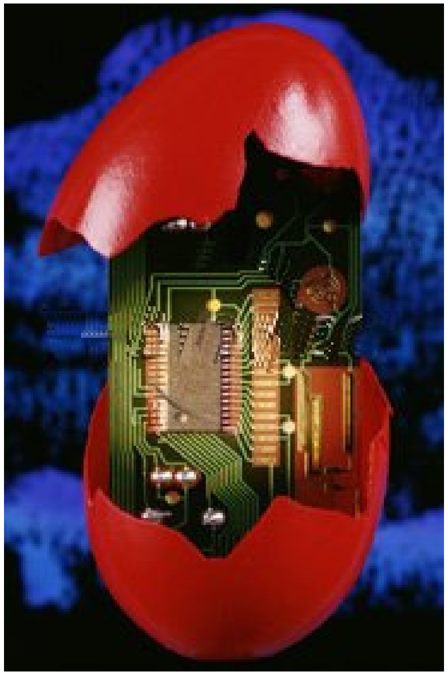
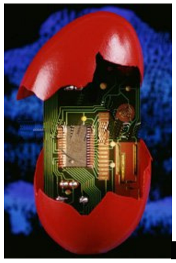
Web Services

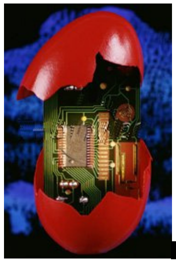
Research Capsule





Contents

- What are web services ?
- What are the sample uses ?
- Why do I need web services ?
- What do I need ?
- What are the implications ?
- What are the risks ?
- How do I implement ?
- How much is it going to cost ?
- How do I make money ?
- TCO & ROI Analysis
- Main architectures – Microsoft v. J2EE
- Main vendors in the marketplace
- Sources of information
- Sales perspective
- Market trends



What are web services ?

Basic

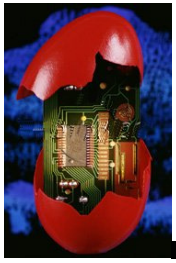
- **A web service is a loosely coupled, asynchronous interface that is exposed and invoked using platform-independent technology**

Business

- **Conceptually, web services represent a model in which small, loosely coupled pieces of application functionality are published, consumed, and combined with other applications over a network**
- **Web services are a practical implementation of a service-oriented architecture (SOA). (They are not the only one, but are the most viable)**

Technical

- **Specifically, web services are a stack of emerging standards that define protocols and create a loosely coupled framework for programmatic communication among disparate systems**
- **Scope:**
  - The core web services communication protocols are UDDI, WSDL, SOAP and XML-RPC. All web services software should support these standards
- **Key elements:**
  - Addresses the problems of communication between legacy systems, by setting standards
  - Is a key element in reducing the costs of integration (EAI, B2B etc.)
  - Enables new capabilities for the “networked world” – configuration of disparate applications/services “on-the-fly”, search and selection of “perfect” service
- **Intended audience**
  - Web services can be used by any enterprise system and any size enterprise



What are the sample uses ?

Business

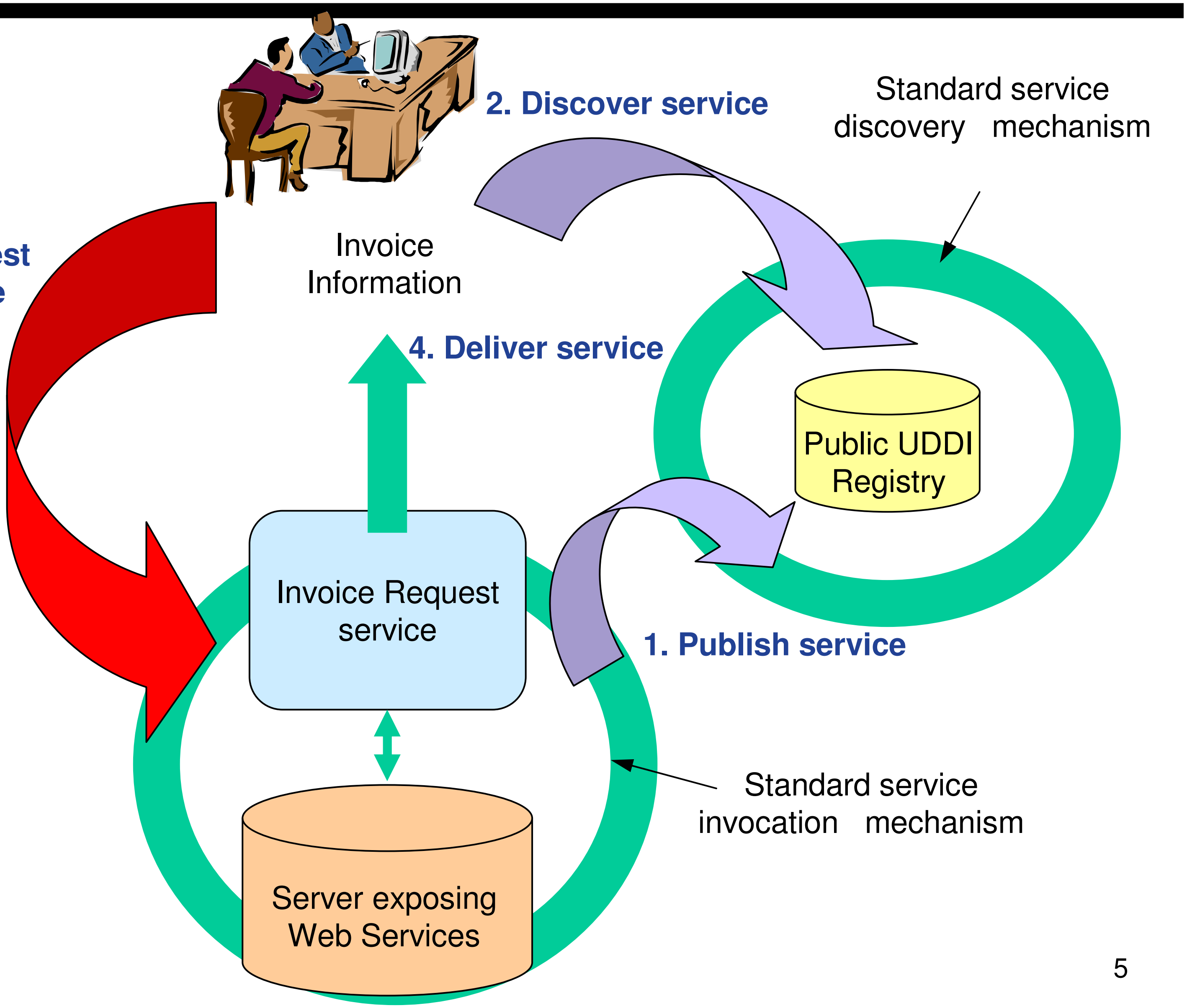
Business challenges being solved today with web services

