E6998-04:
Web Application Servers –
Architecture and Design

Dr. Donald F. Ferguson,
IBM Fellow
Chief Architect, IBM Software Group
dff@us.ibm.com, or donff2@aol.com
BLOG:
http://www.ibm.com/developerworks/blogs/page/donferguson

Agenda

- Session I: 11:00 – 11:5
  - Some Logistics
  - Continuation from last w
  - Major subsystems Over
  - Application Component
    - Will use Java 2 Enterp
    - Concepts apply to other Web servers
  - An example -- Servlets
- Break/Discussion
- Session II: 12:00 – 12:50
  - An overview of some concepts
    - Begin to get our minds about what an app. Server does
    - Selected and almost random
  - First Assignment

This is going to be overwhelming and apparently random.

Helping you understand why an app server is more than an OS process and JVM.

Will cover a bit more gently and progressively in later classes.

Some Logistics

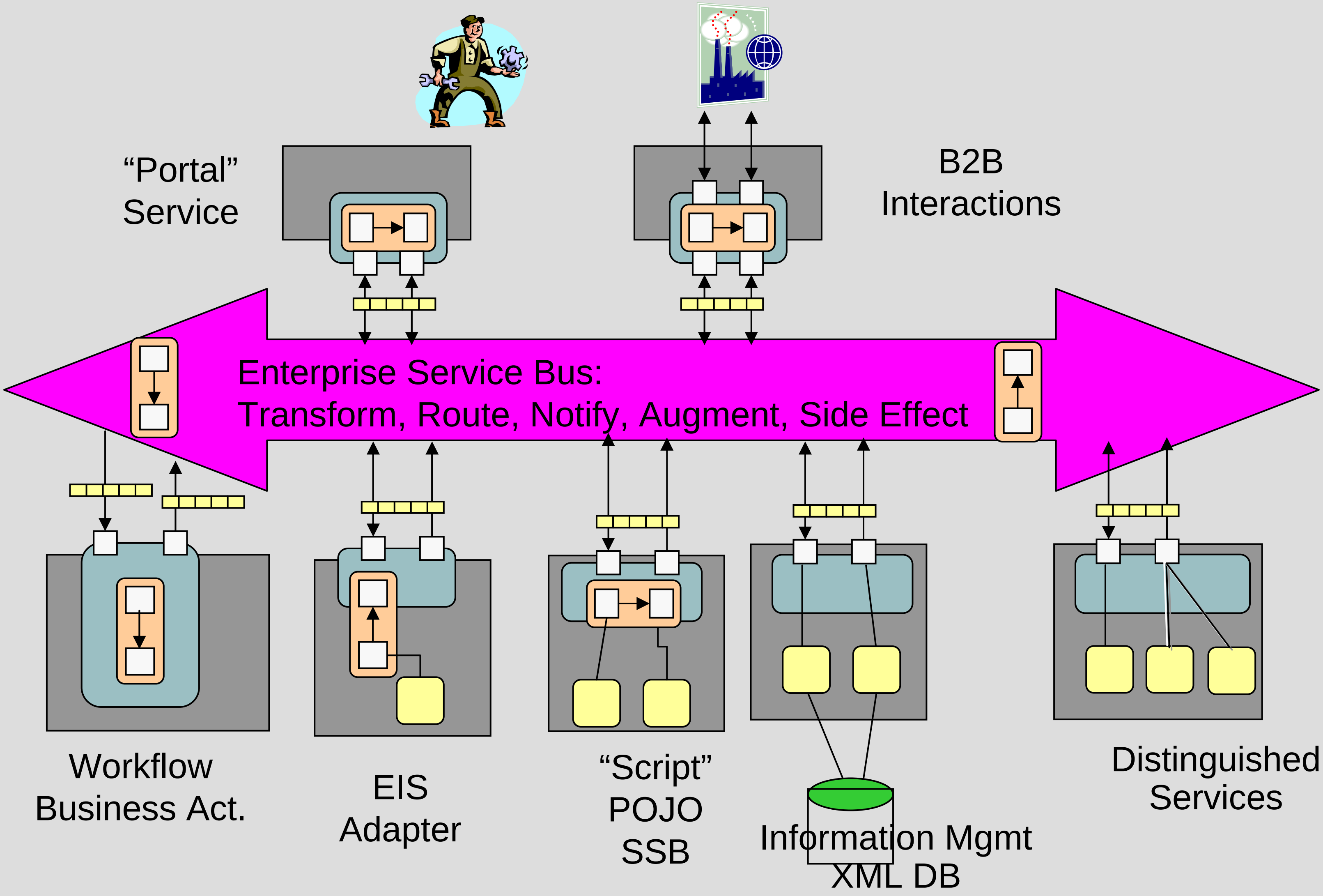
- I have a homepage at www.cs.columbia.edu/~dff
  - Course outline, which will evolve
    - Enumeration of content
    - Mapping to class session will evolve, based on progress
  - Will contain links to slides and reference material
  - Currently a static Web site
    - Will add a document database
    - Comments and discussion
- Preparing for course assignments
  - Papers and presentations
    - Start thinking about component you want to design.
    - Form teams of 2-3 people.
    - Concept Design Document
    - System Architecture and Design Document
  - Precision will matter
    - PowerPoint is imprecise visual notation.
    - Recommend using open source UML Tool

