Cloud Services and Service Oriented Architectures

Software Engineering Considerations

Len Cayetano
Personal Information

- Software Engineer, KYOCERA Document Solutions Development America
- Email: lcayetano@alumni.usc.edu
- 20 Plus Years in Software Engineering
  - Embedded Systems, Internet, Storage, Corporate Tax Software, Program Management, Engineering Management, SOA
- Double Alumnus, University of Southern California
  - MS Software Engineering, Viterbi School of Engineering
  - MBA International Business, Marshall School of Business
- Loyola Marymount University, Los Angeles
  - MS Computer Science
- DePaul University, Chicago
  - BS, Dual Major in Mathematics & Computer Science with Honors
TOPICS

Important Trends in Information Technology

Cloud Services

Service Oriented Architecture (SOA)

Case Study on Cloud R&D Development
SECTION 1

IMPORTANT TRENDS IN TECHNOLOGY
General Thoughts on IT Trends

1. There are significant trends and driving forces that will affect the future of engineering and indeed society
2. The Cloud is an important technology and it is creating new and exciting opportunities for innovation
3. Service Oriented Architectures (SOA) is maturing and is proving to be very scalable and reliable
4. Mobility is a trend that will impact the continued proliferation of embedded devices
5. Smart Machines may be the next disruptive technology
6. Privacy and security will continue to be ongoing concerns
7. Software Engineering will continue to play an increasing role in the advancement of technology
Evidence of the New Technological World Order

Mobile Diversity

3D Printing

Mobile Apps

October 12, 2015
Additional Evidence of the New Technological World Order

Smart Machines

October 12, 2015

Cloud Services and Software Oriented Architectures-7
Further Evidence of the New Technological World Order

Your Body

Your Home

Your City

Industry

Environment

References: Gartner, 2013. Published by Michael Cooney in InfoWorld. See [1]
Also see http://postscapes.com/internet-of-things-examples/

October 12, 2015
Four Driving Forces (Gartner)

- These driving forces will:
  - Continue to **drive change** and create **new opportunities**
  - Create **high demand** for advanced programmable infrastructure
  - Create the need for programmable infrastructure to execute at **Web-Scale**

Insightful Observations on Big Data
(John E. Kelly and Steve Hamm)

- Society is faced with a torrent of data
- Data includes:
  - Structured data such as transaction records
  - Unstructured data such as still images, video, audio, and sensor data
- Biggest new source of data is Internet of Things (IOT)
  - Data produced by sensors but harvested via the Internet
- Sensors range from:
  - Radio Frequency Identification (RFID) tags retailers use to track merchandise
  - Video cameras that capture the flow of traffic
- Every day, human beings generate about 3 exabytes of data
- By 2020, there will be a data universe of 40 zettabytes (1 followed by 21 zeroes)

Source: John E. Kelly and Steve Hamm, See Reference [1], Pages 43-67
Additional Observations on Big Data
(John E. Kelly and Steve Hamm)

- 1 zettabyte holds approximately 250 billion 2-hour High Definition (HD) movies
- Flood of information should be extremely valuable
- Profusion of data is difficult to capture, make sense of, and move around
- Less than 1% of digital data have been collected and actually analyzed
- New generation of tools must be designed to handle the 4 Vs of Big Data
  1. Volume: Amount of data is increasing rapidly
  2. Variety: Video, Geospatial, Web Pages, Speech
  3. Velocity: May have to analyze data in motion
  4. Veracity: Accuracy of data & conclusions are critical

- Big Data is a digital expression of life in the raw

Source: John E. Kelly and Steve Hamm, See Reference [1], Pages 43-67
Strategic Outlook for 2020 (Gartner, Kelly, Hamm)

Total economic value add for “Internet of Things” will be $1.9 trillion dollars in 2020.

There will be up to 30 billion devices connected with unique IP addresses.

By 2020, there will be a data universe of 40 zettabytes (Kelly, Hamm).

Will benefit and impact wide range of industries including healthcare, retail, transportations.

References: Gartner, 2013. Published by Michael Cooney in InfoWorld. See [2]. Also see Kelly, Hamm in Reference [1].

