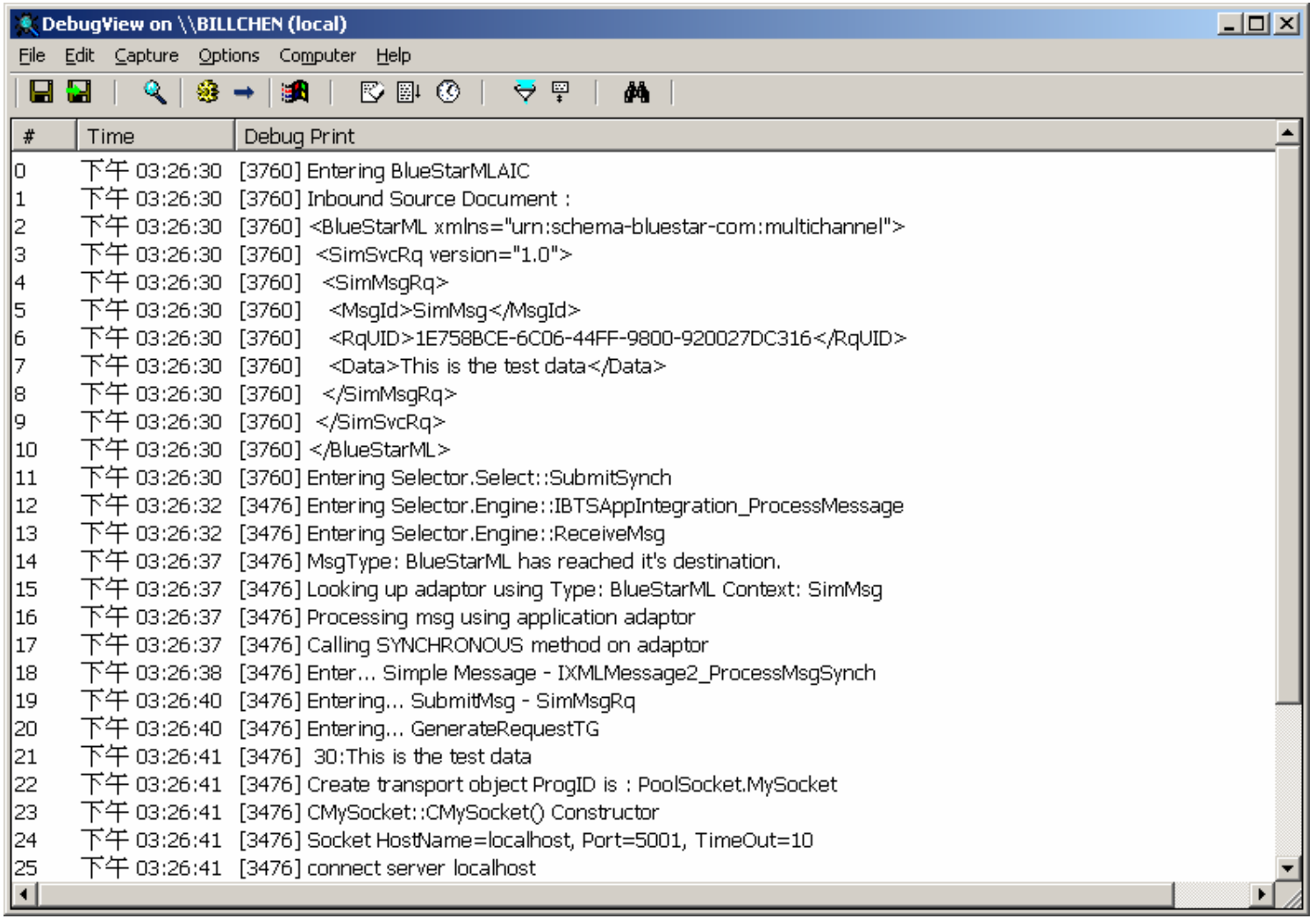
Event Monitoring

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

