



Web Services Using Microsoft .NET

Copyright © 2004 Accenture All Rights Reserved. Accenture, its logo, and High Performance Delivered are trademarks of Accenture.

Purpose

- What are Web Services?
- Why are they important?
- What is .NET anyway?
- How does .NET help implement Web Services?

Copyright © 2004 Accenture All Rights Reserved.

Web Services Overview

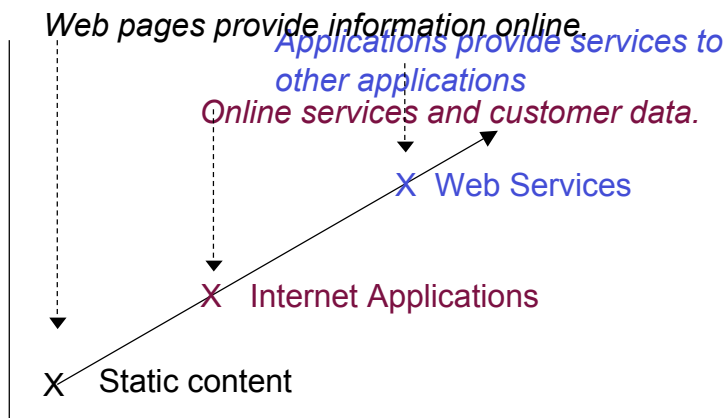


- **The Internet is evolving**
 - ☒ From today's Web sites that deliver HTML pages and centralized services to browsers
 - ☒ To next generation of programmable Web sites that directly link organizations, applications, services, and devices with one another
- **These programmable Web sites become more than passively accessed sites –**
They become reusable, intelligent [Web Services](#)

Copyright © 2004 Accenture All Rights Reserved.

3

The Evolution of the Internet



Copyright © 2004 Accenture All Rights Reserved.

4

Web Services



They are a set of technologies to standardize how applications talk to each other.

- The purpose is to connect systems, business partners, and customers over the web.
- To do this, they use open standard protocols to communicate over the Internet.
- This makes business-to-business integration more possible.

Web Services are:



- Platform independent
- Language neutral
- Web services “expose” functionality to potential users through emerging Web technology standards (XML, SOAP, WSDL, and HTTP)
- The services are computer to computer (not computer to browser)

Web Services are:



- XML Web services provide a way to describe their interfaces in enough detail to allow a user to build a client application to talk to them. This description is usually provided in an XML document called a Web Services Description Language (WSDL) document.
- XML Web services are registered so that potential users can find them easily. This is done with Universal Discovery Description and Integration (UDDI).

Copyright © 2004 Accenture All Rights Reserved.

7

Web Services Will Generate Value in Three Areas



I. The facilitation of communication between applications within an enterprise.

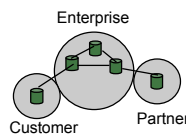


Enterprise

The potential for ubiquitous integration

- ✓ All systems speaking to one another without the complexity of eAI.
- ✓ Single framework for both internal and external interface requirements.
- ✓ Faster and cheaper to deploy and maintain solutions.

II. The interconnection of applications between enterprises.

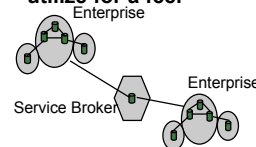


Customer Partner

Partner and customer data interchange

- ✓ Reduces the cost and complexity to establish data interchange.
- ✓ Common standards enable new trading partner integration.
- ✓ Faster and cheaper to deploy and maintain solutions.

III. Enabling applications to be "published" as a service for others to utilize for a fee.



Service Broker Enterprise

Publish applications as a "service"

- ✓ Companies can publish applications that can be utilized by others for a subscription fee.
- ✓ Buyers can incorporate specialized applications as outsourced service.
- ✓ Enables the vision of B2B Exchanges and hosted applications.

Copyright © 2004 Accenture All Rights Reserved.

8

Integration Example



Central Registration

Mainframe/DB2
COBOL

- Filing Requirements
- Locations and Addresses

Customer Accounting

UNIX/Oracle
Java

- Financial Transactions
- Penalty and Interest Calculations

CRM

Windows 2000/SQLServer
C#

- Contacts

Collections

Windows 2000/SQLServer
(VisualBasic)

- Billing
- Case Management
- Workflow

Copyright © 2004 Accenture All Rights Reserved.

9

Types of Integration



Client-to-client:

“Smart” clients (devices) can host and apply Web services that allow data to be shared anywhere, and any time.

Client-to-server:

XML Web services can share data from a server application to a desktop or mobile computing device via the Internet.

Server-to-server:

XML Web services provide a common interface between existing applications within an environment of independent servers.

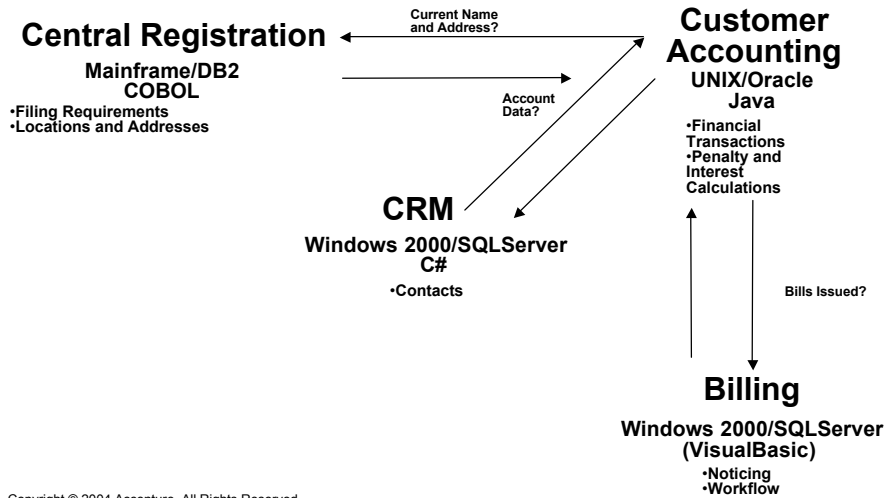
Service to Service:

XML Web services can work together in sequence to create a more complex data operation.