- **Re-use and syndication**
  - Software re-use, syndication of systems
- **Automation and productivity**
  - Employee productivity, reduced errors
- **Visibility into operations**
  - Digital dashboards
- **Exploring new business models**
  - New subscription or pay-per-use models

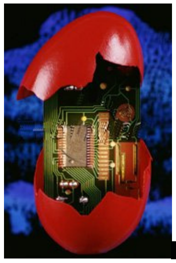
Overall, web services provide a common integration structure – both within and external to the enterprise

What are the sample uses ?

Technical



2. Discover service
Standard service discovery mechanism
3. Request service
Invoice Information
4. Deliver service
Public UDDI Registry
Need all outstanding invoices for our account
Invoice Request service
1. Publish service
Standard service invocation mechanism
Server exposing Web Services



Why do I need web services ?

Business

The next wave of enterprise computing is service-oriented

The current business requirements of IT in the enterprise are:

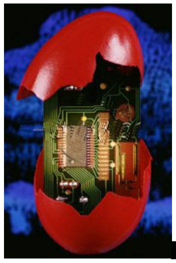
- Increased flexibility in response to changing complex business processes
- Scalable integration of disparate systems (inter- and intra-enterprise)
- Cost effectiveness
- Event-driven architecture

Web services provide the best (but not the only) means of achieving these requirements

Why do I need web services ?

Technical

- **Loose technology coupling**
  - Removes technology dependency between provider and requestor platforms
- **Transport independent**
  - Can be used with a variety of transport mechanisms to carry the message, and support alternative messaging and transaction styles
- **Late binding**
  - Links between systems resolved at run-time
- **Dynamic inspection**
  - Availability and functionality discovered at run-time
- **Programmable**
  - Call functions across the Internet
- **Use of standard protocols**
  - Not multiple proprietary technologies
- **Industry consensus**
  - On the application of web services.

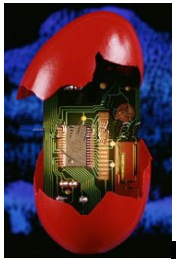


Why do I need web services ?

Statistics

- More than 90% of corporate software projects fail to achieve their intended business goals (Forrester).
- For every $100 spent acquiring software applications (build or buy), $80 is spent annually to maintain and update that application (Gartner).
- About 47% of the effort in software maintenance involves understanding the software to be modified (IEEE Computer Society Press).
- The Global 3000 allocate about 60% of their annual IT budget for application maintenance (Forrester).

- It is the next major shift in enterprise architecture



What do I need ?

Business

Movement to an SOA requires revamping existing capabilities and adding new ones

NEED:

Technology-neutral view of platform services

- ■ Your platform may not be at the centre of the universe
- ■ Still need to maintain security and service levels

SOLUTION:

An extra layer

- ■ Most current infrastructure will not fully support web services standards
- ■ Building capabilities into each component is time-consuming and not as “clean”

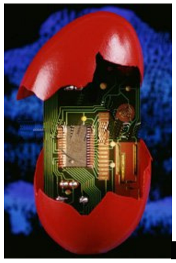
What do I need ?

Technical

Movement to an SOA requires revamping existing capabilities and adding new ones

- Authentication and authorisation
- Testing & Certification (through-out lifecycle)
- URL switching and service self-testing
- Role of UDDI registry
- Logging and event management
- Accounting
- Transformation
- Service aggregation
- Performance and availability management
- Conversation management

