

Simple Designs

- When engineering a service, your foremost consideration should be simplicity. Strive to create the simplest solution that satisfies all the requirements.

- **Simple Services**

Maintenance	Expansion	Integration
▪ Easiest to maintain	▪ Easiest to expand	▪ Easiest to integrate with other systems

- **Complex Services**

Confusion	Mistakes	Usage	Slower	Costly
▪ Leads to confusion	▪ Prone to mistakes	▪ Difficult to use	▪ Makes everything slower	▪ More expensive in setup cost and maintenance costs.

- During the engineering phase, create multiple design proposals. Consider each of them for its simplicity, manageability, cost, and so on. Revise, revise, revise.
- Good engineering is not about having the ability to create perfection in the first draft, but rather about being able to iterate over many revisions, improving each one until we have the design we want.

Use Vendor-Certified Designs

➤ Vendors of IT products have the engineering guidelines.

- They proscribes minimum CPU, RAM, and disk requirements.
- They also publish certified engineering designs or vendor best practices.
- There may be separate engineering recommendations for use with up to one dozen, one hundred, or one thousand users.

➤ Vendor designs are

best practice
engineering designs

Real-world
experience

labs and real
world Tested

Stress
Tested

➤ Trust vendor designs but verify them

- Straying too far from the vendor's recommendations will result in the loss of support.

➤ But even with the use of the certified engineering design exactly, some local decisions are part of local engineering design such as;

hardware
vendor

model
number

specific
configurations,

IP
addresses

network switch
connections

deployment
racks

Optional
parameters

Dependencies

- **The reliability of a service is greatly influenced by its dependencies.**
 - A system is never more reliable than the most unreliable system it has a hard dependency on.
- **The larger and more complex a system becomes, the more dependencies it has.**
 - These dependencies can be managed during the engineering process in ways that greatly improve the resulting system's reliability
- **Primary Dependencies**
 - Primary reliability refers to the core system reliability.
 - SAs do not have much control over reliability of software itself especially for 3rd party products.
 - SAs can control are their decisions about the hardware it runs on.
 - Generally, enterprise software tends to be written assuming the hardware will never fail thus for SAs, the options are reliability features such as mirrored boot disks, ECC RAM, dual NICs etc.

External Dependencies

- **SAs can control many of the external dependencies that influence a service's reliability.**
- **Know what those dependencies are. A dependency matrix is a list of which subsystems depend on other subsystems.**
 - Usually the dependency matrix resembles a tree diagram, with each level relying on services in the levels below, and no cross-dependencies.
 - Used to find high-risk
 - Useful for operational tasks such as basic debugging as well as disaster planning
- **Types of External Dependencies**
 - **Hard dependency**
The dependent service will fail if the other service fails or is unavailable
 - **Soft dependency**
does not directly cause a failure, though it may reduce performance
 - **Graceful degradation**
if that server is unavailable, the application may work around it by using cached data, disabling a specific feature, or requiring the data to be manually entered
- Improve the reliability by identifying hard dependencies and either eliminate them or reengineer them to be soft dependencies

Dependency Alignment

- Another engineering trick is to realign dependencies into smaller failure domains.
- A failure domain is all the services, locations, and users that are adversely affected when a particular component fails or is taken out of service.
- Good engineering requires taking the time to think through how we can shrink failure domains, or divide large failure domains into smaller, isolated failure domains.

Hardware Failure Domains

- Make Redundant domains including power strips, CPUs, and storage and VPN devices

Service Failure Domains

- Align all dependencies on same machine
- All startup scripts should be on the same machine
- Rack aligned dependencies

Location Failure Domains

Make sites self dependent

Decoupling Hostname from Service Name

- **Decouple the purpose of the machine from the name of the machine.**
- **A better choice is to design your systems to use aliases for the service name**
- **do not hardcode services to IP addresses**

Support

- **During planning and engineering phase, consider how service will be supported during its lifetime.**
- **Support includes**
 - **Technical aspects**
detecting problems, patching, upgrading, performing backups, and scaling the service
 - **People and processes**
identifying and training the support groups, developing mechanisms for customers to report problems and submit requests, and providing documentation for the customers and the support teams.
- **Support planning aspects can be divided into followings**

Monitoring

Support Model

**Service Request
Model**

Documentation

Monitoring

- **It isn't a service if it isn't monitored.**
- **A service should be monitored for availability problems, performance, and capacity-planning.**
- **The helpdesk, or front-line support group, must be automatically alerted to problems with the service.**
- **The SA group should monitor the service on an ongoing basis from a capacity-planning standpoint.**
- **Followings must be defined and developed**

what to monitor and
what events to
trigger notifications.

the process for adding new
components into the
monitoring system as the
system is scaled up or
upgraded.

how component are
removed from the
monitoring system.

Automation needs to be
developed, and the
processes need to be
clearly defined and
documented

Support Model

- **The support model needs to specify who supports the various components of the service, and what the OLAs and SLAs for each component are.**
- **The OLA and SLA for the service need to consider the OLAs and SLAs of the constituent parts, as well as the design of the service itself.**
 - In small or midsize company, many of these roles will be performed by the same people, who all know each other.
 - In large companies, there will be different teams supporting each component, they won't all know each other, and the processes and communication paths may not be clear

Service Request Model

- **The local engineering plan should also include a service request model (SRM).**
 - The SRM defines
 - which requests users can make relating to this service,
 - who can make those requests,
 - how they make those requests,
 - which approvals are required
 - who acts on the requests
 - which change control policies apply
 - which SLA is used for turning around those requests.
- **Work with the vendor and stakeholders to define which tasks are required, document them, and then practice them in a test environment.**
- **Try to estimate the volume of requests of each type and understand from the stakeholders what would trigger a request of each type and what kind of turnaround time they would expect on each request.**

Documentation

- **As part of support planning, operational procedures must be defined.**
- **These are different for every service, but generally include**
 - backups and restores,
 - Business continuity or disaster recovery plans,
 - tasks related to onboarding new users
 - Disconnecting users who are leaving
 - performing periodic tasks
 - Anything else required to keep the service running.
- **For each routine task, there must be a runbook that details how the task should be performed. Define the runbooks by performing each task in a test environment.**

IT601 – System and Network Administration

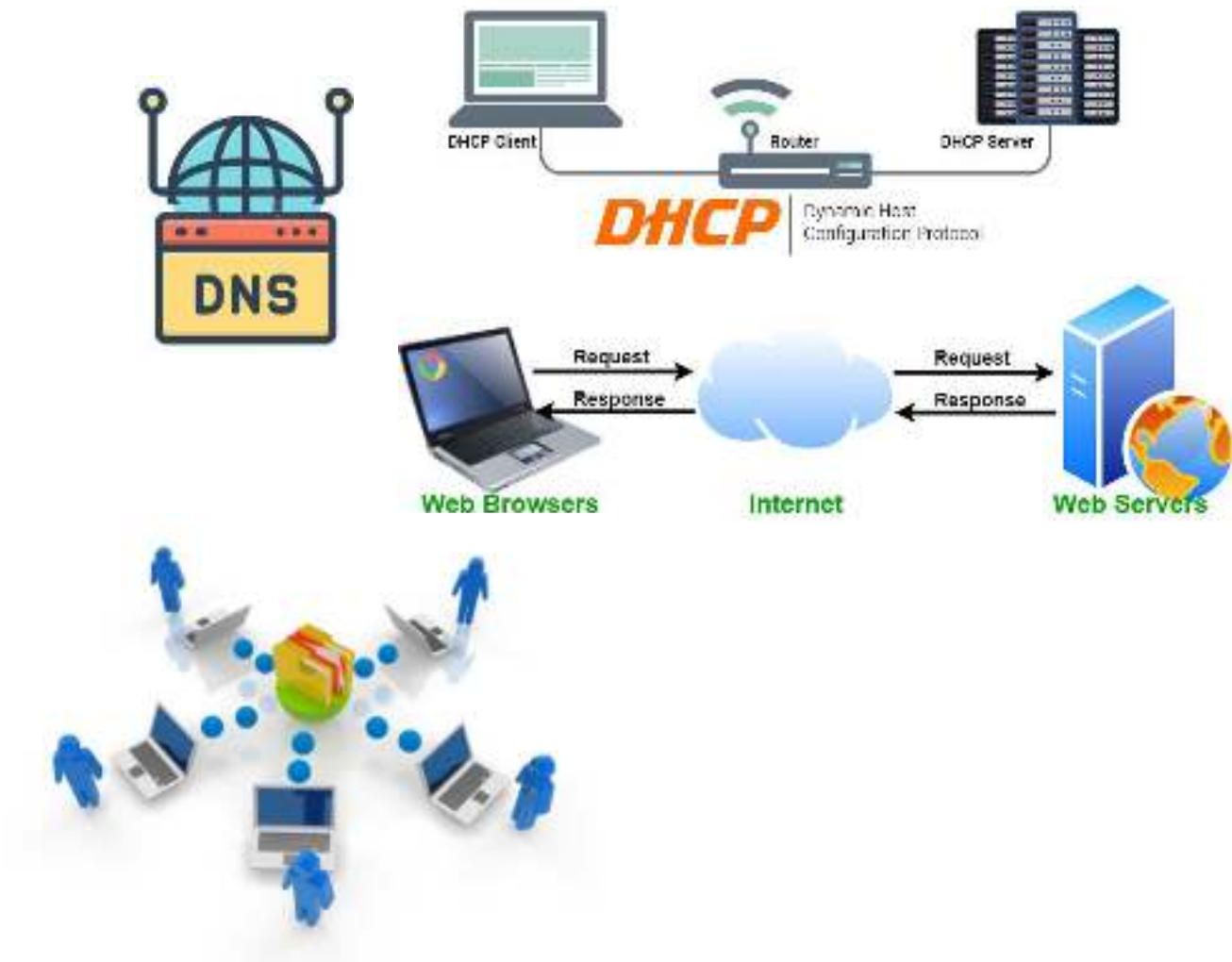
Week 9 (Part 2)



