CYBERSECURITY
SPECIAL INTEREST GROUP
3RD MEETING

MAY 27, 2021

5:30 PM – 6:30 PM

AGENDA

1. Opening Remarks – Dr. Arshad Ali – 10 Min
2. Security Management at University – Dr. Saad A. Malik, Namal – 15 Min
3. Setting up a SOC in University – Mr. Rizwan Ali, SPS – 25 Min
4. General Discussion – 10 Min

AGENDA

1. Opening Remarks – Dr. Arshad Ali – 10 Min
2. Security Management at University – Dr. Saad A. Malik, Namal – 15 Min
3. Setting up a SOC in University – Mr. Rizwan Ali, SPS – 25 Min
4. General Discussion – 10 Min

Security Management at a University Part-II

by

Saad A. Malik*

SPS/SPINN Lab Special Interest Group on Cyber Security Weekly Talks

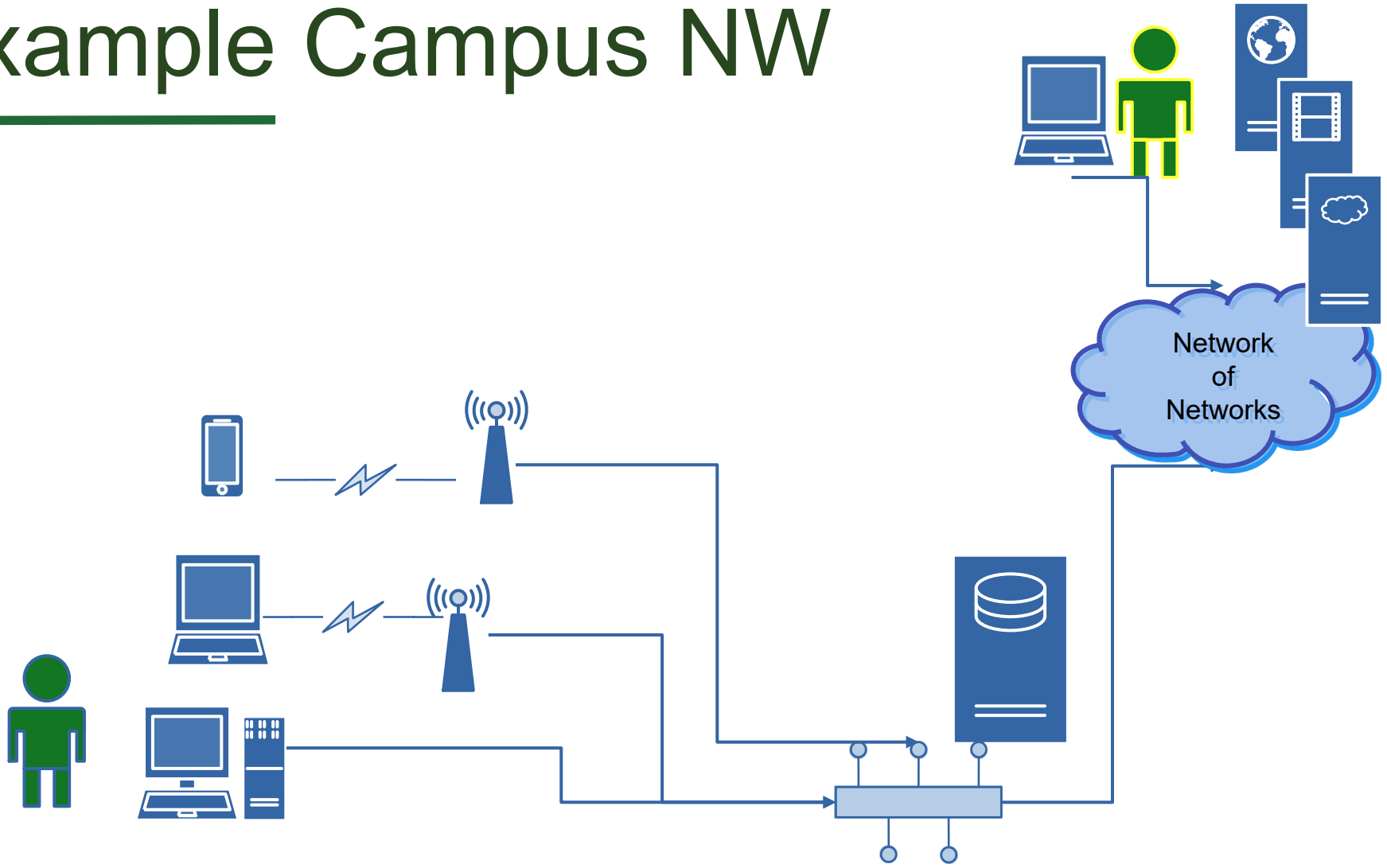
Dated: 27 May 2021

*Saad A. Malik (Ph.D., Engr.)
Asst. Prof. CS Department,
Head ITSC & SDC,
Namal Institute Mianwali, Pakistan.
Email: saad.ali@namal.edu.pk; saad.malik@spsnet.com
Mob: +92 (0) 332 860 7168

Outline of the talk

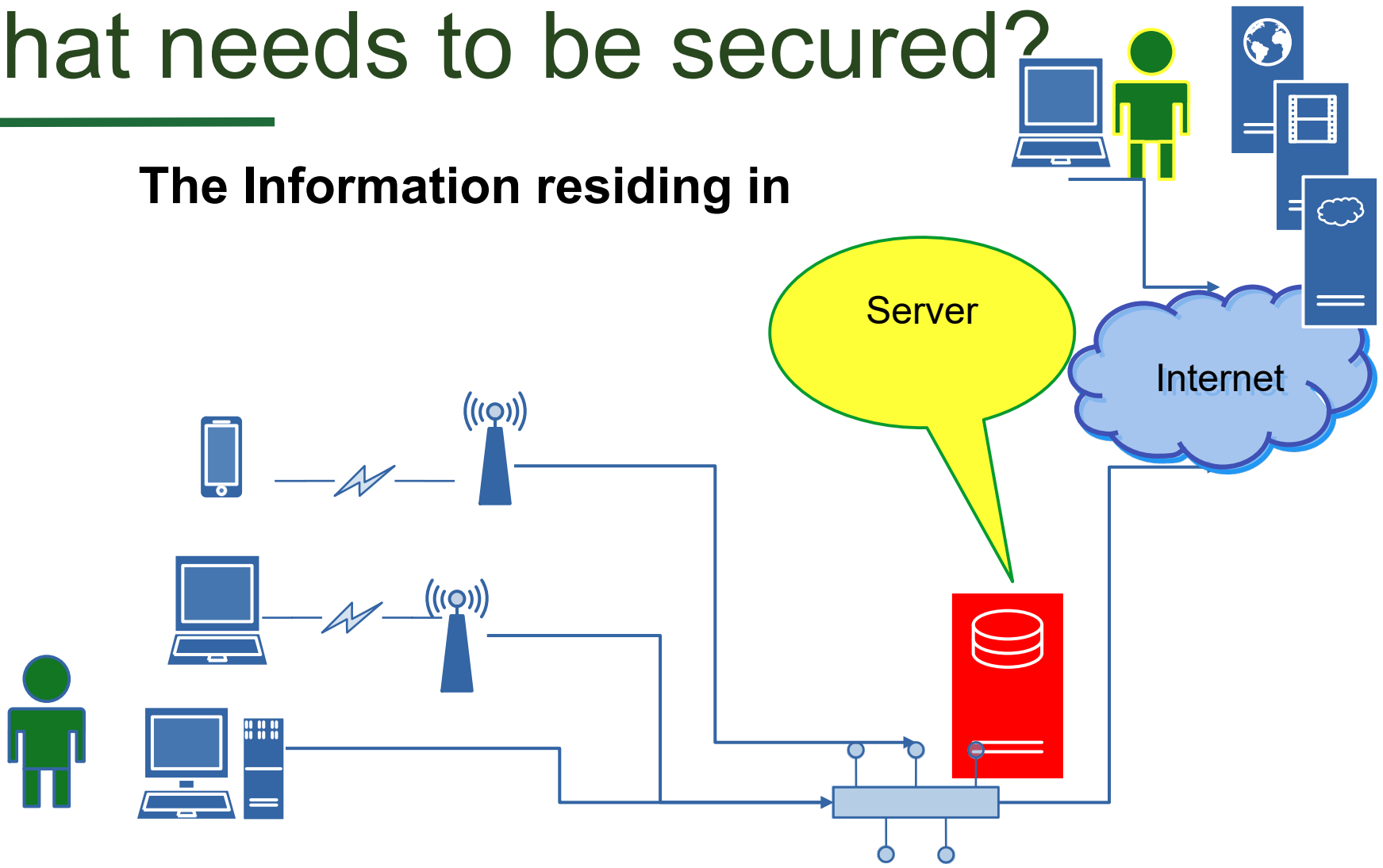
- Review a campus network for a Denial of Service Attack.

Example Campus NW



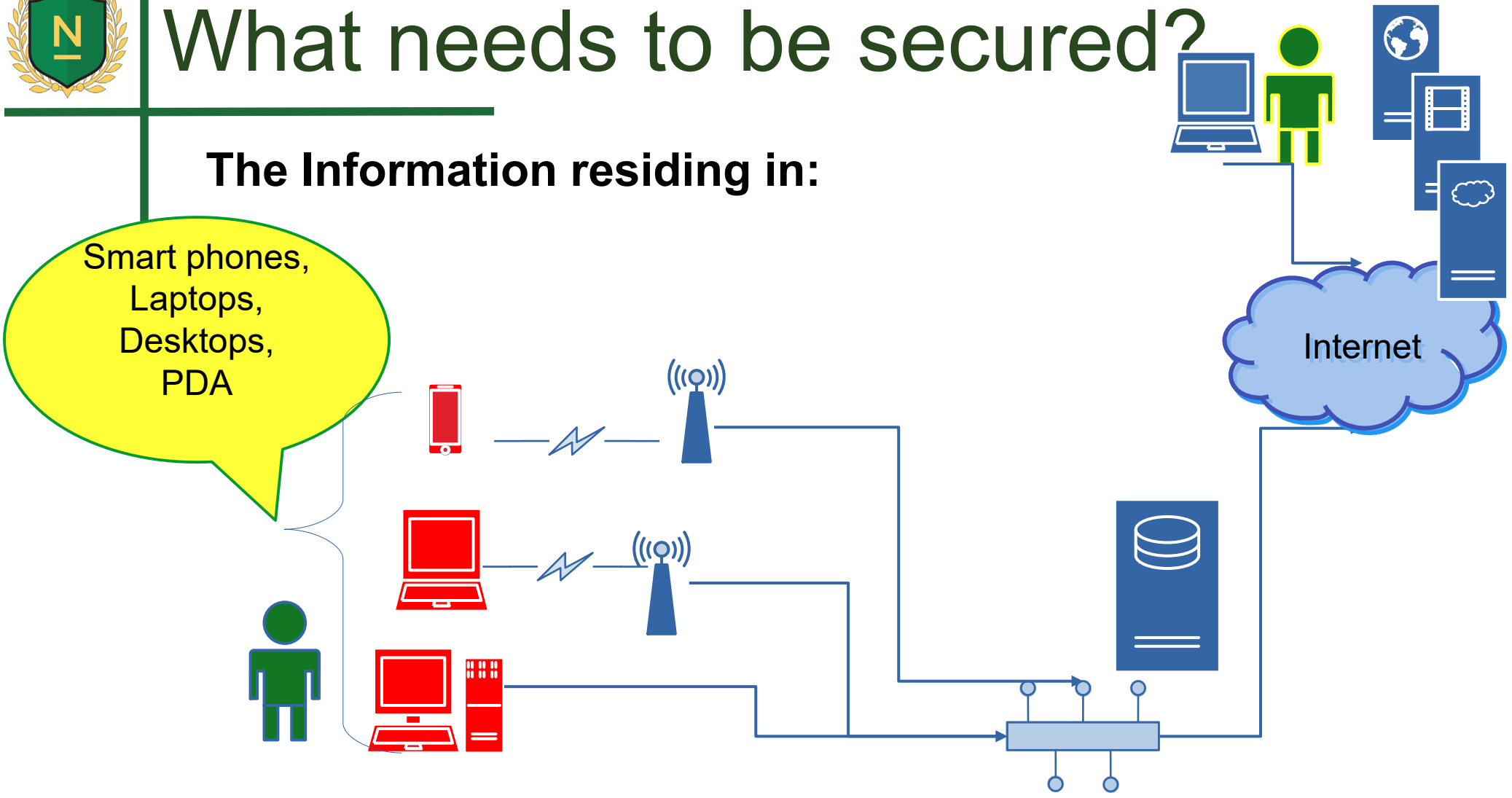
What needs to be secured?