Continuation From Last Week

Continuation

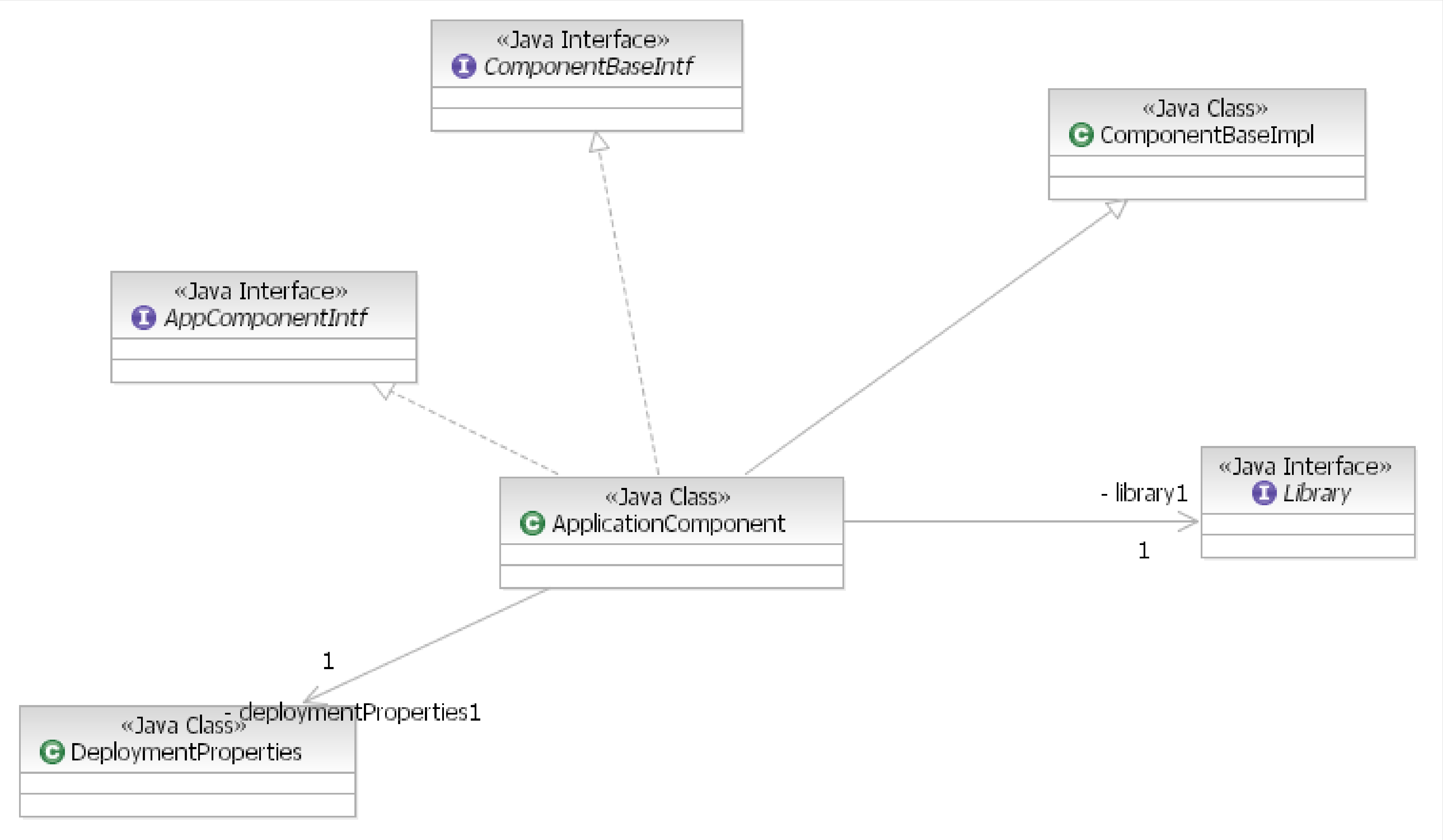
(Switch Presentation)

End-to-End Architecture



Application Components

Logical Model



Application Components (I)

- Plain Old Java Object (Class)
- Servlet
  - Handles HTTP POST, GET, …
  - Implements a controller pattern
  - Calls business logic
  - Selects response template
- JavaServer Page
  - Mostly HTML
  - Embedded scripts
  - JSP Tags
  - Passed result objects to format page
- Enterprise JavaBean™
  - Presentation independent business logic
  - SessionBean – Set of verbs
  - EntityBean – Noun, with a set of operations
  - Deployment descriptor

Application Components (II)

- Business Process Components
  - Long running business processes – hours, days, weeks
  - Servlets, EJBs, … are basically method lifetime
  - Process component
    - Incoming message
    - Reactivate
      - Process state variables
      - Current state
    - Run some activities
      - Update state variables
      - Change state
    - Return
    - Save
- Message Components
  - Destination
    - Queue (FIFO) data model
    - Ordering, priority, iterate, etc.
    - Multiple “put” components
    - Multiple “get” components
  - Topic
    - URL
    - Event format
    - Publish
    - Subscribe with predicate/filter

Application Components

- Portlet
  - Extends concepts of JavaServer Page
  - Event model between portlets
  - Minimize, properties, …
- Adaptor/Connector
  - Plug-in for protocol to remote system
  - Maps
    - Abstract verbs
    - To protocol specific verbs
  - Format conversion, e.g.
    - COBOL to
    - XML
- Gateway
  - Make internal services available outside the enterprise and vice versa.
  - Authorization, audit
  - Protocol mapping
- Distinguished services
  - Authentication, Authorization
  - Transaction manager
  - Log
  - …

Servlets

- Servlet
  - doGet, if the servlet supports HTTP GET requests
  - doPost, for HTTP POST requests
  - doPut, for HTTP PUT requests
  - doDelete, for HTTP DELETE requests
  - init and destroy, to manage resources that are held for the life of the servlet
  - getServletInfo, which the servlet uses to provide information about itself
- Servlet Config
  - getInitParameter(java.lang.String name)
  - getInitParameterNames()
  - getServletContext()
  - getServletName()

Servlet – Other Classes

- ServletContext
  - getRealPath
  - log()
  - … …
- HttpRequest and HttpResponse
  - Get parameters
  - MIME type
  - Browser type
  - IP address
  - Get/save cookies
  - … …
- HttpSession
  - Scratch pad between requests
  - Save information for next request processing

What Does the Application Server Do?

- Implement classes that business logic can call
  - HttpSession
  - ServletContex, e.g. log()
- Manage Servlet
  - Read configuration information
    - Singleton versus stateful
    - URLs → Servlet classes
  - Change management
  - Monitor
  - Operate
- Provide services, e.g. authorization
- Parse and manage incoming HTTP request and response
- … …

