Application Service Providers: System Development Using Services over the Net

Kenji Takahashi
Information Sharing Platform Labs.
NTT
Tokyo, Japan
http://www.kokono.net/~kt

Wolfgang Emmerich, Anthony Finkelstein & Sofia Guerra
Dept. of Computer Science
University College London
Gower St, London WC1E 6BT
http://www.cs.ucl.ac.uk/ssegroup.html

1

"Five years from now, no one will be buying and running their own software. We'll look back on those days and laugh"

ASP - Internet Research Group

2

Outline

- Introduction
- Case Studies
- Requirements
- Architectures
- Enabling Technologies
- Development
- Conclusions

3

Learning Contract

- Understand what an application service provider is
- Appreciate the key technical and management choices
- Gain insight into system development with application service providers over the Internet, intranet, and extranet.
- Understand advantages, pitfalls and prospects of this growing and rapidly developing area

4

Introduction

5

What is an ASP?

- An Application Service Provider is a third party organization that provides access to applications to multiple customers over network based on rental or lease contracts. It deploys and manages application software, system hardware, and networking at a centralized facility on behalf of the customers

6

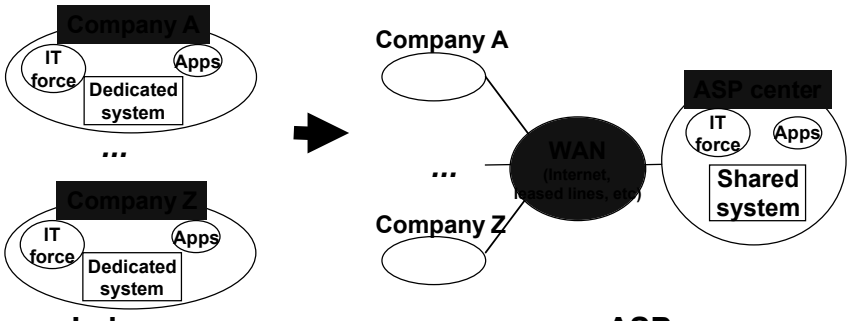
What is an ASP?

- Others said...
  - "ASPs are service firms that provide a contractual service offering to deploy, host, manage, and rent access to an application from a centrally managed facility" (Meredith McCarthy, IDC 1999)
  - "An application service provider manages and delivers application capabilities to multiple entities from data centers across a wide area network. An ASP may be a commercial entity, providing a paid service to customers or, conversely, a not-for-profit or government organization supporting end users." (ASP Consortium, 1999)

7

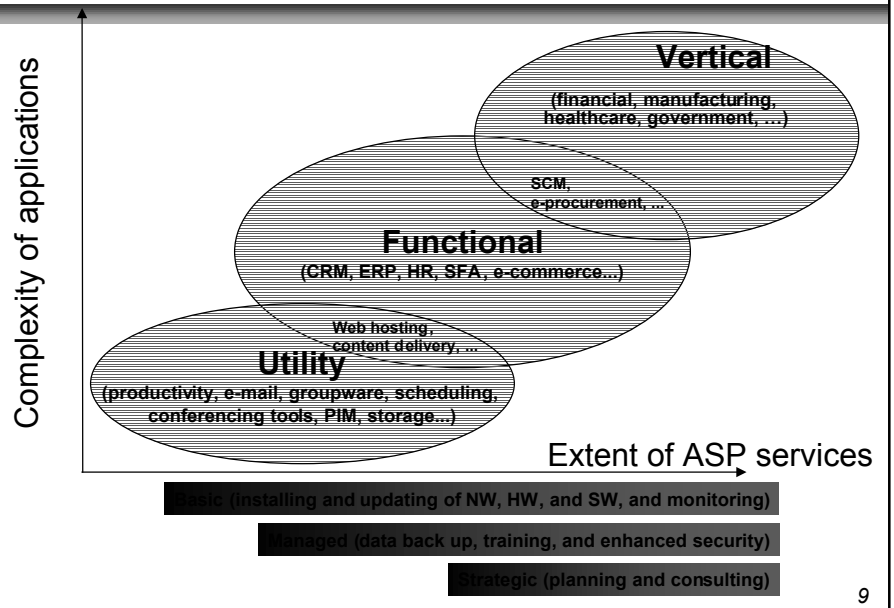
What is an ASP? (cont.)

- One to many
- One stop shopping for all IT needs applications, network, computers, and operation services)
- Rent or lease contracts



8

Taxonomy



9

Taxonomy (Cont.)