The Information residing in



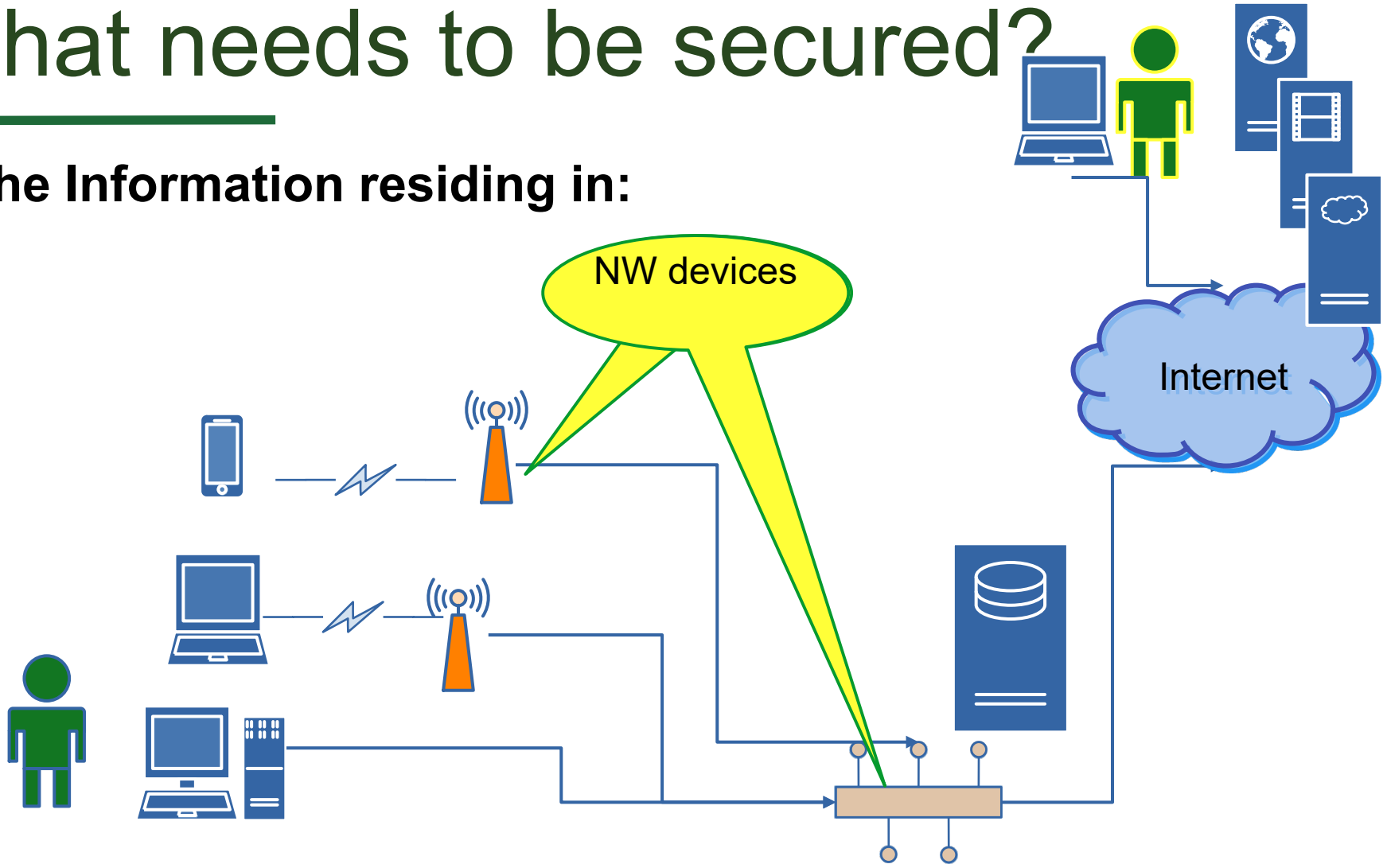
What needs to be secured?

The Information residing in:

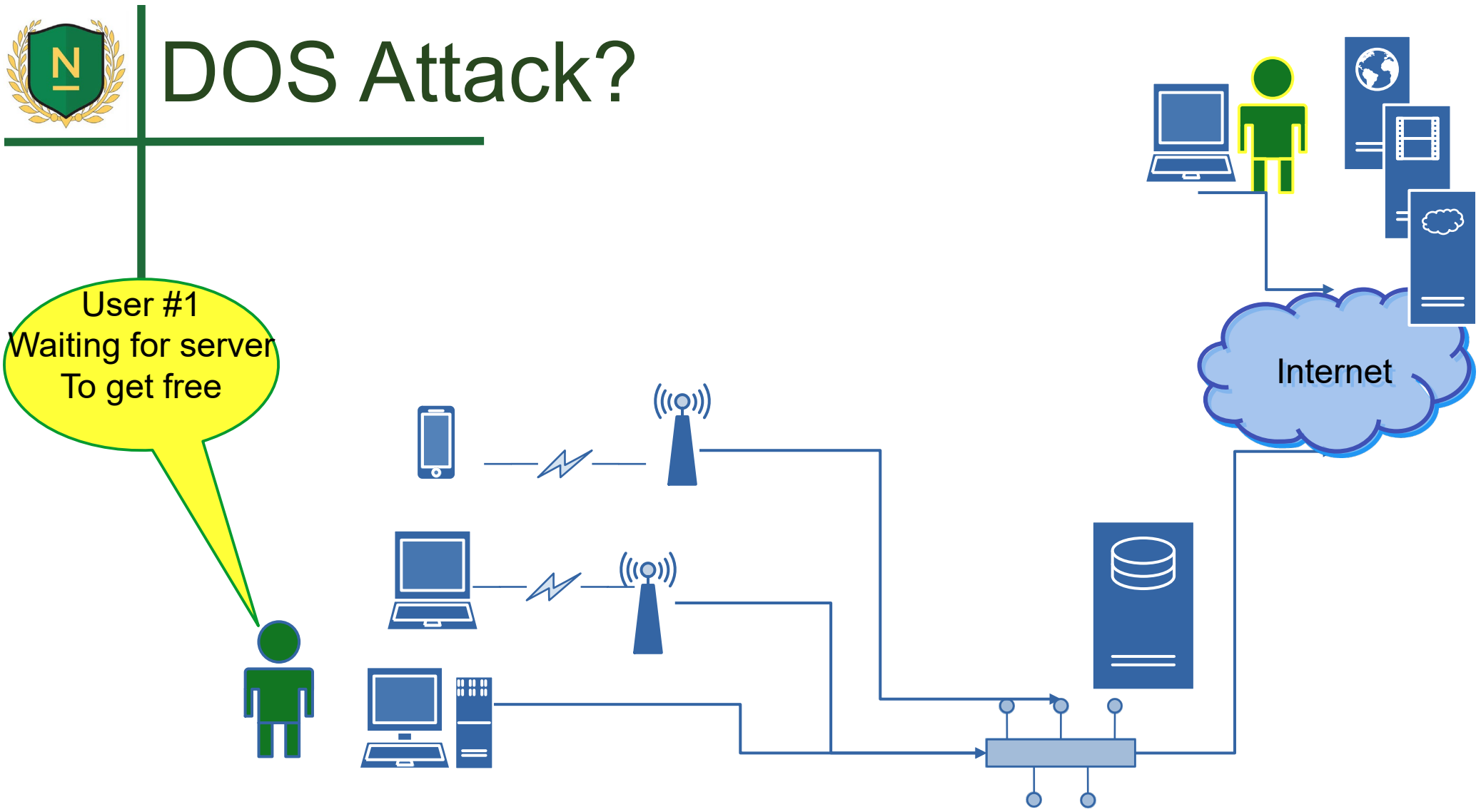


What needs to be secured?

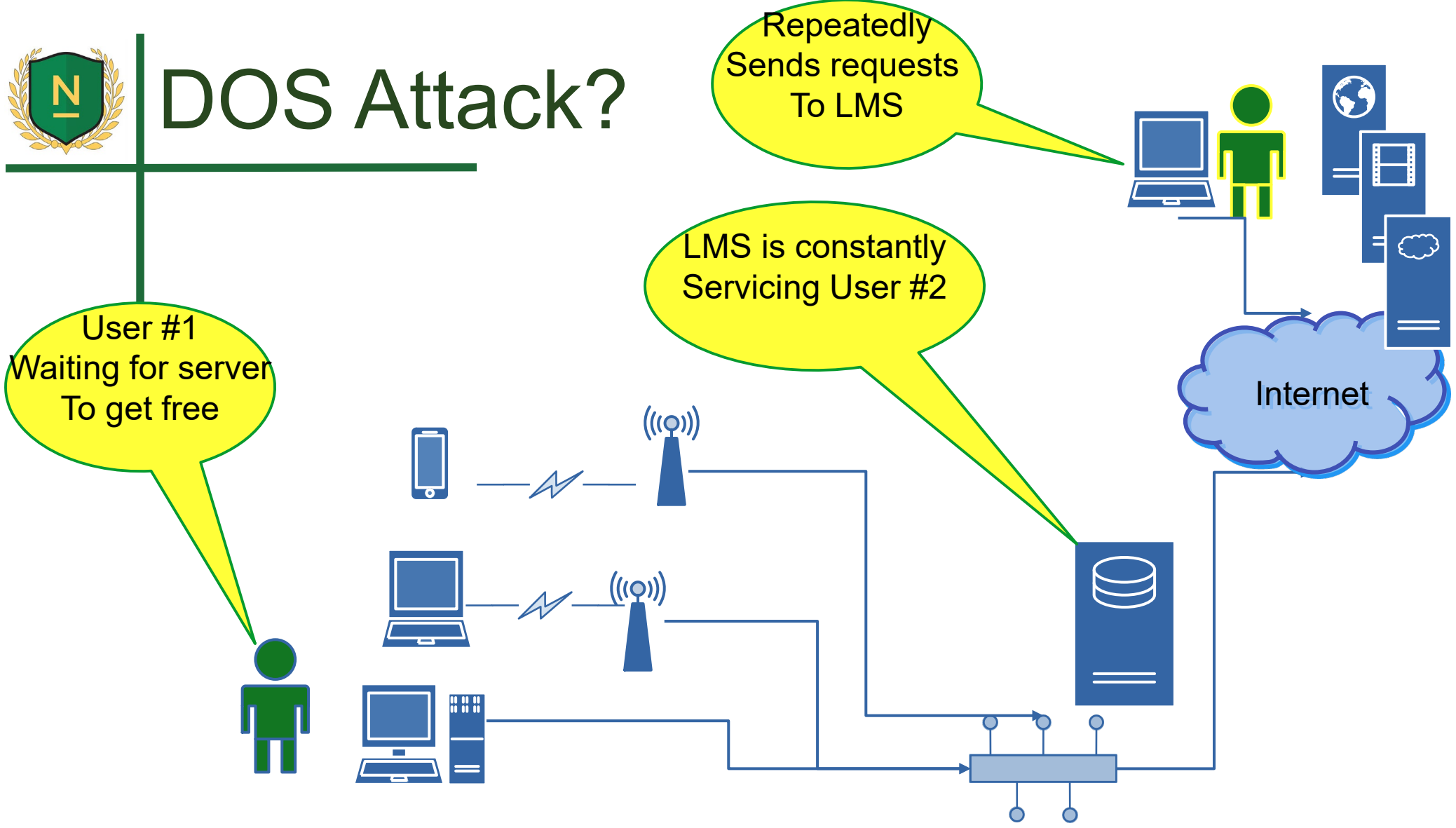
The Information residing in:



DOS Attack?