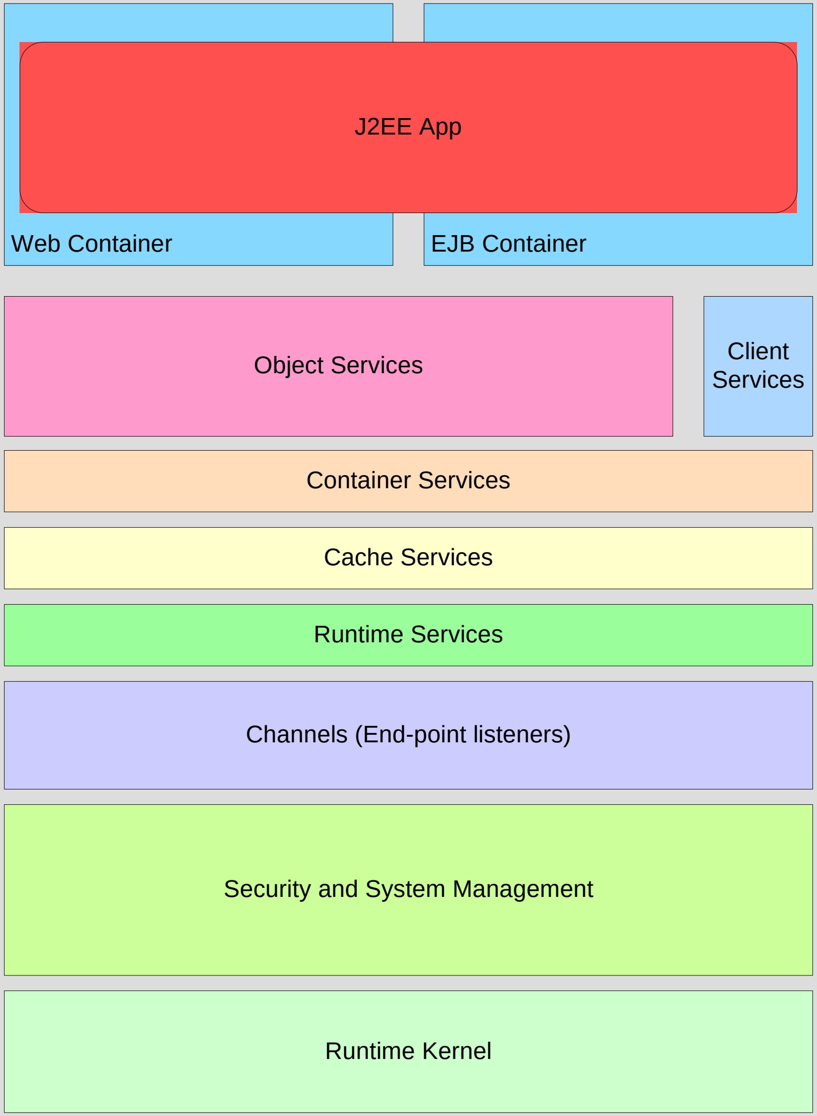
Agenda

- Session I: 11:00 – 11:50
  - Some Logistics
  - Continuation from last week
  - Major subsystems Overview
  - Application Components
    - Will use Java 2 Enterprise Edition
    - Concepts apply to other Web application servers
  - An example -- Servlets
- Break/Discussion
- Session II: 12:00 – 12:50
  - An overview of some concepts
    - Begin to get our minds about what an app. Server does
    - Selected and almost random
  - First Assignment First Assignment

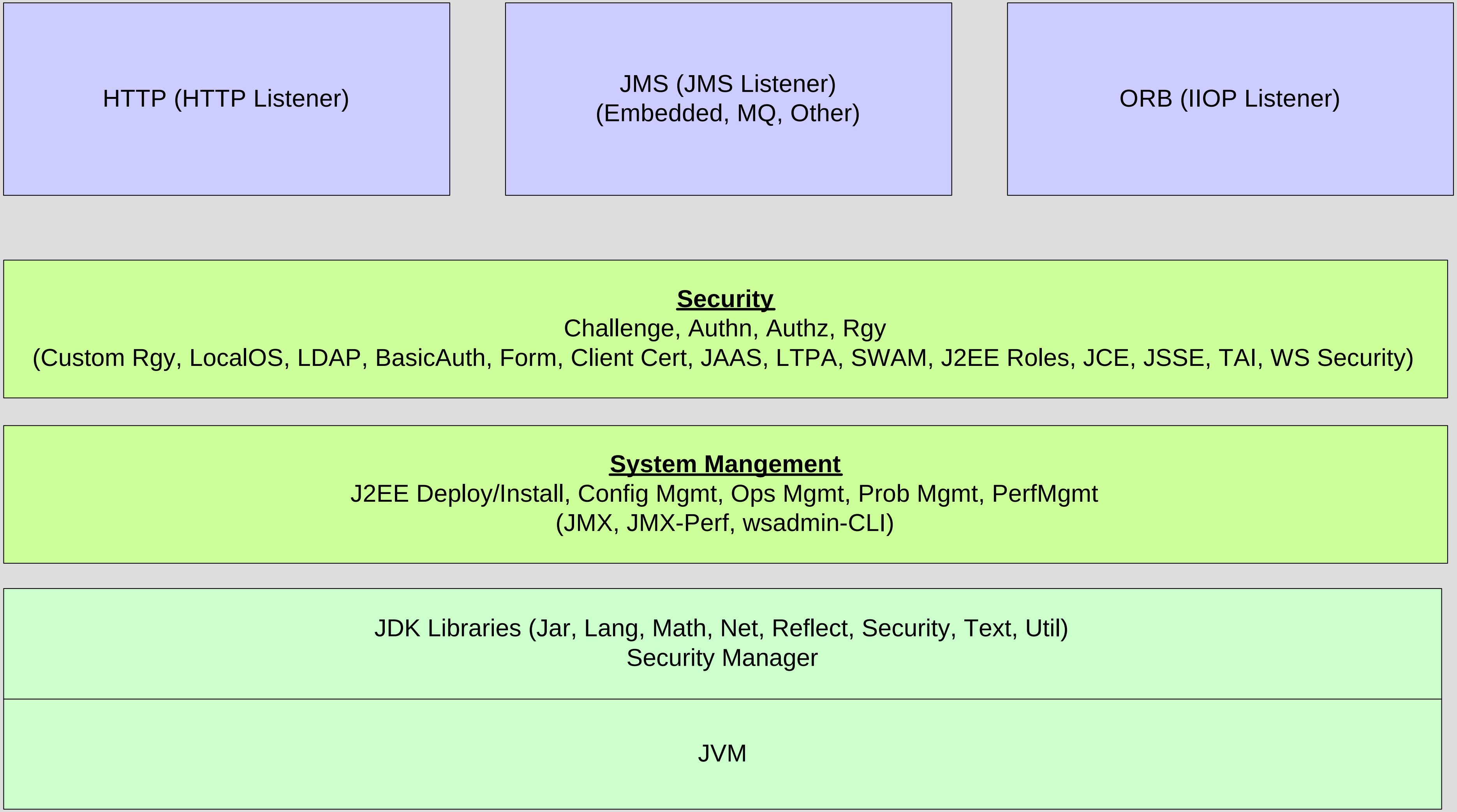
Agenda

- Session I: 11:00 – 11:50
  - Some Logistics
  - Continuation from last week
  - Major subsystems Overview
  - Application Components
    - Will use Java 2 Enterprise Edition
    - Concepts apply to other Web application servers
  - An example -- Servlets
- Break/Discussion
- Session II: 12:00 – 12:50
  - An overview of some concepts
    - Begin to get our minds about what an app. Server does
    - Selected and almost random
  - First Assignment

Application Server Base



Application Server Components (I)



HTTP (HTTP Listener)

JMS (JMS Listener)
(Embedded, MQ, Other)

ORB (IIOP Listener)