- **Taxonomy based on "how services are provided"**
  - **Web Software Vendors**
  - **Service Aggregators**
  - **Full Service Providers**
  - **Application Infrastructure Providers**

(source: "Application service providers: A market overview", Internet Research Group, 2000)

10

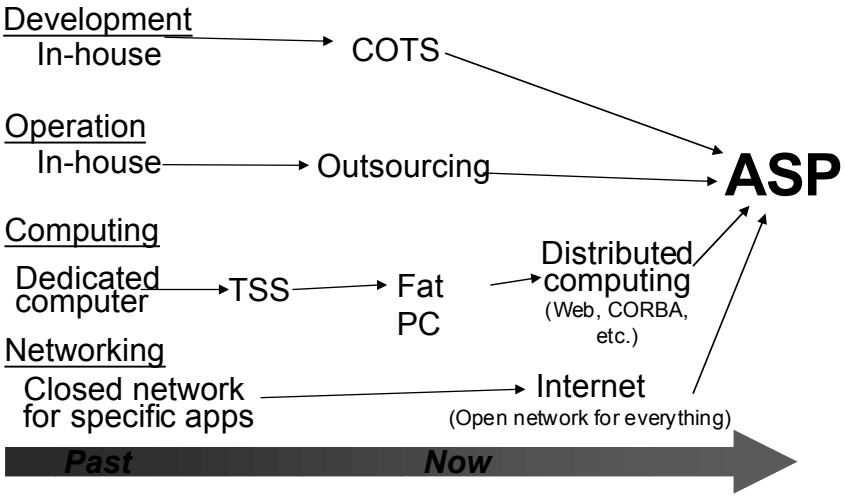
Taxonomy (Cont.)

- **Taxonomy based on "what applications are provided"**
  - **Enterprise ASPs**
  - **Local/regional ASPs**
  - **Specialist ASPs**
  - **Vertical Market ASPs**
  - **Volume Business ASPs**

(source: "ASPnews.com")

11

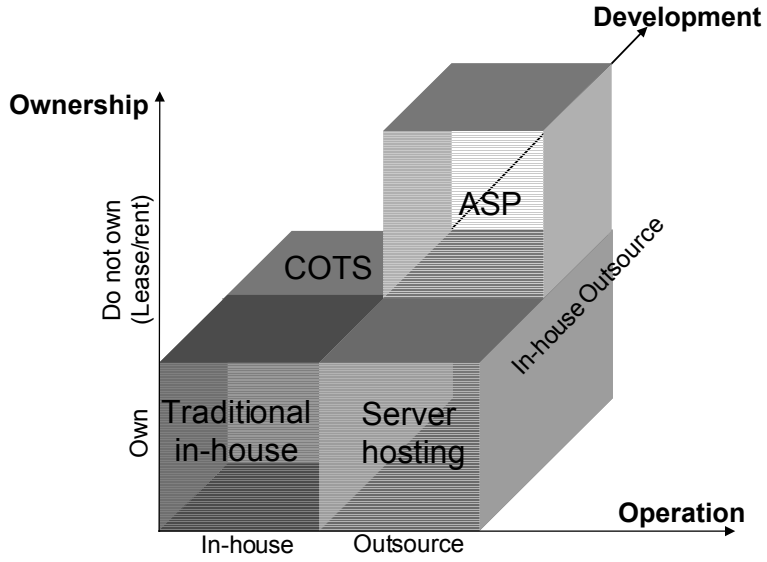
History



- ***Economic pressures to outsource non-competencies***
- ***Advances in enabling technologies, e.g. WWW, and distributed computing***

12

Software Engineering Landscape



13

Bespoke vs. ASP-based Development

Bespoke	ASP
In-house development	Packaged software
In-house management	Outsourced management
Specify everything	Specify requirements and choose implementation
Customized architecture	Standardized architecture
Computing locally	Networked computing
Traditional big firms	Rapidly growing SMEs
Large intial investment	Small set-up fee
Fixed cost	Pay per usage

14

Overview of ASP Industry

- Young and attracting many companies
  - The term "ASP" is just two years old.
  - Over 300 companies called themselves ASPs as of Jan. 2000.

    (source: "Application service providers: A market overview", Internet Research Group, 2000)
  - In 2000, 1000 ASPs will join the market.

    (source: Summit Strategies)

15

Overview of ASP Industry (cont.)

- Big companies have jumped on the bandwagon
  - AT&T
  - Microsoft
  - IBM
  - Sun
  - Oracle
  - SAP
  - …
- ASP Industry Consortium
  - Nearly 400 members world-wide as of March 2000

16

Overview of ASP industry (cont.)

