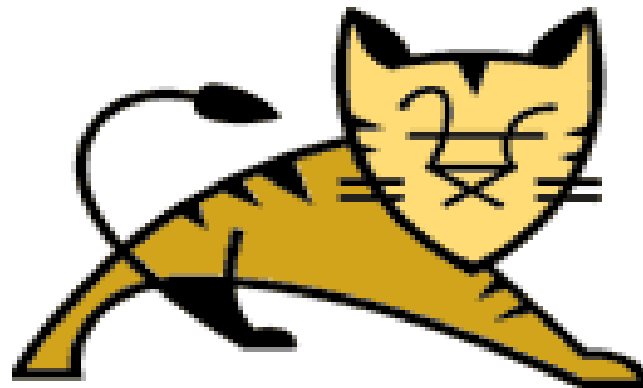


# TWINCLING Saturday Meet (TSM)



Tomcat Servlet Container

*Ver 5.5.9*

**TWINCLING Society** Presentation

Saturday, July 16, 2005  
Hyderabad AP India

# Request to Audience



No Eatables



Please switch off  
your mobile

# Outline

- Generic Component Model
- Component Model Interfaces
- Servlet Component Architecture
- Servlet Container Architecture
- Key concepts (SSCERFHC)
- Request Processing Pipeline
- Global config (server.xml)
- Context config (web.xml)

# Outline ...

- Evolution of Tomcat
- Installation
- Administration
- Logging
- Java Servlet
- Java Server Pages (JSP, JSPEL)
- Deployment, HOT deployment, RE deployment

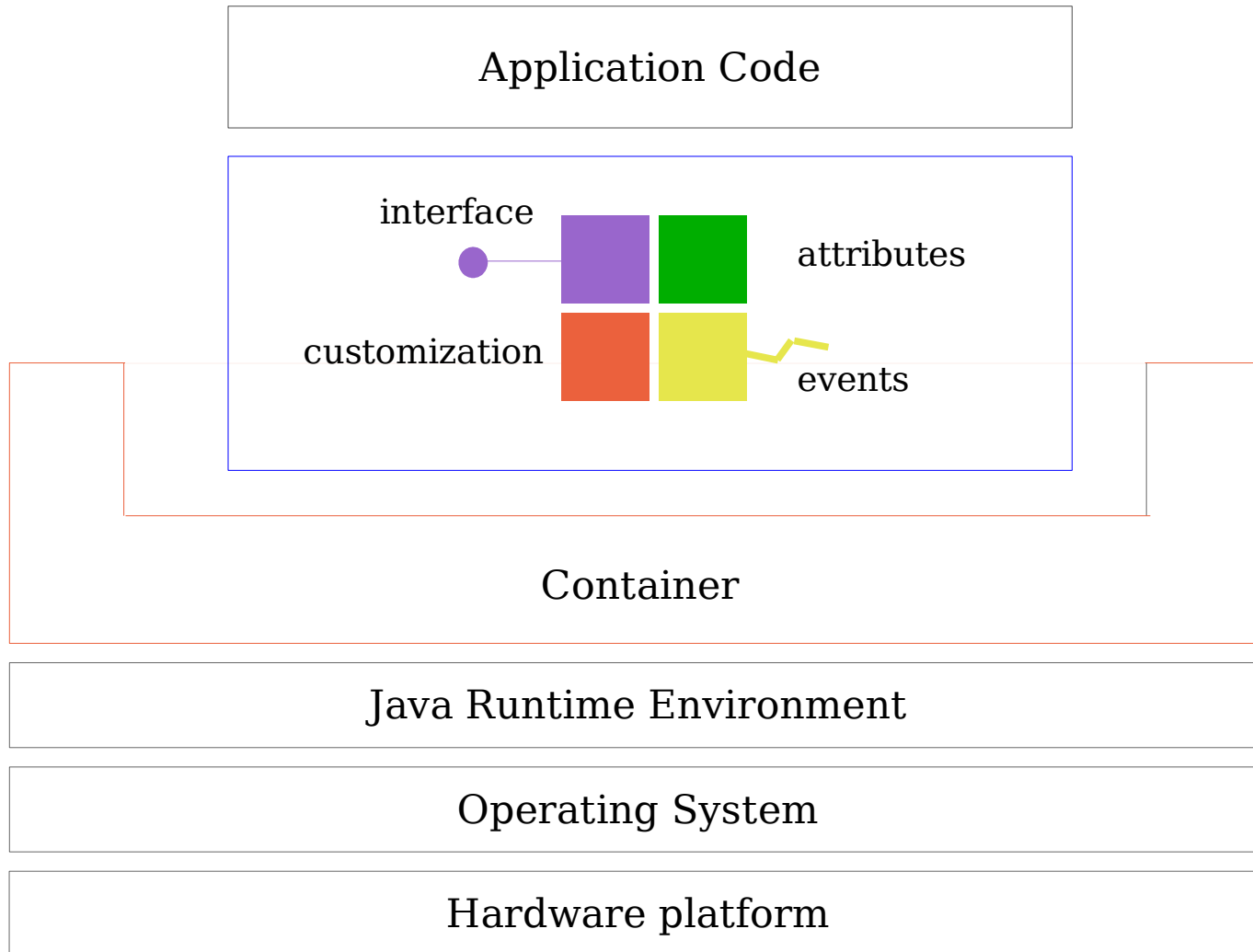
# Outline ...