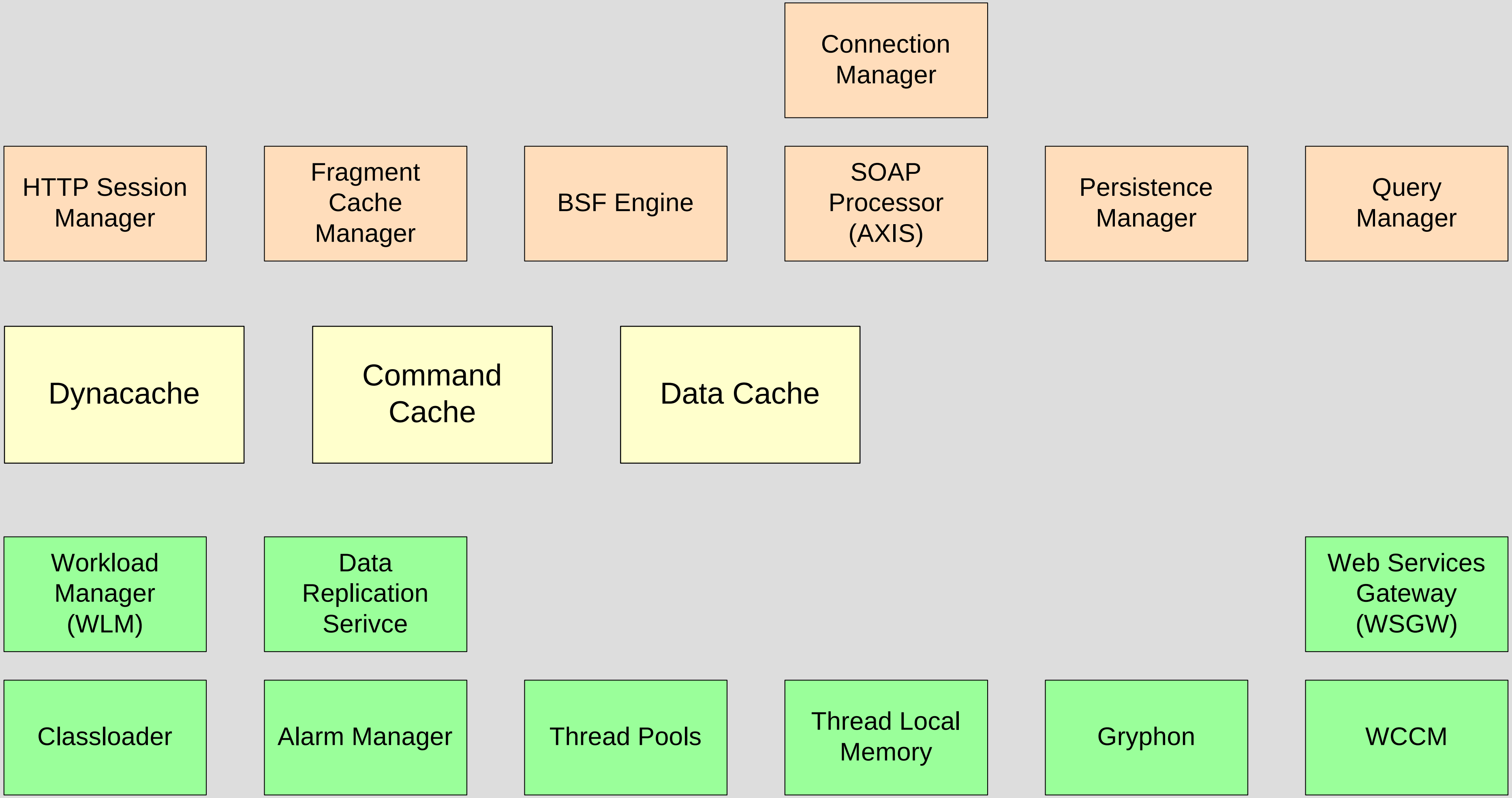
Security
Challenge, Authn, Authz, Rgy
(Custom Rgy, LocalOS, LDAP, BasicAuth, Form, Client Cert, JAAS, LTPA, SWAM, J2EE Roles, JCE, JSSE, TAI, WS Security)

System Mangement
J2EE Deploy/Install, Config Mgmt, Ops Mgmt, Prob Mgmt, PerfMgmt
(JMX, JMX-Perf, wsadmin-CLI)

JDK Libraries (Jar, Lang, Math, Net, Reflect, Security, Text, Util)
Security Manager

JVM

Application Server Components (II)



Connection Manager

HTTP Session Manager

Fragment Cache Manager

BSF Engine

SOAP Processor (AXIS)

Persistence Manager

Query Manager

Dynacache

Command Cache

Data Cache

Workload Manager (WLM)

Data Replication Serivce

Web Services Gateway (WSGW)

Classloader

Alarm Manager

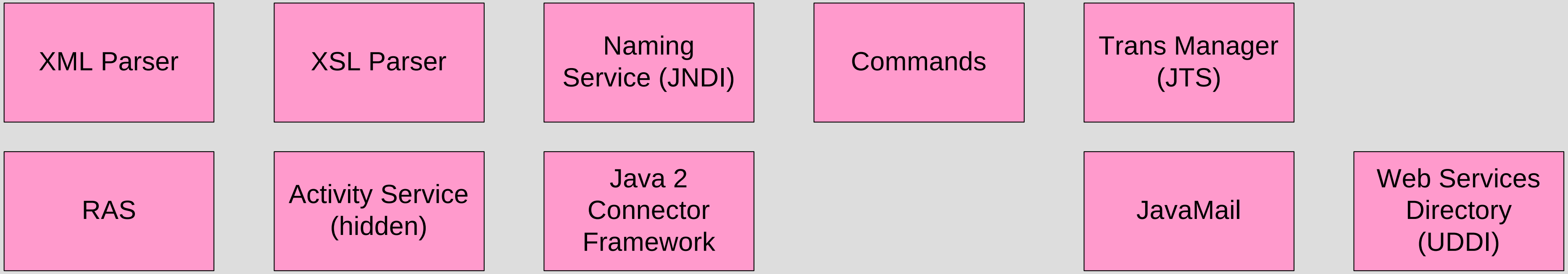
Thread Pools

Thread Local Memory

Gryphon

WCCM

Application Server Components (II)



XML Parser

XSL Parser

Naming Service (JNDI)

Commands

Trans Manager (JTS)