- Estimated market size (world wide)
  - IDC
    - US$296 millions (1999)
    - US$7800 millions (2004)
  - Gartner Group's Dataquest
    - US$889 millions (1998)
    - US$ 2,700 millions (1999)
    - US$ 22,700 millions (2003)

17

Business Models

- ASPs sell "services", not "applications"
- Customers buy services but do not own applications
- Partnership is the name of game
  - share risks (initial investment) and revenues
  - gather expertise in different areas
- ASPs serve as a primary contact for customers
- Major model is monthly fee per user.
  - Per transaction
  - Supported by advertising - free to end-users
  - Revenue sharing with partners

18

Example of ASP Arrangement

- Utility applications (e.g. groupware and e-mail)
  - Installation : $0- 1,000 (within a few days)
  - Monthly fee: $20 per user
  - Term : 1 - 12 months
- Enterprise applications (e.g. ERP)
  - Installation : $100,000 (within a few months)
  - Monthly fee: $800 per user
  - Term : 1 - 3 years
- Customers are seeking usage-based fee structure and volume discounts

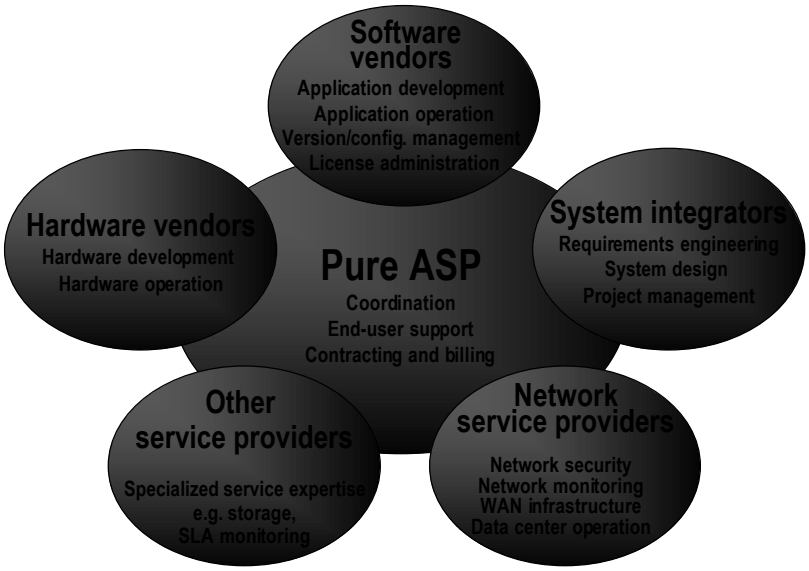
19

Service Level Agreements (SLA)

- SLA are a means for customers to have service levels guaranteed and for ASPs to distinguish their services
- ASPs are still struggling to define SLA and manage the service levels
- Typical SLA of ISPs include availability, latency, and time for error notification
- Also important are problem resolution speed and resources
- Refunds are made only upon customer claim in many cases

20

Players



21

Pure ASPs

- **Pure ASPs focus on application services**
  - **Full service providers, e.g.**
    - **USinternetworking**
    - **Breakaway Solutions**
  - **Service aggregators, e.g.**
    - **Corio**
    - **Futurelink**

22

Software Vendors

- Some application vendors are trying to become ASPs
  - Enterprise: mySAP.com, Siebel and Oracle BOL
  - Office/personal: Lotus Notes and Microsoft Office
  - Also they are starting to license their products on lease and rental contracts
- Software used by ASPs include:
  - Web servers and performance monitoring tools (e.g. BMC)

23

Hardware Vendors

- Provide hardware for ASPs
  - Computers (e.g. servers, PCs, thin clients)
  - Network equipment
  - Storage devices
- Enter ASP market through new business models
  - Revenue sharing
    - e.g. HP, Qwest, and SAP alliance
  - Pay-as-you-grow
    - e.g. SUN's "Capacity on demand"

24

Network Service Providers

- Telephone companies
  - Are becoming ASPs and/or provide network services to ASPs
  - e.g. AT&T, US West, Qwest, and NTT
- Internet service providers (ISPs)
  - Also provide Web hosting services and try to get ASPs as their customers
  - e.g. Verio, Uunet, and Concentric
- Data center operators
  - Focus on server hosting services
  - e.g. Exodus and AboveNet

25

System Integrators

- System integrators provide consulting services to introduce ASPs to enterprises
- Faster installation of standardized applications changes the role integrators play in system development
  - e.g. EDS/C2O
  - Deloitte and
  - Andersen Consulting

26

Other Service Providers

- Storage service providers
  - e.g. Storage Network provides data storage services over network
- SLA monitoring
  - e.g Candle monitors ASP performance for ASP customers and NSP performances for ASPs

27

Market Needs

- Time to services is critical - huge first movers' advantages in e-commerce
- Avoid risks in IT recruitment and retention
- Constant pressures to focus on core competencies and reduce total cost of ownership of information systems
- For start ups, renting and leasing are preferable options to improve cash flows
- Increased acceptance of IT outsourcing
- Web and Internet access is ubiquitous

28

Case Studies

Corio, USInternetworking, Futurelink & mySAP

29

Case Studies: Corio

- www.corio.com
- Service aggregator
- Provides applications including
  - SAP (ERP)
  - PeopleSoft (Finance and HR)
  - Siebel (CRM)
  - CommerceOne (e-procurement), etc.
- Architecture: client/server

30

Case Studies: Corio (cont.)