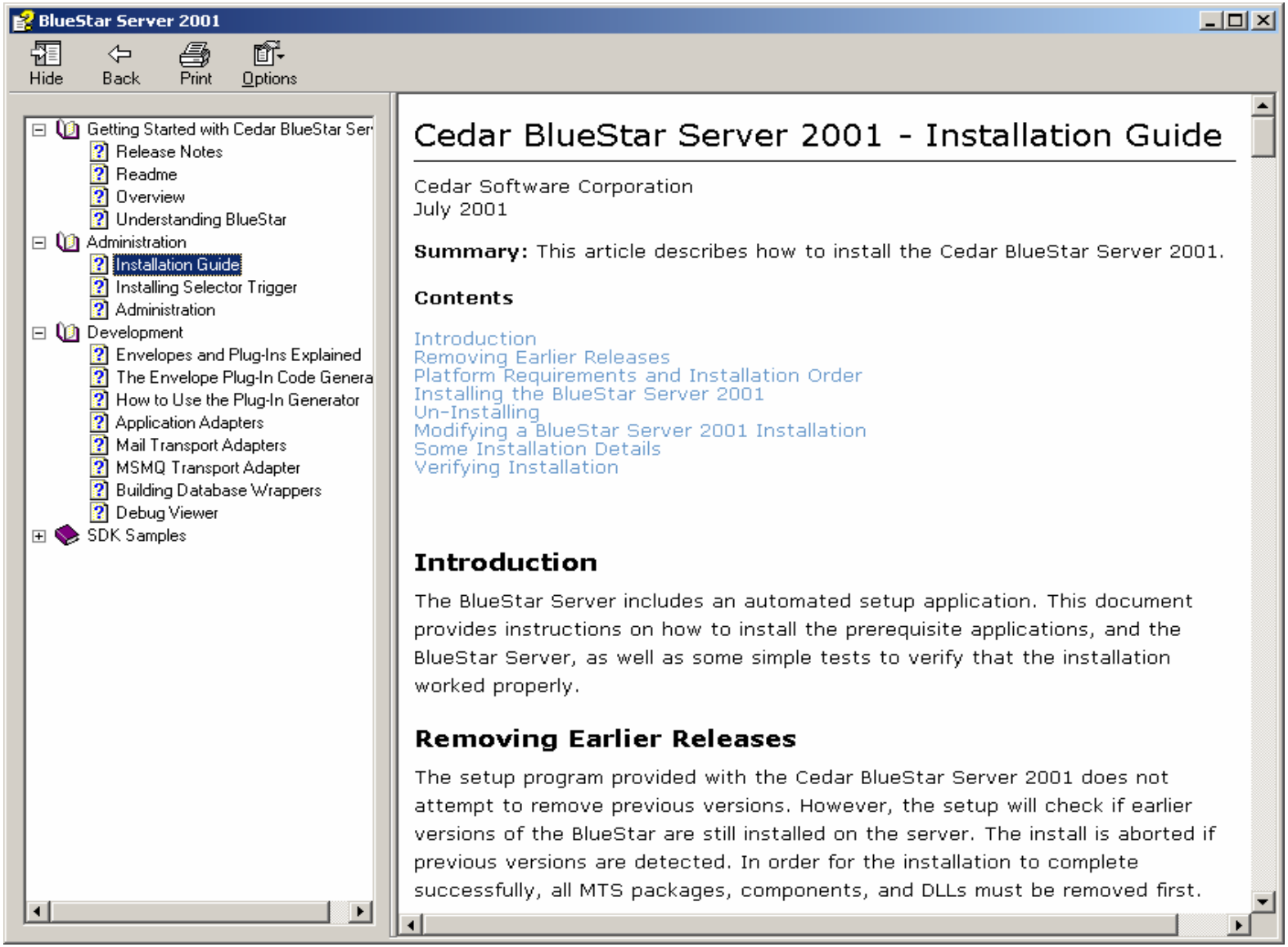


Debugging Tool

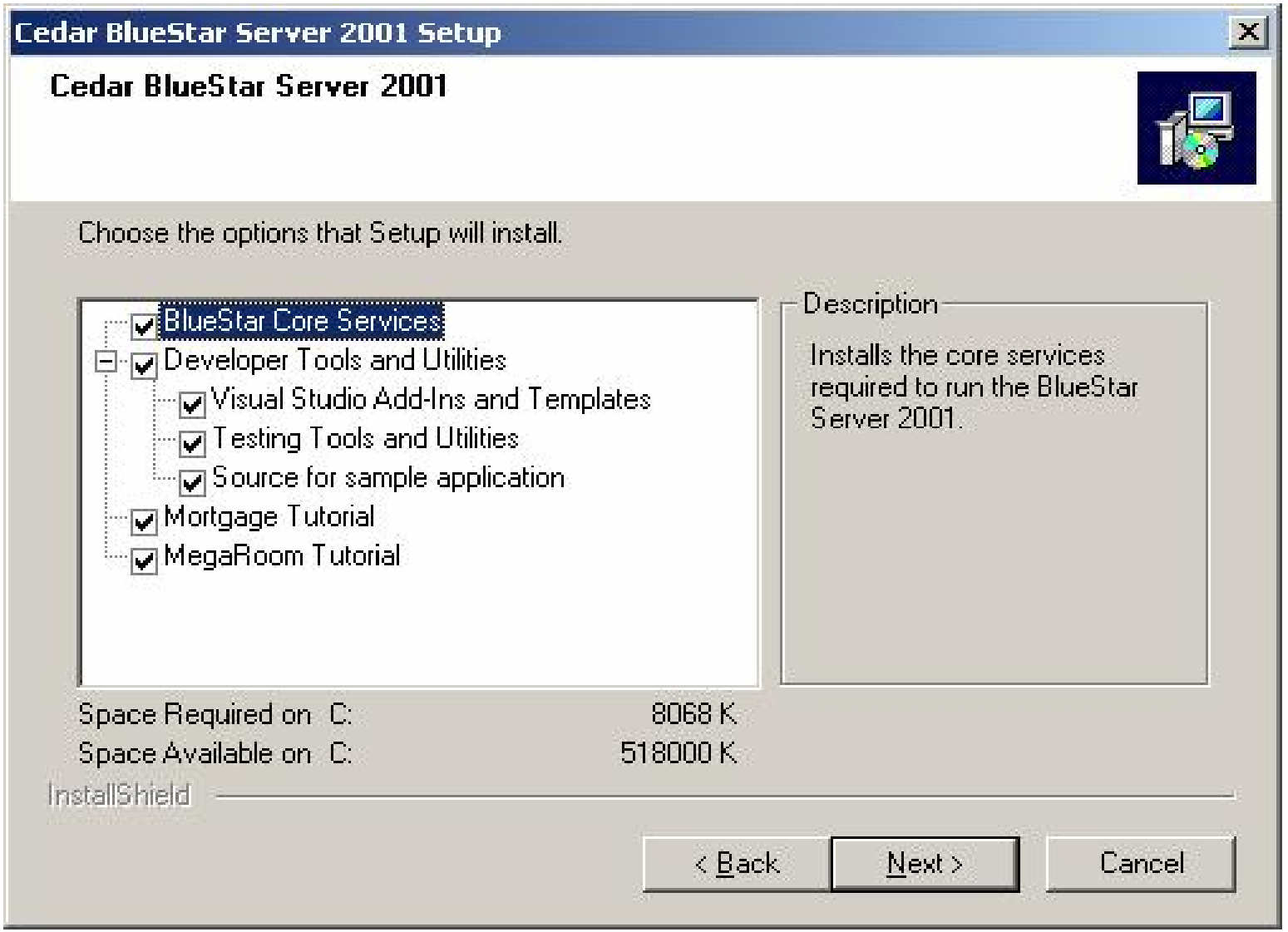
- Debugging Tool for application debugging