October 12, 2015
Gartner’s Top 10 Trend for 2015

1. Computing Everywhere
2. The Internet of Things
3. 3-D Printing
4. Advanced, Pervasive and Invisible Analytics
5. Context-Rich Systems
6. Smart Machines
7. Cloud/Client Computing
8. Software-Defined Applications and Infrastructure
9. Web-scale IT
10. Risk-Based Security and Self-Protection

Three Main Insights from Gartner’s 2015 Technology Trends

- Real World and Virtual World Are Merging!
- There will be Intelligence Everywhere!
- New IT Realities are Emerging!

## Review Questions

1. What are the four driving forces that will drive technological change in 2015?

2. What are the top ten megatrends in technology in 2015? Briefly describe each megatrend.

3. Describe four driving forces that are spurring rapid changes in innovation.

4. What is the difference between the “Internet of Things” and the “Internet of Everything?” What are the implications for the future of embedded systems?

5. What is an exabyte of data? What is a zettabyte of data? What is a petabyte?

6. Define Big Data. What are the four Vs of Big Data? Why are they important?
REFERENCES


SECTION 2

CLOUD SERVICES
## History of Computing

<table>
<thead>
<tr>
<th></th>
<th>Cloud 1</th>
<th>Cloud 2</th>
<th>Cloud 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E-business Services</td>
<td>IT as a Service</td>
<td>Everything as a Service</td>
</tr>
<tr>
<td>1990</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1995</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Primary Forcing Function
- 1990: Internet-based supply chain integration and e-commerce
- 1995: Consumer Internet services
- 2000: Low Cost IT
- 2005: Pervasive business and consumer services
- 2010: Pervasive business and consumer services
- 2015: Pervasive business and consumer services
- 2020: Pervasive business and consumer services

### Technology Orientation
- 1990: Web-based app design
- 1995: EAI & message bus integration
- 2000: Internal protocols
- 2005: 3-tier architecture
- 2010: Cloud-based technology platforms
- 2015: Data oriented, context aware services
- 2020: Data oriented, context aware services

### IT Organization Design
- 1990: Organized around technology domains
- 1995: Technology-centric
- 2000: Organized around service supply chain
- 2005: Service-centric
- 2010: Service-centric
- 2015: Service-centric
- 2020: Service-centric

**Source:** Reference [1], Page 5 (Figure 1.1 - History of Computing) and Page 6 (Figure 1.2 - Clouds Past, Present, and Future).

October 12, 2015
Definition of the Cloud

- A network of data centers, each composed of many thousands of computers working together
- These computers in the Cloud perform the functions of software on a personal or business computer by providing the users access to powerful applications, platforms, and services delivered over the Internet
- It is a set of network enabled services that is capable of providing scalable, customized and computing infrastructure on demand
- These services can be accessed in a simple and pervasive way by a wide range of geographically dispersed users
- The Cloud also assures application based Quality-of-Service (QoS) guarantees to its users

Source: Reference [1], Pages 3-4.
Definition of Cloud Computing

- National Institute of Standards and Technology defines Cloud Computing as:

  "A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." (See Reference [1], page 3. Also see original source in Reference [2]).

- Key features of Cloud Computing
  - Agility – need to be able to re-provision resources rapidly and inexpensively
  - Location Independence – need to access resources from anywhere and everywhere
  - Multi-tenancy – resources are shared among a large pool of users
  - Scalability – provisioning of resources must be done dynamically
  - Reliability – access to resources and computation must be dependable
  - Maintenance – maintenance is handled by service providers

Source: Reference [1], Pages 3. Also see Reference [2].

