1. REST
   1. Representational State Transfer
   2. Software architecture
      1. Element
      2. Constraints
      3. Rationale
   3. Uniform interface
      1. Resources are identified by ONE resource identifier mechanism
      2. Access methods are the same for all resources
      3. Resources are manipulated by exchanging representations
      4. Representations are in self-descriptive message
      5. Hypermedia acts as the engine of application state
   4. Rest rationale
      1. Maximize reuse
      2. Minimize coupling
      3. Eliminate partial failure condition
      4. Scale without bound
      5. Simplify
   5. Design Restful service
      1. Identify resource
      2. Design URIs
      3. Expose a subset of the uniform interface
      4. Design the representations from and to the client
      5. Integrate this resource into existing resources, using hypermedia links and forms
      6. Consider typical flows
      7. Consider error condition
   6. Terminology
      1. Focus on Resources (URI)
         1. Lightweight
         2. Re-Use existing protocols (HTTP, HTTPS, XML)
      2. Use of HTTP commands
         1. GET
         2. PUT
         3. DELETE
         4. HEAD
         5. POST
      3. Powerful when combined with XML
         1. JS and XML
         2. Lightweight
         3. Uses existing technology
   7. Open issues
      1. Serialization
      2. Transactions
      3. Inflation of resources
      4. Semantics of HTTP commands
      5. WADL is still work in progress
2. Service-Oriented Architecture
   1. Service registry (UDDI)
      1. Provide a public listing of registries
      2. Service providers can register there services
      3. Service consumers can search for services and get information on how to connect to those service
   2. Service consumer
      1. Finds a web service needed
      2. Connect to service
   3. Service provider
      1. Offers an implementation of web service
      2. Makes it available for use
   4. Why SOA
      1. For programmer
         1. Easy way to integrate existing software
         2. Focus on atomic service
         3. Performance
         4. RPC
         5. Importance of workflow and repositories
      2. Manager
         1. Empowering
         2. Focus on workflows
         3. Love service repositories
         4. Able to adapt to change
         5. Downsize department
3. Web service technologies
   1. XML (describes message)
      1. Data storage
   2. SOAP (Processes XML)
      1. Simple Object Access Protocol
      2. Format for registration, searching and connecting messages in XML sent over HTTP
      3. Communications protocol for web service
      4. Defines the XML format for message
      5. Supports remote procedure calls (RPC) and message exchange
      6. Transport protocol (HTTP, SMTP, tec) binding
      7. Silent on security issue
   3. WSDL (Endpoint)
      1. Web Services Description Language
         1. Abstract definitions
         2. Concreate description
      2. XML document that describes a single web service
      3. Standardize way to describe web service interface
      4. Contract between client and server
      5. An XML document to describe SOAP message and how to exchange them
   4. UDDI (be discovered)
      1. Universal description, discovery and integration
      2. Registry listing of companies, web services they offer, and information on how to connect to those web service
      3. Each registry entry is an XML document
4. ESB
   1. Routing
   2. Message transformation
   3. Message enhancement
   4. Protocol transformation
   5. Service mapping
   6. Choreography
   7. Orchestration
   8. Transactions
   9. Security
   10. Management
5. Three type service
   1. Atomic
   2. Compounded
   3. Composite
6. BPEL
   1. XML based standard for describing a business process
   2. Enabling layer of service
7. HTTP communication layers
   1. Physical layer
   2. Data link layer
   3. Network/internet layer
   4. Transport layer
   5. Application layer
8. TCP
   1. Transmission control protocol
      1. TCP
      2. Core protocol
      3. Provide reliable, order communication
      4. Web runs on it
      5. Guaranteed delivery
      6. 2 ways commication
9. UDP
   1. User Datagram protocol
   2. Fast, compact but no guarantees
   3. One way communication
   4. No guaranteed delivery
   5. Often used for VoIP, DNS etc
10. HTTP commands
    1. HEAD
    2. GET
    3. POST
    4. PUT
    5. DELETE
    6. TRACES
    7. OPTIONS
    8. CONNECT
11. HTTP CODE
    1. 1xx
       1. informal
    2. 2xx
       1. client request successful
    3. 3xx
       1. redirection
    4. 4xx
       1. client request incomplete
    5. 5xx
       1. server error
12. Service Novel Aspect
    1. Scale
       1. Web app must scale
    2. Development speed style
       1. Very short development cycle
       2. Heavy use of platform, tool and lib
       3. Support
       4. Integration
    3. Different delivery and access of applications
    4. Constrained Brower, N-Tier Server Side
13. JSP Stack
    1. JSP
    2. Servlet (Older and low level construct and shows basic steps in processing request)
    3. Servlet Container
    4. Java
14. Server
    1. Web Server (Apache HTTP server) receives HTTP requests and return HTTP response
    2. Servlet container (Apache Tomcat) is an environment for hosting servlet (mostly HTTP)
    3. Application server (Apache Geronimo) consists of servlet container plus tools for transactions, messaging, EJB server …
15. .NET
    1. C# Source (.cs)
    2. C# Compiler
    3. Portable executable
    4. CLR (JIT compiler)
    5. OS
16. Key Element in .NET
    1. CLS (common language specification)
    2. CLI (Common language interface) like Common language runtime
       1. All basic type
       2. Threads
       3. Code execution
       4. Memory management
          1. Write code
          2. Compile into IL
          3. Store IL as a assembly file
          4. Activate assembly
          5. Compile assembly into native code
          6. Execute