DOS Attack?


Repeatedly
Sends requests
To LMS
LMS is constantly
Servicing User #2
User #1
Waiting for server
To get free
Internet

Back to the Basics:

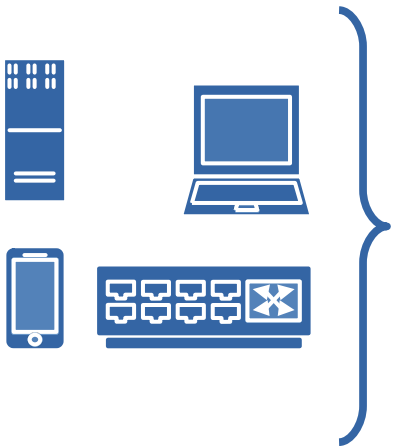
- how information is accessed over the network?
– – How information is stored in a computer?
– – How it is made available on the network?
– – how DOS attacks work!

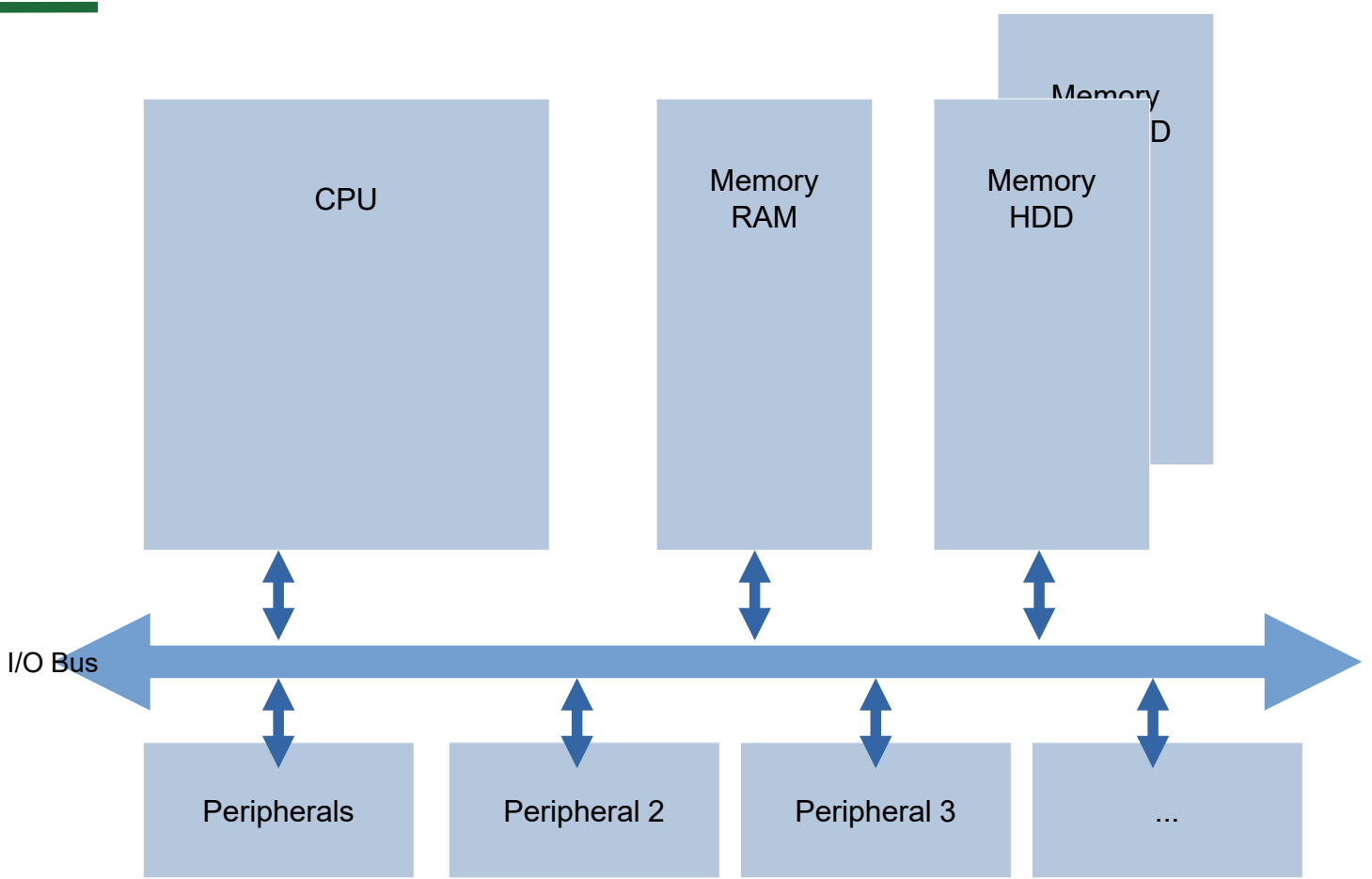
Computers are Everywhere!

An internal Computer keeps each of the devices operational!

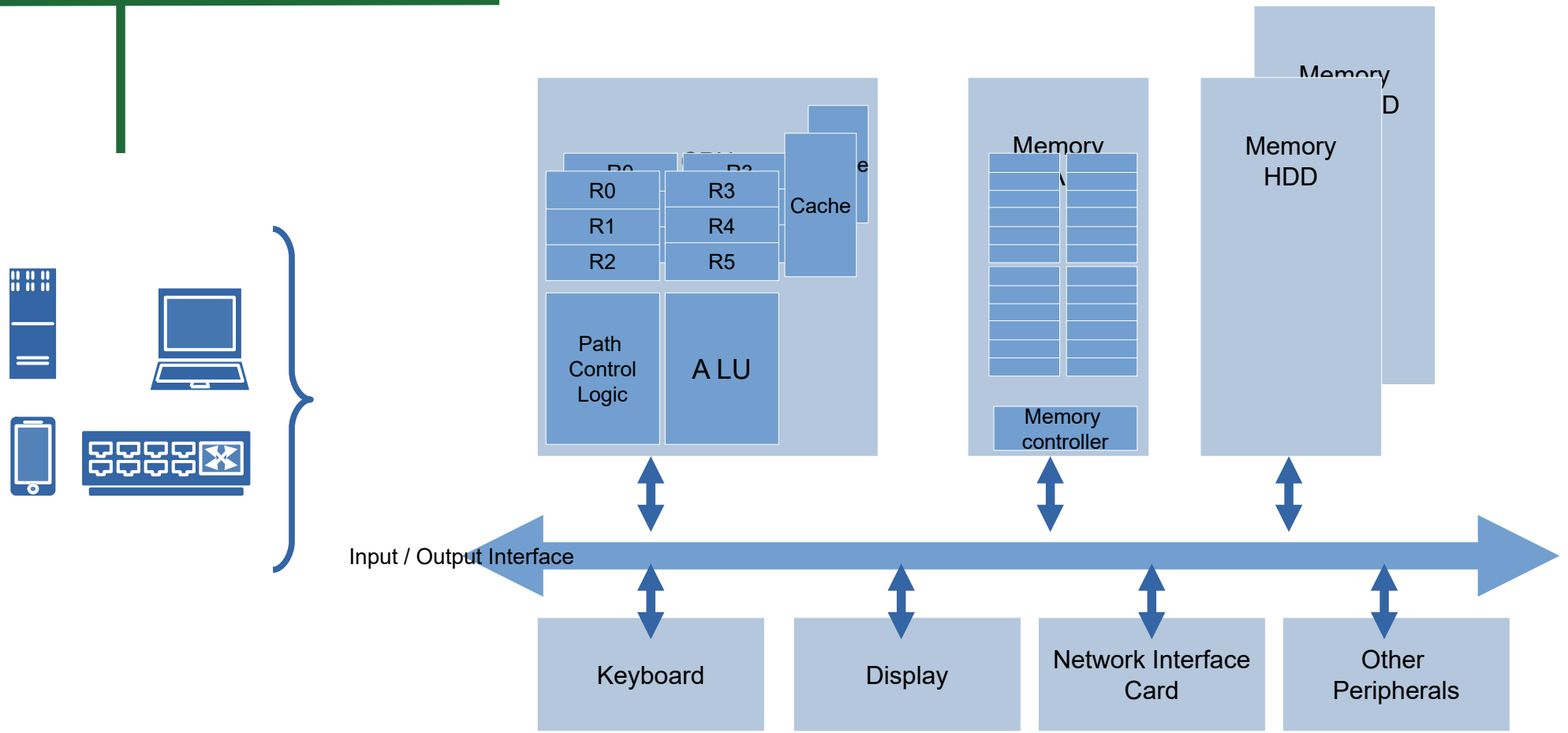
- processes information from data.
- executes actions, based on commands.
- e.g desktops, laptops, routers, smart phones.

Building blocks of a Computer!