Online Documentation



InstallShield Support for Installation process



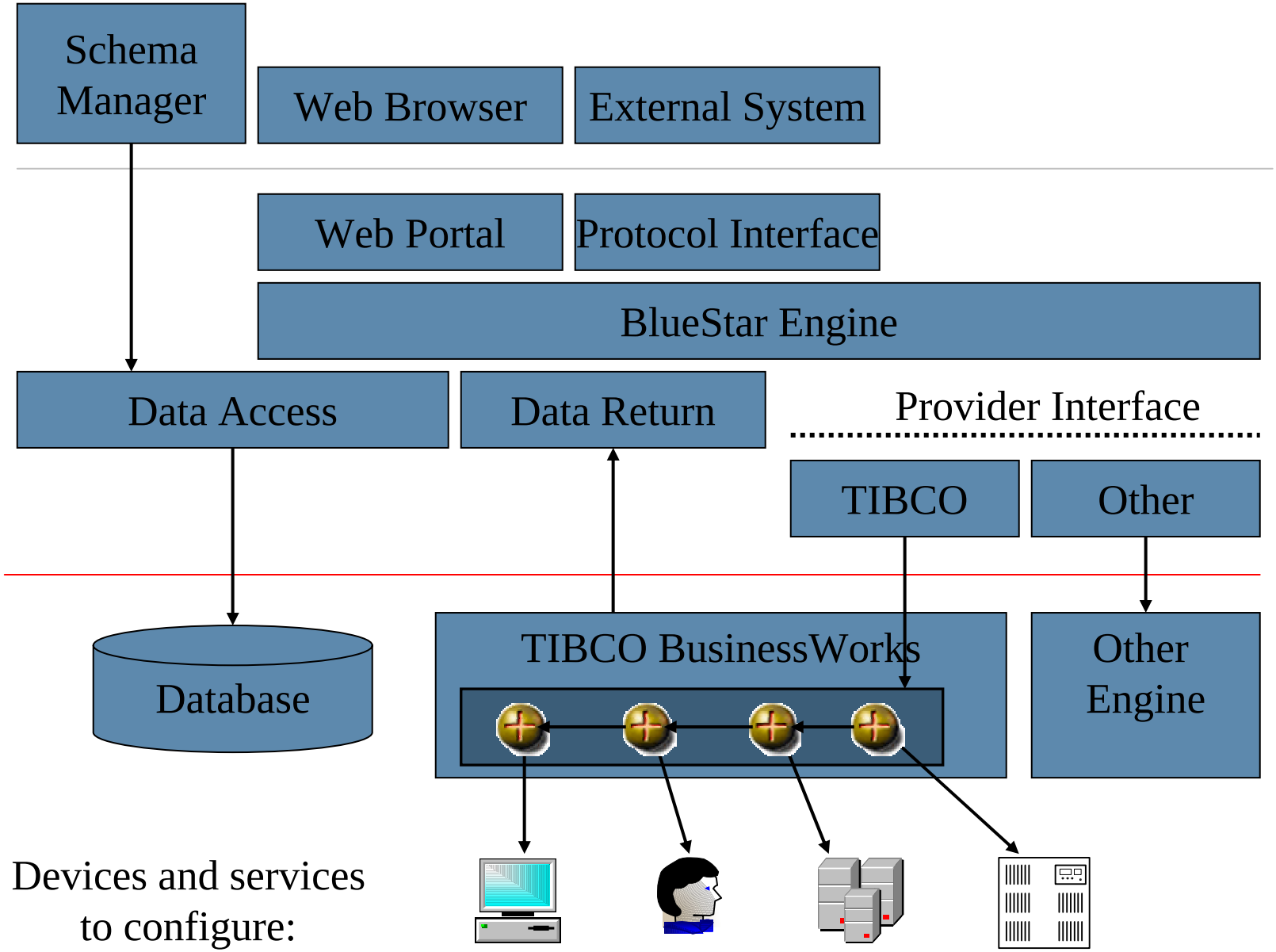
BlueStar's Key Design Characteristics

- 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
- TIBCO Extensions for BlueStar
  - BlueStarML AIC
    - State Management, Transaction context, logging for TIBCO Transactions, Auditing/Event management
  - COM+ interface for hosting scripts
    - Simplified object model for script writers