RAS

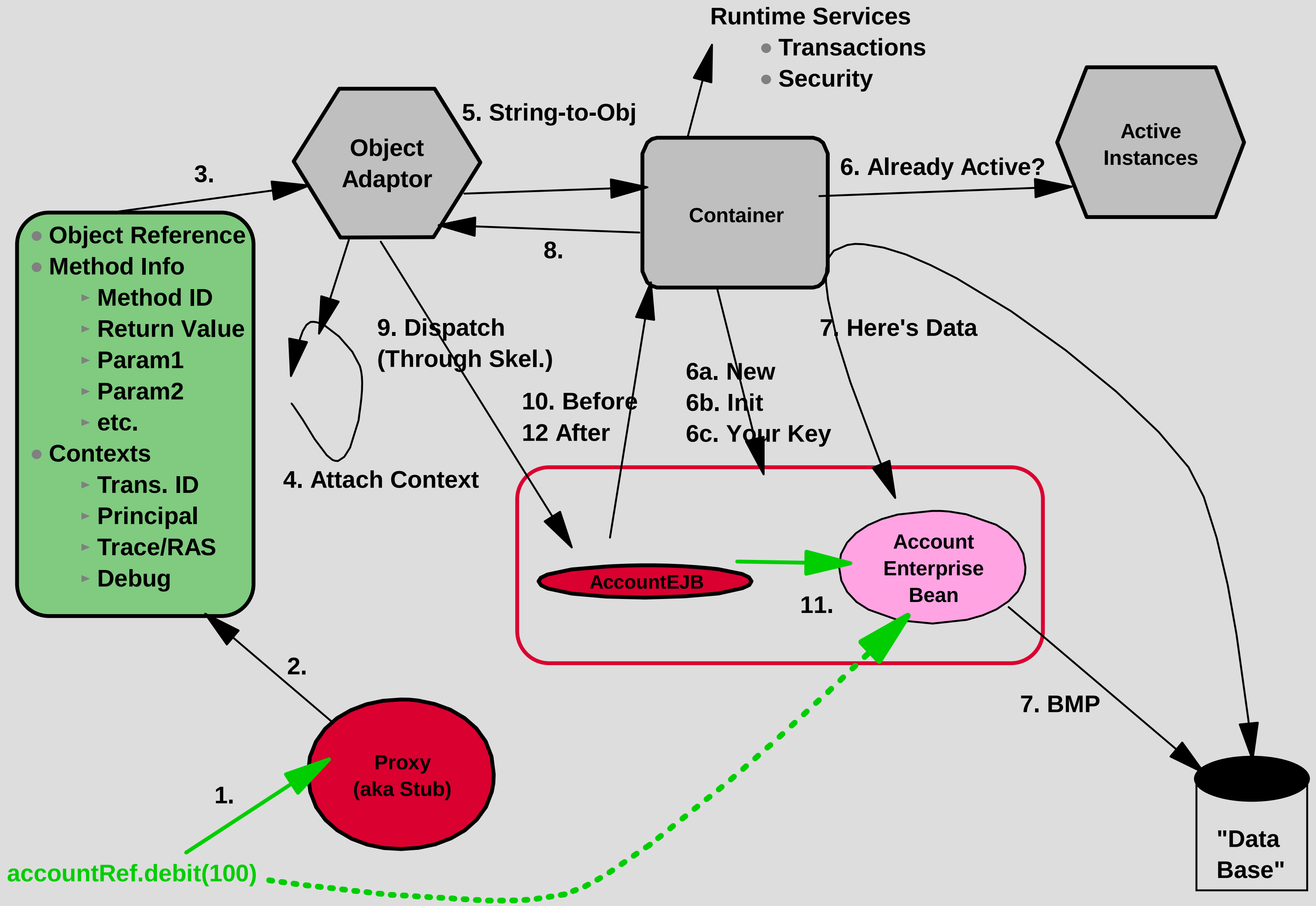
Activity Service (hidden)

Java 2 Connector Framework

JavaMail

Web Services Directory (UDDI)