- Clustering
- Session Replication
- Future Developments
- Resources
- Books

# Generic Component Model

- A component is an encapsulation of the code utilized in an application
- A component model is a set of requirements to which components and their environments adhere
- A component model
  - **defines** a “known” behaviour for interfaces
  - **specifies** interface(s) that a component implements
  - **guarantees** a certain level of interoperability among components
  - **operates** in an application environment

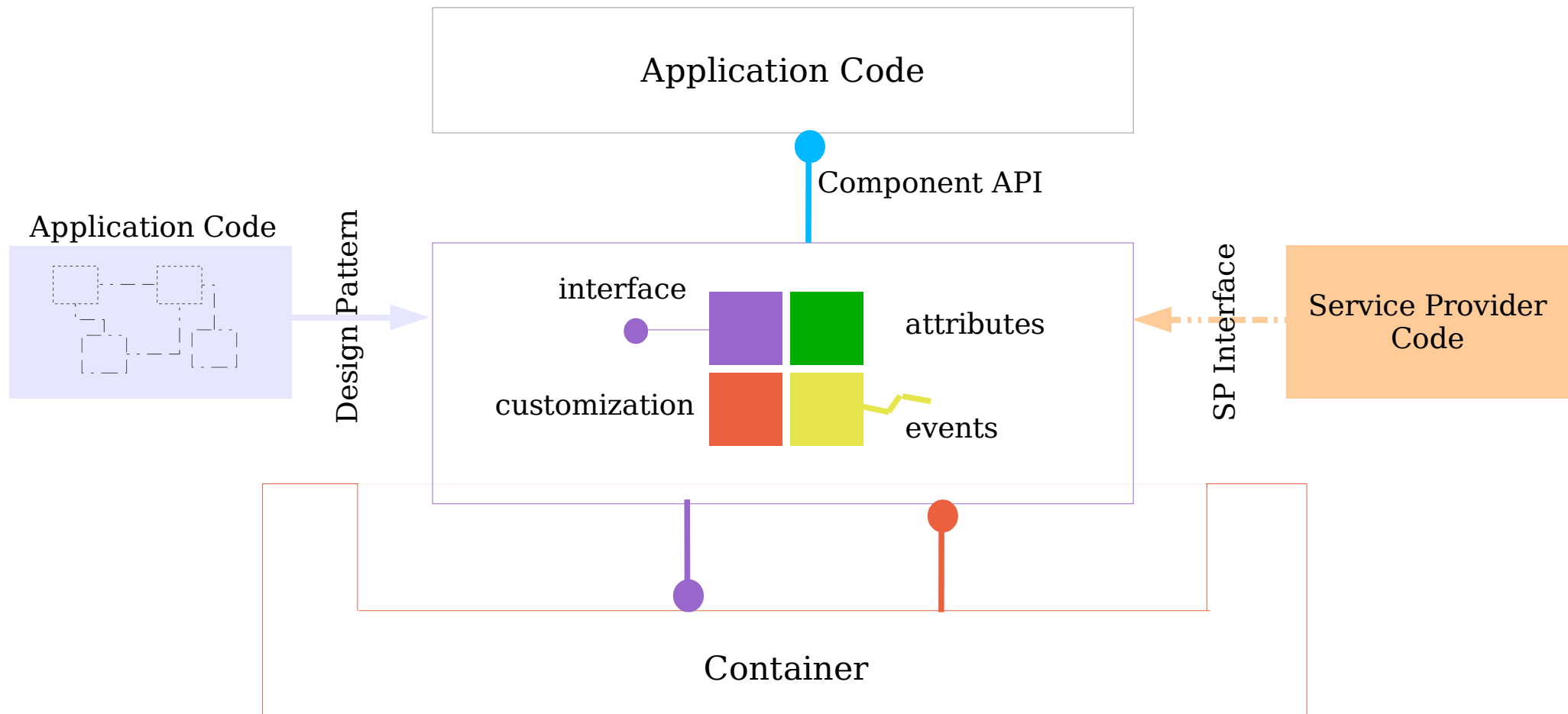
# Generic Component Model



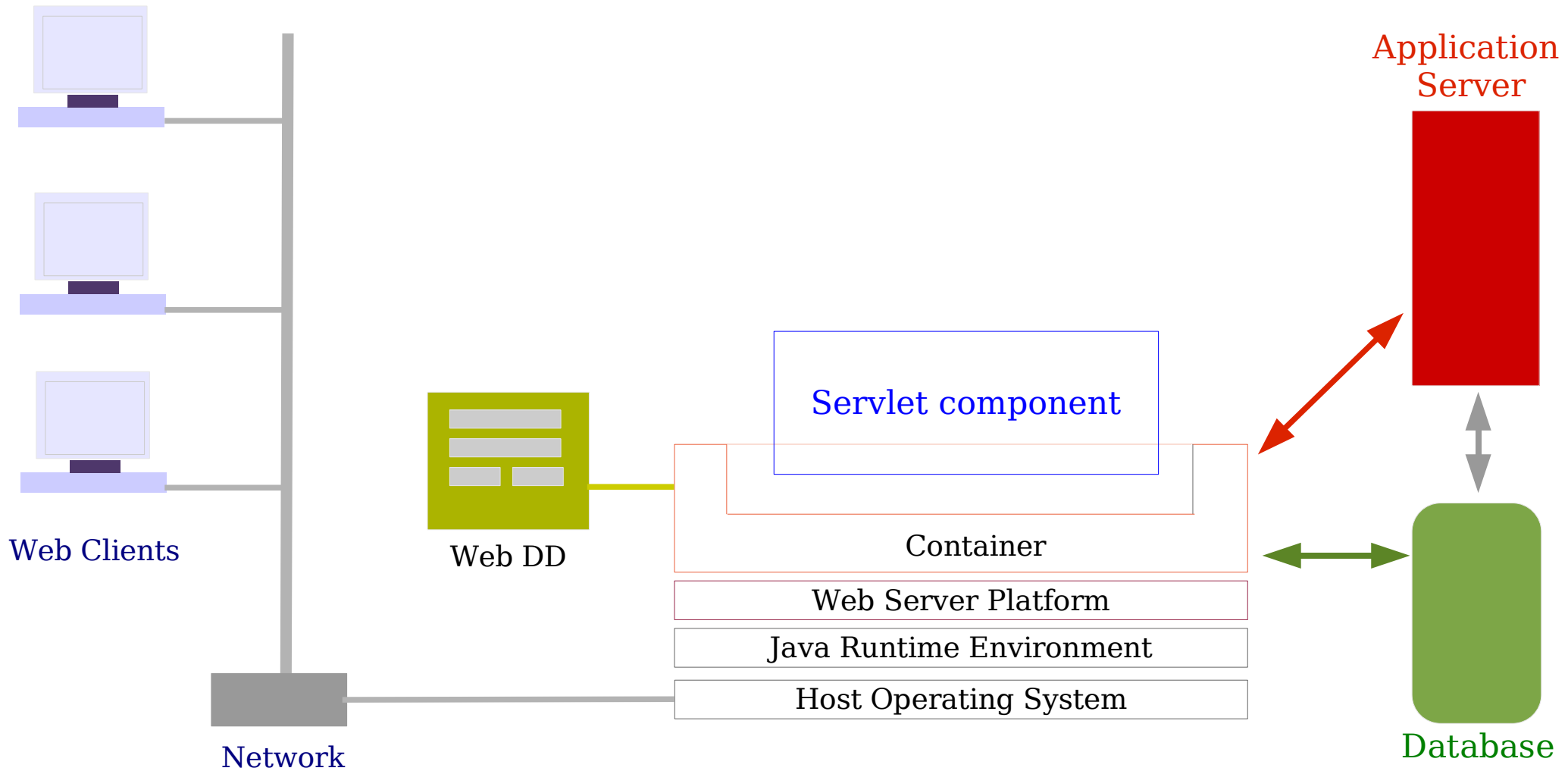
# Component Model Interfaces

- **Component APIs**
  - used by applications to use services of component
- **Design Patterns**
  - generic solution to a class of problems
- **Component SPIs**
  - interface implemented by service provider
- **Component-to-Container interfaces**
  - interface to Container environment assumed by component
- **Container-to-Component interfaces**
  - interface to component as assumed by Container