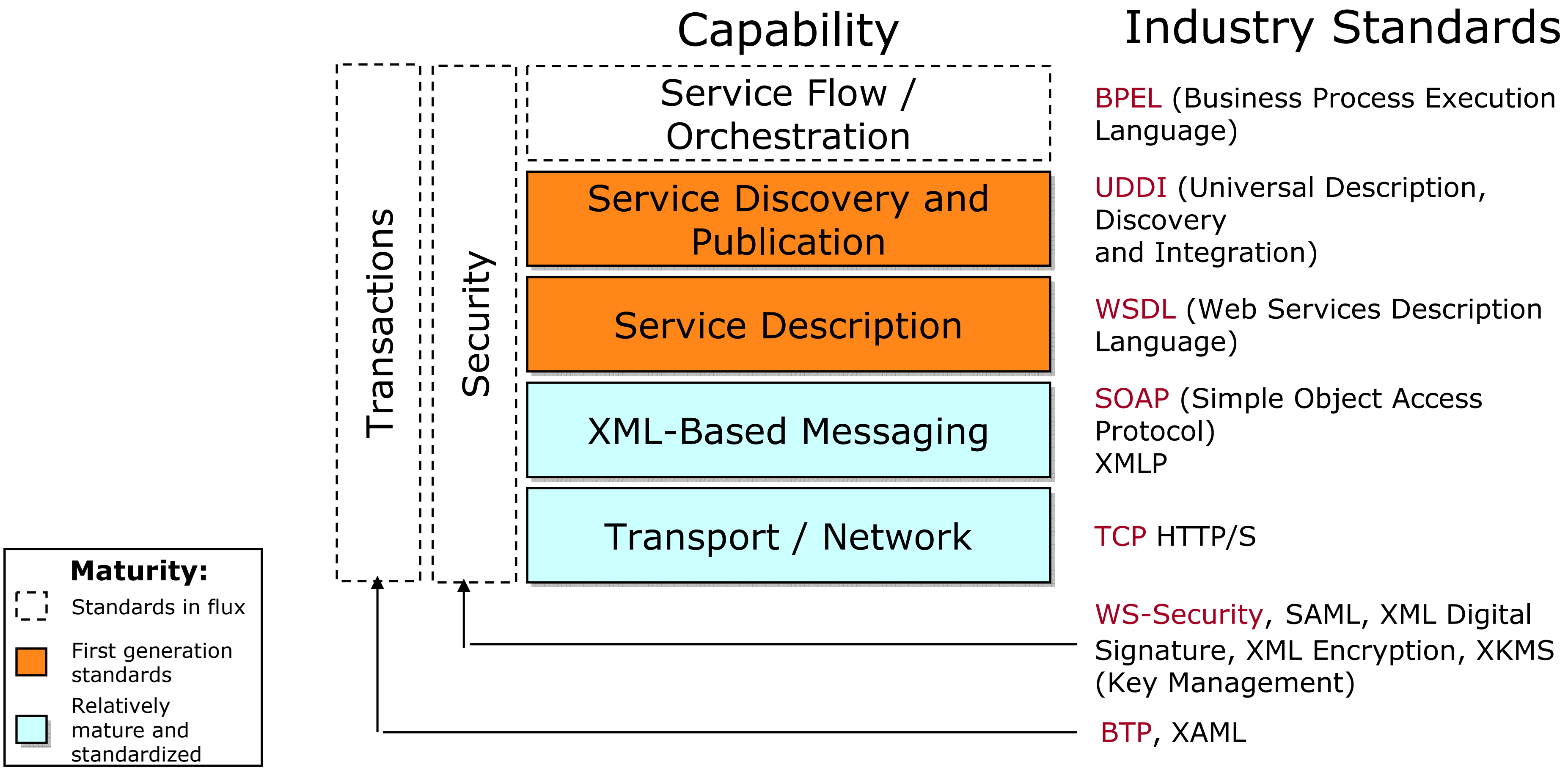
These functions are available in many packages today, but may not support web services protocols or service-oriented architecture



What do I need ?

Technical

New capabilities required for the Web Service Internet Protocol Stack



Capability

Industry Standards

Transactions

Security

Service Flow / Orchestration

Service Discovery and Publication

Service Description

XML-Based Messaging

Transport / Network

BPEL (Business Process Execution Language)

UDDI (Universal Description, Discovery and Integration)

WSDL (Web Services Description Language)

SOAP (Simple Object Access Protocol)
XMLP

TCP HTTP/S

WS-Security, SAML, XML Digital Signature, XML Encryption, XKMS (Key Management)

BTP, XAML

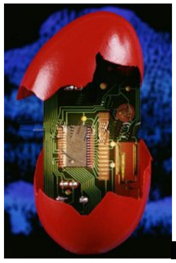
Maturity:

⬚ Standards in flux

🟧 First generation standards

▢ Relatively mature and standardized

See notes page for high-level picture of differences in architecture

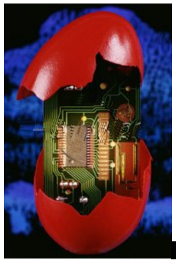


What are the implications ?

Business

- **Web services are a new communications technology**
  - Legacy apps will need to be exposed as web services
- **Web services infrastructure will be as pervasive as software itself**
  - Planning must be just as pervasive
- **Cannot over-rely on non-standardized portions of the web services stack**
  - Transaction and security protocols have only recently been ratified
- **Requires architecture underneath**
  - New capabilities for web services management
- **Requires understanding of contracts between services**
  - Design by contract
- **There are many holes to fill in and many competitors**
  - Stay with defined standards and maintain expectations during the next two years

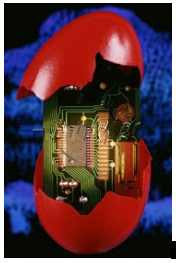
“Web services ease integration, but their current state does not replace existing EAI functionality” – *META Group*



What are the implications ?