Building blocks of a Computer

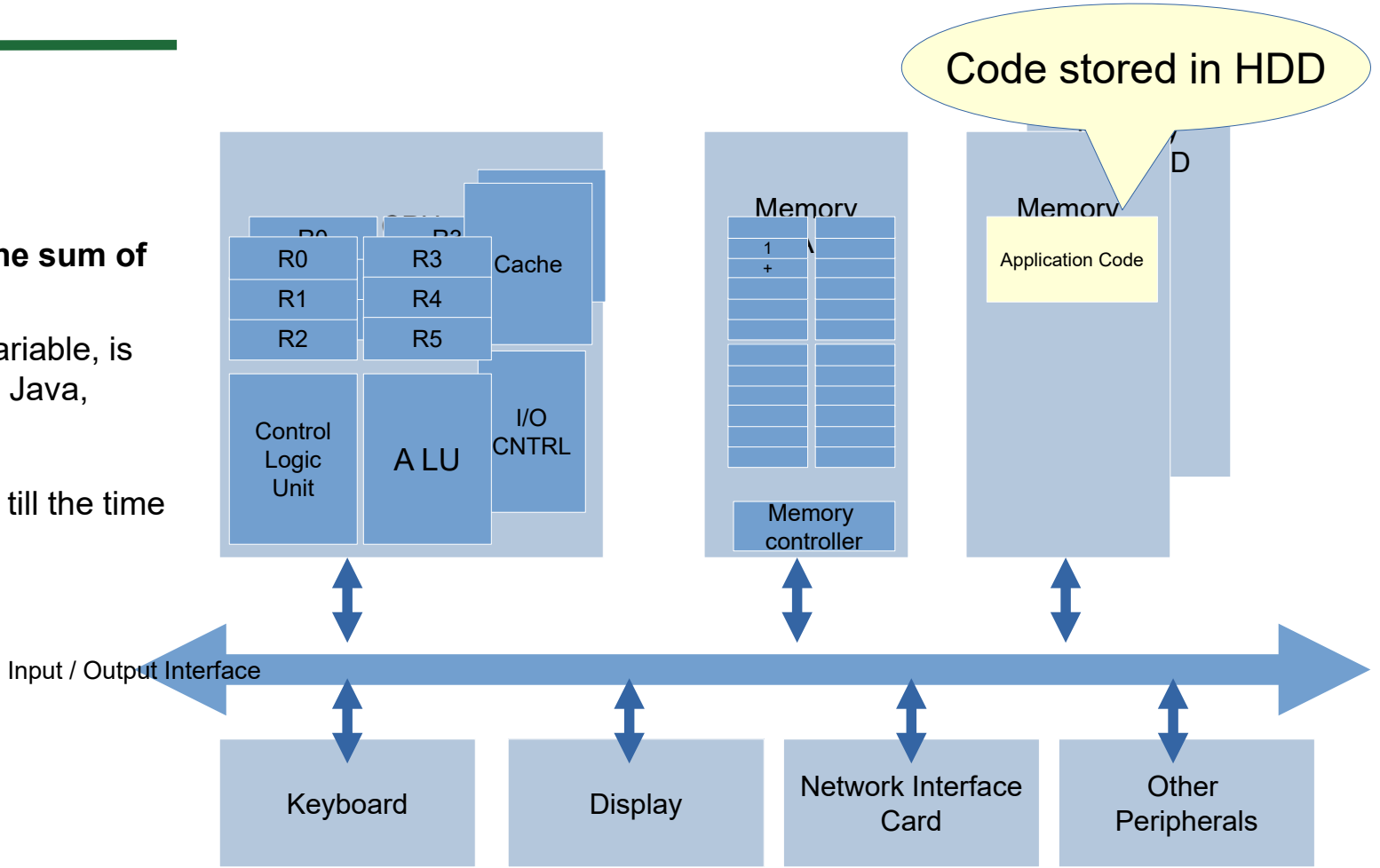


How does code execute?

E.g. ask computer to find the sum of two numbers say a + b.

A code which can sum two variable, is written in either Assembly, C, Java, Python etc.

The Code resides in HDD, till the time it is executed.



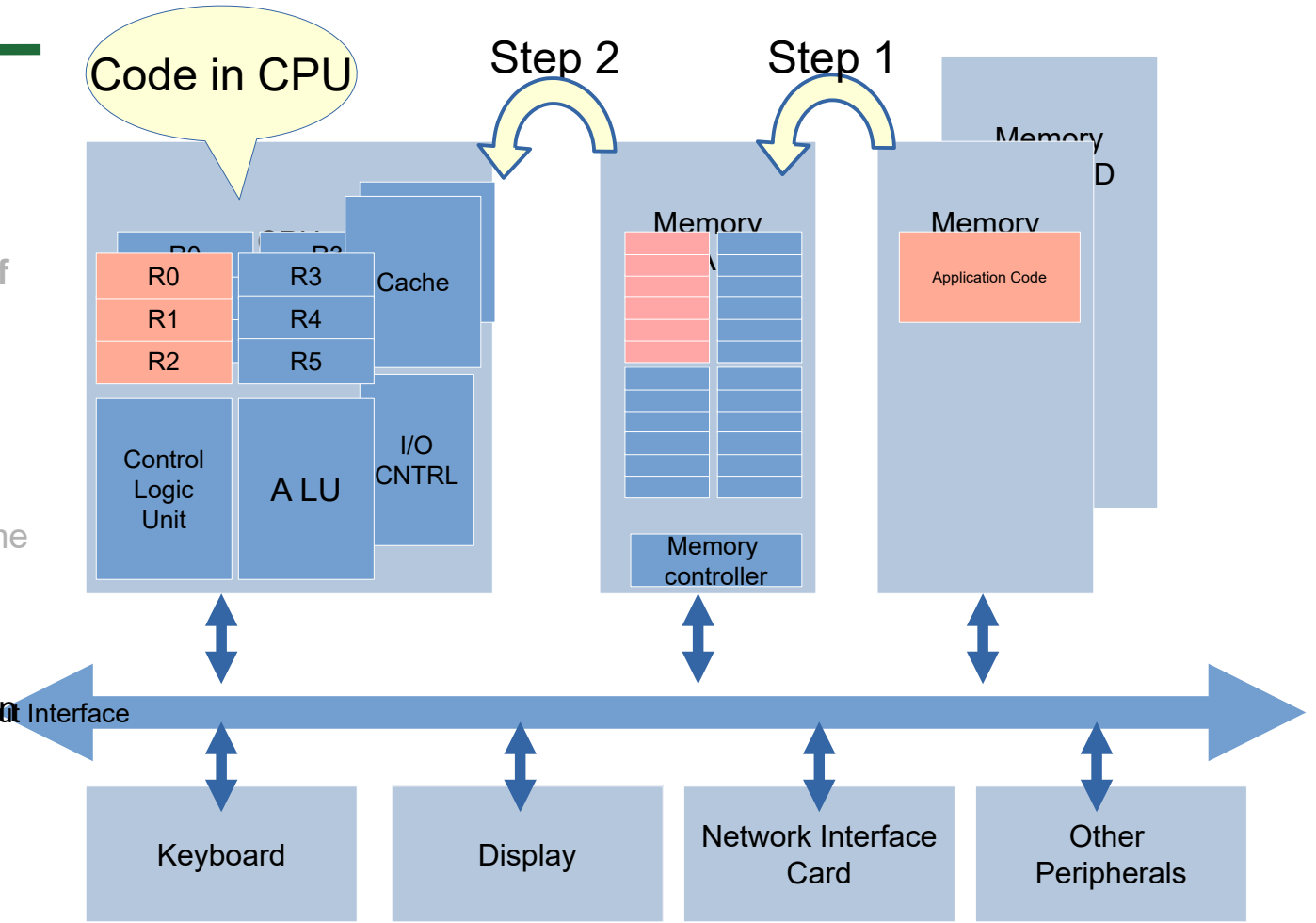
How does a code execute?

E.g. ask computer to find the sum of two numbers say a + b.

A code which can sum two variable, is written in either Assembly, C, Java, Python etc.

The Code resides in HDD, till the time it is executed.

STEP 1, 2: On execution, it is fetched into RAM, then to CPU cache, and then into internal registers.



How does a code execute?

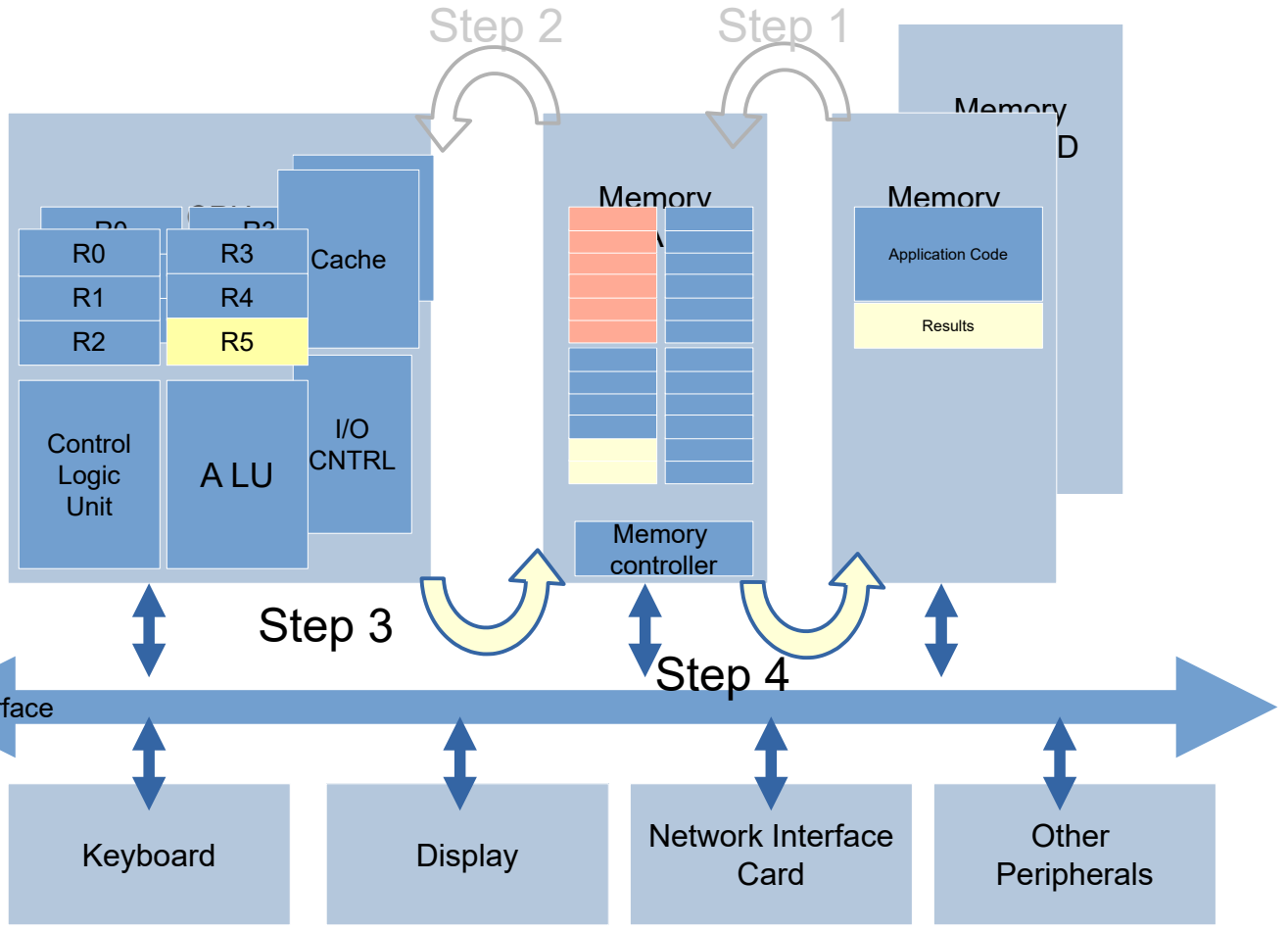
E.g. ask computer to find the sum of two numbers say a + b.

A code which can sum two variable, is written in either Assembly, C, Java, Python etc.

The Code resides in HDD, till the time it is executed.

On execution, it is fetched into RAM, then to CPU cache, and then into internal registers.

STEP 3: ALU then computes the sum, stores it in one of the registers.