# Component Interface types

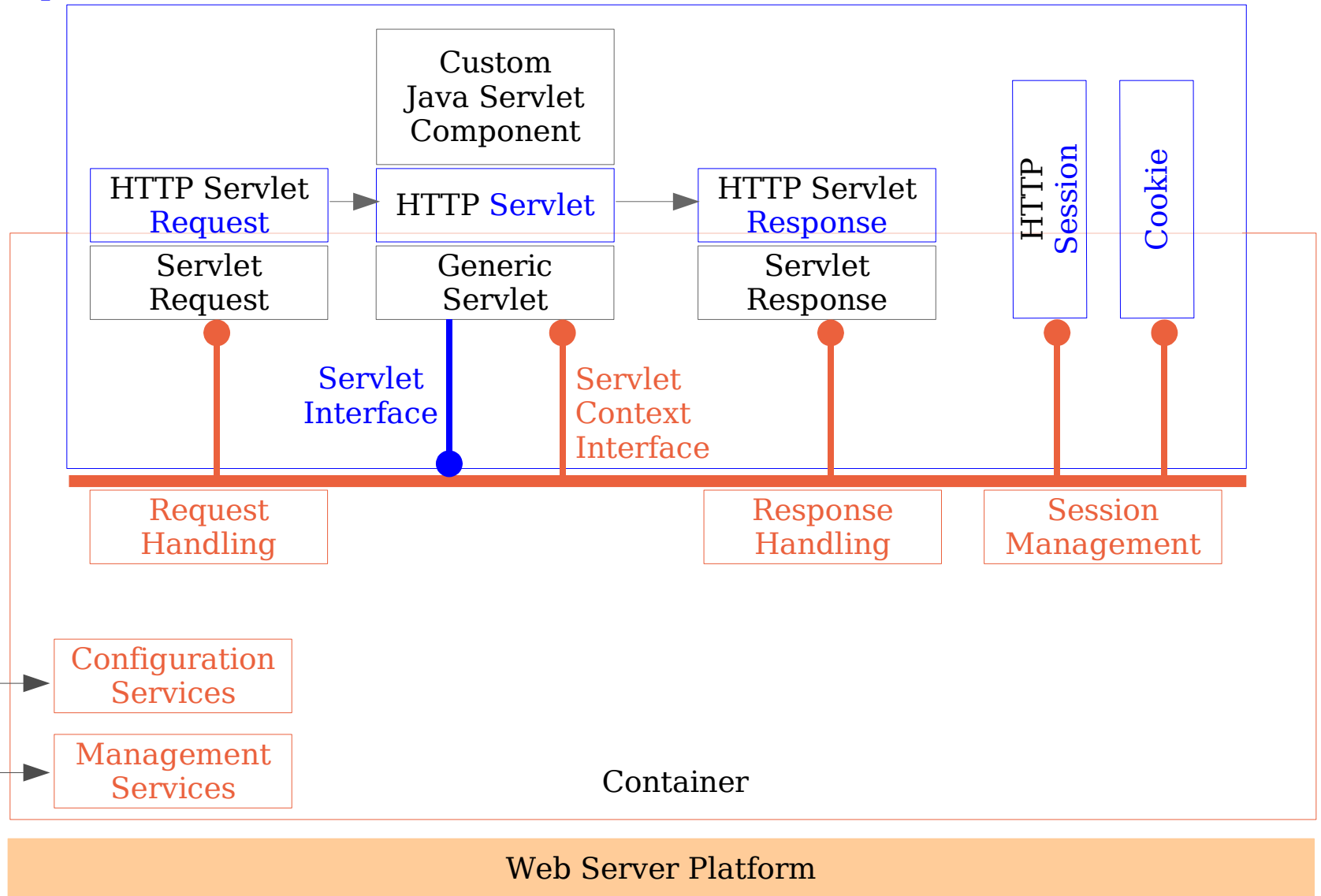


# Servlet Component



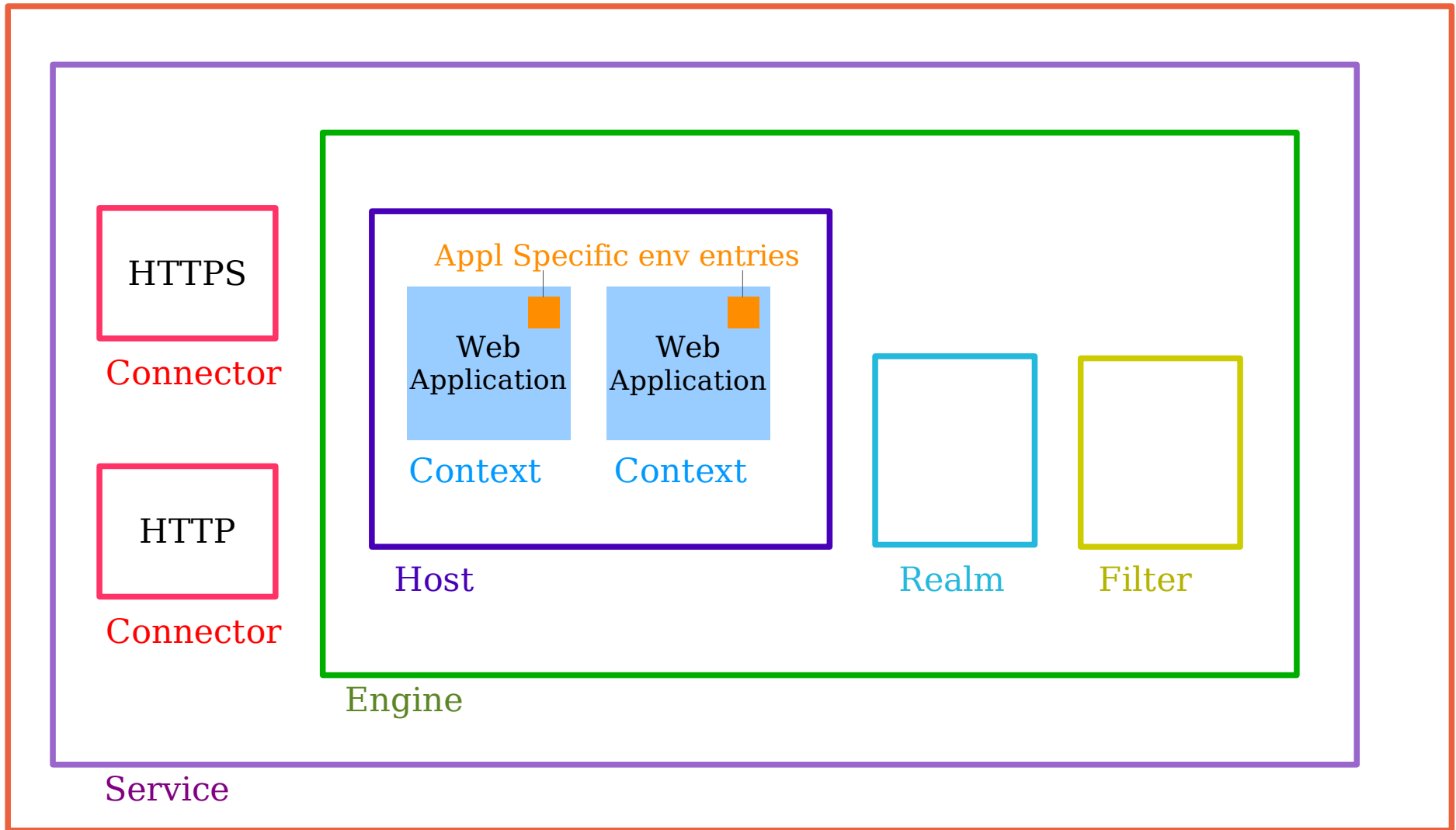
# Servlet Component Architecture

## Servlet component



# Servlet Container Architecture

Nested Containers



Server

# Key concepts

- Server
- Service
  - [ Connector(s) + Engine ]
- Connector
- Engine
  - [ Host(s) + Realm + Filter ]
- Realm
- Filter
- Host
  - [ Context(s) ]
- Context

# Server

- Top level container
- Vendor specific features
- Native performance optimizations

# Service

- Logical abstraction
- Ties Connectors to Engines
- Stable and Standardized

# Connector

- Communication Protocol implementation
  - HTTP
  - AJP
- one connector utilizes one port
- configuration of connection properties
  - number of processing threads
- Engine may utilize multiple Connectors
  - HTTP
  - HTTPS

# Engine

- Contains
  - Hosts
  - Realms
  - Filters (Valves)
- Provides hooks for AJP load balancing
  - jvmRoute attribute
- Declarative specification for
  - Realms
    - Authentication
    - Authorization

# Realm

- organizing principle for access to resource
  - database of credentials
  - list of roles
- applied to specific domains
  - JDBC Realm
  - Datasource Realm
  - JNDI Realm
  - Memory Realm
  - JAAS Realm

Authentication

Authorization

# Filter

- Map to url-pattern or a specific servlet
- Portable (specified in servlet spec 2.3)
- execution order decided by
  - declaration order in web.xml
- filter operations should be specific and fast
- Example
  - Compression Filter
    - maps to /\*
      - checks every response
      - compress if needed