Technical

PRINCIPLE	IMPACT	CONSEQUENCES
Enforced Encapsulation	Usage is independent of implementation	• Heterogeneity is not a barrier to interoperability • Top down (business driven) projects • Conventional "systems" boundaries no longer relevant
Self Describing	Automation and Simplification	• Automated service publishing/discovery • Focus on contract architecture, management and administration • Federated modeling, assembly and runtime tools
Open	No tower of Babel	• Industry wide loose coupling based on industry accepted protocols • Platform is irrelevant to service provision and usage • Increased consumer choice and lower acquisition cost
Dynamic	Run-time usage and behavioural modification	• Services can be discovered and used at run-time -Run time selection of services -Run time behavioural modification • Programmatic control outside the component container implemented at runtime



What are the risks ?

Business

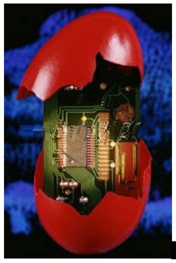
Current apps could be exposed as web services, but with no real benefits

- **If no further componentisation, applications will remain just as monolithic and unwieldy**
  - Problems with maintenance and integration remain
  - Low re-use potential of particular software, and no gains from re-use of common services
- **Applications can be easily exposed, however that does not mean they should be**
  - Intellectual capital protection
  - Security concerns in exposing enterprise to Internet
- **Not everyone will use all the web services standards**
  - Vanilla implementations may not make sense
  - But Web Standard Interoperability (WS-I) is trying to remedy this
- **Web services platforms do not necessarily contain all the features required to provide or consume web services**
  - E.g. billing, identification, encryption
- **Extension of web services capabilities to relevant applications**
  - EAI system
  - Legacy applications

What are the risks ?

Technical

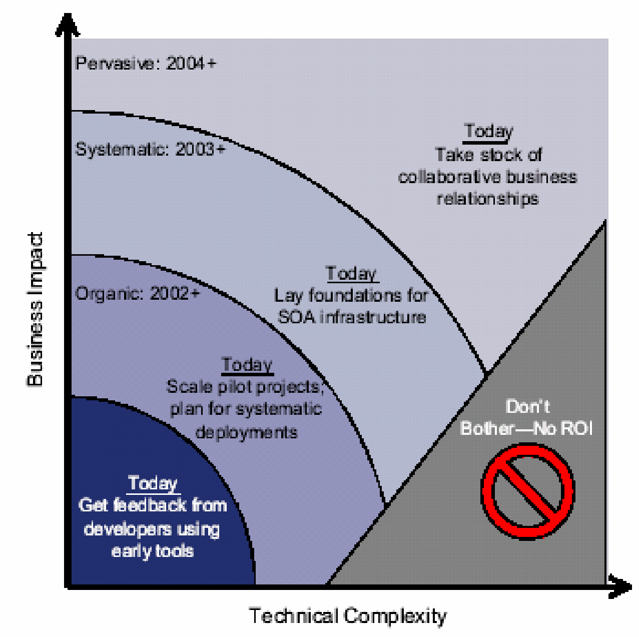
- **Security**
  - Lacking standards for strong authentication, access control, data integrity and non-repudiation though WS-Security specification is starting to address these issues
- **Guaranteed message delivery**
  - HTTP does not provide a guaranteed or reliable messaging mechanism
- **Two-Phase Commit**
  - Difficult to support in a loosely-coupled architecture
- **XML Semantics**
  - Organizations must map their XML semantics before seamless interoperability can occur
- **Complexity of testing, monitoring and administering web service connections**
- **Lack of Internet predictability and reliability**

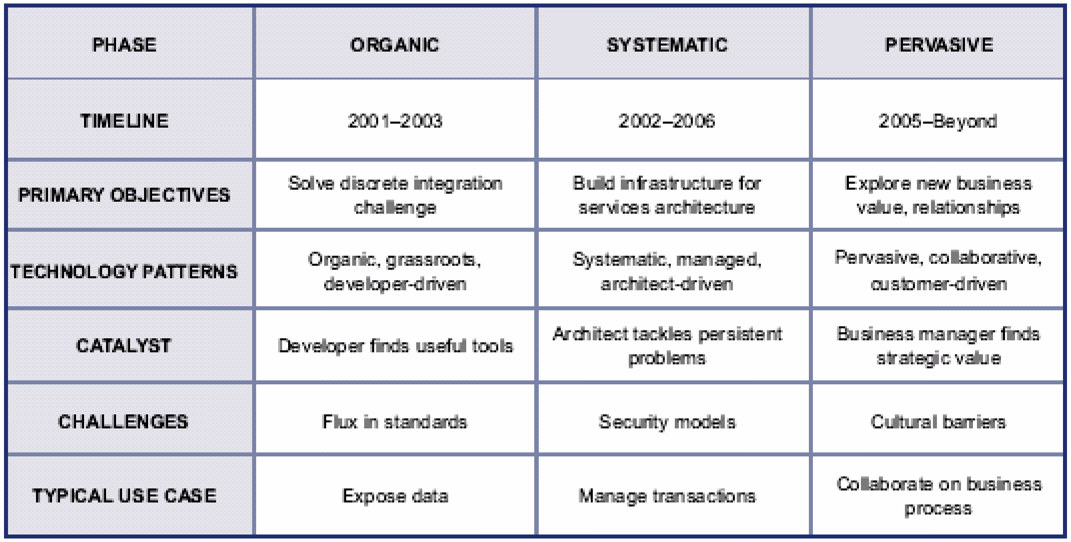


How do I implement ?