  - Parser
    - Parameter driven for legacy application integration

BlueStar Provider Model

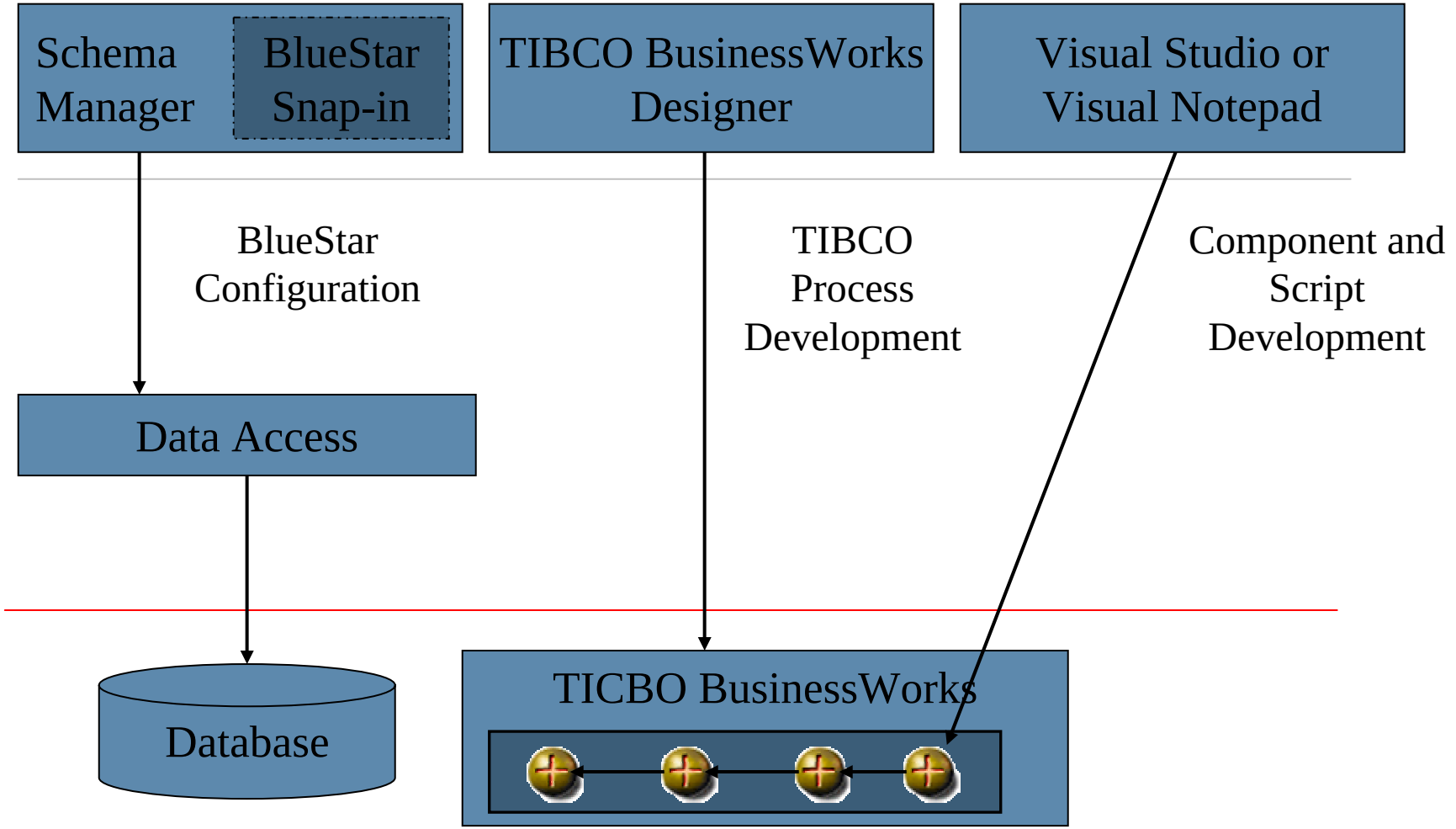
- BlueStar provides a solution framework to support channel integration
- The catalog and interfaces are standard components
- “Engines” actually perform tasks
- Engines receive a standard XML document and may need to transform it
- Engines can support synchronous or asynchronous behavior