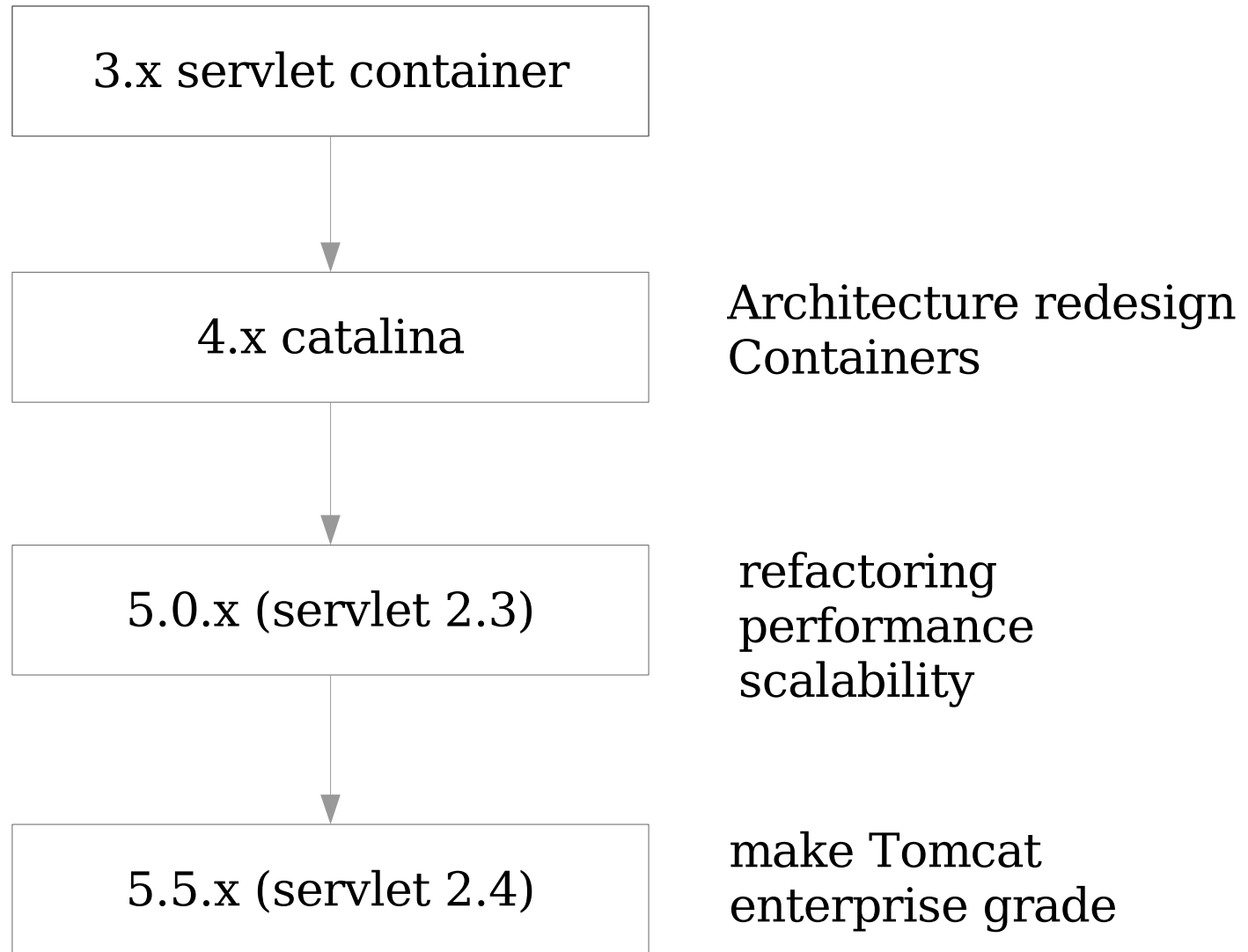
# Host

- Equivalent to Virtual Host
  - Apache VirtualHost
- Container for Contexts (Web applications)
- Configured to control
  - application deployment
  - application startup

# Context

- Single Web application
- Perform Application Specific, Server-side tuning
  - over and above web.xml
- Configure application specific env resources
  - JNDI entries

# Evolution of Tomcat



# Request Processing Pipeline

- Request received on a specific port
  - if processing thread is available
    - placed in queue
    - else REQ rejected
  - one thread process request from start to finish (Servlet spec)
  - thread finds host matching DNS or default
- Host executes any filters before forwarding to Context
  - matching filters executed
  - servlet, JSP, static resource processed
  - matching filters on response
  - response written to client

# Installation

- download binary tar.gz file from [jakarta.apache.org](http://jakarta.apache.org)
- open archive in a directory
- define the following environment variables
  - JAVA\_HOME
  - CATALINE\_HOME
- JDK 1.4.x and above is required
  - CLASSPATH=.:\$CLASSPATH
- Install any 3<sup>rd</sup> party JDBC etc drivers
  - MySQL Connector/J

# Administration

- Basic commands
  - `$CATALINA_HOME/bin/`
- Administration
  - startup, shutdown
  - logging, log analysis
- Upgrades and Updates
  - `common/lib`
  - `shared/lib`
- Deployment
  - `webapps`, `webapp.WAR`

# Deployment

- setting up a Web Application Archive component for loading in a Context
- Direct deployment
- Manager based deployment
- Web based management of deployment

# Manager

- Default
  - Manager access undefined
- Configure
  - `conf/tomcat-users.xml`
- URL
  - `http://localhost:8080/manager/status`