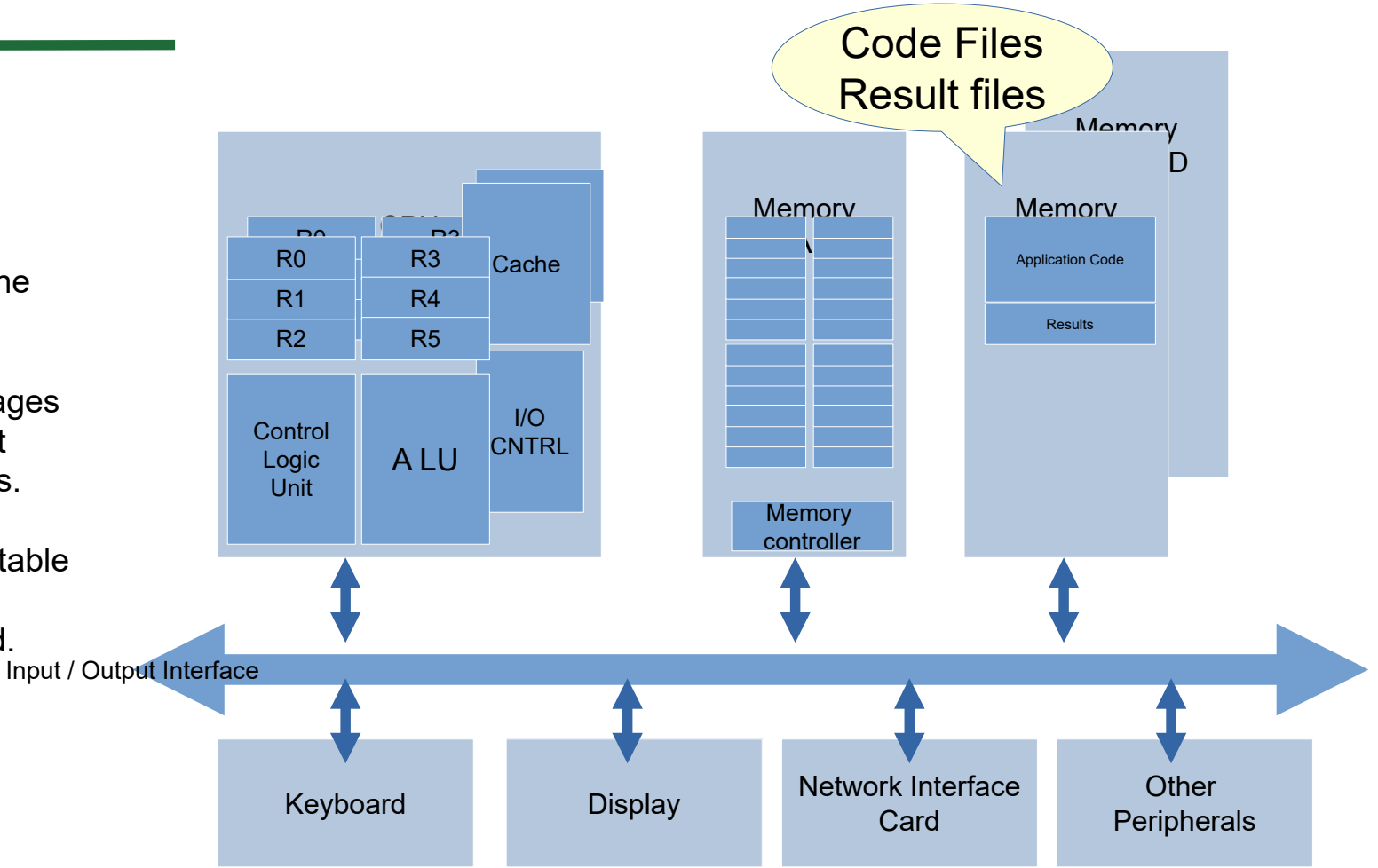


Files the containers of information!

The application code, and the results are stored as **files**.

The operating system manages the storage format of files. It also manages the resources.

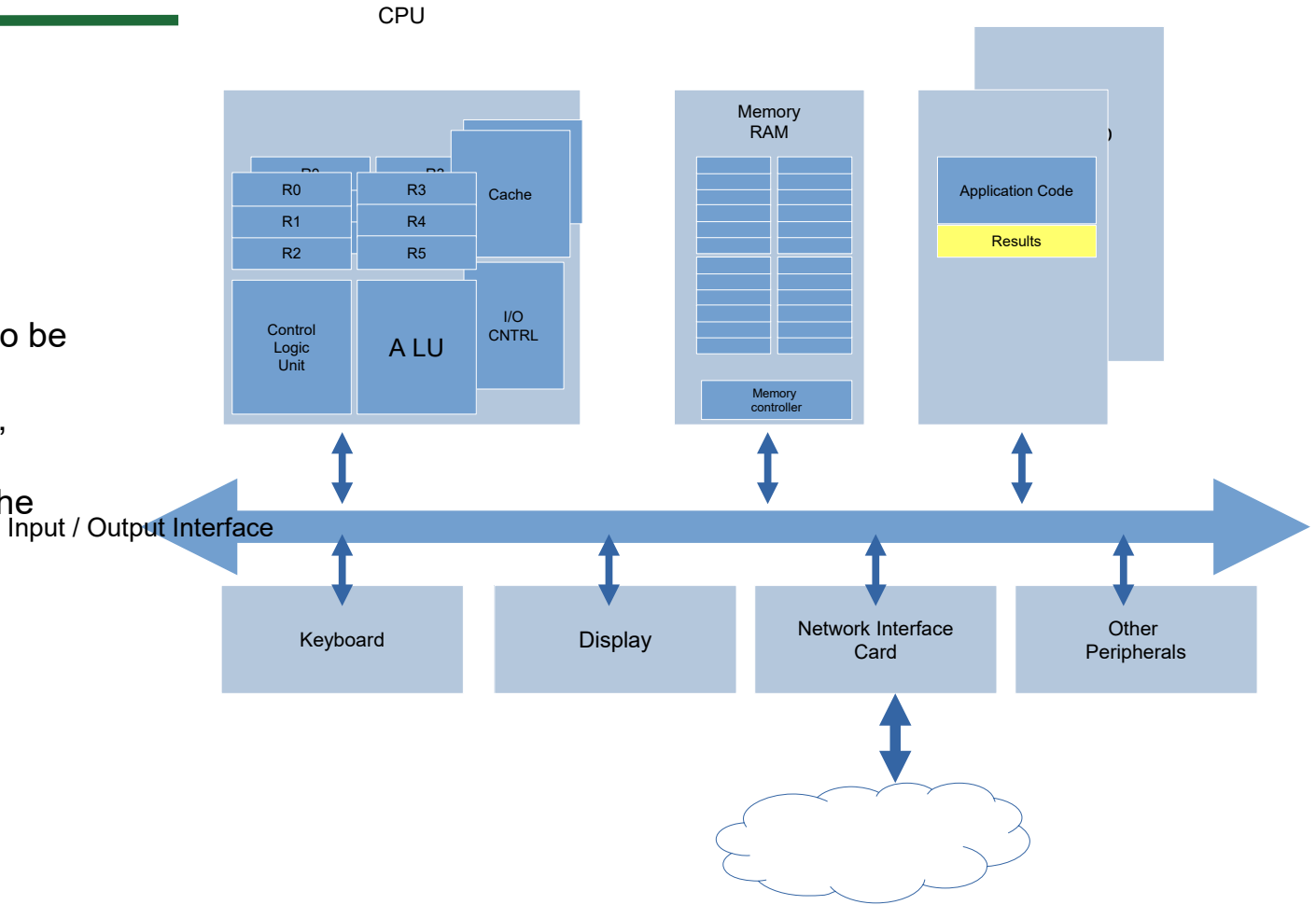
Unix philosophy is behind stable OS's e.g. Red Hat Linux, SystemV, MAC OS, Android.



Send Results over the NW!

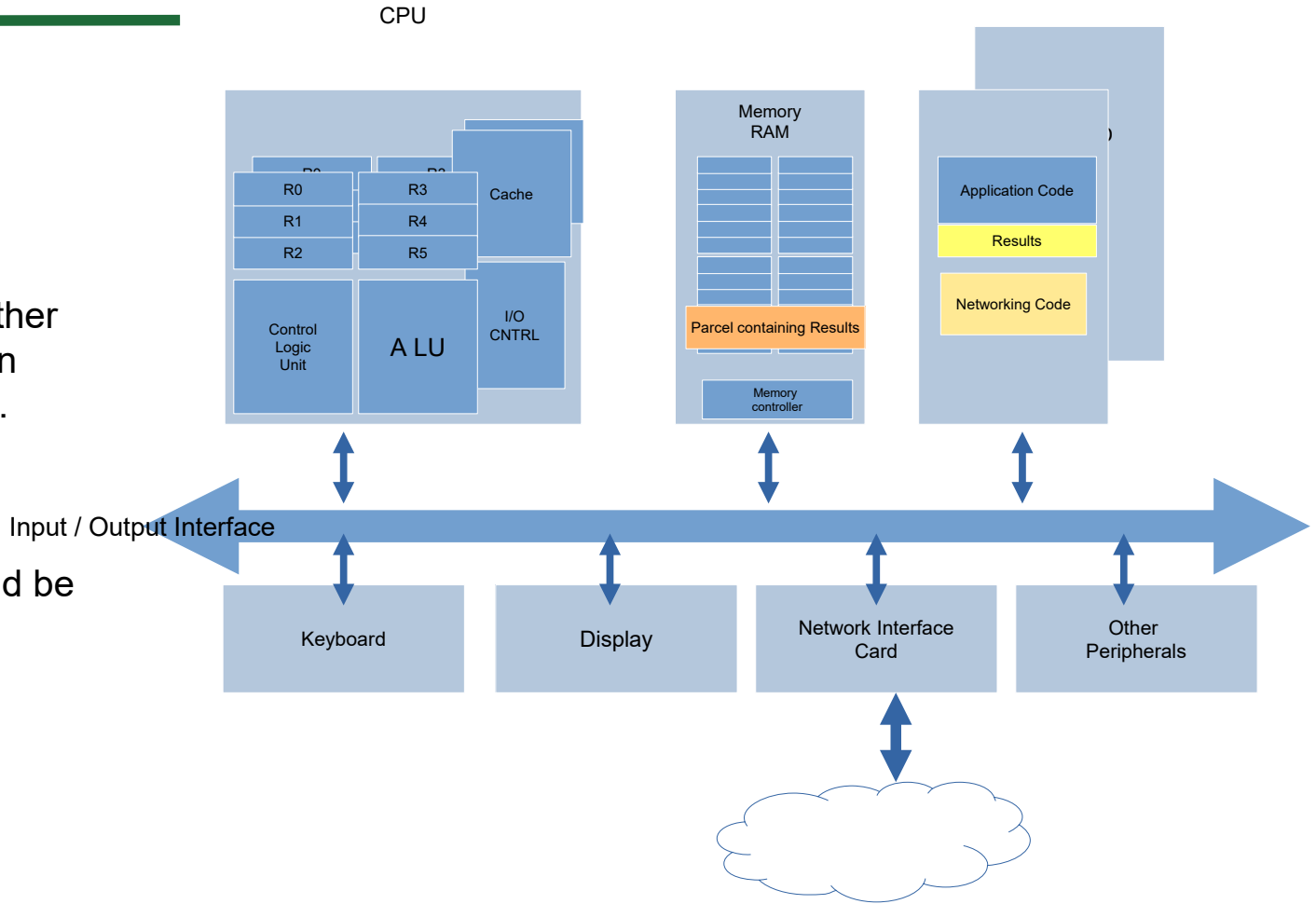
Scenario

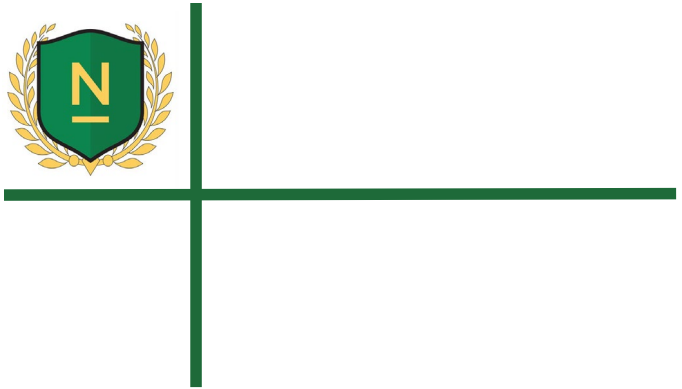
Assuming the results need to be sent to another computer connected over the network, then how the file named “Results” is transferred via the network.