October 12, 2015
Cloud Services and Software Oriented Architectures-20
Basic Cloud Computing Architecture – Simple View

- Involves multiple components which work together
- Supports managing and dispatching resources
- Supports SOA
- Scalable
- Supports interoperability and the sharing of resources
- Ability to support managing resources like servers and storage devices virtually
- Ability to support creation of a virtual organization

Source: Reference [1], Page 9.
Layered Cloud Computing Architecture

- Clients Layer
  - Hardware and/or software that relies on computational capability of the Cloud

- Services Layer
  - Different service models such as SaaS

- Application Layer
  - Actual app that delivers the service over the Internet

- Platform Layer
  - Layer delivers a computing platform and/or solutions stack as a service
  - Facilitates deployment w/o investing in HW/SW

- Storage Layer
  - Hardware and/or software products that are designed for storage of Cloud Services

- Infrastructure Layer
  - Delivers computer infrastructure, typically a platform virtualization environment as a service

Source: Reference [1], Figure 1.4, Page 10.
(Similar to 7-layer OSI model of data networks)
Types of Clouds

- Four types of Clouds: Public, Private, Hybrid & Community

- Provisioned for open use for the public by a particular organization who also hosts the service.

- Shared by several organizations. Typically externally hosted. May be internally hosted by one of the organizations.

- Composition of two or more Clouds (private, community or public) that remain unique entities but are bound together, offering deployment models. It is internally and externally hosted.

- Used for single organization. Can be internally or externally hosted.

Source: The Appacore Blog, 12/18/2012. Also see Reference [1, 2, 3].
Public Clouds

- General characteristics of Public Cloud
  - Available to the **general public** by a Service Provider (SP)
  - Service Provider typically **hosts** the Cloud infrastructure
  - Service Providers typically **own and operate** the infrastructure
  - Service Providers typically **offer access over the Internet**
  - Customers have **no control** over where the infrastructure is located
  - Examples of SPs are Amazon AWS, Microsoft Azure and Google

- Advantages of Public Cloud
  - Customers benefit from **economies of scale**
    - Infrastructure costs are spread across all users
    - Allows each individual client to operate on a low-cost, **pay-as-you-go** model
  - They are typically larger in scale than an in-house enterprise Cloud
  - Provides clients with seamless, on-demand scalability
  - They offer the greatest level of efficiency in shared resources

*Source*: The Appacore Blog, 12/18/2012. Also see Reference [1, 2, 3].
Public Clouds

- Disadvantages of Public Cloud
  - All customers on public Clouds share the same infrastructure pool
  - Customers share the same limited configuration, security protections and availability variances
  - Public Clouds are more vulnerable than private Clouds

Source: The Appacore Blog, 12/18/2012. Also see Reference [1, 2, 3].
Private Clouds

- General characteristics of Private Cloud
  - Cloud infrastructure is dedicated to a particular organization
  - Allows businesses to host applications in the Cloud
  - It is not shared with other organizations and addresses concerns regarding data security and control
  - There are two variations of private Clouds:
    1. On-Premise Private Cloud
    2. Externally Hosted Private Cloud

- Advantages of Private Cloud
  - Private Clouds are more secure when compared to public Clouds
  - Offers the greatest level of security and control

- Disadvantages of Private Cloud
  - A private Cloud project requires a significant level and degree of engagement to virtualize the business environment
  - Requires organization to reevaluate decisions about existing resources

Source: The Appacore Blog, 12/18/2012. Also see Reference [1, 2, 3].
Hybrid Clouds

- General characteristics of Hybrid Clouds
  - Hybrid Clouds are a composition of two or more Clouds
  - The individual Clouds remain unique entities but are bound together
  - Hybrid Cloud architecture requires both on-premise resources and off-site server-based Cloud infrastructure

- Advantages of Hybrid Cloud
  - Can leverage third-party Cloud providers in either a full or partial manner
  - Increases flexibility of computing
  - Augmenting a traditional private Cloud with the resources of a public Cloud can be used to manage any unexpected surges in workload
  - By spreading things out over a hybrid Cloud, you keep each aspect of your business in the most efficient environment possible

- Disadvantages of Hybrid Cloud
  - Have to keep track of multiple Cloud security platforms
  - Need to ensure that all business entities can communicate with each other

Source: The Appacore Blog, 12/18/2012. Also see Reference [1, 2, 3].
Community Clouds