- Main customers are medium and large size companies
- Monthly fee per sheet
- Partnership with more than 27 companies
  - Networking and data centers: Exodus and Cocentric
  - System integrators: Ernst & Young and Cambridge technology partners

Case Studies: USinternetworking

- www.usi.net
- Full service provider
- Provides applications including:
  - PeopleSoft (Finance and HR)
  - Siebel (CRM)
  - Ariba (e-procurement), etc.
- Architecture: client/server

Case Studies: USinternetworking (cont.)

- Main customers are medium and large size companies
- Flat monthly fees
- USi owns and manages data centers and peer-to-peer network

33

Case Studies: Futurelink

- www.futurelink.net
- Full service providers
- Provides applications including:
  - Sales Logix (SFA)
  - Onix (CRM)
  - Greatplains (e-commerce).
- Architecture: server-based

34

Case Studies: Futurelink (cont.)

- Main customers are small and medium size companies
- Fees per hour, per day, per project, per consultant per month, or per sheet per month
- Owns and manages data centers

35

Case Studies: mySAP

- www.mysap.com
- Web software vendor
- Provides SAP products and B2B portal services
- Architecture: Web-based

36

Case Studies: mySAP

- Main customers are small and medium-size companies.
- Used by over 10,000 companies
- Partners provide implementation and hosting services
  - Corio, EDS, eOnline Inc., HostLogic Inc., IBM Global Services, Interpath Communications Inc., Qwest CyberSolutions, and Siemens Business Services

37

Challenges to ASPs

- Business models are still changing - customers tend to wait and see
- Huge early investment, then race to quickly establish ASP infrastructure
- Need to guarantee responsiveness of each of different applications, which are growing in volume and complexity
- Need to measure and guarantee the level of usability and customer satisfaction
- Trade off customization & standardization

38

Requirements

Security, Availability, Reliability, Scalability, Interoperability & Performance

39

Requirements and Architecture

- Non-functional requirements (global service properties) determine architectures (ASP)
- Changes in non-functional requirements can necessitate costly architecture changes (ASP)
- Getting these requirements right to start with is critical (End-user)
- Estimate degree in which they are likely to evolve in the foreseeable future (End-user)

40

Security

- Preserve security for all work processes of customers- from clients to servers
- Good news:
  - ASPs also can function as security information clearinghouses
    - Access to security expertise
    - Provides continuous updates on security issues
    - Education and training are most critical
  - Stronger security at hosting centers physically and operationally
  - Data back up and transaction logs for faster recovery

41

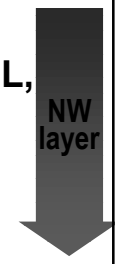
Security (cont.)

- Bad news
  - Communication between end-users and ASP centers is potentially insecure
  - Critical data is in third party hands

42

Network Security

- Trade off among cost, flexibility, and strength
- Generally the higher the layer at which network security functions, the less expensive, the more flexible, and the more vulnerable security becomes.
  - Secure application protocols (HTTPS, SSL, etc.)
  - Virtual private network (network layer)
  - Dedicated leased lines (physical layer)



43

Application Security

- Authentication
  - Identify and permit agents (both human and computer) to execute particular actions (e.g. access network and data, and use of applications)
  - Users
    - (one time) passwords, biometrics, IC card, etc.
  - Servers and files sent by servers (e.g. mobile codes)
    - digital certificates issued by third parties (e.g. RSA), etc.
- Viruses

44

Availability

- Give guarantees for service availability to end-users
- As network services become indispensable and global, services cannot be allowed to stop
- A few ASPs guarantee availability
  - USinternetworking (99.9%)
  - Breakaway Solutions (99.5%)
  - Hitachi (100%)

45

Availability (cont.)

- Typical SLA of ISPs includes:
  - Network access availability (e.g. 100% guaranteed. Return one day's fee for each hour unavailability in a month)
  - 24*7 monitoring and immediate notification of network service problems (e.g. within 15 minutes)

46

Reliability

- Define reliability level of services
- Typically measure unscheduled downtime and mean time to repair
- ASPs claim to provide cost effective, 24*7 maintenance services using shared and secured facilities
- Some ASPs provide data back-up services

47

Reliability (cont.)

- Few ASPs guarantee reliability
- Typical SLA of ISPs include:
  - No more than one hour of unscheduled downtime per year
  - No more than 40 minutes to restore network operation

48

Scalability

- End-users demands high degrees of scalability from their ASPs, because
  - their potential customer base is global
  - growth of electronic markets
- Users want scalability to be provided with smooth cost model
- Scale can grow in
  - Number of customers
  - Total number of transactions
  - Required transaction throughput
  - Transaction complexity with more demanding customers

49

Interoperability

- Interoperability is the capacity of a system to cooperate with other systems
- Service provided by an ASP may have to be integrated into a vertical business process either at user or at ASP end
- Service provision itself may have to be based on existing legacy systems

50

Interoperability (cont.)