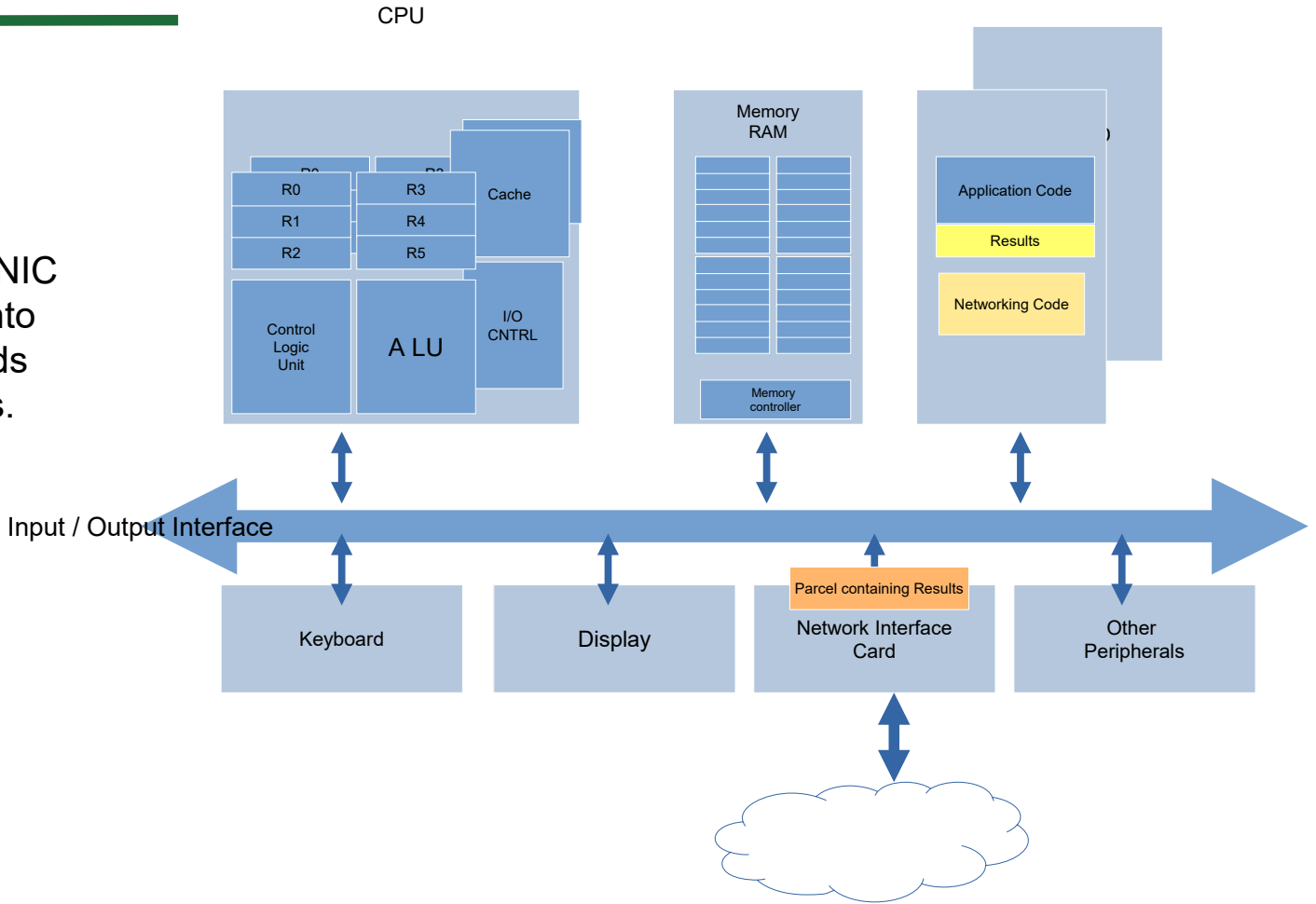
Another application will transform the file into another file, suitable for sending on the internet, called packet.

Note:
The parcel may be sent directly from RAM, or could be stored in HDD.





Parcel is sent to the NIC. NIC converts the digital data into electrical signals and sends them on a wire or wireless.



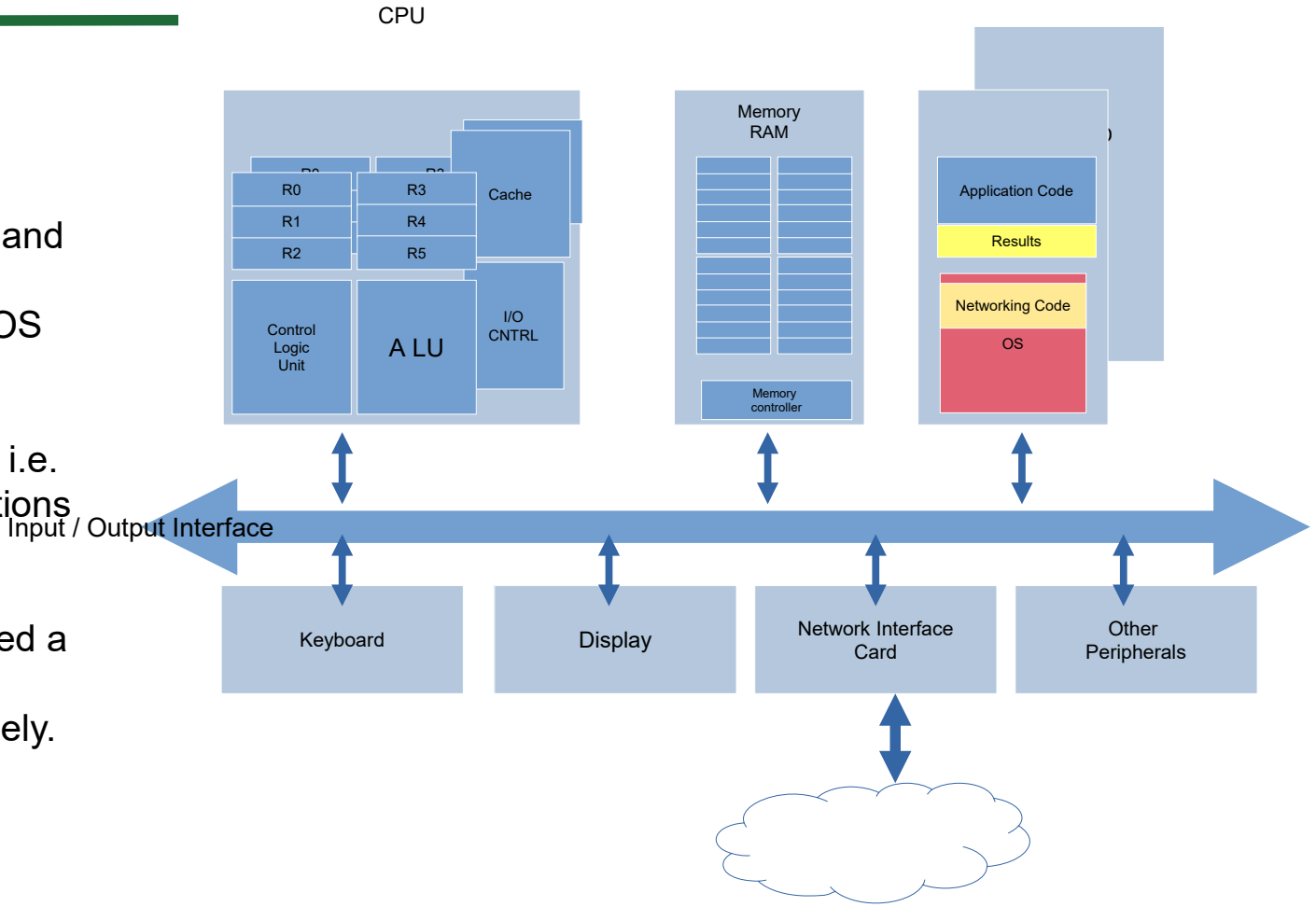
The role of Operating System!

In modern OS's like Unix, and Windows, the networking code is embedded in the OS kernel.

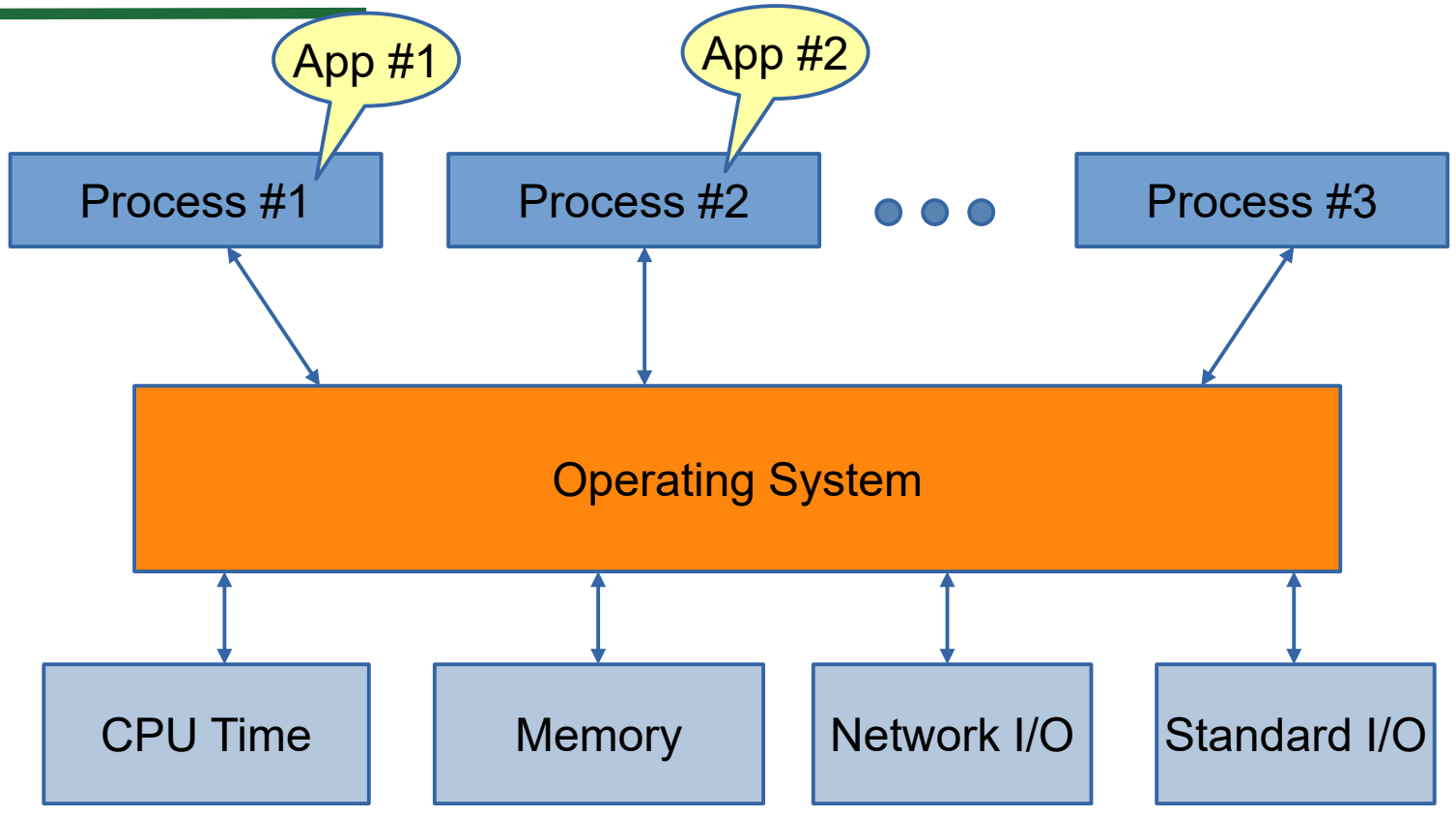
OS supports multi-tasking i.e. it can run multiple applications simultaneously.

An application is considered a process, therefore each process is identified uniquely.