**Department of Computer Science and Information Technology,
Virtual University of Pakistan**



-
-
-
-
-
-

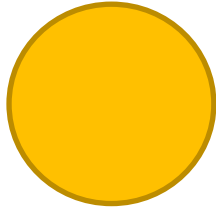


IT601 – System and Network Administration

**Department of Computer Science and Information Technology,
Virtual University of Pakistan**



1 - DHCP Client





—



—



—



—



—



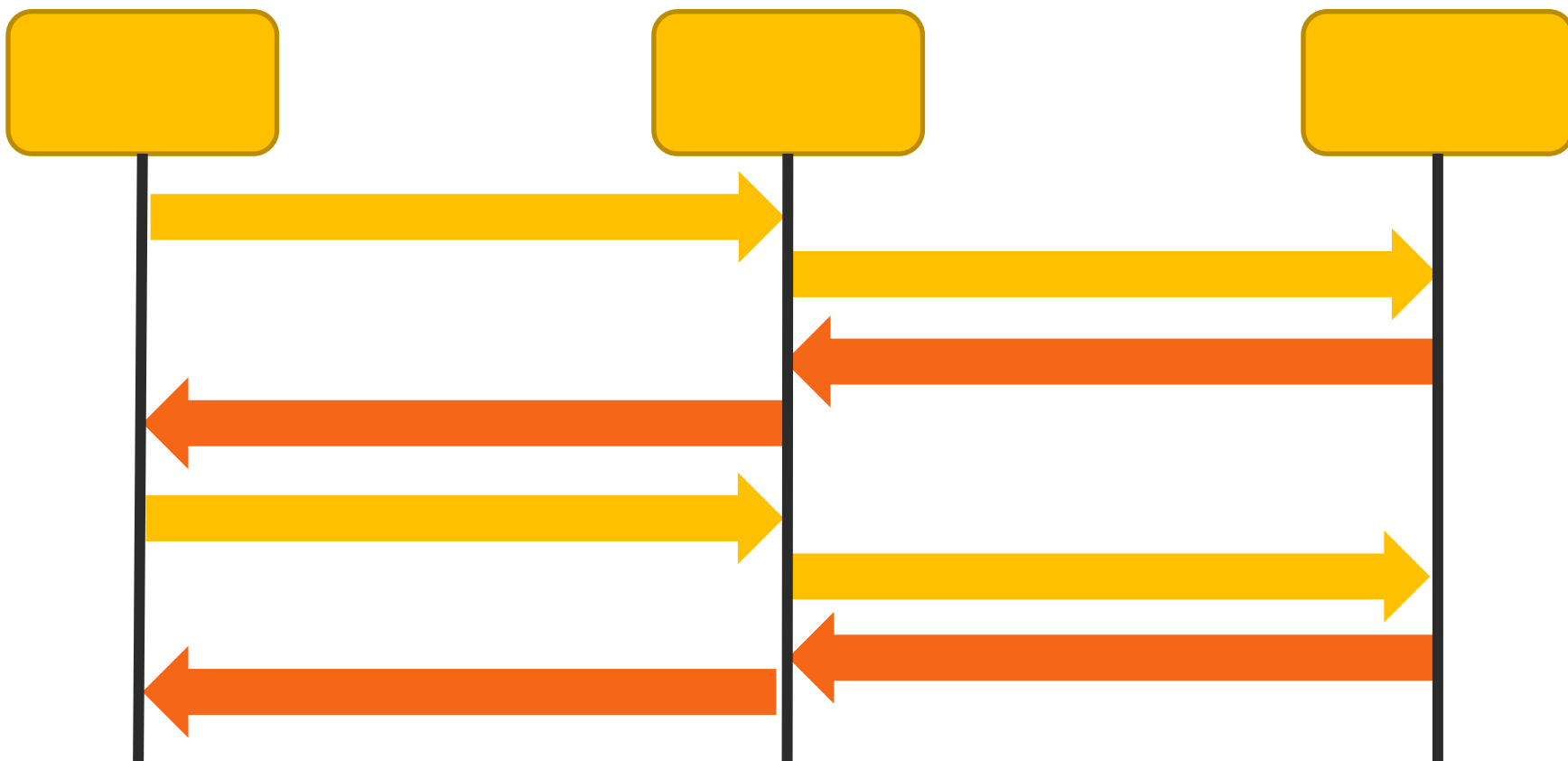
—



—



—





DHCP server, and the configuration is transparent to the computer's user.



1. Client discovers DHCP on switch



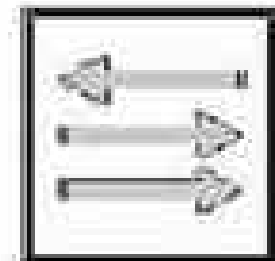
2. Switch offers DHCP to client

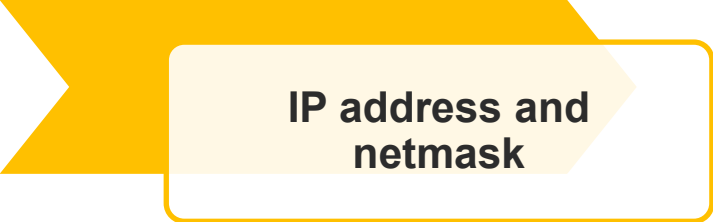


3. Client requests IP address from switch



4. Switch assigns IP address configuration to client

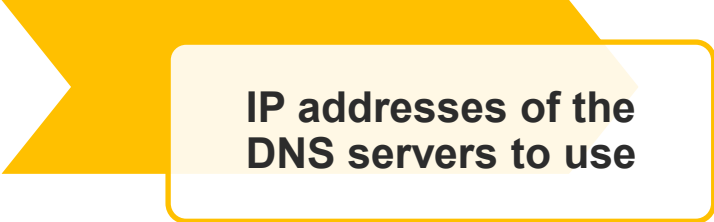




**IP address and
netmask**



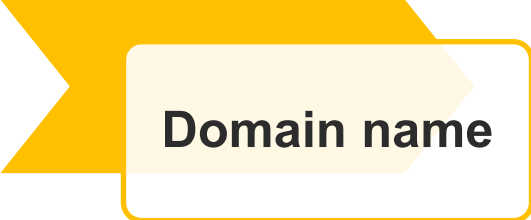
**IP address of the
default-gateway to use**



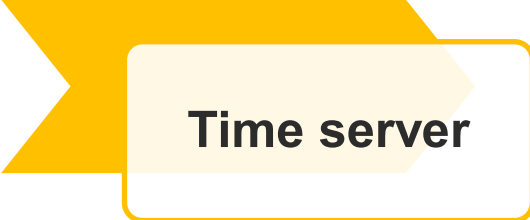
**IP addresses of the
DNS servers to use**