- Requires resolution of hardware and operating system heterogeneity platforms
- Challenge is to achieve interoperability and still be able to make non-functional guarantees!

51

Performance

- Maintain service performance
  - Response time
  - Throughput
  - Time to service
- ASPs claim to be able to flexibly upgrade their capacity to meet performance requirements
- Few ASPs guarantee performance

52

Performance

- Overall usability and customer satisfaction are also important but difficult to quantify
- Typical SLA of ISPs includes:
  - Latency (e.g. less than 120 ms for a transatlantic round trip)
  - Time to service (e.g. less than 40 days after signing contracts)

53

Architectures

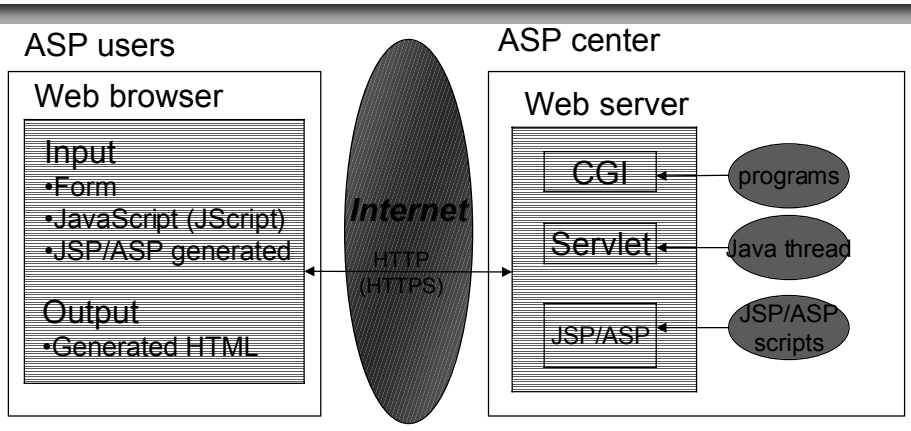
Web-based, Mobile-code,
Server & Client/Server

54

Web-based Architecture

- **Web browsers are used as clients. User interaction is implemented using forms, JavaScript, and Java Server Page (or Jscript and Active Server Page)**
- **Computing is mainly done on web-server side via HTTP/application gateway interface (e.g. CGI) or Servlet API, or server side script API (ASP/JSP)**
- **Internet is usually used for communications**
- **Examples: Web-enabled ERPs (e.g. mySAP.com) and utility tools (groupware, e-mail, personal productivity tools, etc.)**

Web-based Architecture (cont.)



- CGI invokes a process for each request
- Servlet invokes a Java thread for each request
- JSP/ASP scripts have structures similar to output HTML formats

Web-based Architecture

- Advantages
  - Web is ubiquitous
    - browsers are available on virtually any machine
    - HTTP goes through almost any network
  - Reduced education cost and faster learning curve
  - Reduced cost of software installation and maintenance on client side
- Disadvantages
  - Limited user interaction capability
  - No standardized transaction support

57

Mobile Code Architecture

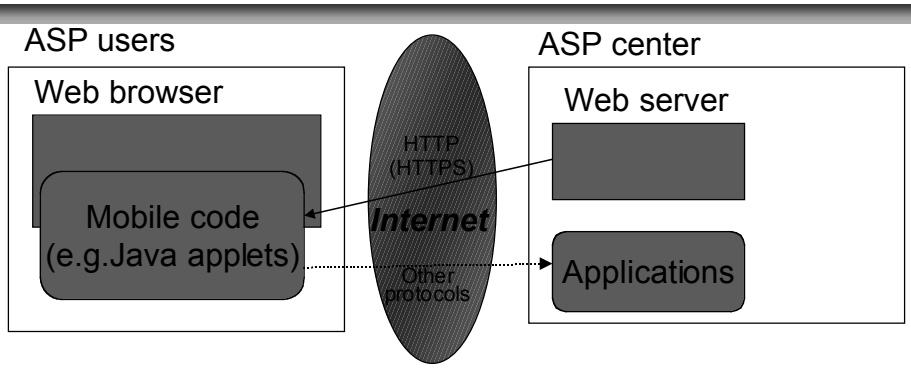
- ASPs send application programs written as mobile code to users' clients
- Mobile code transmitted across IP networks
- Mobile code mainly provides user interface functionality

58

Mobile Code Architecture (cont.)

- **Mobile code can be:**
  - **independent programs (e.g. Java applets and Active X controls).**
  - **embedded in HTML documents (e.g. JavaScript and VBScript)**
- **Examples: highly interactive applications**
  - **conferencing tools (as Java applet)**
  - **validation of complex forms (as JavaScript)**

Mobile Code Architecture (cont.)



- Mobile code is executed independently of browsers
  - GUI is not limited by browser implementation
  - Applets can communicate with other programs
- JSP/ASP can be used together

Mobile Code Architecture (cont.)