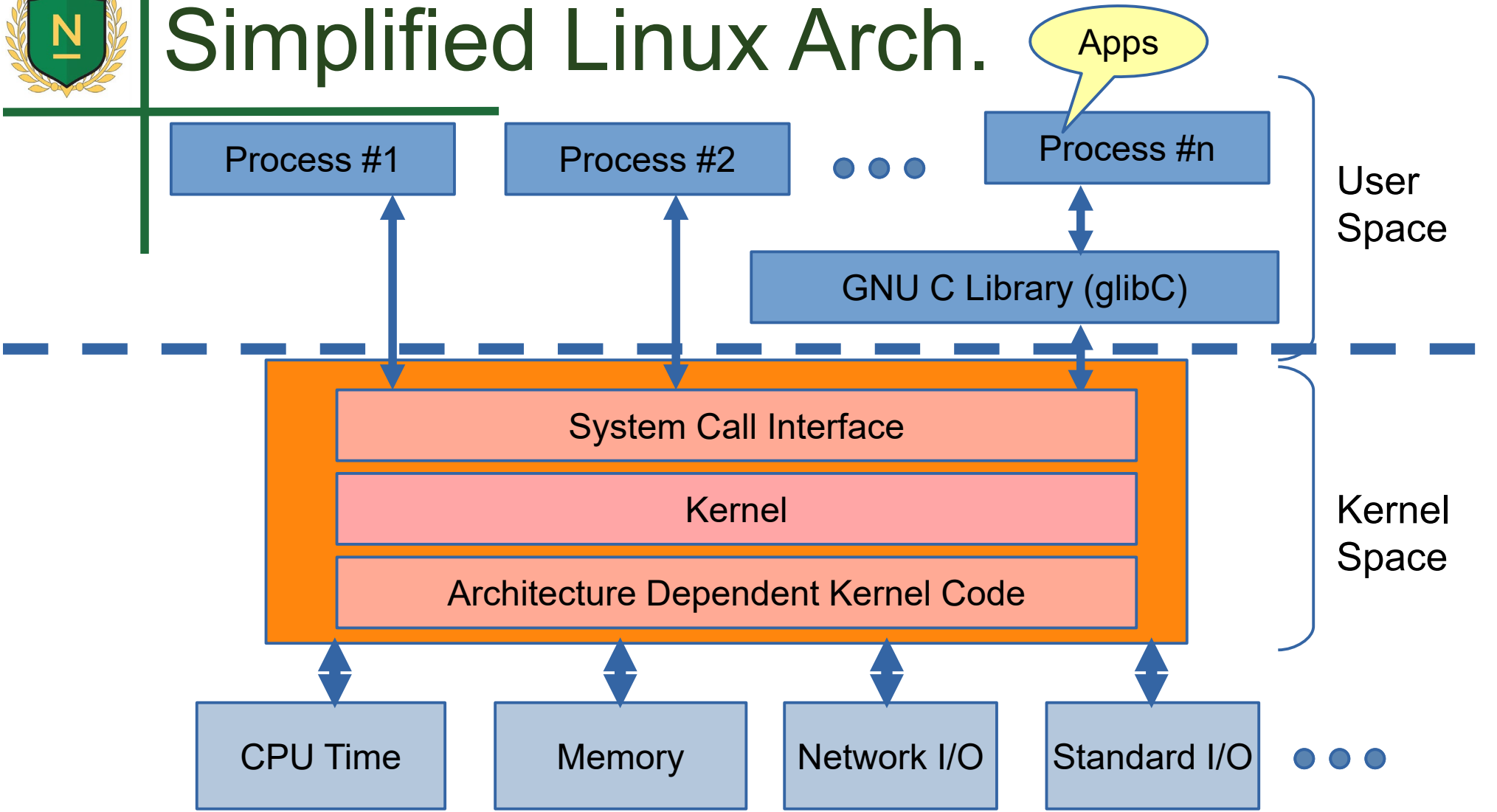
A popular API used for networking applications is Sockets



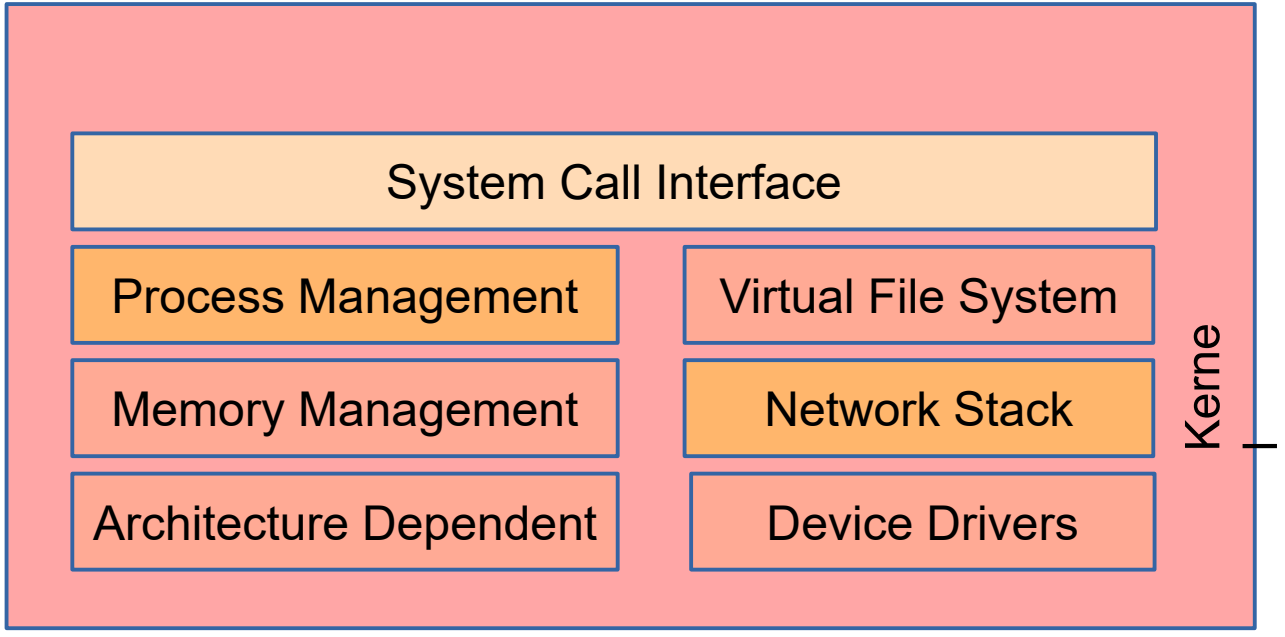
MultiTasking Operating System



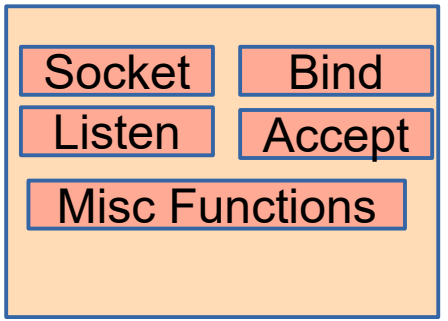
Simplified Linux Arch.