Business

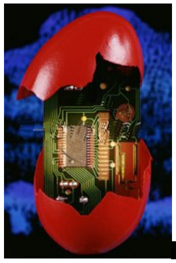
- Phase 1: Organic adoption of web services tools and standards
- Phase 2: Systematic deployment of services infrastructure
- Phase 3: Pervasive use of services in collaborative business processes


Business Impact
Technical Complexity
Pervasive: 2004+
Systematic: 2003+
Organic: 2002+
Today
Take stock of collaborative business relationships
Today
Lay foundations for SOA infrastructure
Today
Scale pilot projects; plan for systematic deployments
Today
Get feedback from developers using early tools
Don't Bother—No ROI



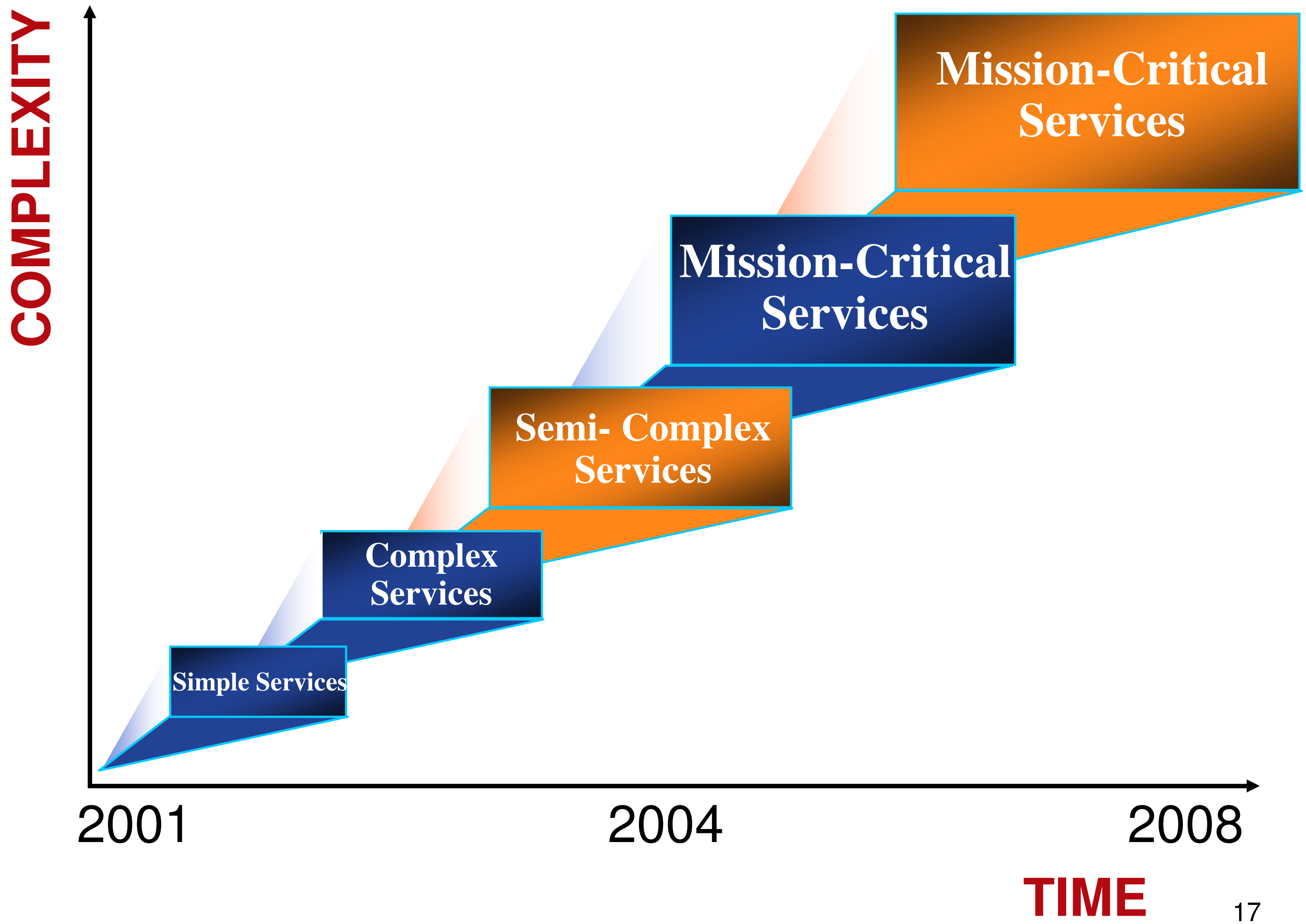
PHASE	ORGANIC	SYSTEMATIC	PERVASIVE
TIMELINE	2001–2003	2002–2006	2005–Beyond
PRIMARY OBJECTIVES	Solve discrete integration challenge	Build infrastructure for services architecture	Explore new business value, relationships
TECHNOLOGY PATTERNS	Organic, grassroots, developer-driven	Systematic, managed, architect-driven	Pervasive, collaborative, customer-driven
CATALYST	Developer finds useful tools	Architect tackles persistent problems	Business manager finds strategic value
CHALLENGES	Flux in standards	Security models	Cultural barriers
TYPICAL USE CASE	Expose data	Manage transactions	Collaborate on business process

- Start simple but plan ahead
- Incremental roll-out
- Look internally before externally



How do I implement ?