BlueStar Architecture



Cedar

Design a BlueStar Scenario



Devices and services to configure:

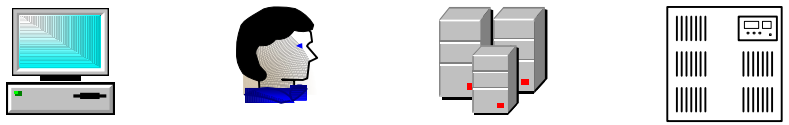
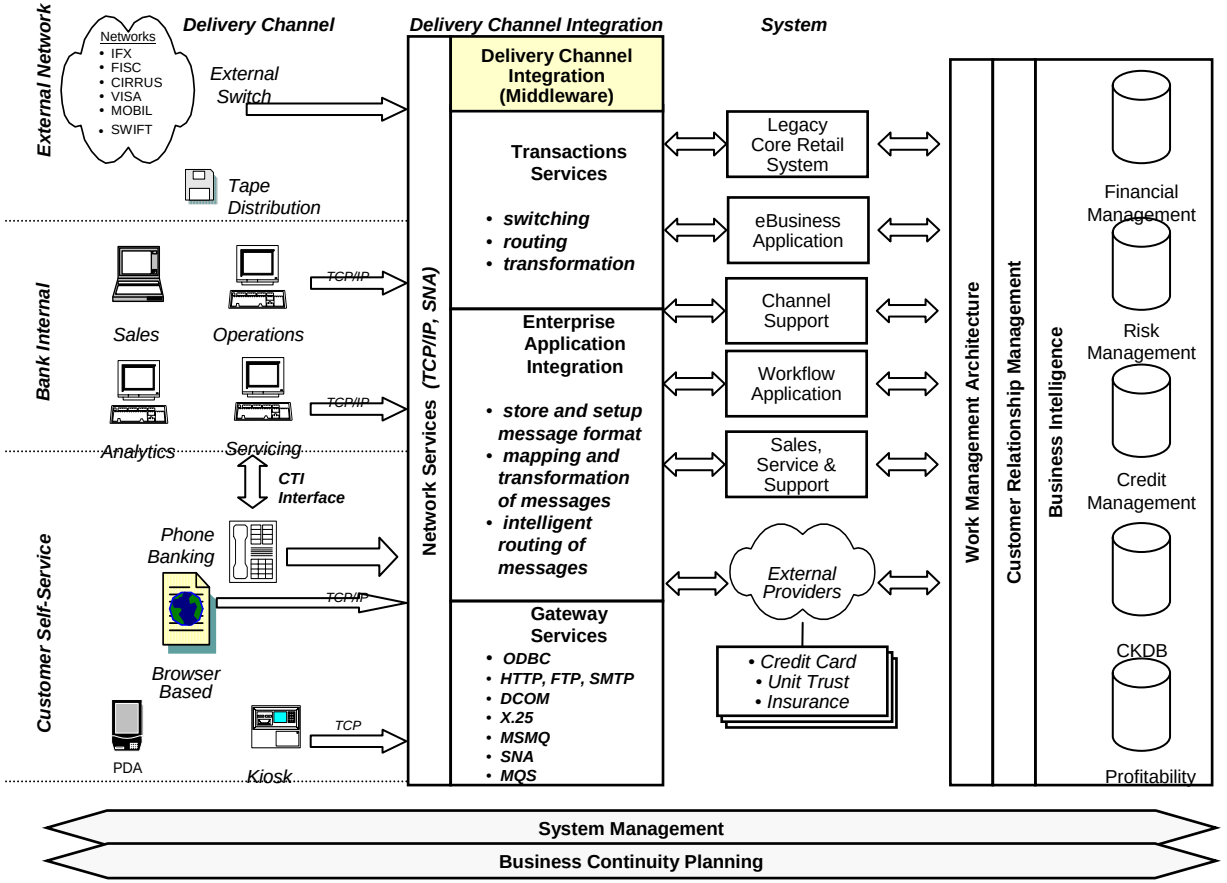