- General characteristics of Community Clouds
  - A community Cloud is a multi-tenant Cloud service model
  - Community Clouds are a hybrid form of private Clouds built and operated specifically for a targeted group
  - These communities have similar Cloud requirements
  - The ultimate goal is for the communities to work together to achieve their business objectives
  - Community Clouds can be either on-premise or off-premise

- Advantages of Community Clouds
  - Participating organizations realize the benefits of a public Cloud
  - Obtain the added level of privacy, security, and policy compliance usually associated with a private Cloud

- Disadvantages of Community Clouds
  - The Community Cloud is only valuable when participating organizations within a targeted group can benefit from sharing infrastructure

Source: The Appacore Blog, 12/18/2012. Also see Reference [1, 2, 3].
Criteria for Cloud Selection

- Criteria for selecting Public Cloud
  - The organization’s standardized workload for applications is used by lots of people, such as e-mail
  - Incremental capacity is needed
  - There is a need to do collaboration projects

- Criteria for selecting Private Cloud
  - Data sovereignty is needed but want to leverage Cloud efficiencies
  - Consistency across services is required
Criteria for Cloud Selection

- Criteria for selecting Hybrid Cloud
  - Organization wants to use an SaaS app but is concerned about security
  - Organization wants to offer services targeting different vertical markets
    - Use public to interact with the clients, keep data secured with private Cloud
  - Organization wants to provide a public Cloud to its customers while using a private Cloud for internal IT

- Criteria for selecting Community Cloud (by examples)
  - Government organizations within a state may need to share resources
  - A group of hospitals or clinics may need to share a private HIPAA compliant Cloud
  - The Telco community may need to share Telco Community Cloud for Telco Disaster Recovery (DR) to meet specific FCC regulations

Source: Reference [1], Pages 17-19
Cloud Services
Definition & Examples

- Cloud Computing products can be broadly classified into **three main Services**:
  - Software-as-a-Service (SaaS)
  - Platform-as-a-Service (PaaS)
  - Infrastructure-as-a-Service (IaaS)

<table>
<thead>
<tr>
<th>SaaS (Software as a Service)</th>
<th>PaaS (Platform as a Service)</th>
<th>IaaS (Infrastructure as a Service)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Salesforce.com (CRM)</td>
<td>• Google App Engine</td>
<td>• Amazon</td>
</tr>
<tr>
<td>• Google Gmail</td>
<td>• Microsoft Azure</td>
<td>• Microsoft</td>
</tr>
<tr>
<td>• Microsoft Hotmail</td>
<td>• Force.com</td>
<td>• Rackspace</td>
</tr>
<tr>
<td>• Google Docs</td>
<td></td>
<td>• GoGrid</td>
</tr>
<tr>
<td>• Microsoft online</td>
<td></td>
<td>• AT&amp;T</td>
</tr>
<tr>
<td>version of Office</td>
<td></td>
<td>• IBM</td>
</tr>
</tbody>
</table>

Source: Reference [1], Pages 12-16
## Additional Examples of Cloud Service Examples

<table>
<thead>
<tr>
<th>Cloud Services</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage-as-a-Service (SaaS)</td>
<td>Nirvanix, Cleverafe dsNET</td>
</tr>
<tr>
<td>Database-as-a-Service (DbaaS)</td>
<td>Amazon, Force.com</td>
</tr>
<tr>
<td>Information-as-a-Service (IfaaS)</td>
<td>IBM, Microsoft</td>
</tr>
<tr>
<td>Process-as-a-Service (PraaS)</td>
<td>Appian, Akemma, Intensil</td>
</tr>
<tr>
<td>Integration-as-a-Service (InaaS)</td>
<td>Amazon SQS, OpSource Connect, Boomi, Mule OnDemand</td>
</tr>
<tr>
<td>Security-as-a-Service (SeaaS)</td>
<td>Cisco, McAfee, Panda Software, Symantec, Trend Micro, Verisign</td>
</tr>
<tr>
<td>Management/Governance-as-a-Service (MaaS)</td>
<td>RightScale, rPath, Xen, Elastra</td>
</tr>
<tr>
<td>Testing-as-a-Service (TaaS)</td>
<td>SOASTA</td>
</tr>
</tbody>
</table>