- Advantages
  - Complex, highly interactive GUIs can be implemented
  - Reduced cost of software installation and maintenance on client side
- Disadvantages
  - Slow at invocation and execution compared to “static” code
  - Security concerns can result in limited capabilities (e.g. cannot read, write, or print local data)

61

Server-based Architecture

- Applications are provided using emulated “thin” clients
- Thin clients emulate desktops (or application windows) running on servers
- Thin clients can be either software or hardware-based
- Computing, including handling of user interaction, is executed on server side

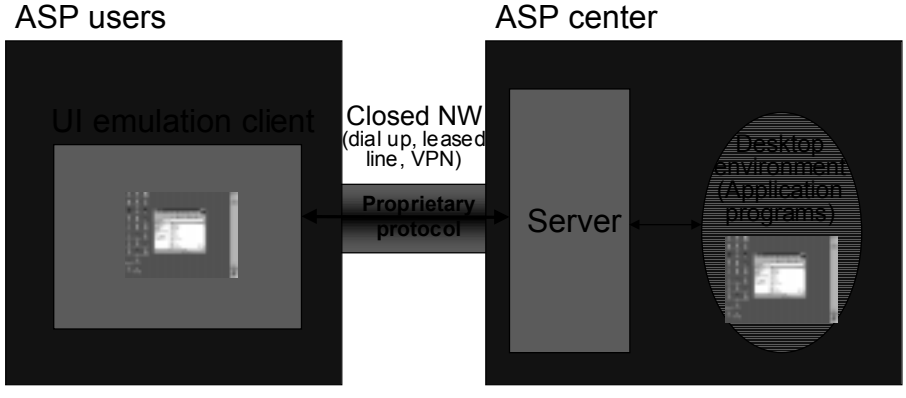
62

Server-based Architecture (cont.)

- Closed network is used for communications to facilitate security and interoperability considerations
- Examples: use of ERP systems with Windows NT Terminal Server Environment and Citrix software on Windows CE thin client machines

63

Server-based Architecture(cont.)



64

Server-based Architecture (cont.)

- Advantages
  - Existing systems can be used from multiple platforms without modification
  - Reduced cost of software installation and maintenance on client side
  - Bandwidth is not a problem
    - e.g. Citrix MetaFrame can work at under 20 Kbps
- Disadvantages
  - Loads are concentrated on server machines
  - Need highly secure network

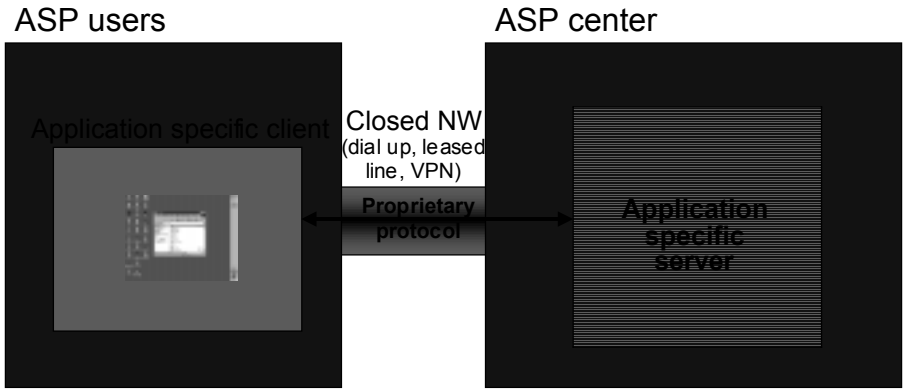
65

Client-server Architecture

- Original clients and servers are used
- Closed network is used for communication to facilitate security and interoperability
- Many enterprise ASPs still take this approach
- Examples: Corio and USinternetworking

66

Client-server Architecture (cont.)



67

Client-server Architecture (cont.)

- **Advantages**
  - Existing systems can be used from multiple platforms without modification
- **Disadvantages**
  - Cost of software installation and maintenance on client side are higher compared to other approaches
  - Need high speed, highly secure network

68

Enabling Technologies

Client, Middleware, XML, Server & Networking

69

Client

- **Web browsers**
  - **Script languages**
    - **Enable more dynamic and interactive web pages**
    - **Proprietary solutions (e.g. JavaScript, Jscript, ECMAscript)**
  - **Mobile computing**
    - **Access to Web via smart phones and PDAs**
    - **Competing standards (protocols, markup languages, and browsers)**

70

Client (cont.)

- Thin client
  - Software-based
    - Unix: Virtual Network Computer (AT&T UK Labs)
    - Windows: Windows NT Terminal Server Edition/2000, Citrix MetaFrame
    - Proprietary protocols
  - Hardware-based
    - Inexpensive thin terminals with thin-client software embedded
    - No local storage

Middleware

- Layered between Application and OS/Network
- Makes distribution transparent
- Resolves heterogeneity of
  - Hardware
  - Operating Systems
  - Networks
  - Programming Languages
- Provides development and run-time environment for distributed systems

Why not use Network Protocols?