Technical



Simple Services

Complex Services

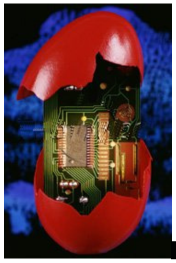
Semi- Complex Services

Mission-Critical Services

Mission-Critical Services

🟦 Internal to enterprise

🟧 External to enterprise



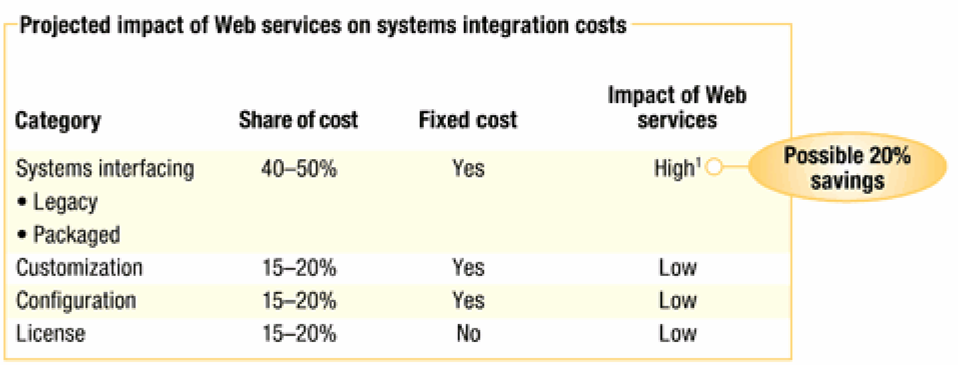
How much is it going to cost ?

- **Depends……**
  - Platform
    - Web services are platform-independent, hence it is not absolutely necessary to purchase a platform. Many applications servers, EAI platforms, have bolt-ons to support integration via web services
  - Developer tools
    - Free tools are available
    - Extensions to current tools are available
    - Platform tools also have extensions
  - Licensing
    - Since it based on standards, there are no licensing costs
  - Development
    - Theoretically applications are very easy to expose – i.e. an average is two weeks
  - Maintenance
    - Little – SOAP is ratified so it should be backwards-compatible
  - Integration
    - Web services standards are not all complete, and currently do not satisfy enterprise requirements – I.e. other technologies besides web services will need to be involved
  - Training
    - Little

How do I make money ?

Business

- **Cost savings**
  - Development
  - Integration



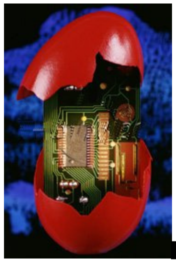
Projected impact of Web services on systems integration costs

Category	Share of cost	Fixed cost	Impact of Web services
Systems interfacing • Legacy • Packaged	40–50%	Yes	High[1]
Customization	15–20%	Yes	Low
Configuration	15–20%	Yes	Low
License	15–20%	No	Low

Possible 20% savings

Source: McKinsey

- **New revenue opportunities**
  - Reduced integration costs means that more inter- and intra-enterprise business process are feasible
    - Exposing applications
    - New composite applications possible
  - Charging for each service possible
- **Extend scope of integration**
  - Include minor business partners since low barriers to entry
  - Leverage automation to all business partners



TCO Analysis

Business

	Cost change
3 main factors that drive TCO	
- Technologies	↑ Minimal
- Processes	↑ Minimal
- People	↑ Minimal
Execution Environment	
- Software costs	↑
- Hardware costs	↑ Slight
Development Environment	
- Tools / IDEs	Remains the same
- Web services lifecycle (human aspect)	↑ Primarily management layer

+ Interoperability

- Inter-enterprise processes (e.g. B2B, outsourcing web services from 3rd parties)

- Web services add additional layer of software

- Increased complexity

ROI Analysis - Benefits

In the long term……

Business

- Faster time to market with new and improved services
- Expanded services development at lower cost
- Vastly improved business agility
- Increased process efficiencies due to B2B collaboration
- Reduced human error due to business process automation