**Source:** Reference [1], Pages 13-16 (Credited to David Linthicum)
Cloud and mobile computing are converging
This will continue to promote the growth of centrally coordinated applications
These applications will be able to deliver data to any device
The Cloud is the control point and system of record

Cloud/Client Computing

- The client is a rich app running on an Internet-connected device
- The server is a set of application services
- Applications are hosted in a scalable Cloud computing platform
- Applications can span multiple client devices
- The client environment may be a native app or browser-based

Issues with Cloud Computing

- **Security**
  - Users lose significant control over data
  - Risk of seizure on Public Clouds

- **Legal and Compliance Issues**
  - Affects both vendors and users
  - 3 main types of issues: Functional, Jurisdictional, Contractual
    - Physical location of data
    - Responsibilities of data, e.g. Disaster Recovery
    - Intellectual Property Rights

- **Performance and QoS Related Issues**

- **Data Management Issues**
  - Scalability
  - Confidentiality

- **Interoperability with Other Clouds**
  - Google, Microsoft, Amazon, IBM do not have interoperability

Source: Reference [1], Pages 17-19
Cloud Computing Methodologies

- Cloud Computing is based on two main techniques:
  - Service Oriented Architecture (We will discuss in detail in Section 3)
  - Virtualization
- Virtualization
  - Concept is to relieve the user from the burden of resource purchases and installations
  - The Cloud brings the resources to the users
- Virtualization may refer to:
  - Hardware
  - Memory
  - Storage
  - Software
  - Data
  - Network

Source: Reference [1], Pages 17-19
Rationale for Virtualization

- Server and application consolidation
  - Server resources can be utilized more efficiently as multiple applications can run on the same server

- Configurability
  - Virtualization provides solution for customized configuration and aggregation of resources

- Increased application availability
  - Virtualization allows quick recovery for unplanned outages
  - Why? Virtual environments can be backed up and migrated with no interruption in services

- Improved Responsiveness
  - Resource provisioning, monitoring and maintenance can be automated
  - Common resources can be cached and reused

- Virtualization can facilitate Cloud to achieve stringent SLA (Service Level Agreement)

Source: Reference [1], Pages 7-8
<table>
<thead>
<tr>
<th>Review Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Describe the four different types of Clouds. What are the advantages and</td>
</tr>
<tr>
<td>disadvantages of each type of cloud?</td>
</tr>
<tr>
<td>3. The Cloud is not a Silver Bullet. What are some of the major issues with</td>
</tr>
<tr>
<td>using the Cloud?</td>
</tr>
<tr>
<td>4. What is a Cloud Service? What are the three main types of Cloud Services?</td>
</tr>
<tr>
<td>5. Although there are three main types of Cloud Services, David Linthicum</td>
</tr>
<tr>
<td>identified other types of Cloud Services. List all of them.</td>
</tr>
<tr>
<td>6. Can you identify at least one company that does each type of Cloud Service</td>
</tr>
<tr>
<td>(including from the long list defined by David Linthicum)?</td>
</tr>
<tr>
<td>7. What are the characteristics of the Cloud/Client Computing trend defined by</td>
</tr>
<tr>
<td>Gartner?</td>
</tr>
<tr>
<td>8. What is the primary software architectural style used for Cloud Computing?</td>
</tr>
<tr>
<td>Why is it suitable for Cloud Computing?</td>
</tr>
<tr>
<td>9. Imagine that your boss asked you to write a proposal to develop an application</td>
</tr>
<tr>
<td>that leverages the Cloud. What would be your cost justification? What would</td>
</tr>
<tr>
<td>your development approach be? Assume that you would be proposing The</td>
</tr>
<tr>
<td>Incremental Commitment SPIRAL MODEL.</td>
</tr>
<tr>
<td>10. Define Virtualization? Why is it important in Cloud Computing?</td>
</tr>
</tbody>
</table>
SECTION 3

SERVICE-ORIENTED ARCHITECTURE (SOA)
What is SOA? First, Understand “Tight Coupling”

- Data and functionality typically reside on more than one, often **disparate** systems.