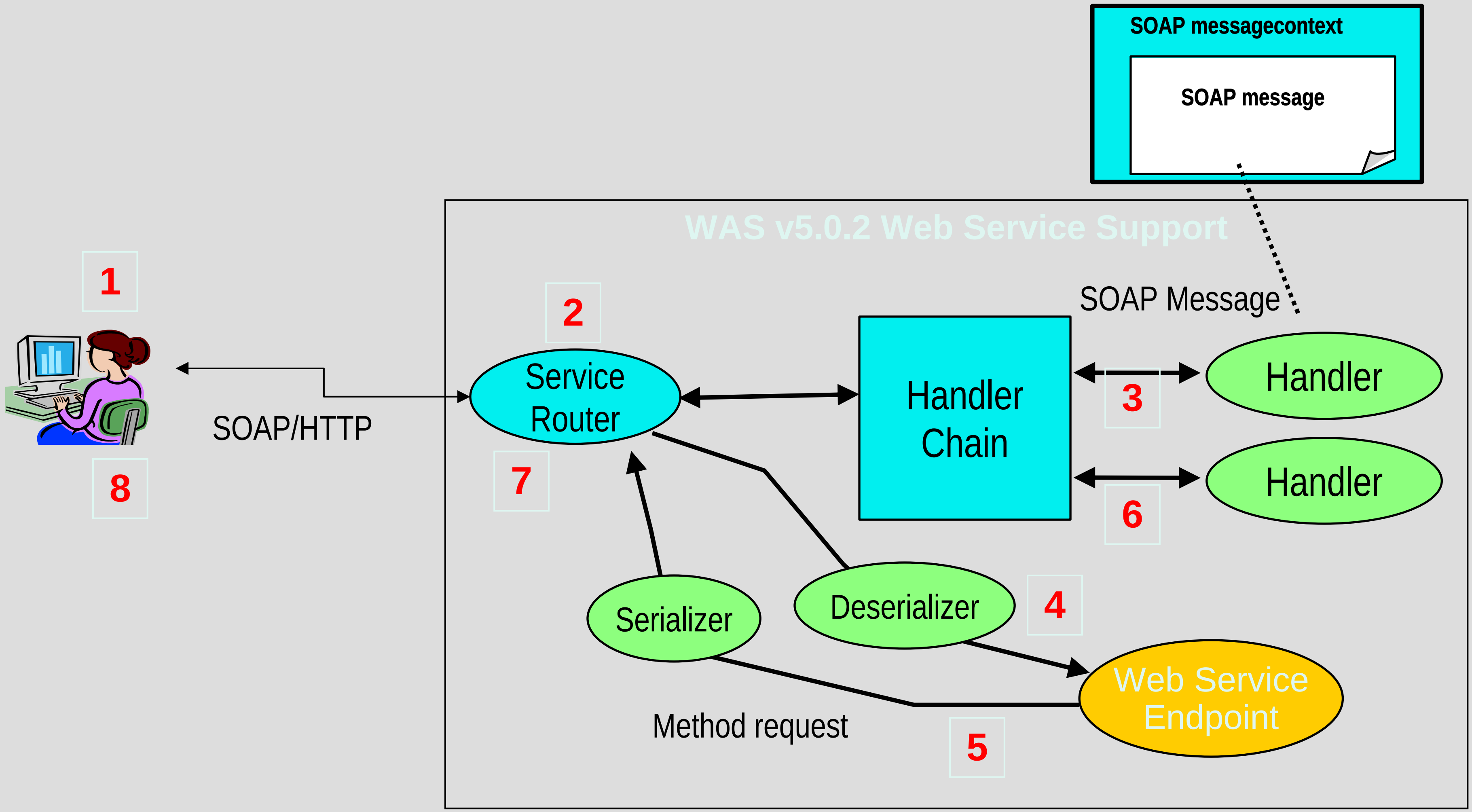
EJB Container



Runtime Services
- Transactions
- Security

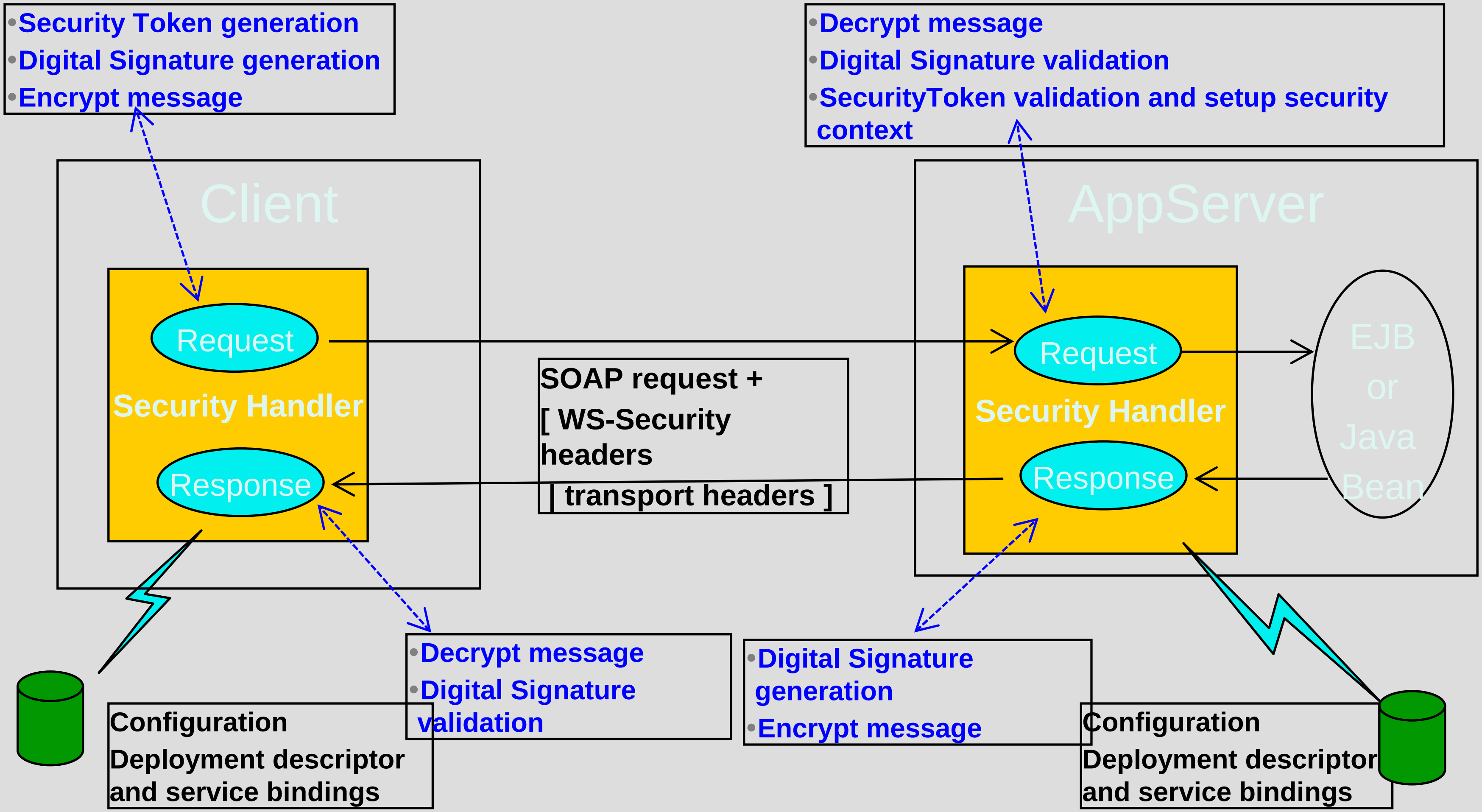
- Object Reference
- Method Info
  - Method ID
  - Return Value
  - Param1
  - Param2
  - etc.
- Contexts
  - Trans. ID
  - Principal
  - Trace/RAS
  - Debug

End to End Flow for SOAP/HTTP



Steps 1-4 handle Request
Steps 5-8 handle Response

WS-Security High Level Architecture



- Security Token generation
- Digital Signature generation
- Encrypt message

- Decrypt message
- Digital Signature validation
- SecurityToken validation and setup security context

SOAP request +
[WS-Security headers
[transport headers]