Hostname



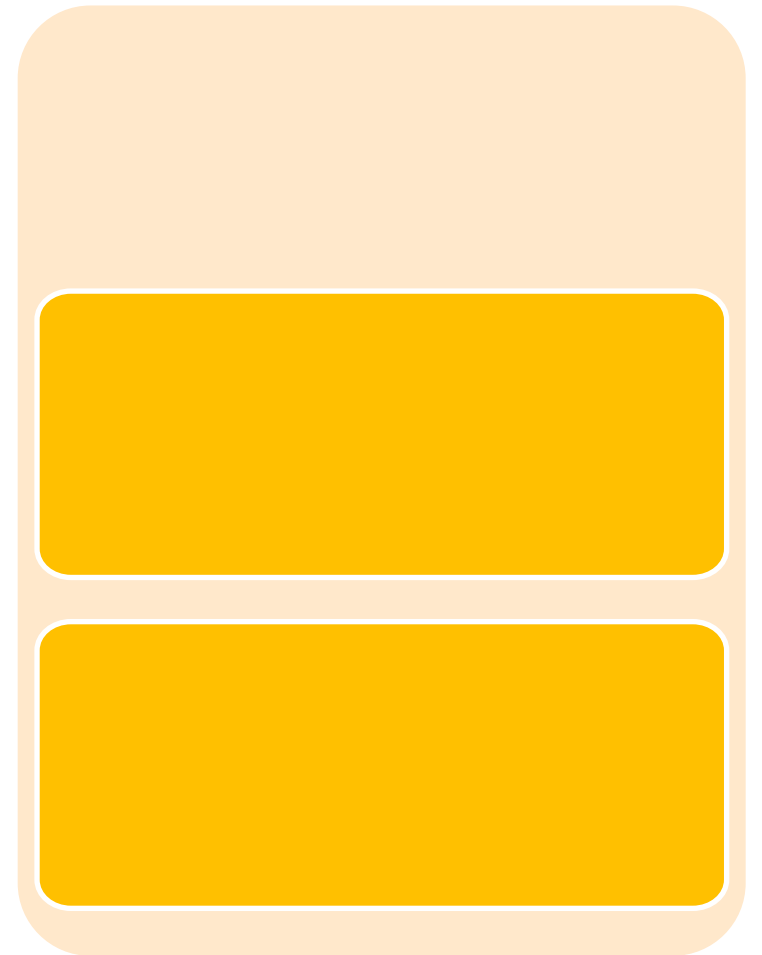
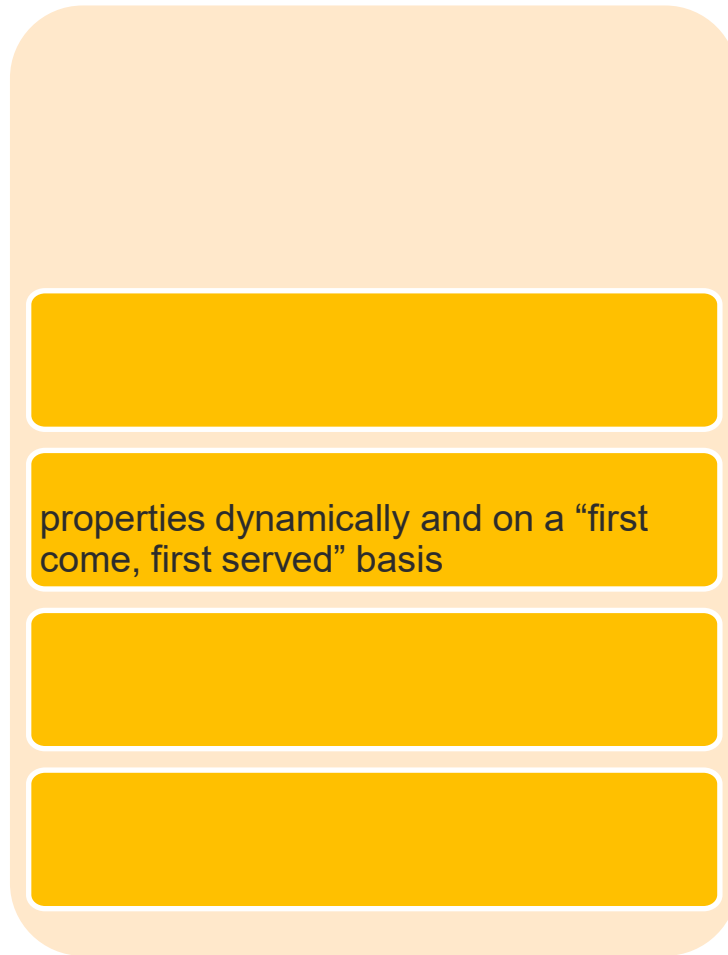
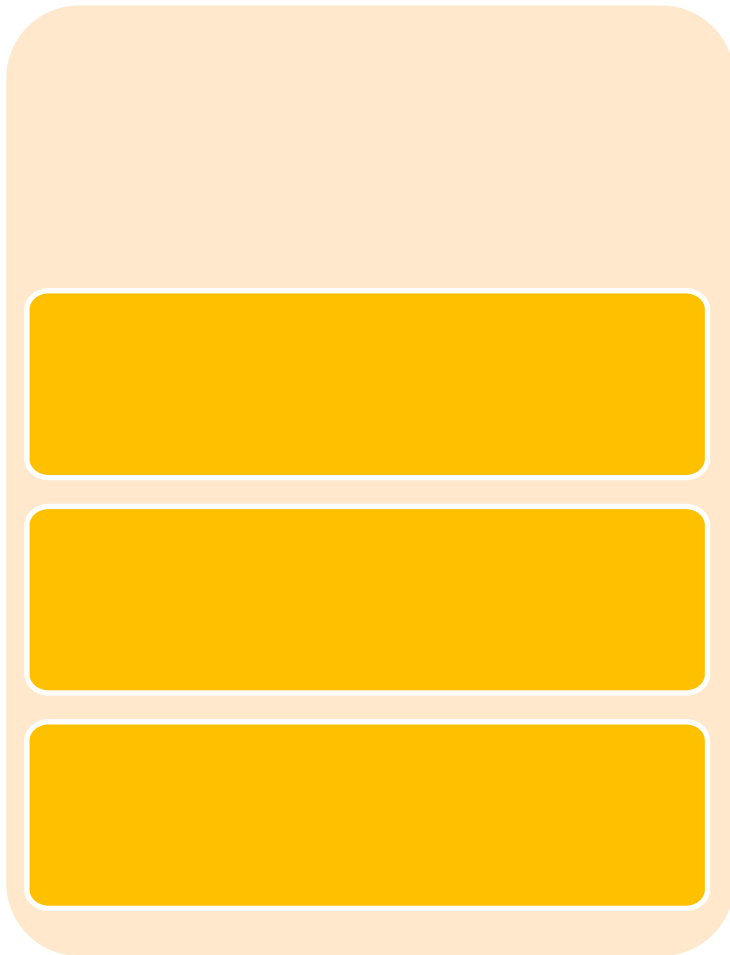
Domain name