- Applications need to be able to **“talk to each other”**.

- Status quo: **Proprietary** or **custom** communication interfaces between applications.

Challenges with Tight Coupling

- It is **costly** to **maintain**
- Slow and costly to **change**
- **Consequence:** Cost and complexity of managing and changing a tightly coupled architecture makes **business agility** difficult
- IT can’t keep up with business needs, but it’s not their fault
- **Does not support REUSE!**
  - Reuse is a challenge industry wanted to solve for many years!
- Evolution of reuse solutions reflects industry’s concerns
  - **Header files, inheritance** and **polymorphism** at the object level, frameworks
  - **CORBA** (Common Object Request Broker Architecture)
  - **Microsoft COM** (Component Object Model)
  - **EAI** (Enterprise Application Integration)
  - **Web Services**

SOA: The Ideal of Open Interoperability (Loose Coupling)

SOA – A Definition

- **An IT architecture composed of software that has been exposed as “Services”** – i.e. invoked on demand using a standard protocol.

- **“Web Services”** – software available as a “service” using Internet protocols.

- **One software application talking to another using a standards-based (i.e. non-proprietary) language over a standards-based communication protocol.**

- **An IT architecture** that enables “loose coupling” of applications

Core SOA Definitions

- XML – Extensible Markup Language
- SOAP – Simple Object Access Protocol
- WSDL – Web Services Description Language
- UDDI - Universal Description, Discovery and Integration
- ESB – Enterprise Service Bus

Key Concepts
- Network Transparency
- Virtualized endpoint
- Self-describing software
- Universally discoverable software
- Universally understood software
- Machine to machine interaction

Standards Organizations
- (World Wide Web Consortium) W3C (www.w3.org)
- Organization for the Advancement of Structured Information Standards (OASIS) (www.oasis-open.org)

SOA Usage & Supporting Platforms

- **SOA Usage**
  - B2B
  - Enterprise Application Integration (EAI)
  - Application to Application
  - Government
  - Embedded Systems
  - SaaS (Software as a Service)
  - Cloud Computing (Computing/Infrastructure as a Service)

- **Major Players in SOA Space**
  - IBM: WebSphere SOA Product Suite
  - Oracle: WebLogic
  - Oracle: Fusion Middleware
  - Microsoft: .NET
  - SAP: NetWeaver

## What Makes Web Services Appealing?

<table>
<thead>
<tr>
<th></th>
<th>CORBA</th>
<th>JAVA RMI</th>
<th>ONC(SUN) RPC</th>
<th>WEB SERVICES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Data Encoding</strong></td>
<td>Common Data Representation (CDR)</td>
<td>Serialized Java/CDR</td>
<td>Extended Data Representation (XDR)</td>
<td>XML (WS-I doc-literal, SOAP Encoding)</td>
</tr>
<tr>
<td><strong>2. Message Format</strong></td>
<td>IIOP (GIOP)</td>
<td>RMI Protocol/IIOP</td>
<td>RPC RMS</td>
<td>SOAP</td>
</tr>
<tr>
<td><strong>3. Transport Protocol</strong></td>
<td>TCP</td>
<td>TCP</td>
<td>UDP TCP</td>
<td>HTTP</td>
</tr>
<tr>
<td><strong>4. Description Language</strong></td>
<td>CORBA IDL</td>
<td>Java Interface/Class</td>
<td>RPC IDL</td>
<td>WSDL</td>
</tr>
<tr>
<td><strong>5. Discovery Mechanism</strong></td>
<td>COS Naming</td>
<td>RMI Registry</td>
<td>Undefined</td>
<td>UDDI</td>
</tr>
<tr>
<td><strong>6. Invocation Method</strong></td>
<td>CORBA RMI</td>
<td>Java RMI (method call)</td>
<td>RPC</td>
<td>Undefined</td>
</tr>
</tbody>
</table>