- Decrypt message
- Digital Signature validation

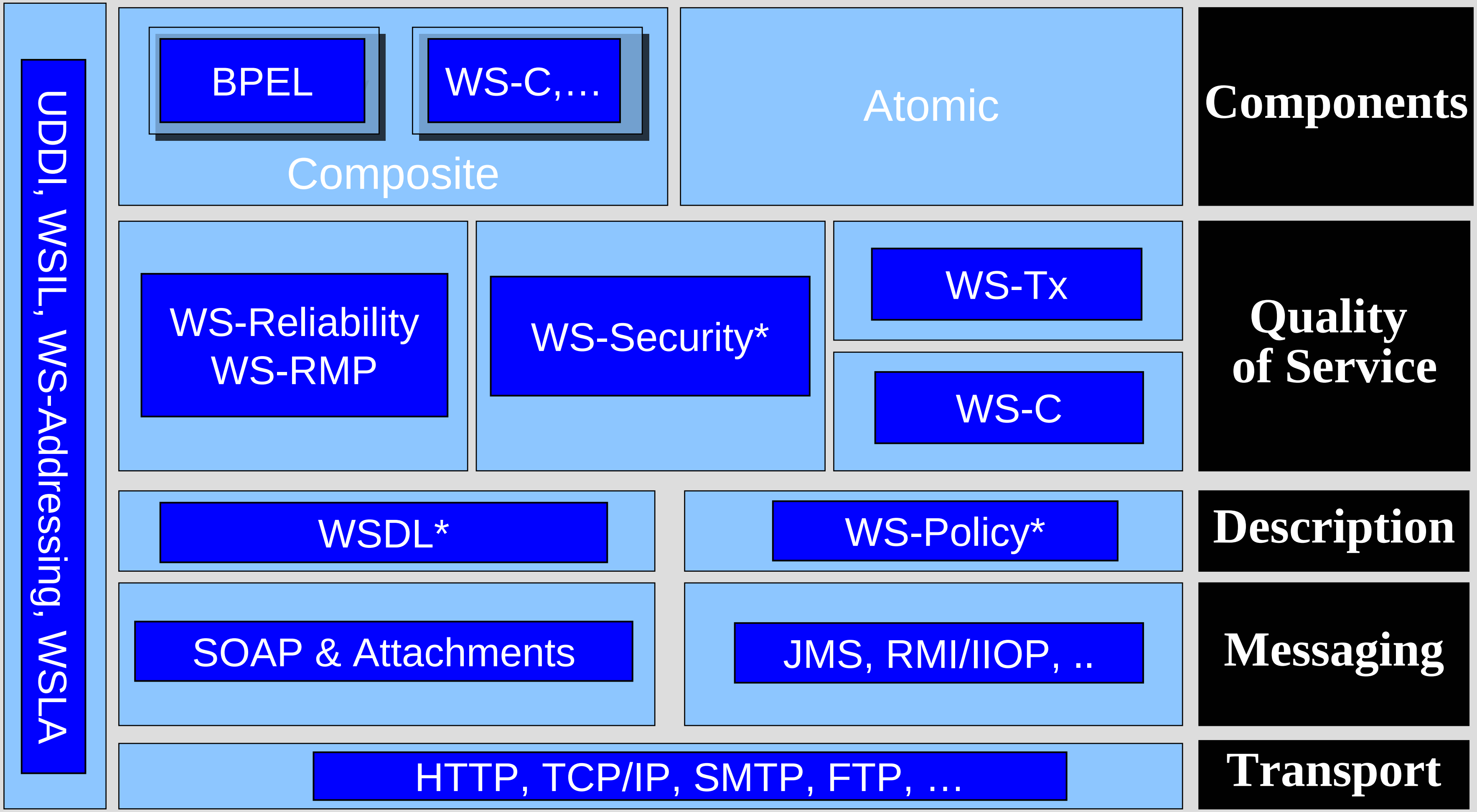
- Digital Signature generation
- Encrypt message

Configuration
Deployment descriptor and service bindings

Configuration
Deployment descriptor and service bindings

WS-* Specifications

Web application server provides implementations, and simplifes application logic.



UDDI, WSIL, WS-Addressing, WSLA

BPEL

WS-C,…

Composite

Atomic

Components

WS-Reliability
WS-RMP

WS-Security*

WS-Tx

WS-C

Quality of Service

WSDL*

WS-Policy*

Description

SOAP & Attachments

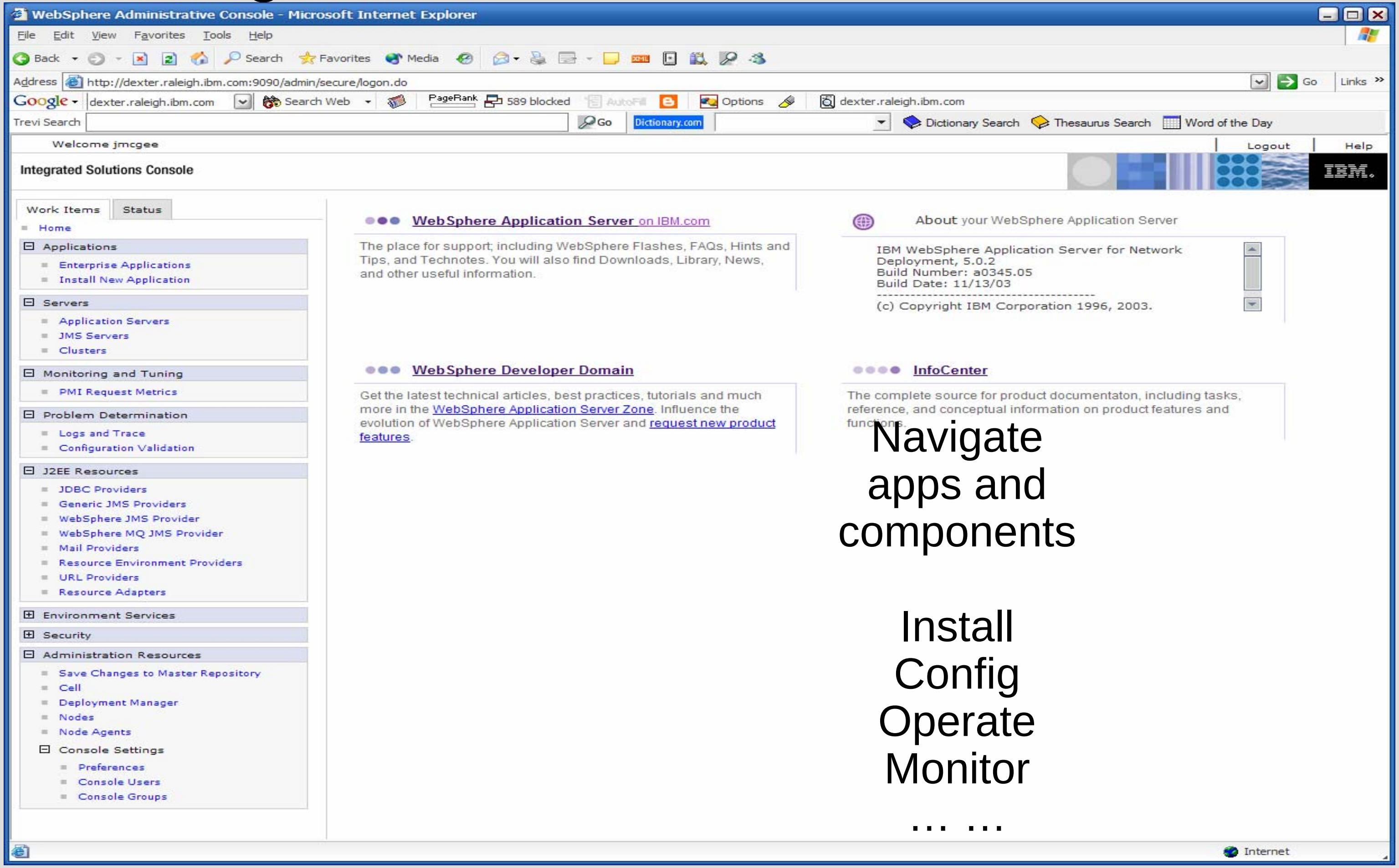
JMS, RMI/IIOP, ..