Time server



Print server

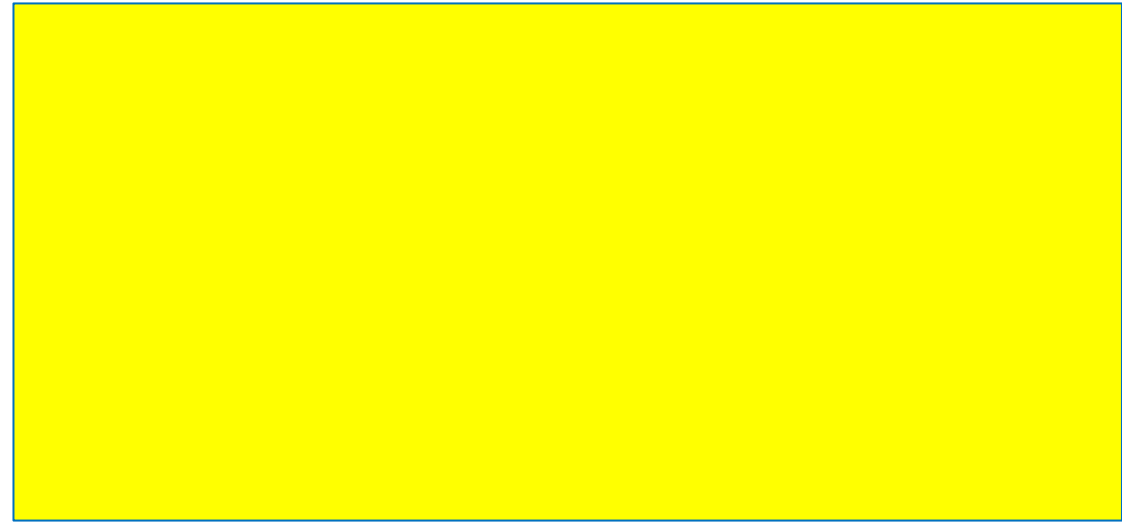


1. Install DHCP Server

-

2. Configure the DHCP Server

-
-
-



```
[21:13:42] # ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens60f0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default
    qlen 1000
    link/ether 00:90:25:b2:b4:c0 brd ff:ff:ff:ff:ff:ff
3: wlp5s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:91:53:3f:56:9b brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.218/24 brd 192.168.1.255 scope global dynamic noprefixroute wlp5s0
        valid_lft 83870sec preferred_lft 83870sec
    inet6 fe80::e0b:3c9e:8aeb:7/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

3. Bind the DHCP Server to an interface

4. Restart the DHCP Server

5. Check the status of the DHCP Server

```
LinuxForDevices: systemctl status isc-dhcp-server.service
● isc-dhcp-server.service: ISC DHCP IPsec Server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enab
   Active: active (running) since Fri 2020-09-18 16:50:50 UTC; 79s ago
     Docs: man:dhcpd(8)
   Main PID: 18083 (dhcpd)
    Tasks: 4 (limit: 2282)
   Memory: 4.0M
   CGroup: /systemd/system/isc-dhcp-server.service
           └─18083 dhcpd -user dhcpd -group dhcpd -f -4 -cf /run/dhcp-server/dhcpd.pid -cf /etc