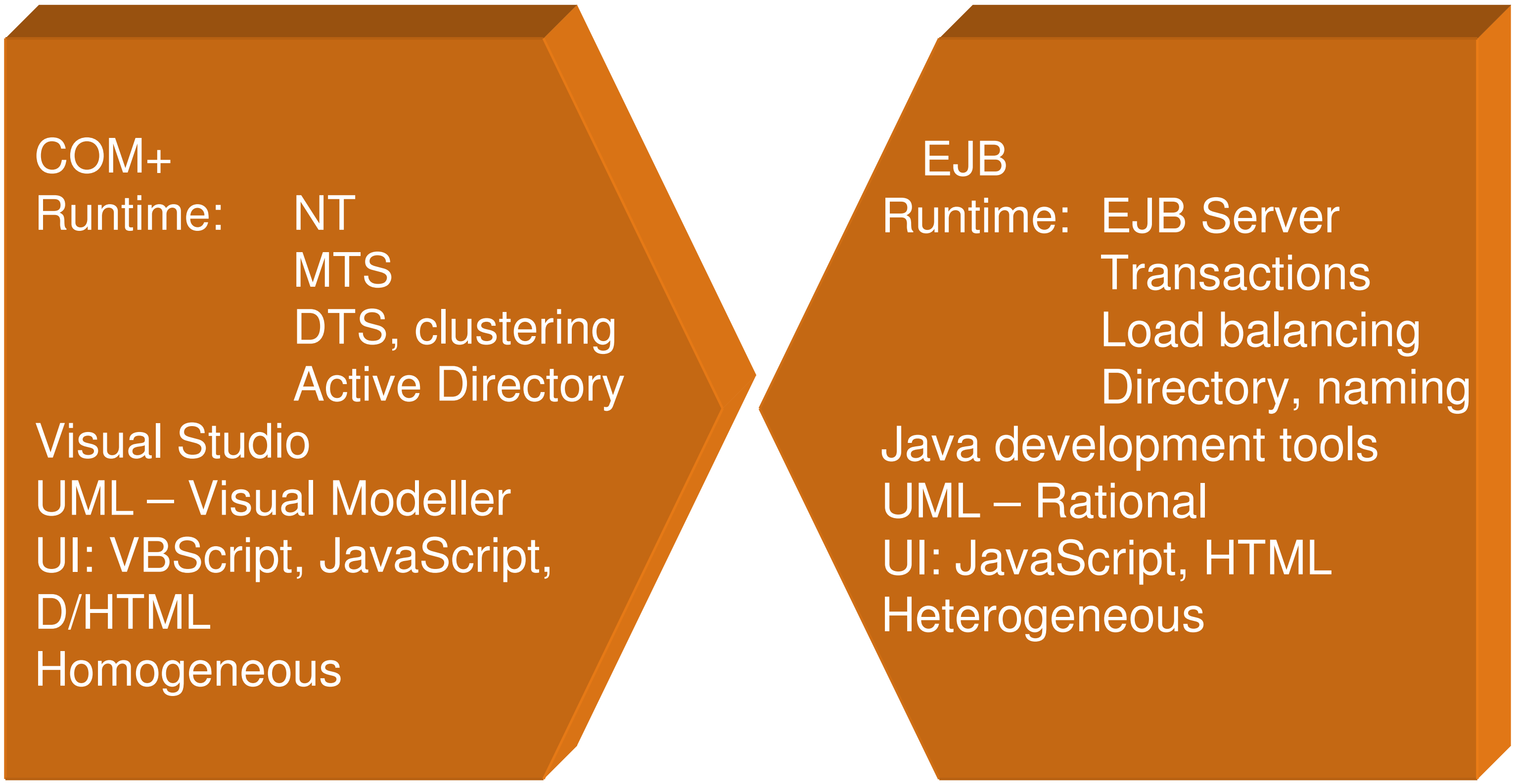
Technical

- Lower integration costs, mainly attributable to a broad base of public standards
- Increased speed of deployment as a software services industry gains momentum
- Less coding requirements needed, owing to the ability to licence more services from 3rd party vendors
- Less internal human resources required

- Enterprise can build understanding around the Services-Oriented Architecture
- Technical interoperability

Main Architectures – Microsoft v. J2EE

Technical



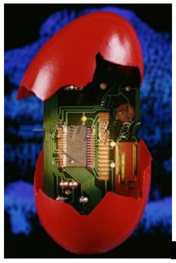
Microsoft

COM+
Runtime: NT
MTS
DTS, clustering
Active Directory
Visual Studio
UML – Visual Modeller
UI: VBScript, JavaScript, D/HTML
Homogeneous

The World

EJB
Runtime: EJB Server
Transactions
Load balancing
Directory, naming
Java development tools
UML – Rational
UI: JavaScript, HTML
Heterogeneous

“Infrastructure decisions must include choosing an application architecture – Microsoft or J2EE” – *META Group*



Main vendors in the market

Business

Incumbent heavyweights

- BEA
- IBM
- Microsoft
- Oracle
- Sun

Incumbents who have focused on supporting web services

- Actional
- Altoweb (Dev Tool)
- Epionet (Dev Tool)
- Iona
- Iopsis
- OpenLink Software
- SilverStream

New Entrants

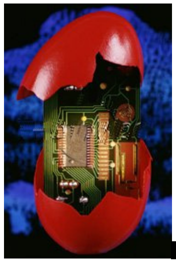
- Avinon
- Blue Titan
- Bowstreet
- Flamenco Networks
- Grand Central
- Infravio
- Interkeel
- Primordial
- Systinet
- Talking Blocks
- West Global

Development Tools

- Bind Studio
- Cape Clear

Sources of information

- CBDiforum – gives detailed information about component-based development issues and web services. Fairly technical, but applies knowledge to a business perspective
  - www.cbdiforum.com
- Stencil Group – focused on web services, and effect on the enterprise. More business-focused than CDBiforum
  - http://www.stencilgroup.com/
- WebServices.org – provides news, very useful analytical articles, and profiles of companies
  - www.webservices.org



Sales perspective

- **Triggers:**
  - Integration
  - Reducing TCO
  - Reducing time to deploy applications
  - Reduce cost of modifying current applications
  - Extending business processes outside the enterprise / firewall