Messaging

HTTP, TCP/IP, SMTP, FTP, …

Transport

Systems Management



Address: http://dexter.raleigh.ibm.com:9090/admin/secure/logon.do

Google: dexter.raleigh.ibm.com

dexter.raleigh.ibm.com

WebSphere Application Server on IBM.com

The place for support; including WebSphere Flashes, FAQs, Hints and Tips, and Technotes. You will also find Downloads, Library, News, and other useful information.

About your WebSphere Application Server

IBM WebSphere Application Server for Network Deployment, 5.0.2
Build Number: a0345.05
Build Date: 11/13/03

(c) Copyright IBM Corporation 1996, 2003.

WebSphere Developer Domain

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Application Server Zone. Influence the evolution of WebSphere Application Server and request new product features.

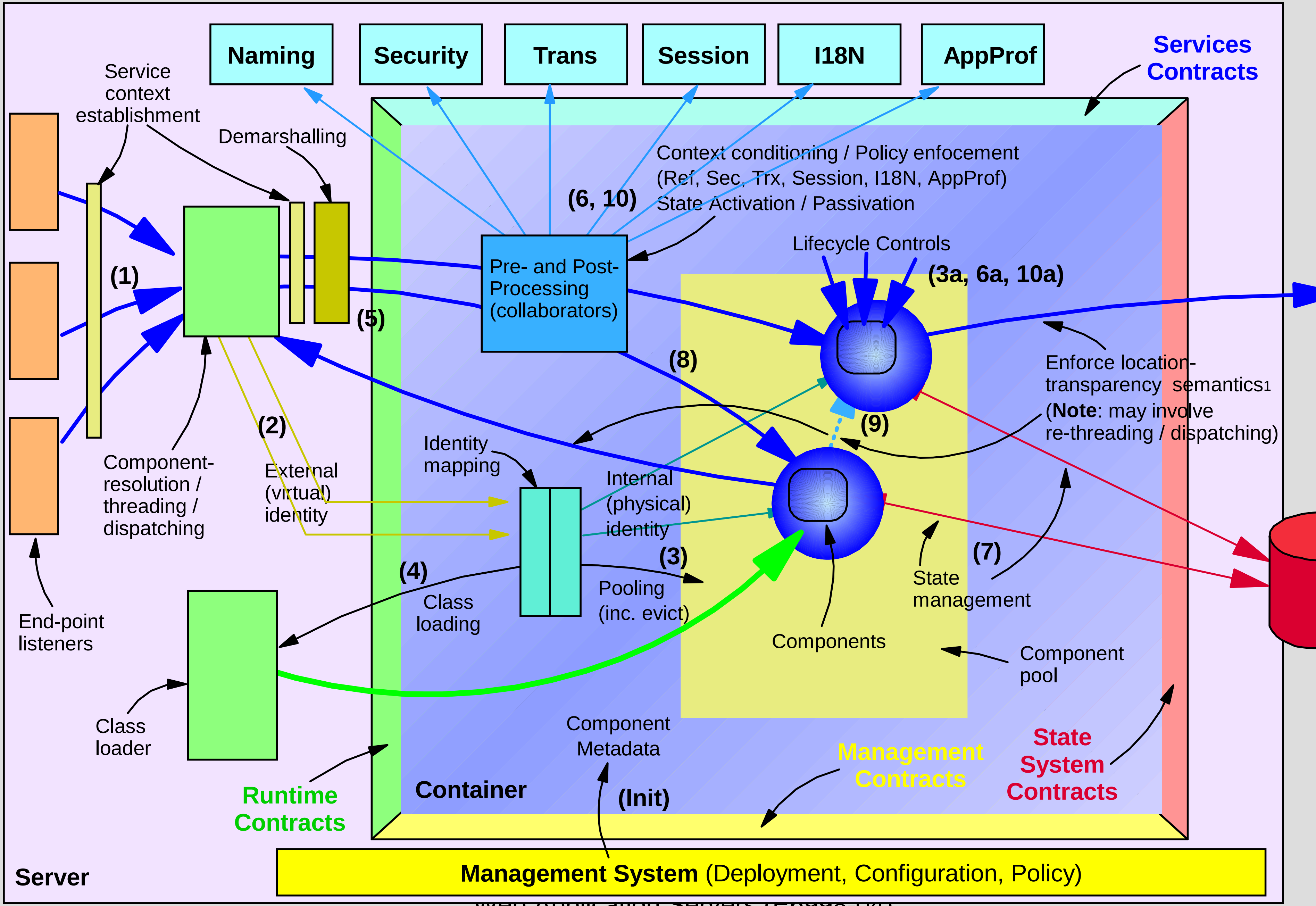
InfoCenter

The complete source for product documentaton, including tasks, reference, and conceptual information on product features and functions.

Navigate apps and components

Install
Config
Operate
Monitor
… …

- Single point of management
- Single system image



Naming

Security

Trans

Session

I18N

AppProf

Services Contracts

Service context establishment

Demarshalling

Context conditioning / Policy enfocement (Ref, Sec, Trx, Session, I18N, AppProf)

State Activation / Passivation

(6, 10)

Pre- and Post-Processing (collaborators)

Lifecycle Controls

(3a, 6a, 10a)

(1)

(5)

(8)

Enforce location-transparency semantics1

(**Note**: may involve re-threading / dispatching)

(9)

(2)

Identity mapping

Internal (physical) identity

Component-resolution / threading / dispatching

External (virtual) identity

(3)

(7)

(4)

Class loading

Pooling (inc. evict)

State management

End-point listeners

Components

Component pool

Class loader

Component Metadata

State System Contracts

Management Contracts

Runtime Contracts

Container

(Init)

Server

Management System (Deployment, Configuration, Policy)

Agenda

- Session I: 11:00 – 11:50
  - Some Logistics
  - Continuation from last week
  - Major subsystems Overview
  - Application Components
    - Will use Java 2 Enterprise Edition
    - Concepts apply to other Web application servers
  - An example -- Servlets
- Break/Discussion
- Session II: 12:00 – 12:50
  - An overview of some concepts
    - Begin to get our minds about what an app. Server does
    - Selected and almost random
  - First Assignment

First Assignment (due in one week).

- A simple paper
  - Five questions
  - For each question
    - One slide (picture, bullets)
    - One page of text answering question
  - Combine into one document
- Be prepared to present one of your slides.
- Five questions
  1. Why would a site configure more than one application server on a node/system?
  2. Review the interfaces and protocols for the Java Transaction Service. What functions would the application server provide to business logic components?
  3. Systems and application management is an important function of an application server.
     1. Why do we separate management out of applications?
     2. Why use a separate management daemon on a machine instead of implementing the management functions in the application server.
  4. Performance requires workload management, and load balancing is an important example. What other functions comprise workload management.
  5. Java applications typically have a single classpath. Why would an application server support multiple, configurable, overlapping classpaths?