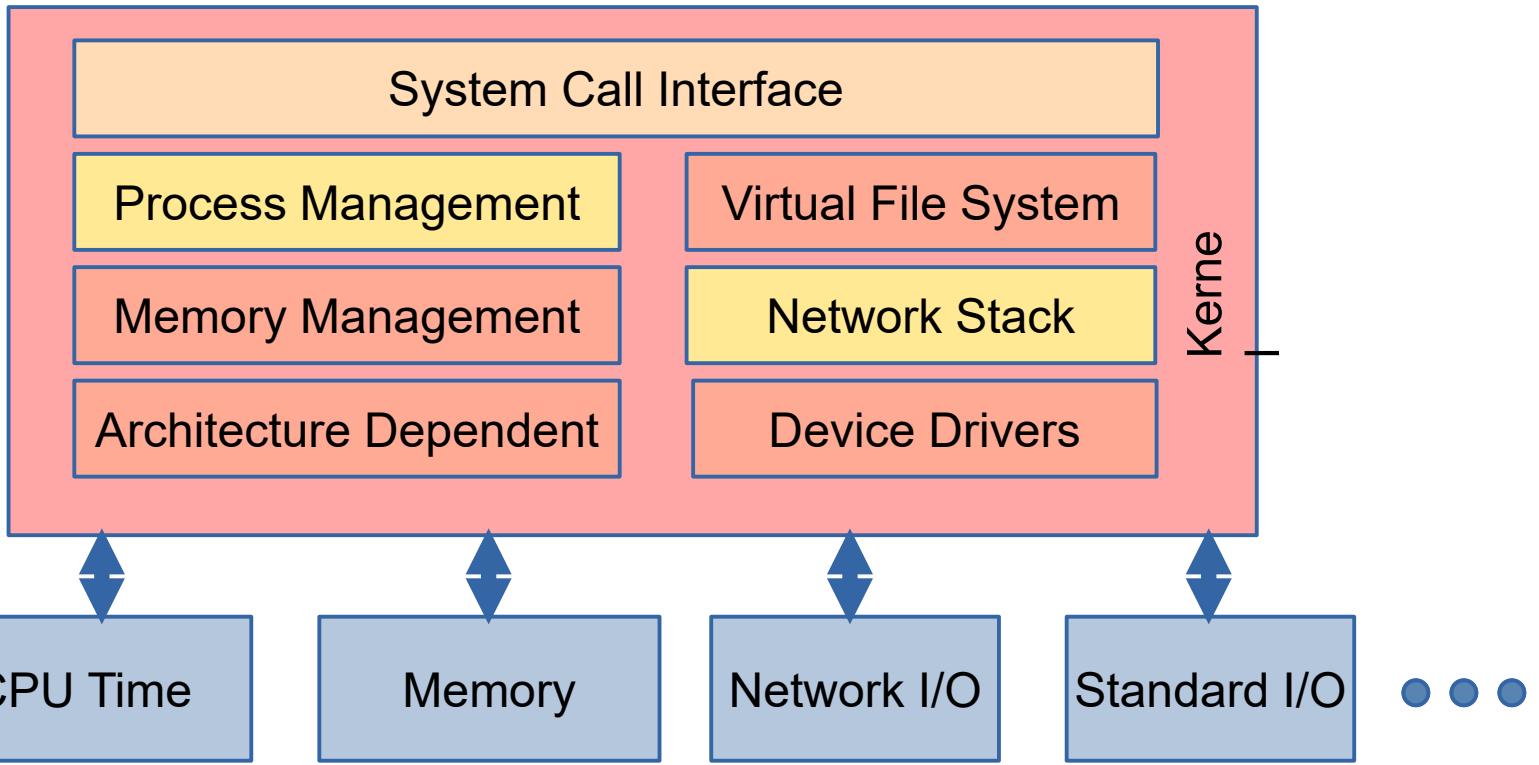
Simplified Kernel Sub-Systems

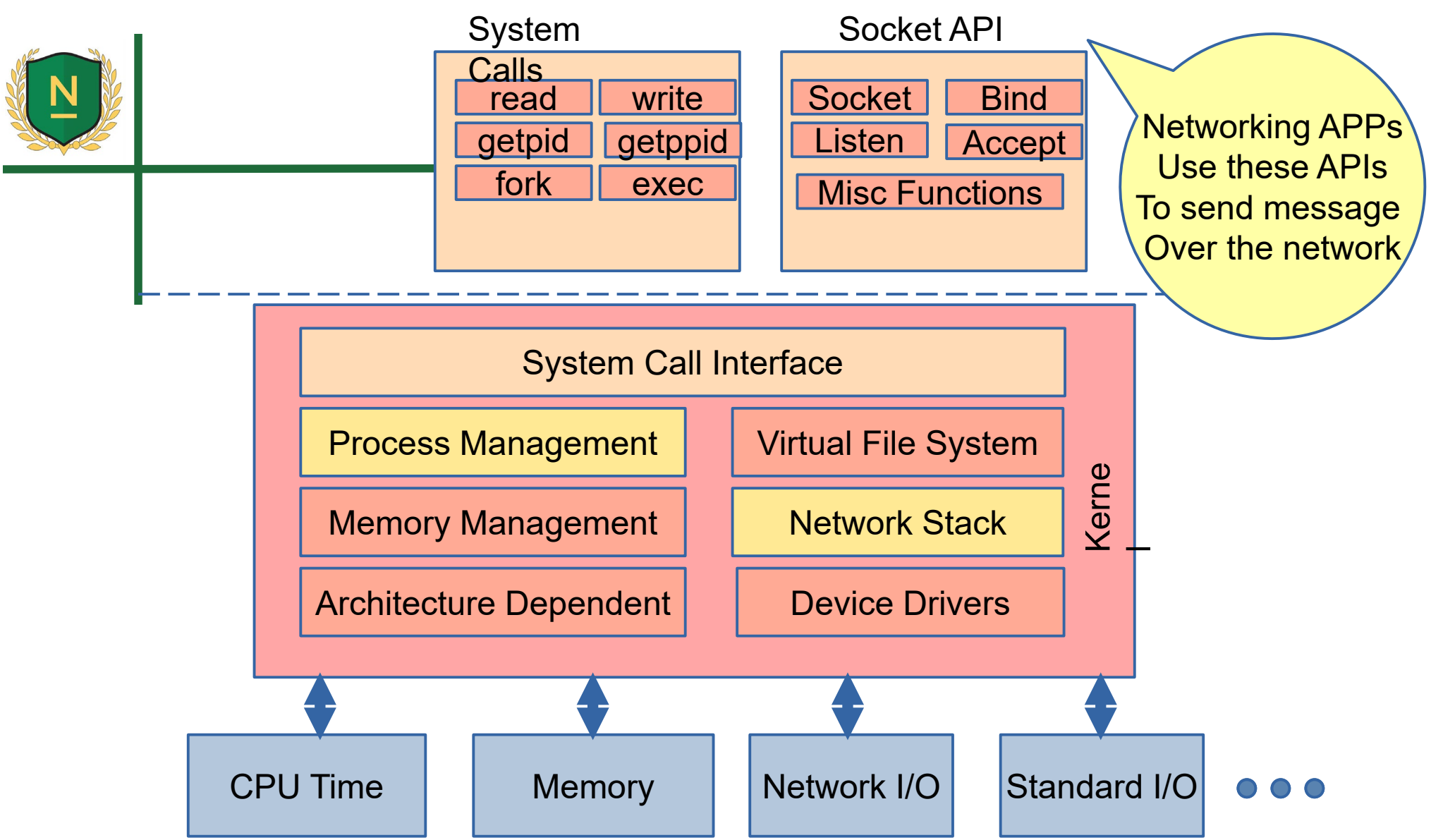


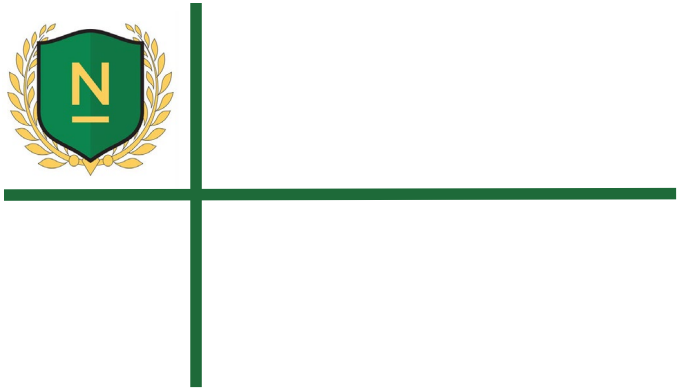
Socket API



Socket Application Programmer Interface




System Calls
read
write
getpid
getppid
fork
exec
Socket API
Socket
Bind
Listen
Accept
Misc Functions
Networking APPs
Use these APIs
To send message
Over the network
System Call Interface
Process Management
Virtual File System
Memory Management
Network Stack
Architecture Dependent
Device Drivers
Kerne
CPU Time
Memory
Network I/O
Standard I/O

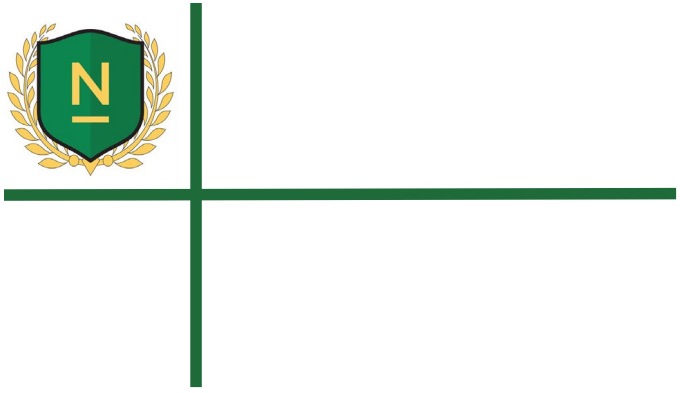


Lets delve in to coding.

```c
#include <stdio.h>   /* standard I/O library */
#include <stdlib.h>  /* exit */
#include <netdb.h>
#include <sys/socket.h> /* berkley socket library */
#include <sys/types.h> /*  */
#include <stdarg.h>
#include <syslog.h>
#include <errno.h>
#include <strings.h>  /*mem set */
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h> /*sockaddr*/
#include <time.h>
#include <string.h>

#define MAXLINE 1024
#define BACKLOG 128 /* 4096 for kernel 5.4 */
#define SA  struct sockaddr /* Used for type casting  */


int
main (int argc, char **argv)
{
```



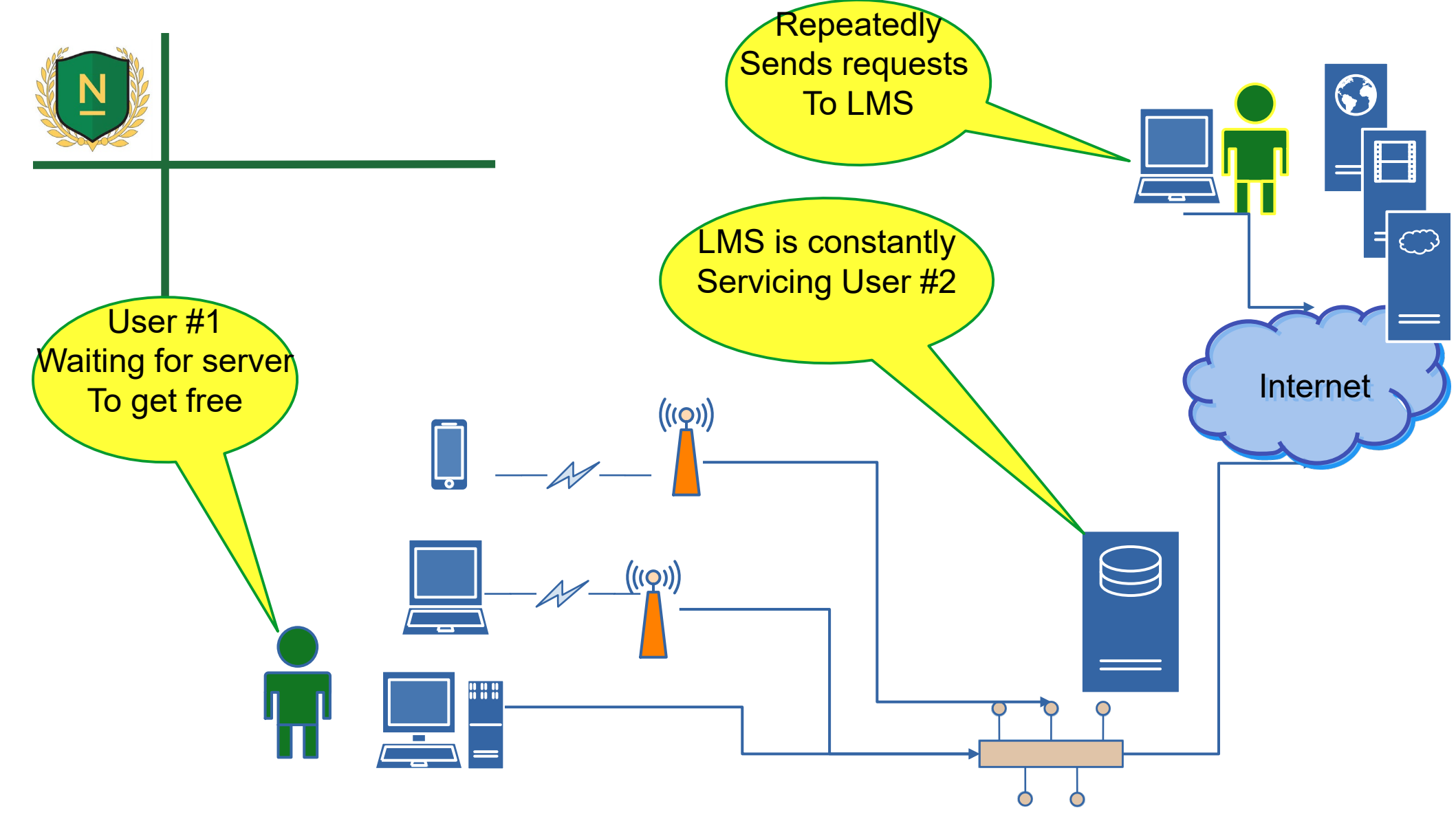
```c
{
  int listenfd, connfd, n_lfd, n_cfd;
  char buff[MAXLINE+1];
  struct sockaddr_in servaddr; /* 6 is mentioned*/
  /*struct sockaddr *SA //another way to define pointer to struct*/
  time_t ticks;

  /* Calling socket function */
  listenfd = socket(AF_INET, SOCK_STREAM, 0);
  if(listenfd < 0)
  {
          printf(" Socket error ");
          exit(-1);
  }

  /* Populating the Addresses */
  bzero(&servaddr, sizeof(servaddr));
  servaddr.sin_family = AF_INET;
  servaddr.sin_port = htons(5500); /* daytime server*/
  servaddr.sin_addr.s_addr = htonl(INADDR_ANY);

  /* Binding the socket to the port servaddr.sin_port*/
  n_lfd = bind(listenfd, (SA *) &servaddr, sizeof(servaddr));
  if(n_lfd ==-1)
    {
      printf("Error could not bind\n");
      exit(-1);
    }
  /* Listening to valid coonections*/
    n_cfd = listen (listenfd, BACKLOG);

  /* Accept requests again and again */
  for ( ; ; ) {
    connfd = accept(listenfd, (SA *) NULL, NULL);
    ticks = time(NULL);
    snprintf(buff, sizeof(buff), "%.24s\r\n", ctime(&ticks));
    write(connfd, buff, strlen(buff));
    close(connfd);
    sleep(1);
```


Repeatedly
Sends requests
To LMS
LMS is constantly
Servicing User #2
User #1
Waiting for server
To get free
Internet

Thank you !

- Questions, queries etc.

AGENDA

1. Opening Remarks – Dr. Arshad Ali – 10 Min
2. Security Management at University – Dr. Saad A. Malik, Namal – 15 Min
3. Setting up a SOC in University – Mr. Rizwan Ali, SPS – 25 Min
4. General Discussion – 10 Min

GENERAL DISCUSSION

(SPINNLABS@SPSNET.COM)

THANK YOU!

spinnlabs@spsnet.com

BACK UP SLIDES