Table of Content

- 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.



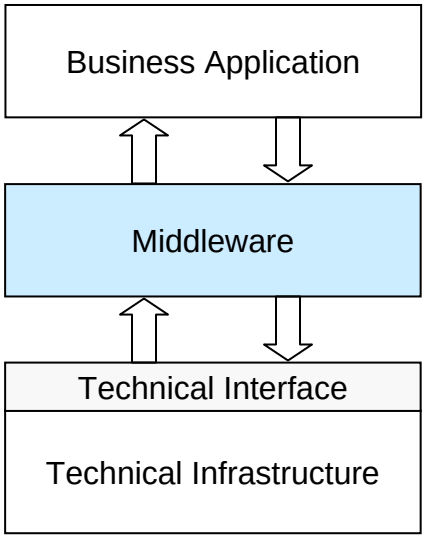
Addressing these business imperatives :

- 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

Middleware Definition and Benefits



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 to 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.

	Middleware Categories	Description
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

Overview

•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 - BlueStar

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

	Investment	Details
1	Hardware	- 2 Production Servers' requirement as Intel Server with 1GHz CPUs and 1GB Memory
2	Software	- SQL Server 2000 - Host Integration Server 2000 (Optional) - Microsoft Visual Studio .NET - 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.

	Risk	Description	Mitigation
1	Schedule	Roll-outs of EAI interfaces depend on various roll-outs of external systems and task may not start on time	- 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	- 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	- 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	- Vendor to conduct formal training - Knowledge transfer during development with the vendor
5	Operation	High availability	- Thorough testing of the proposed system recovery solution and high availability solution