# tomcat-users.xml

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="tomcat" />
  <role rolename="role1" />
  <role rolename="standard" />
  <role rolename="manager" />
  <user username="tomcat" password="tomcat"
        roles="tomcat,standard,manager" />
  <user username="both" password="tomcat"
        roles="tomcat,role1" />
  <user username="role1" password="tomcat"
        roles="role1" />
</tomcat-users>
```

Deploy, Undeploy sample.war using manager

# Active deployment

- HOT deployment
  - turned on by default
  - performance impact
  - startup time longer
- RE deployment
  - done via manager app
  - application stop is graceful
  - Increased server load
  - Avoid due to memory leaks (ref)

# Clustering

- Pure Java clustering sans hardware, native app
- Combinable with
  - load balanced solution
  - high availability solution
- HttpSession key issue

# Session Replication

- HttpSession
  - Key to clustering Java Web apps
- Cluster ready
  - All HttpSession attributes implement Serializable interface
- To avoid
  - Large object trees
  - large number of session attributes

# Future Developments

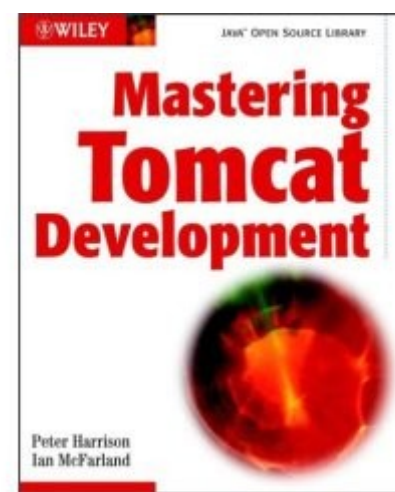
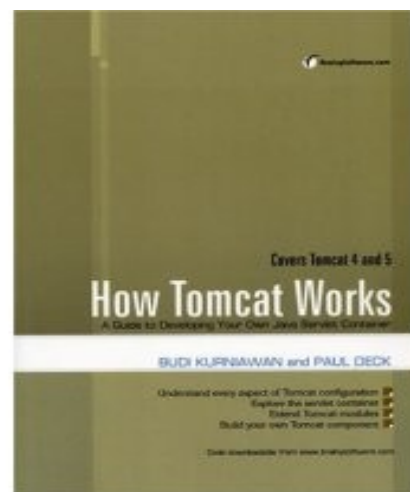
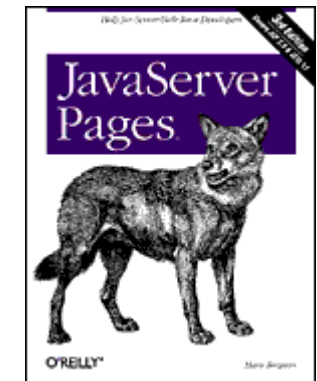
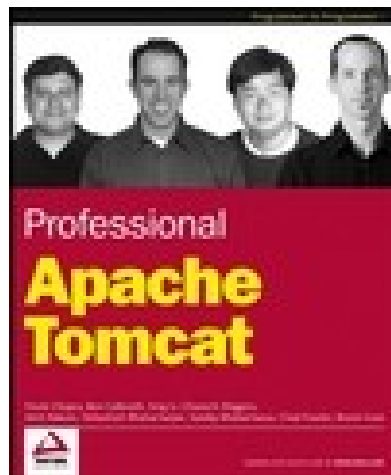
- APR connector
  - higher performance than java.nio
- Clustering
  - primary-secondary replication mode
  - deploy to one server, distribute to all nodes
- JDT compiler
  - instead of javac for JSP
  - tomcat runs with JRE

# Resources

- Tomcat

<http://jakarta.apache.org/tomcat/>

# Books



# Acknowledgement



Special Thanks to Euclid India Pvt. Ltd  
for coordinating with TWINCLING Society  
towards the TSM

[www.euclid.com](http://www.euclid.com)

# About Us

## TWINCLING Society

A “not-for-profit” Registered Society under A.P. Societies Registration Act.

- India's first
- independent
- "not-for-profit"
- OpenSource software
- development and promotion organization

[www.twincling.org](http://www.twincling.org)

[groups.yahoo.com/group/twincling/](http://groups.yahoo.com/group/twincling/)