Sep 18 16:50:50 localhost: dhcpd(18083): ** Ignoring requests on docker0. If this is not what
Sep 18 16:50:50 localhost: dhcpd(18083): you want, please write a subnet declaration
Sep 18 16:50:50 localhost: dhcpd(18083): in your dhcpd.conf file for the network segment
Sep 18 16:50:50 localhost: dhcpd(18083): to which interface docker0 is attached. **
Sep 18 16:50:50 localhost: dhcpd(18083):
Sep 18 16:50:50 localhost: dhcpd(18083): Listening on 192/eth0: /24
Sep 18 16:50:50 localhost: dhcpd(18083): Sending on 192/eth0:/
Sep 18 16:50:50 localhost: dhcpd(18083): Sending on Socket/railback/ta:192.168.1.1
Sep 18 16:50:50 localhost: dhcpd(18083): Sending on Socket/railback/ta:192.168.1.1
Sep 18 16:50:50 localhost: dhcpd(18083): Server starting service.
lines 1-20/20 (END)
```



- **Have an appropriate amount of established IP addresses.**



- **Avoid overlapping static addresses.**



- **DHCP security best practices**

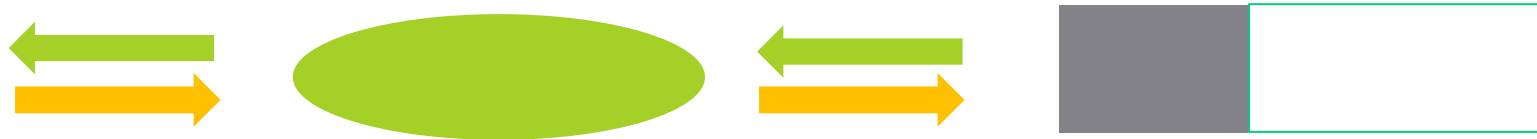


IT601 – System and Network Administration

Week 10



**Department of Computer Science and Information Technology,
Virtual University of Pakistan**



1 - Open Standards

-
-
-

2 - Benefits

-
-

3 - Usage

-

Static web server

documents that don't

CGI servers

Database-driven web sites

Multimedia servers

➤ **Small Vs Large sites**

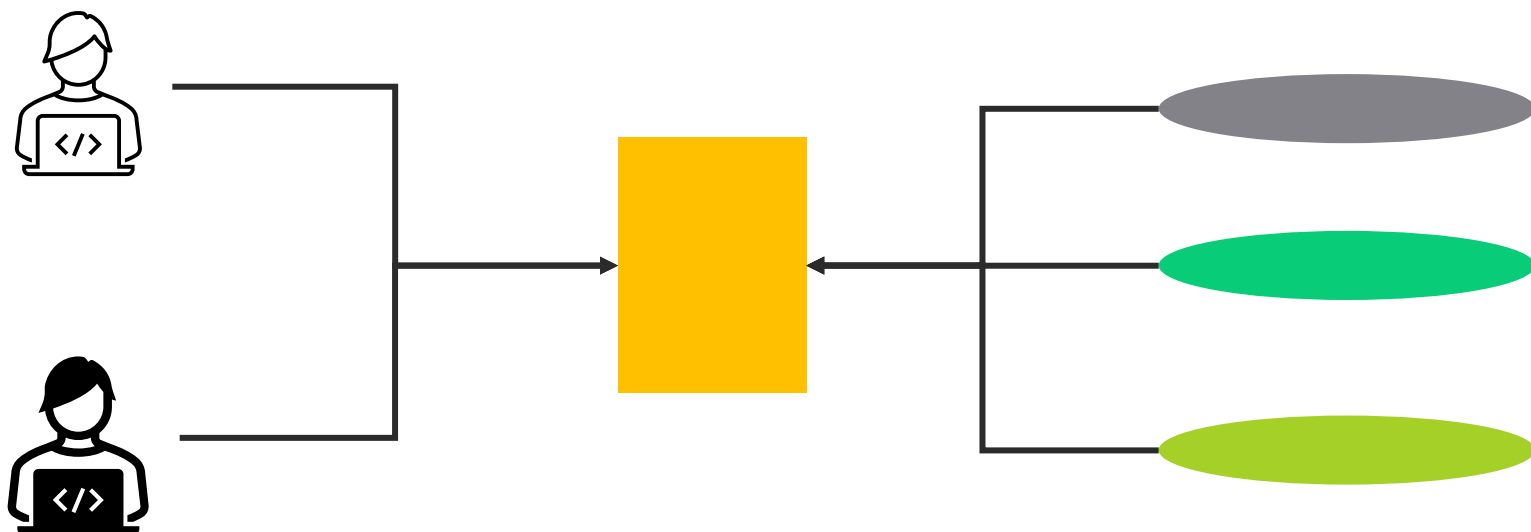
Small Sites	Large Sites

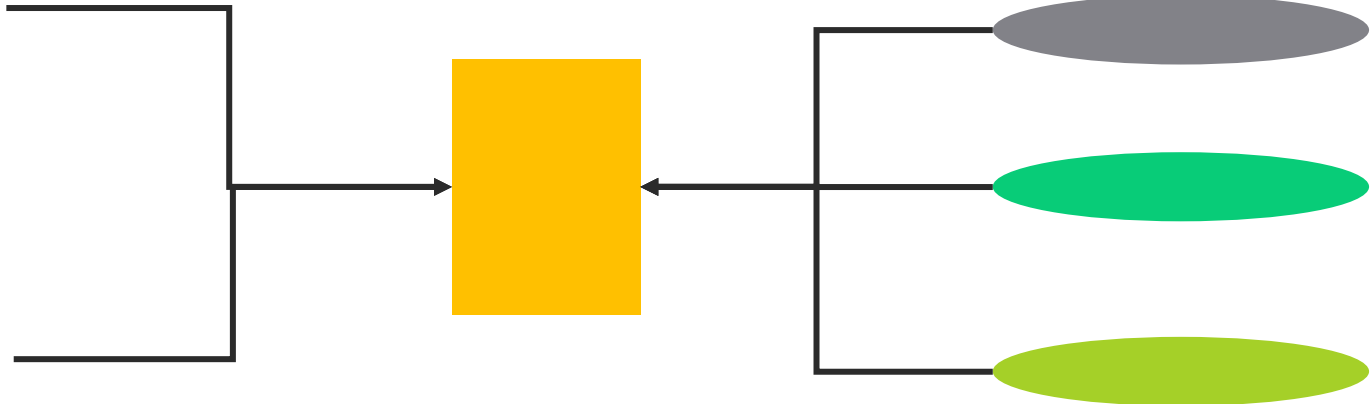
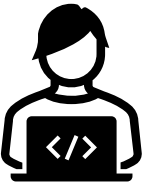
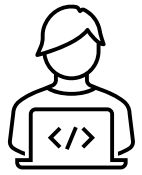
➤ **Multiple Web service Problems**

Single Port	
Multiple IP Addresses	

1 - Virtual hosting

■





➤ Catch-22 Problem

➤ Techniques to Address Catch 22

Share Certificate

for all domains, which isn't usually a problem.

Server name indication

- **A web service needs an SLA.**
- **Generally, web as a 24/7 critical service, but the SLA of an individual web service might be quite different.**
 - ❖
- **Ideally, as with any SLA, the service level should be set by collaborating with the customer community.**



➤ **Why monitoring web services?**

-
-
-

➤ **Be specific about which web-specific elements to your monitoring.**

- Web server errors are most often related to problems with the site's content and are often valuable for
- site's scripts. Other errors may indicate an intrusion attempt. Such scenarios are worth

➤ **Typically, web servers allow logging of the browser client type and of the URL of the page containing the link followed to your site (the referring URL).**

➤ **Web servers may have server-specific information that would be useful as well, such as data on active threads and per-thread memory usage.**

➤ **Become familiar with any special support for extended monitoring available on your web server platform.**

“Scaling is the only problem on the Internet. Everything else is a subproblem.”

➤ **slashdot effect**

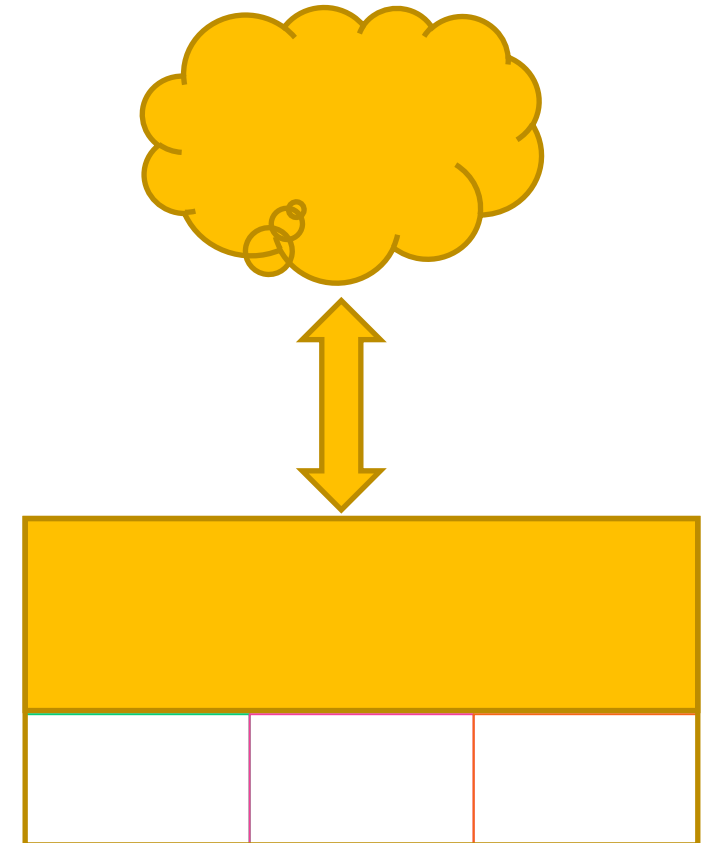
-
-

➤ **Single machine web server Scaling**

- A small organization with basic needs can improve a web server's

➤ **Multi machine web server Scaling**

- - ❖
 - ❖



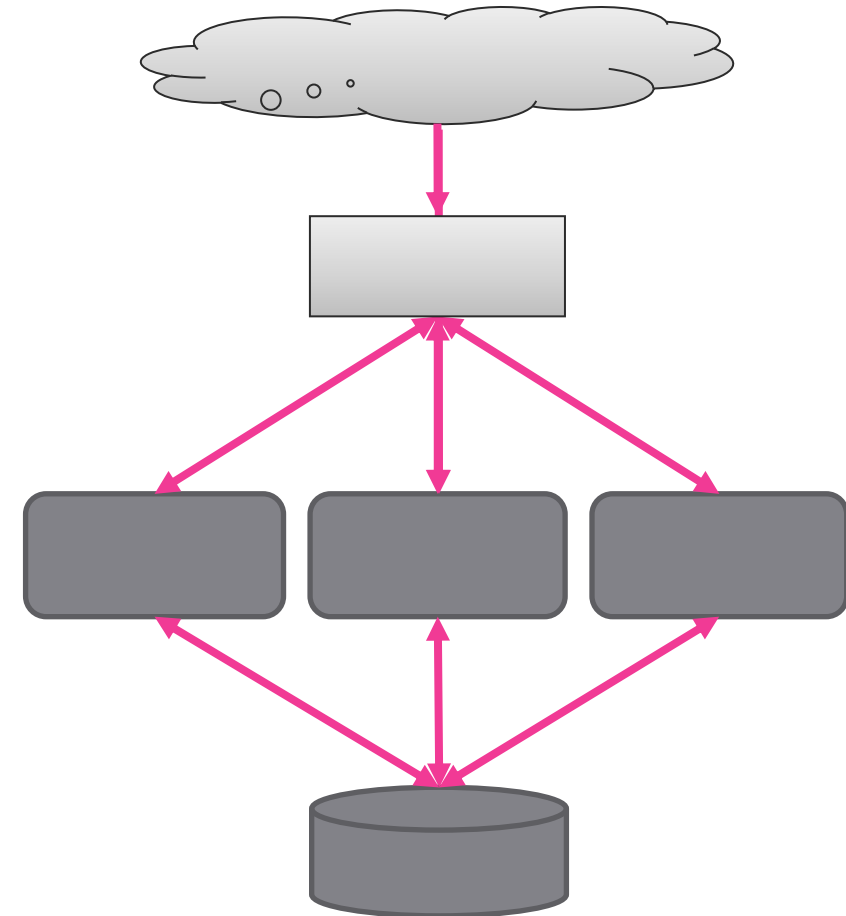
- In horizontal scaling, a web server or web service resource is replicated and the load is divided among the replicated resources.

- **Example**

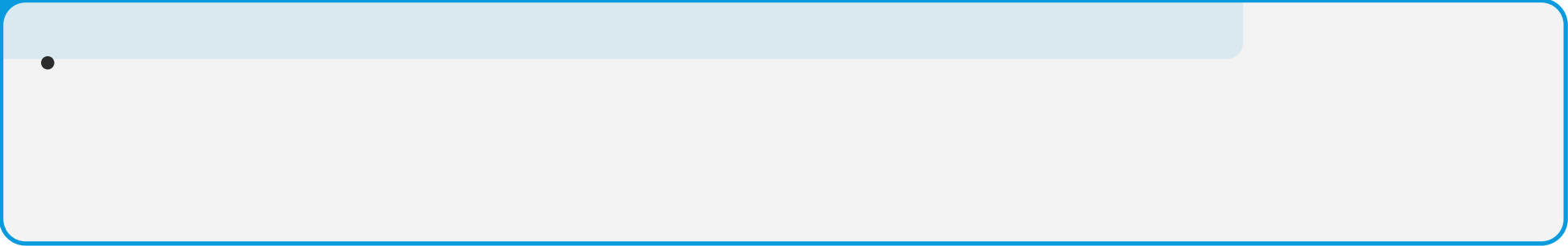
-
-
-
-
-

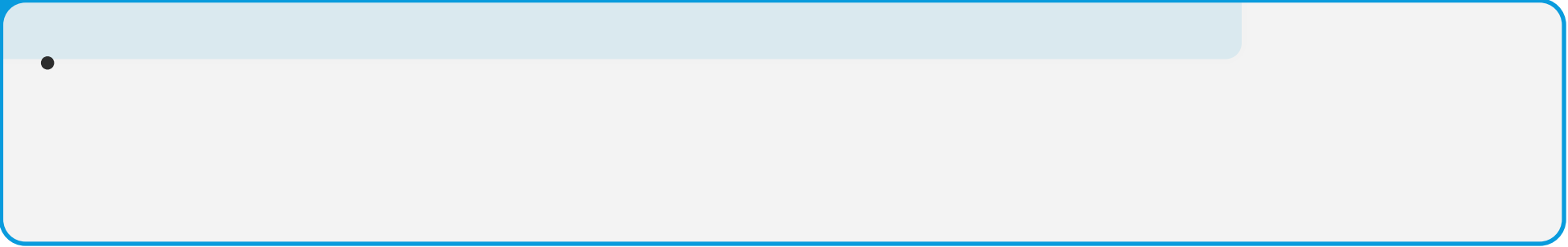
- Load balancers are often general-purpose protocol and traffic shapers, routing not only HTTP but also other protocol requests, as required.

-



- **Separates out the various kinds of subservices, rather than duplicating a whole machine.**
 - **It allows you to create an architecture with finer granularity, so that you can put more resources at the most intensively used stages of page creation.**
 - **It also keeps different types of requests from competing for resources on the same system.**
- **Examples**
 -
 -
 -
- **As you might guess, horizontal and vertical scaling can be combined. The video survey web site might need to add another video clip server before it would need to scale the survey form application.**





-

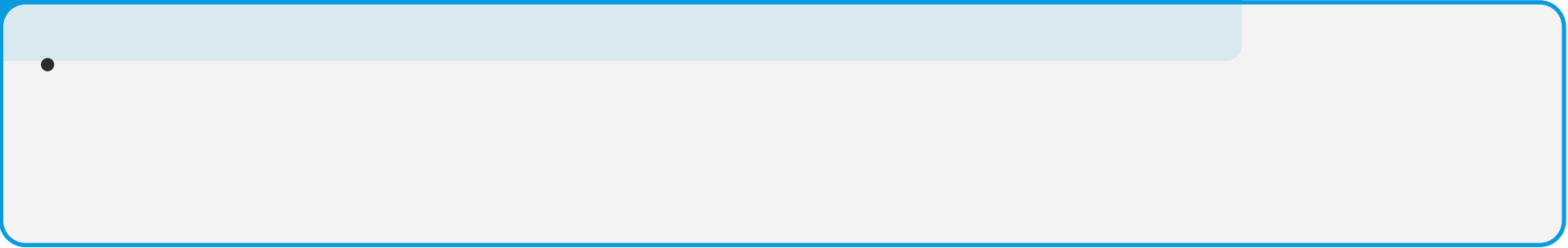
-

-

-

- Databases usually require the lion's share of system resources.

-



-

-

-



➤ Importance of Web server security

-
- Security is a problem because people you don't know are accessing your server.
-
-

➤ The Uses of Web Server Attacks