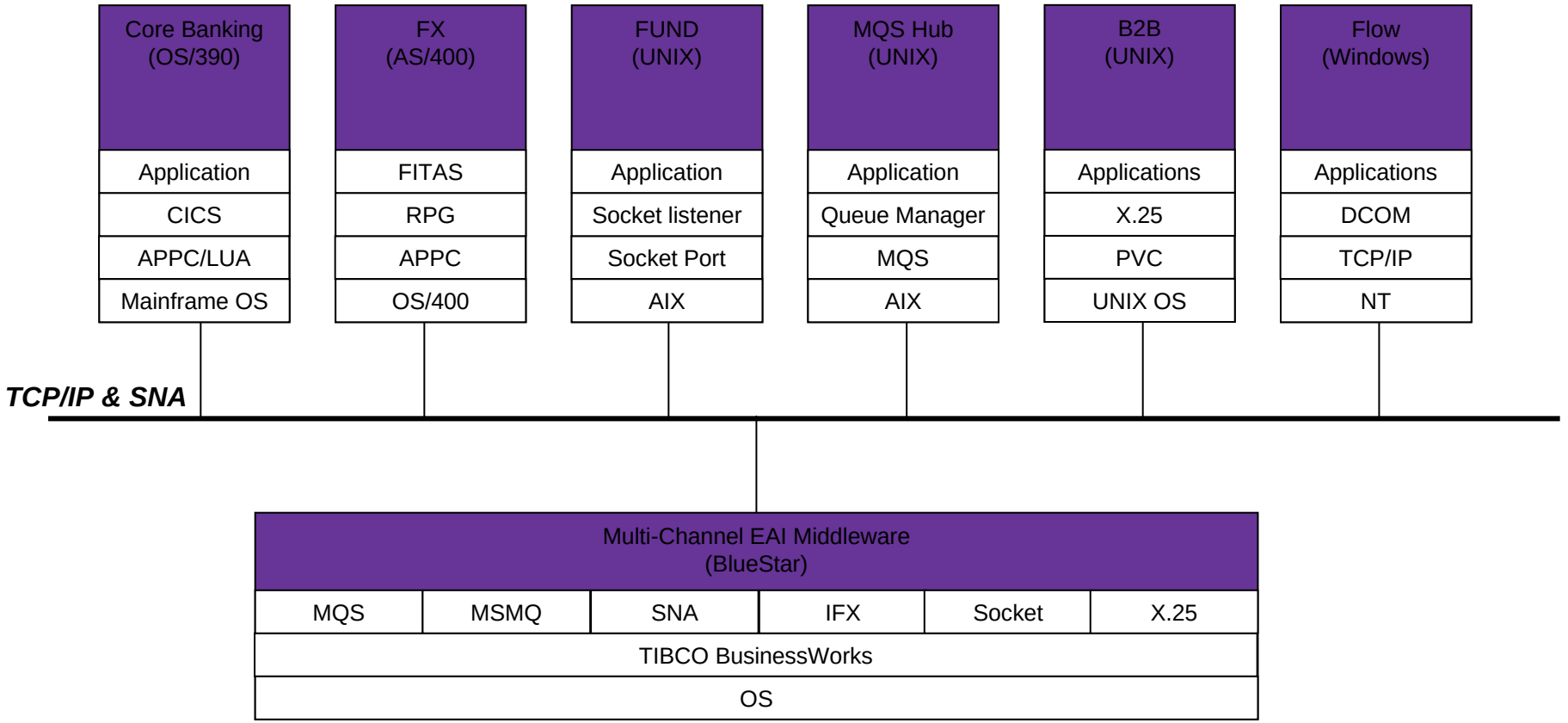
Impact on Current Applications

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

	Impact	Implication
1	Alight to Common Interface Approach	- Some exiting applications rely on synchronous communication - Applying asynchronous approach may require a re-design of existing process - Performance concern has to be addressed
2	Standardised Messaging Formats	- 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

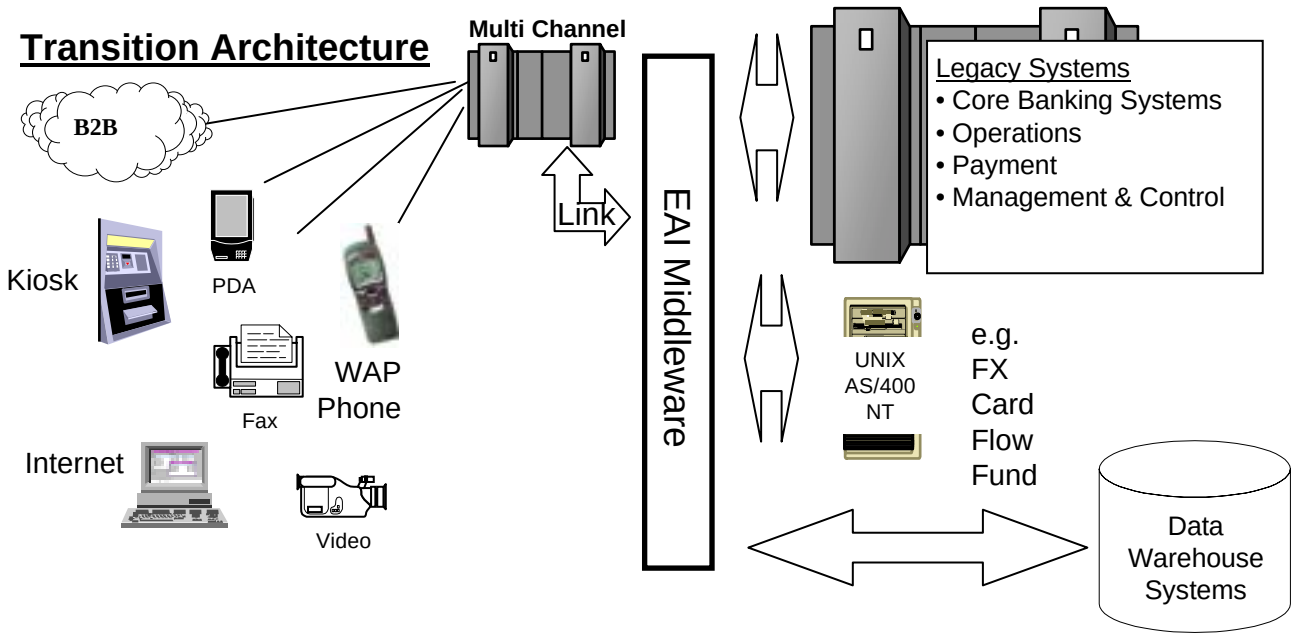
•The high level architecture component diagram for the key strategic applications that come out from the implementation phase is illustrated.