**Source**: See [56], Page 4.
Cross-Enterprise Solution Architecture

- Figure illustrates today’s e-business solution architecture
- **Multiple clients:** Suppliers, Customers, Enterprise, Employees
- **Multiple Actors**
  1. Web Services
  2. Mobile Services
  3. Mobile Employees
  4. Application Software
  5. Security

*Source:* Mobility, Security and Web Services: Technologies and Service-Oriented Architectures for a new Era of IT Solutions by Gerhard Wiehler, p. 46.
Example: Business to Business, External Provider

Two enterprises: A & B
- 4 Web Services
- 3 WS from external providers
- 4 Scenarios

Scenario 1 (S1):
- Enterprise B’s Purchasing accesses Enterprise A’s Inventory Web Service

Scenario 2 (S2):
- Enterprise A’s Accounting accesses Payment Web Service

Scenario 3 (S3):
- Both enterprises’ security access the same authentication Web service

Scenario 4 (S4):
- Enterprise B’s Production accesses Logistics Web Service

Why SOA Governance?

- How do you develop Web Services in an organized and predictable way?
  - Is a Web Service being considered?
  - How are you going about it?
  - Where is a Web Service in its life cycle?
    - Concept?
    - Development?
    - QA? Testing?
    - Deployed?

- Questions if you have a complex ecosystem of services
  - How do you manage them operationally?
    - What services are up/down, for how long, etc.
    - Are the services load balanced?
  - What are policies for accessing the endpoints?
  - How about security? How do we guarantee security?
SOA Needs Critical Mass

- Value of individual service is **low** until there are enough consumers (reuse) to accelerate return on services
- Value of an SOA increases as the volume of services and consumers increase
- Volume needs to hit critical mass
- SOA *network effects* kick in at that time
- SOA critical mass is:
  - Point where there are enough available reusable services
  - Such that one business process can be orchestrated from them

**Competitive Value in the SOA ROI Threshold Model**

**Source:** Service Oriented Architecture: A Planning and Implementation Guide for Business and Technology

October 12, 2015

Cloud Services and Software Oriented Architectures-51
SOA Needs Critical Mass

SOA projects over time... result in many services... that are used and reuse... until SOA critical mass is achieved... which accelerates SOA business value...

SOA Projects

S S S

S S S

S S S

S S S

S S S

Service & Process Orchestration

Integration Avoidance

More Services Use and Reuse

Information & Transaction Latency

Competitive Value in the SOA ROI Threshold Model

Source: Service Oriented Architecture: A Planning and Implementation Guide for Business and Technology, pp.339
SOA: All Hype?

- In a profound sense, the industry hype about SOA is actually true.
  - It does work
  - It is being used in major deployments
  - It does cut costs and enable agility
  - It’s an incremental shift that is possible to adopt without scrapping earlier IT efforts

SOA Is Not a Silver Bullet

- Assumes costs and challenges inherent in reuse
- SOA does not make politics go away
- Your IT organization still has to master it
- Governance is a major challenge
- Security can be a big issue
- Vendors may not necessarily cooperate in an effort that commoditizes their products
- Vendors may be embedded in your organization, rendering some of the theoretical benefits of SOA moot
- Getting started with SOA may require longer and more expensive project cycles the first time around
  - Need high reuse potential & reuse aptitude
- Some SOA standards are still immature, leading to confusion and vendor-driven proprietary creep

Benefits & Limitations of SOA

- **Benefits**
  - Flexibility in new software design
  - Reuse of business components in networks
  - Interoperability and integration capability
  - Ease of assembling new business processes

- **Limits and Open Issues**
  - Not a universal remedy for today’s mix and match architectures
  - It is not a solution for all upcoming challenges
  - Not best practice for long-running asynchronous processes
  - Natural strengths in real-time request-response exchanges (asynchronous and synchronous)
  - SOA requires an environmental framework
    - .NET, SAP NetWeaver, IBM WebSphere, Oracle WebLogic
    - Platform independence not yet achieved
Benefits & Limitations of SOA

- Limits and Open Issues Cont’d
  - Most critical issues are pending **security issues**
    - Physical Network
      - Need Intra- and Inter-organizational security
    - SOAP Messages
      - Need to protect content of SOAP Messages
    - Endpoint (Web Service) Security
      - Need Intra- and Inter-organizational security
    - Extensive security framework worked out
    - Applicable products on the market
    - For mission critical processes, security measures still an issue
  - Not valuable for applications whose business logic components are in a closed application domain
  - Not valuable if there is no intention for reuse
Review Questions

1. Define Service Oriented Architecture.

2. What are the benefits of SOA? What are the risks?

3. In what situations would you be extremely cautious in recommending SOA? Hint: Would you recommend it for long asynchronous processing jobs?