-
- use the site's resources.
- **Common uses of hijacked sites and bandwidth**
 -
 -
 -
 -

Internal web services should be secured as well. Although you may trust employees of your organization, there are still several reasons to practice good web security internally:

-

-

-

-

-

➤ **Usually web sites are accessed using unencrypted, plaintext communication.**

-
- Use encryption to prevent casual eavesdropping on our customers' web sessions even if they are connecting via a

➤ **Implementing HTTPS on a web server is relatively simple, depending on the web server software being deployed. Properly managing the cryptographic certificates is not so easy.**

-
-

➤ **Places to store private keys**

--	--	--	--

- **A cryptographic certificate is created by the web system administrator using software that comes with the encryption package; OpenSSL is one popular system.**
- **The certificate at that point is “self-signed,” which means that when someone connects to the web server using HTTPS, the communication will be encrypted, but the client that connects has no way to know that it has connected to the right machine.**

- Anyone can generate a certificate for any domain. It is easy to trick a web client into connecting to a “man in the middle” attacker instead of to the real server; the client won’t know the difference. This is why most web



- ❖ When it connects to the web site, a client reads the signed certificate and knows that the site’s certificate can

- **A variety of malicious efforts can be directed against the web server itself in an attempt to get login access to the machine or administrative access to the service.**
- **Any vulnerabilities present in the operating system can be addressed by standard security methods.**
- **Web-specific vulnerabilities can be in multiple layers of the web server implementation such as**
 -
 -
 -
- **The best way to stay up-to-date on web server security at those layers is through vendor support.**
 -

- **Some web-intrusion attempts are directed toward gaining access to the content or service rather than to the server..**
 -
- **SAs should educate themselves on current exploits via Internet security resources.**
 -
 -



•

•

•

•

•

•

•

•

•

•

•

•

•

•

•



- **Providing a content-management system (CMS) empowers users to perform self-service updates of the web content.**

-

- **It is not a good idea for an SA to be directly involved with content updates. Assuming this responsibility not only adds to the usually lengthy to-do list of the SA, but also creates a bottleneck between the creators of the content and the publishing process.**

-

- “voices” from each group are already working with the webmaster or the SA who is being a temporary webmaster.

-

- **This process should have a specific policy on updates. Ideally, the policy should distinguish three types of alterations that might have different processes associated with them:**

-

-

-

➤ There are several web server package available for Linux and windows. The commonly used packages are...

-
- It's an open
-
-
-
-

-
-
-
-
-
- can't disable some of

- option that's specifically
-
-
-
-

-
-
-
-

➤ Install apache2 Package

-
-

➤ By default, Apache comes with a basic site

-
- -
 -
 -
 -
 -
 -
 -

➤ Operational Commands

- *start|restart|reload|stop*
-
-
-
-
-

➤ **Active Apache modules**

➤ **Commonly used Apache modules**

-
-
-
-
-
-

Mod_security



-
-
-
-



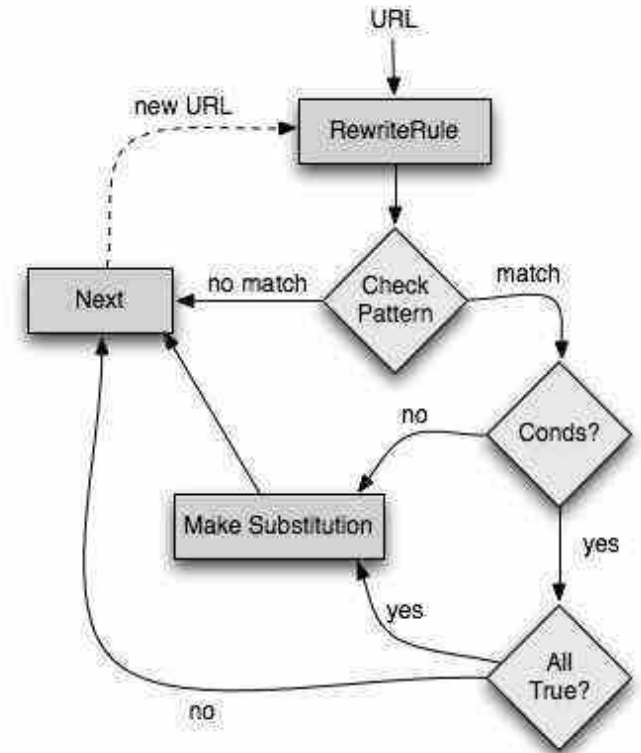
mod_rewrite



-
-



that starts with 'http' to 'https' is given below.



https://httpd.apache.org/docs/2.4/mod/mod_rewrite.html

mod_deflate



mod_cache



■

■

■

■

mod_proxy

➤ **This is an optional Apache module.**

-
-

➤ **To enable this feature, a set of modules will need to be loaded onto the server. This include the following.**

mod_ssl



- **The .htaccess file in Apache is a tool that allows configurations at the directory and subdirectory level.**

- **Using .htaccess enables you to configure website permissions without altering server configuration files.**

- **Enabling .htaccess**

-
-

- **Common Uses**

-
-
-
-

➤ Listen:

-
-

Listen 8000

➤ ServerName:

-

ServerName YourHostNameOrIPAddress:8000

➤ **ServerRoot:**

-

Assume that Apache HTTP server is installed in “/Project/apache2”

ServerRoot “/Project/apache2”

➤ DocumentRoot:

-

```
DocumentRoot "/Porject/apache2/htdocs"
```

```
<Directory "/Porject/apache2/htdocs">
```

```
Options Indexes FollowSymLinks
```

```
AllowOverride None
```

```
Order allow,deny
```

```
Allow from all
```

```
</Directory>
```

IT601 – System and Network Administration

Week 11 (Part 1)



**Department of Computer Science and Information Technology,
Virtual University of Pakistan**





Identification or Authentication

-
-

Authorization

-
-

Administration and Management

-
-

Monitoring and Auditing

- a user's activities.
-

Security and Protection

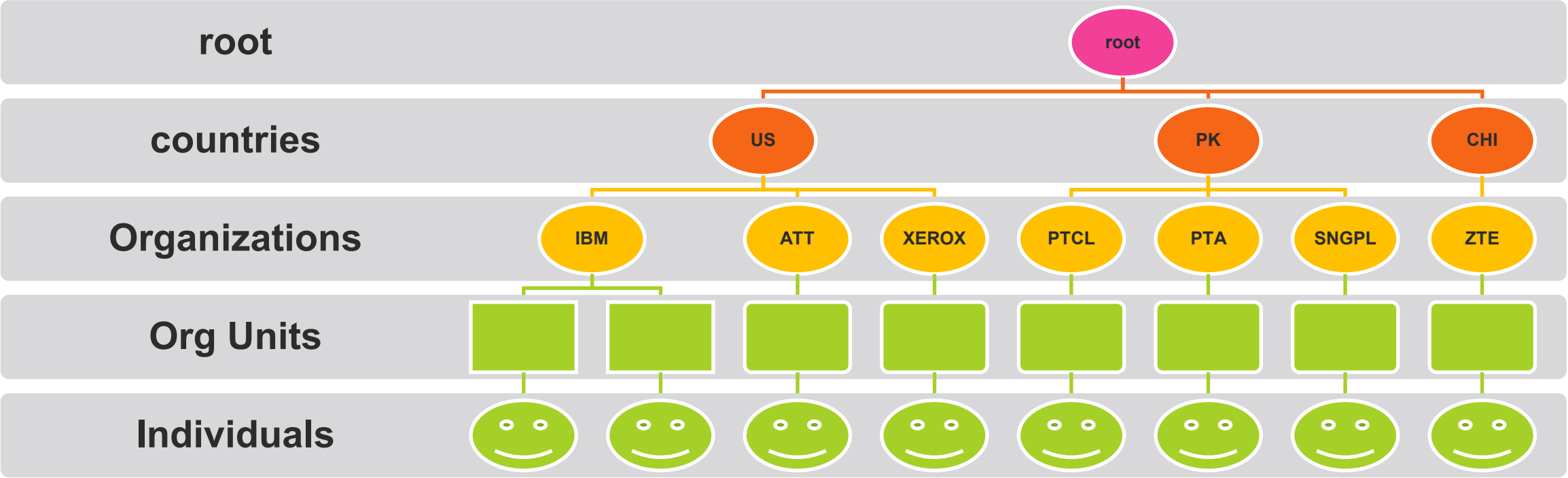
-



IT601 – System and Network Administration

**Department of Computer Science and Information Technology,
Virtual University of Pakistan**

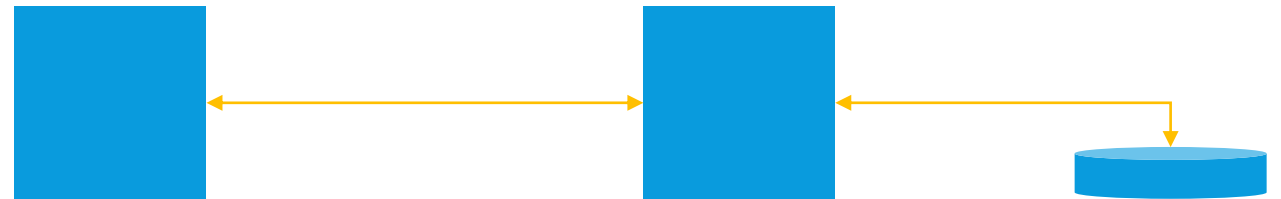
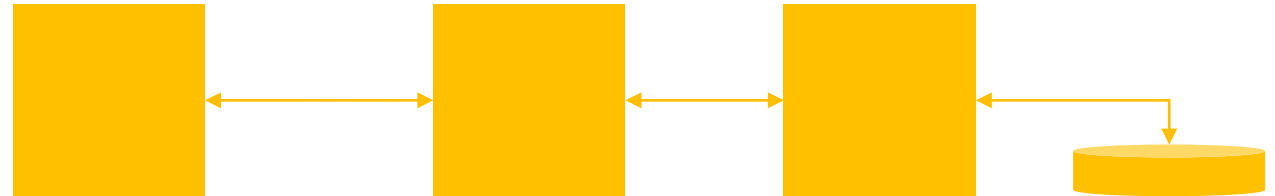






others...

- Uses strings rather than DAP's ASN.1



➤ Information

-

➤ Naming

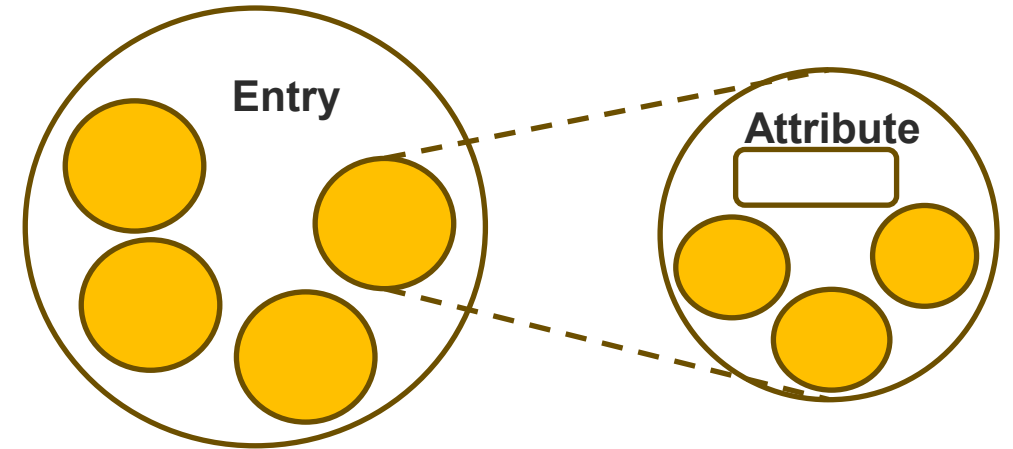
-

➤ Functional / Operations

-

➤ Security

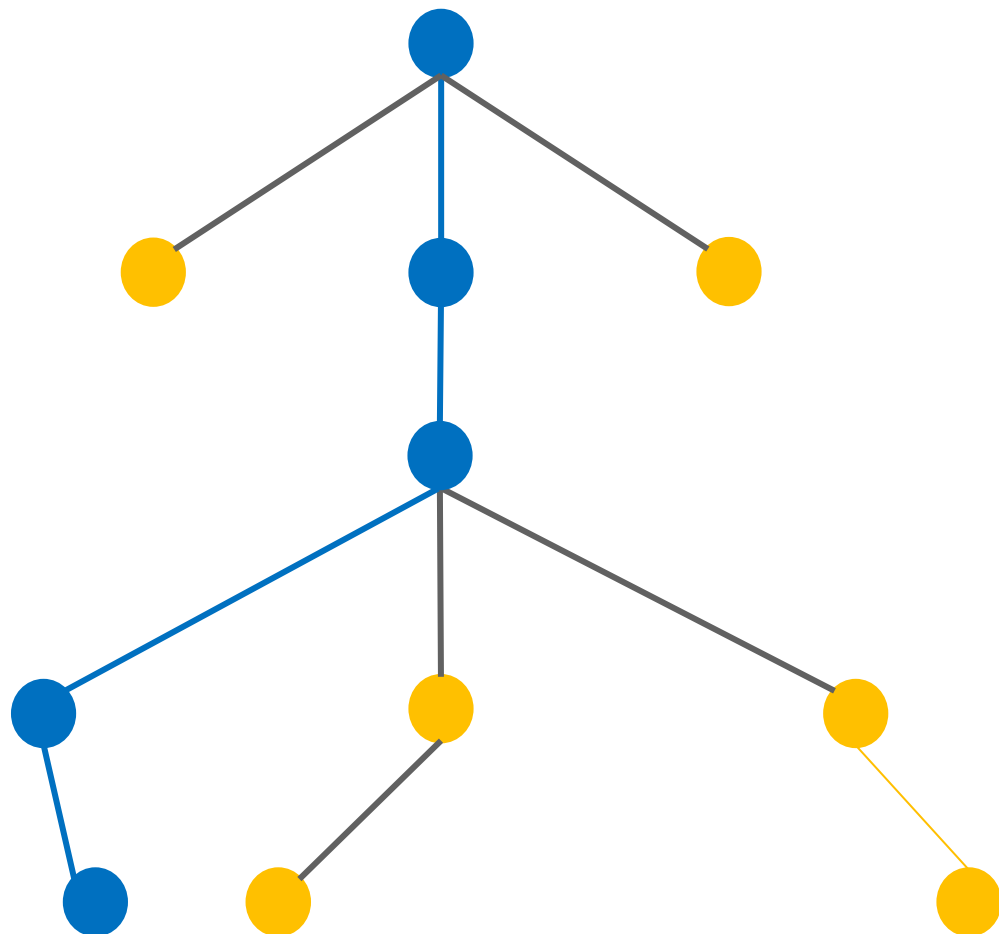
-





“

=vu”

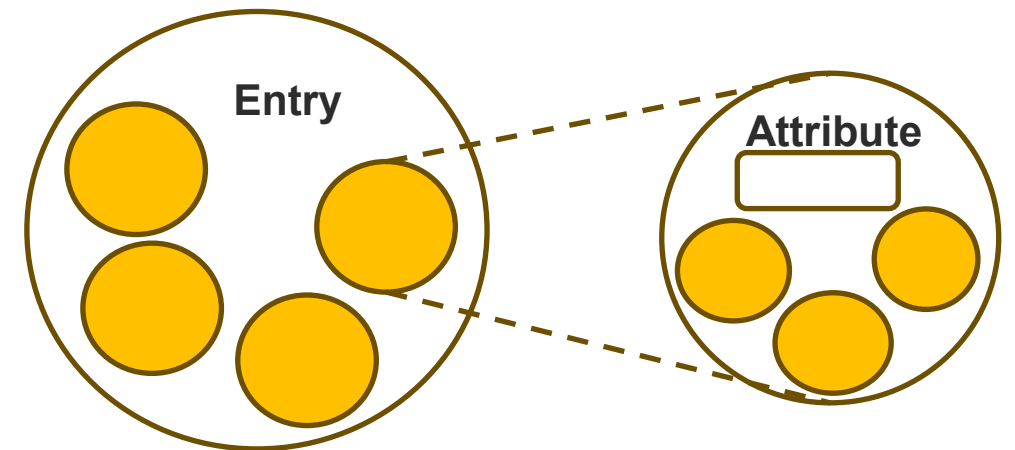


Schema

-
-
-
-
-



- -
 -
 -
- -
-



•

•

•

•

•

•

•

•

•



-
-
-
-
-
-

**Client establishes session
with server (BIND)**

-
-
-
-
-



Client performs operations

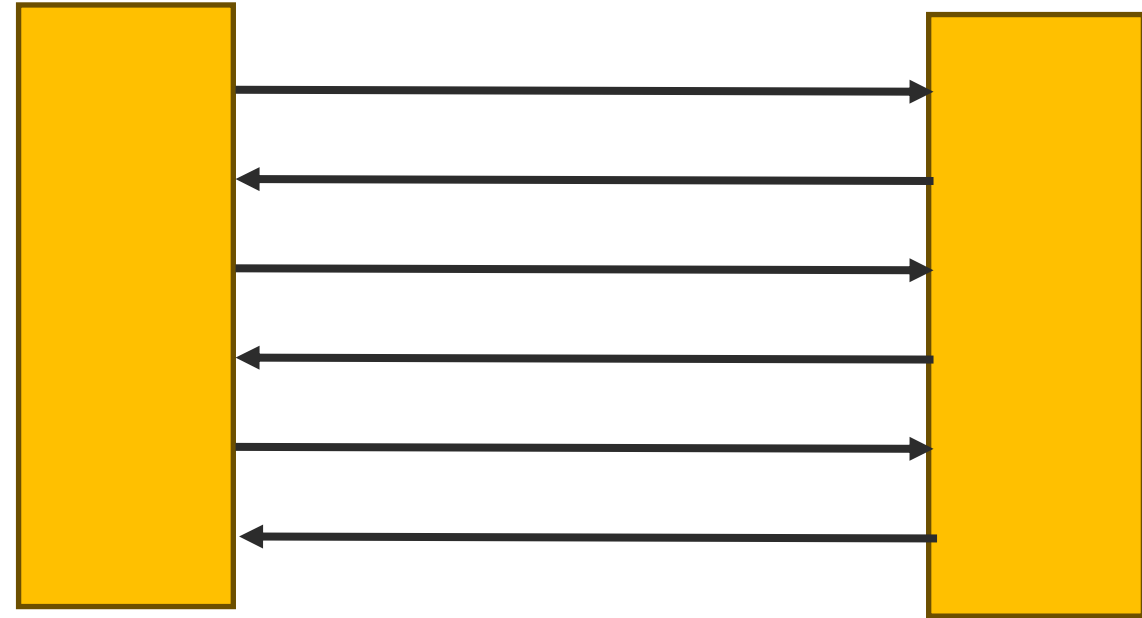
-
-
-



**Client ends the session
(UNBIND)**

**Client can ABANDON the
session**

➤ LDAP using Simple Authentication and Security Layer (SASL)With SSL/TLS



➤ Three types of LDAP Operations are Authentication, Query and Update

 **Search**

 **Compare**

 **Add**

 **Modify**

 **Delete**

 **Rename**

 **Bind**

 **Unbind**

 **Abandon**

 **Extended**

➤ Idapbind

-
-
- **Usage**
- **Scenario**



➤ Idapsearch

- Usage
- Scenario