Current Market

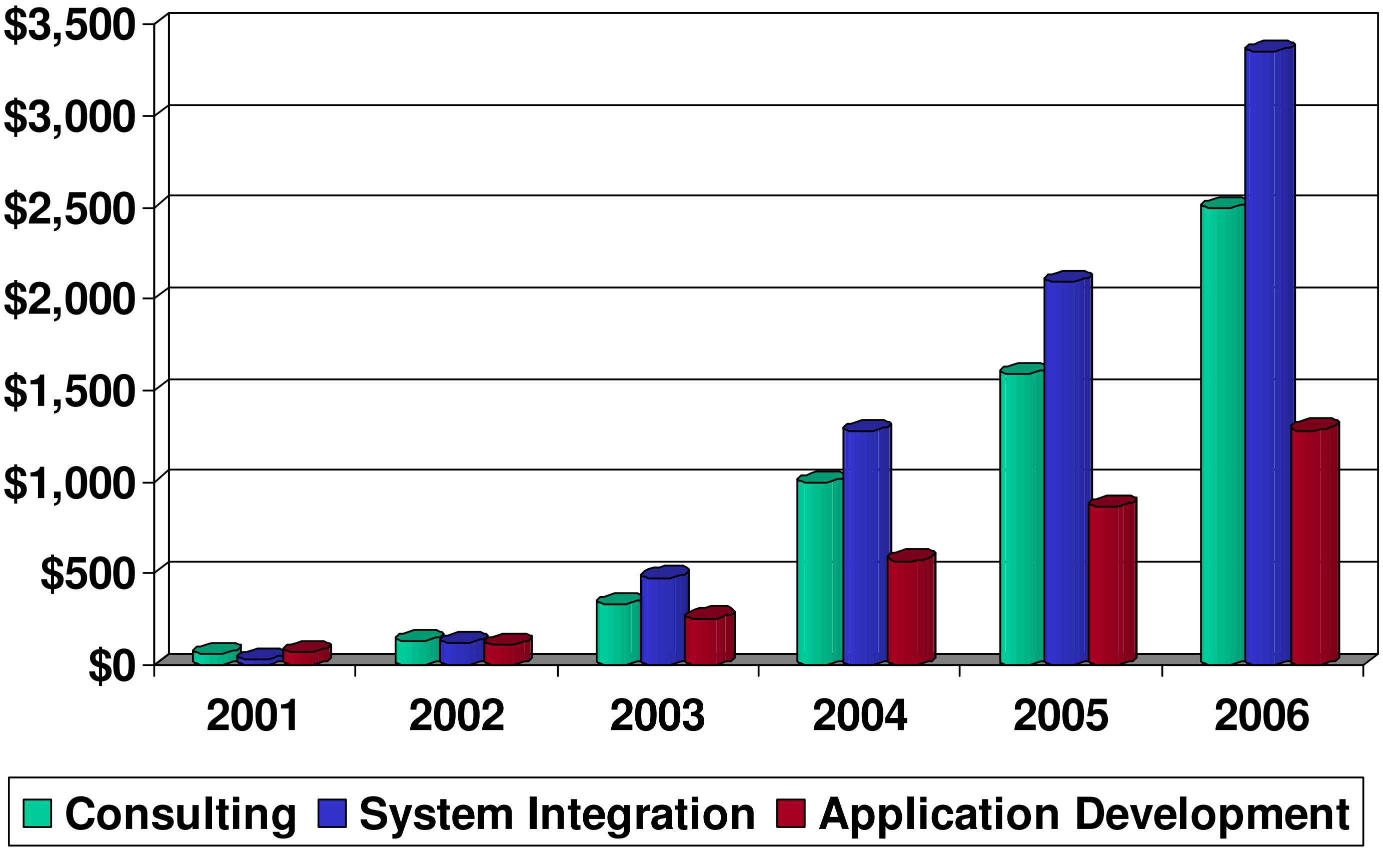
- **Research shows that web services adopted across many industries, but particular adopters are:**
  - Financial services, high-tech manufacturing, and telecom - usual suspects
  - But also automotive manufacturing, insurance, and utilities—are a bit surprising.
- **What do these different groups of companies have in common? Several factors are significant:**
  - Information intensive business processes
  - Widely dispersed suppliers and customers
  - Mission-critical applications built atop diverse, legacy IT assets

Source: The Stencil Group, April 2002

Market Trends

Business Development

Web services-related Professional Services Forecast by Engagement Types ($M)

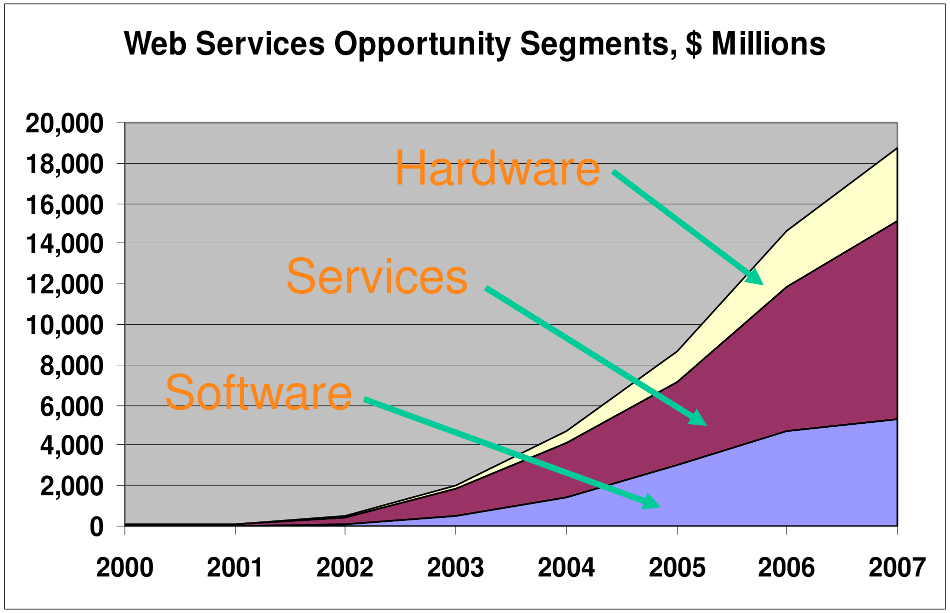


Source: IDC

Market Trends

Business Development

- Web services will become the predominant API for applications and integration by 2004 (Meta Group)
- Web services software market will reach $1.7 billion in 2003 (Gartner)
- Web services hype will outrun web services' ability during the next two years – (META group)



Source: IDC (03/02)