4. There is a debate regarding when to use SOAP and when to use REST. What are your thoughts on this? Think in terms of advantages, disadvantages, scalability, and performance.

5. What are the key technologies used in a Service Oriented Architecture product?

6. List two technologies that preceded SOA.

7. When was SOA introduced as an architectural style?
References

References

References

References


CASE STUDY

EXAMPLE OF CLOUD R&D USING THE INCREMENTAL COMMITMENT SPIRAL MODEL
Statement of Problem

Currently Memory and Disk are Limiting Factors!
General Solution

Remove Memory and Disk as Limiting Factors and Redirect Data & Computing to the Cloud!
Scenario #1 – File and Data Structure in Memory

Memory Allocated for Core Components in the Embedded Printer Environment

Memory Allocated for File#1

File#1 must be in memory for duration of the job

Memory Allocated for Data Structure 1

Must be in memory for the duration of the page

File#1 relocated to the Cloud

DS1 relocated to the Cloud
Scenario #2 – File and Data Structure on Disk

- Memory Allocated for Core Components in the Embedded Printer Environment
  - File #1 must be on disk for duration of the job
  - Must be on disk for the duration of the page

- Core Components
  - File #1 relocated to the Cloud
  - DS1 relocated to the Cloud

- Disk space allocated for File #1
- Disk space allocated for Data Structure #1
Interactions with the Cloud

• What are **distinguishing** interactions with the Cloud?
  - Data is **stored in** or **pushed to** the Cloud
  - Data is **retrieved from** or **pulled from** the Cloud
  - Computation is limited to embedded operating environment
• There is no computation in the Cloud (although this is possible)
Development Approach

- General Approach (Boehm et al. See Reference [1])
  - Incremental development with stakeholders’ commitment
  - Get evidence to fortify concepts
  - Engineering Modeling to reduce risks
  - Always be open to discard/exit if not feasible!

- Explore four Use Cases in the following order of priority:
  1. Use Case #1: Windows-based Applications
  2. Use Case #2: Embedded-based Applications
  3. Use Case #3: Core Technologies to Cloud
  4. Use Case #4: Mobile-based Applications to Cloud

Prototyping Plan

- Plan is to develop each use case in the following order
  1. Use Case #1 (Highest Priority)
  2. Use Case #2 (Second Highest Priority)
  3. Use Case #3 (Low Priority)
  4. Use Case #4 (Low Priority)

- Each Use Case will be developed **incrementally using modeling**

- Why modeling?
  - It will help us **identify and reduce** risks before we make a huge commitment
  - This will make it possible to involve our stakeholders at each critical phase
  - It will help us determine if development should continue

- **Question we need to ask:** Should we take the off-ramp now?
#1: Model of Basic Cloud Interaction

- Develop a model that illustrates capability to interact with Microsoft Azure from a “typical” programming environment

Figure 1: Basic Model
Integrate relevant elements of the model in Figure 1 into a Windows Simulator and make those components interact with an application.

Figure 2: Model of Interaction between the Windows Simulator and the Cloud
#3: Model of Writing a File to the Cloud that Can be Accessed Randomly

- Demonstrate that an application can process a PDF file when a key data structure is written to the Cloud

Figure 3: Model of Windows Simulator Writing/Reading temporal data structures to/from the Cloud
#4: Model of Writing a Complex File that Can be Accessed Randomly

- Demonstrate the capability to write complex file to the Cloud

Figure 4: Model of Windows Simulator Writing/Reading complex files to/from the Cloud
References