```
b "ou=people,dc=vu,dc=com" "objectclass=*" 
```

➤ Idapadd

- **Usage**
- **Scenario**



➤ Idapdelete

- Usage
- Scenario



➤ Idapmodify

-
-
- -
 -
 -
 -
- **Usage**
- **Scenario**



➤ Idapmoddn

- -
 -
- **Usage**
- **Scenario**



➤ **Current LDAP version supports**

-
-

➤ **Other authentication methods possible**

➤ **SASL support added in version 3**

-
-
- Kerberos deemed stronger than SASL...

- **Security based on the BIND model**
- **Clear text -> ver 1**
- **Kerberos -> ver 1,2,3 (depr)**
- **SASL -> ver 3**
 -
 -
- **Proposal for Transport Layer Security**
 -

➤ No Authentication

➤ Basic Authentication

-
-

➤ SASL (RFC 2222)

-
-
-
-
-

➤ **C Library API**

- RFC 1823 'The LDAP API'
- —

➤ **Java JNDI**

➤ **LDAP v3 uses the UTF-8 encoding of the Unicode character set.**

➤ **HTTP to LDAP gateway**

➤ **LDAP to X.500 gateway – ldapd**

➤ Referrals

-

➤ Security

-

➤ Internationalization

-

➤ Extensibility

-



➤ **Step 1: Set hostname for the Ubuntu server**

➤ **Step 2: Install OpenLDAP Server**



➤ **Step 3: Add base dn for Users and Groups**

-

➤ **Step 4: Add User Accounts and Groups**

-

-

-

-

➤ Step 5: Install LDAP Account Manager

-

-

-

-

IT601 – System and Network Administration

Week 11 (Part 2)



**Department of Computer Science and Information Technology,
Virtual University of Pakistan**

- **Domain Name Service (DNS)** is an Internet service that maps IP addresses and fully qualified domain names (FQDN) to one another.
- **DNS service** alleviates the need to remember IP addresses. Computers that run DNS are called **name servers**.
- **Ubuntu** is shipped with **BIND (Berkley Internet Naming Daemon)**, the most common program used for maintaining a name server on Linux.
- **Fully Qualified Domain Name (FQDN)**

[Redacted]

[Redacted]

[Redacted]

[Redacted]

• DNS Terminology

-
-
-
-
-
-
-
-
-
-

DNS Components

-
-
-
-
-