Copyright © 2004 Accenture All Rights Reserved.

10

Service-to-service Integration



Delivering Value



- Lower the costs of application development, system integration, and application maintenance.
- Flexibility to accommodate future technology changes by using universally agreed upon interfaces.
- Share information and applications easily and quickly with partners.
- Focus on core competencies.

What is holding us back?



Our own creativity!

As business and innovation leaders, we need to think beyond the way business is currently conducted.

- What services do we provide?
- Can we provide these services automatically?
- Who wants our data?
- How can we expose our information securely to provide those services?

Revenue Ideas?



- What are your core competencies?
 - Document processing
 - Customer accounting
 - Billing
 - Collections
- What internal applications do you wish shared data better?
 - Licensing and Tax Registration
 - Customer Registration with CRM
 - Taxpayer Accounting with Billing or Workflow
 - Taxpayer Accounting with your General Ledger
- What services could you expose to customers or suppliers?
 - Payment plans
 - Return history
 - Confirm estimated payments and credits.

The .NET Vision



- .NET is Microsoft's vision for the second generation of Internet computing. The key elements of this vision are
 - Software as a service (Web Services based on XML, SOAP)
 - Internet as delivery mechanism (HTTP)

The .NET Products



- .NET is Microsoft's products for Web-services based connectivity.
 - Server Software to host
 - Developer tools to build
 - Applications to use.
- .NET provides an IT architecture for web-services.

Key .NET Components



- .NET Platform Windows 2000 Operating system
- .NET Servers Enterprise leading Servers
- .NET Framework Comprehensive class libraries
- VisualStudio.NET Advanced development environment

.NET Servers



- Microsoft Host Integration Server: Access/connectivity to mainframe and UNIX applications and data.
- Microsoft BizTalk Server: XML-based business processes across applications.
- Microsoft Application Center: Deploy and manage Web applications. Managing, securing, and monitoring a cluster (set of servers that provide the same content, yielding high availability).
- SQL Server: Relational database system, supporting XML, data warehousing and data mining.
- And more...

.NET Framework



.NET Framework is the programming model for the .NET platform:

Common Language Runtime

Run-time services, including language integration and memory management

- .NET languages compile to Microsoft Intermediate Language (MSIL).
- Provides multiple language support
- Cross language reuse
- Reliable code (No memory leaks)

Class Libraries

Pre-fabricated code for the developer.

- File operations,
- Date operations,
- Data operations,
- Etc...

VisualStudio.NET



- Build applications and XML Web services that render in any browser and on virtually any device.
- Program using many different languages
 - Leverage current development skills
 - Potential to leverage existing code
- Advanced development toolset provides
 - Increased productivity
 - End-to-end Web development
 - A shorter time to delivery.

What Can I Build?



Visual Studio.NET supports development of:

- Thin-client application (Web Forms)
 - A new and improved web-based development tool
- Thick-client applications (Traditional Windows Forms)
- Web Services
 - XML-based communication to Internet accessible functions
- Mobile Applications
 - Using smart interfaces that adapt to the mobile device being used
- HTML-based Web applications
- Database Integration Applications

Copyright © 2004 Accenture All Rights Reserved.

21

Where would it apply to you?



- **New Internet Applications:** Many of you are building or are considering building Internet applications. You should consider .NET because:
 - Provides a good infrastructure
 - Programmers and vendors are readily available
 - It is a relatively inexpensive technical infrastructure.
- **Mobile Applications:** .NET supports mobile application development using the same development tools and features as is provided for Internet or client development.
- **Migrating Existing applications:** As the solution can run multiple languages and interoperate, you can leverage current application code to migrate to a more modern architecture.

Copyright © 2004 Accenture All Rights Reserved.

22

What is the Difference Between .NET and J2EE?



- Without doubt, they are similar. The primary difference is that J2EE is platform independent while .NET is application code independent.
 - While .NET is platform dependent, but it does provide interoperability.
- JAVA is the only language for programming in J2EE, resulting in developer lock in.
- Keep in mind, you typically are not writing applications to be hosted on multiple environments. Therefore, why not write with better programming tools for a less expensive environment?

Key Points about .NET



- Common Language runtime – program in any language
- Extensive library of prefabricated code
- No more memory leaks – “garbage collection”
- Support community and programmers!

- You write better code with fewer bugs
- .NET = Developer Productivity

Summary



- Your IT strategy should consider Web Services for integrating applications.
- Be innovative.
- .NET provides an integrated, multi-language platform for rapid development of enterprise applications
 - Any CLS compliant language can target the CLR
 - .NET provides cross-language integration
- .NET incorporates a comprehensive vision of web services
 - Completeness of Vision
 - Maturity of Offering
 - Interoperability
 - Comprehensive .NET server offerings

Copyright © 2004 Accenture All Rights Reserved.

25

Thank you for your attention!

Guy.B.Kirtley@accenture.com

Copyright © 2004 Accenture All Rights Reserved.

26