**** Physical location to be determined.***

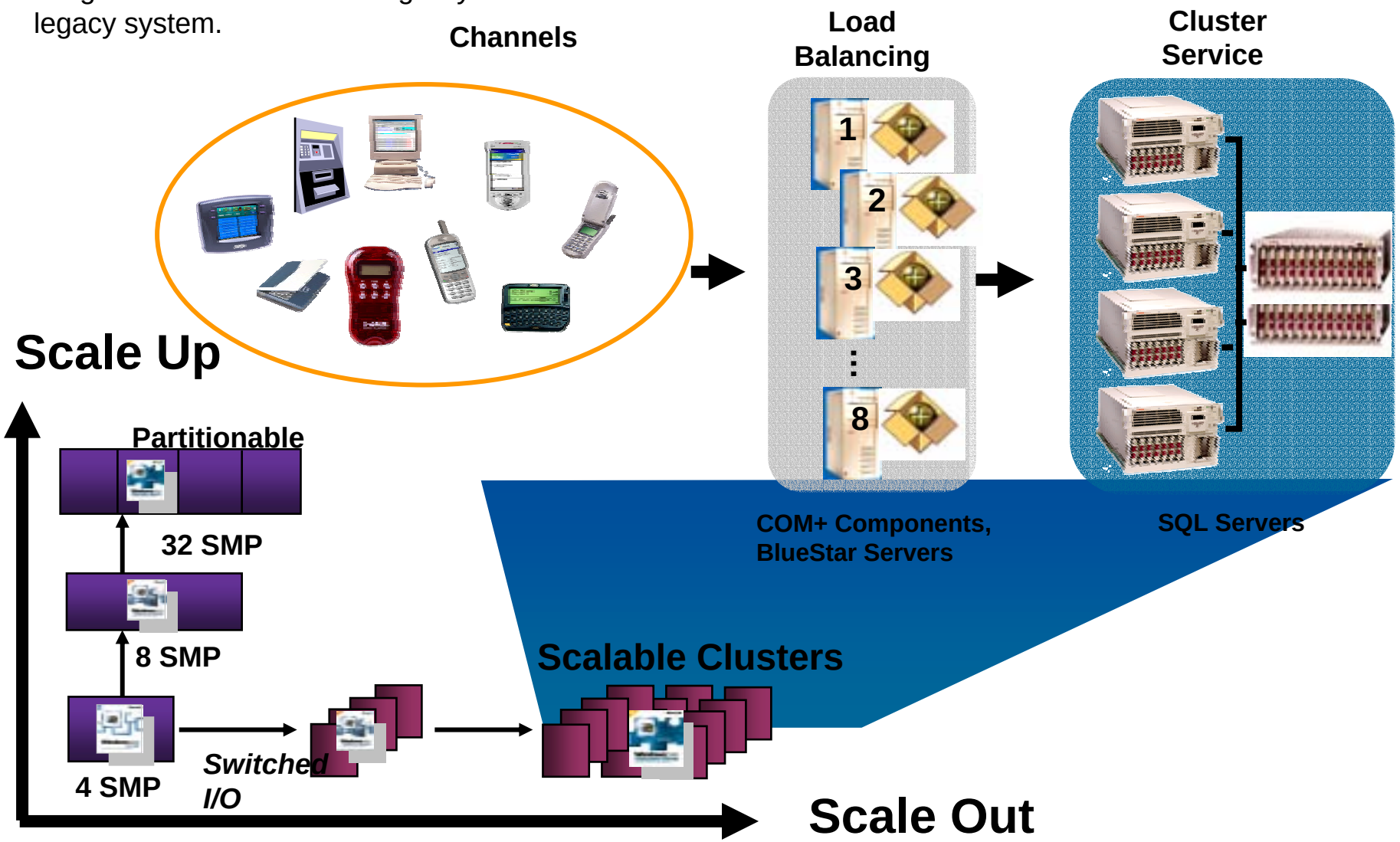
Delivery Channel Integration Architecture

•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.