- Manual mapping of complex request parameters to byte streams
- Manual resolution of data heterogeneity
- Manual identification of components
- Manual implementation of component activation
- No guarantees for type safety
- Manual synchronization of interaction between distributed components
- No quality of service guarantees

Forms of Middleware

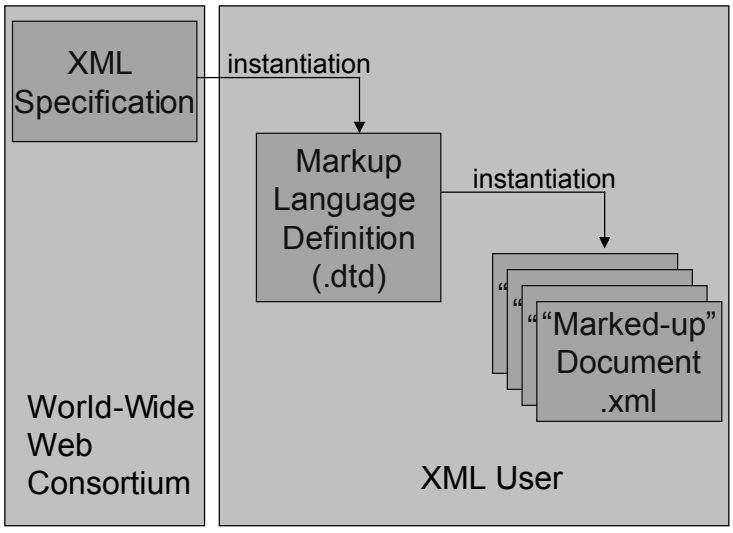
- Transaction-Oriented
  - IBM CICS
  - BEA Tuxedo
  - Encina
- Message-Oriented
  - IBM MQSeries
  - DEC Message Queue
  - NCR TopEnd
- RPC Systems
  - ANSA
  - Sun ONC
  - OSF/DCE
- Object-Oriented
  - OMG/CORBA
  - DCOM
  - Java/RMI

XML

- XML - eXtensible Markup Language
- Based on SGML and developed by the W3C Architecture Domain
- Enables definition of markup schemes for a large variety of “documents”
- Provides
  - Extensibility
  - Structure
  - Validation

75

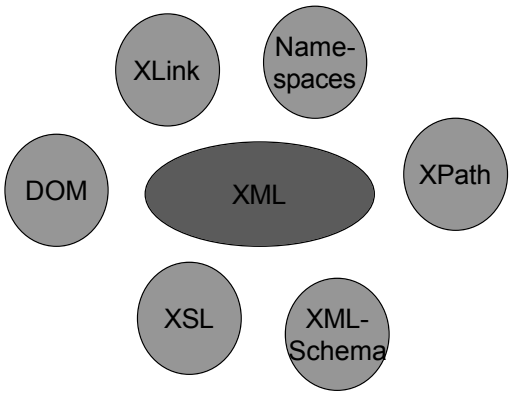
XML (cont.)



76

XML Related Technologies

- XML is not a stand-alone technology!



77

Server: Web Servers

- Applications interact with Web servers:
  - via API (e.g. CGI and Java Servlet API)
  - via server-side script computing (e.g. MS Application Server Pages/Java Server Pages)
- Scalability
  - Major sites handle more than 100 million pages per day (e.g. Yahoo handles around 235 million page views per day)
  - Various technologies are available (e.g. application servers, monitoring tools, replication)

78

Server: Web Servers (cont.)

- **Availability**
  - The server is the bottle-neck in ASP architecture
  - Our web dependent society will not tolerate Web servers to stop at any time

Server: Thin client server

- **Thin client server software handles session management, load balancing, and fail-over**
- **Application requests are processed by original application programs**
- **Scalability**
  - A high-end machine can handle 10 - 20 simultaneous users

Networking

- Dedicated leased lines
  - LANs at end-user sites are connected to LANs of ASP centers with dedicated leased lines
  - Security and QoS guaranteed at lower network layer
  - Comparatively more expensive and less flexible
  - Most enterprise ASP customers use dedicated leased lines for security reasons

81

Networking (cont.)

- Virtual private network (VPN)
  - A VPN is a private network that is virtually established on an open network (e.g. the Internet) by using encryption and tunneling technologies
  - Customer LANs are connected to ASP LANs via VPNs
  - Security and QoS guaranteed at higher network layer (L3)
  - Comparatively less expensive and more flexible
  - Alternative to dedicated leased lines

82

Networking (cont.)

- Secure Internet protocols
  - Web-based ASPs use secure Internet protocols (SSL) and digital certificates (RSA)
  - Current web browsers and servers support such protocols and certificates
  - Each user directly accesses ASP centers
  - No guarantee of end-to-end QoS
  - Inexpensive and very flexible

83

Networking (cont.)

- Storage area network (SAN)
  - Storage management in very large data centers is a significant technical challenge
  - Storage servers can be connected via optical fibers for distributed and replicated data storage
  - Extension of SAN to WAN is actively being developed to provide storage services on demand over Net

84

Development

Specification, Search, Selection, Integration & Evolution

85

Specification

- Characterize service goals and non-functional requirements
- Focus on differentiators and prioritize
- Specify acceptance tests to be transformed into SLA
- Requirements management remains responsibility of end user throughout service life
- Procurement process that emphasizes partnership and risk sharing

86

Search

- There may be more than one provider for the same service
- Complexity is likely to increase as services become more disaggregated
- Need for publication of service type descriptions (meta-data)
- Need for trading / brokering of service provision (yellow pages for ASPs)

87

Selection

- Over-specification may constrain choice unnecessarily
- Iterative process informed by the services existing in the marketplace
- Users may have to adapt their requirements to ASP offerings
- Service shopping versus service matching
- Accept service with minimal changes to gain benefits while minimizing risk

88

Integration

- Use middleware to achieve interoperability and resolve distribution
- Middleware selection problem
- May require component wrapping techniques to provide legacy components with middleware interfaces
- Bridging techniques to deal with heterogeneous middleware

89

Evolution

- Requirements of each end-user change
- Requirements of the ASP change
- Requirements changes demand evolution
- User and ASP evolution are dependent on each other
- They need to be kept in-line
- Important to define exit strategy
- Demands agreement by user and ASP on how evolution is managed and controlled

90

Conclusions

Prospects, Pitfalls and Research Challenges

91

Prospects for End-Users

- Reduced total cost of ownership
- Consistent and efficient updating of applications
- Reduced in-house IT staff
- Shorter time to start using services
- Flexible and scalable deployment of applications
- Applications available to wider user range
  - For example, small organizations can lease complete ERP systems that would not be affordable otherwise

92

Prospects for End-Users

- Corio claims average 56% reduction of total cost of ownership (TCO)
  - e.g. Vertical Networks estimates 62 % saving of TCO
    - Financials, Distribution, SFA, Integration with other company's system
    - 15 active users
    - 1120 employees
    - In house: $5.5 mm ↔ASP: $2.1 for five years
- Gartner expects 50 - 70% reduction of TCO

93

Prospects for ASPs

- Vendors
  - ASPs are a new sales channel (Vendors may give up current revenue streams)
  - Opportunities to sell products to a wider range of customers
- Network service providers
  - ASPs generate more traffic
  - Opportunities to sell value-added services such as data management
- Other service providers
  - ASPs are main sales channel or customers

94

Problems

- Immaturity of market
- Viability of ASPs
- Managing evolution and perceived loss of control
- Writing a good SLA and liability management

95

Future Trends

- Two major trends:
  - Server-based architecture for enterprise applications
  - Web-based architecture for E-commerce applications
- Widespread multi-channel computing
- Immediate challenges include
  - Defining and guaranteeing SLAs for ASPs
  - End-to-end security
  - Customization of application services
  - Migration of application services from legacy

96

Research Challenges

- Development processes
- New code mobility paradigms and technology
- Specifying non-functional requirements
- Service composition

97

Guide to ICSE 2000 - ASP Topics

- Tutorials:
  - Scalability Issues in CORBA-based Systems
  - Understanding Code Mobility
- Future of Software Engineering:
  - Software Engineering and Middleware
  - Software Engineering for Mobility
  - Software Engineering for Security
  - Software Engineering for Performance
  - Software Reliability and Dependability

98

Guide to ICSE 2000 - ASP Topics

- **Workshops:**
  - Component-based Software Engineering
  - Continuing Collaborations for Successful COTS Development
  - Software Engineering Over the Internet
  - Economics-driven Software Engineering Research
- **Technical Programme Sessions**
  - Components and COTS
  - Component-based Software Engineering and the Issue of Trust - Panel

How to find out more?

- **Contact the authors**

Kenji Takahashi
(kt@nttlabs.com)

Wolfgang Emmerich
(w.emmerich@cs.ucl.ac.uk)

Anthony Finkelstein
(a.finkelstein@cs.ucl.ac.uk

References

- Web sites
  - ASP Industry Consortium
    - http://www.aspindustry.org
  - ASP Island
    - http://www.aspisland.com
  - ASP News Review
    - http://www.aspnews.com
- Textbook

  "Engineering Distributed Objects" by W. Emmerich

  (http://www.distributed-objects.com)

101

References (cont.)

- The following companies provide comprehensible, free white papers and case study reports
  - Corio: www.corio.com
  - USinternetworking: www.usi.com
  - Futurelink: www.futurelink.net
  - Breakway Solutions: www.breakaway.com
  - IDC: www.idc.com/Store/Free/forms/five_form.htm
  - Internet Research Group: www.irgintl.com/publications.shtml

102