

---

# Layered Style

---

Jing Dong

CS6362 Software Architecture and Design

1

---

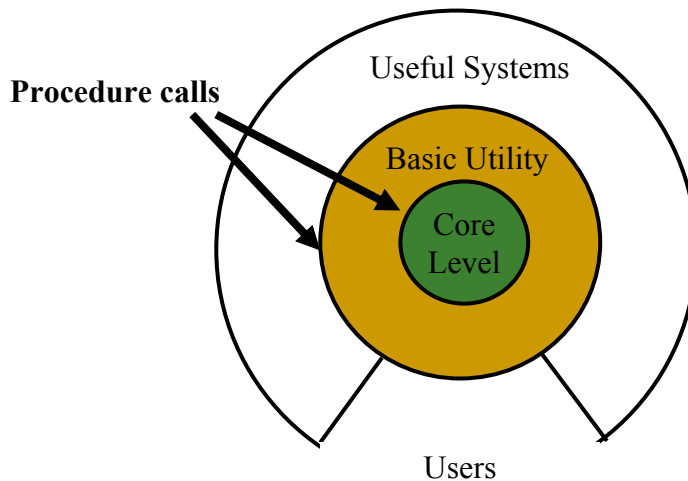
# Layered Style

- Suitable for applications that involve distinct classes of services that can be organized hierarchically.
- Each layer
  - provides service to the layer above it
  - serves as a client to the layer below it
- Only carefully selected procedures from the inner layers are made available (exported) to their adjacent outer layer.
- Layers may be hidden to all except adjacent layers or layers may be partially hidden.
- Components: are typically collections of procedures.
- Connectors: are typically procedure calls under restricted visibility.
- Applicability: a large system that is characterized by a mix of high and low level issues, where high level issues depend on lower level ones.
- Invariants: limit layer (component) interactions to adjacent layers (in practice this may be relaxed for efficiency reasons)

---

2

## Layered Systems



3


## Layered Style Examples

- Layered Communication Protocols:
  - Each layer provides a substrate for communication at some level of abstraction.
  - Lower levels define lower levels of interaction, the lowest level being hardware connections (physical layer).
- Operating Systems
  - Unix
- Version Management

4

# OSI Reference Model

**Application** email (SMTP, MIME), ftp, telnet, gopher, usenet (NNTP) , rlogin, mosaic, netscape, ...

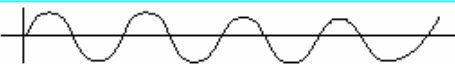
**Presentation**  architecture  
Data Rep. (ASCII, EBCDIC), Compression, Encryption

**Session** graceful data exchange, dialog mgmt, (re)synchronization

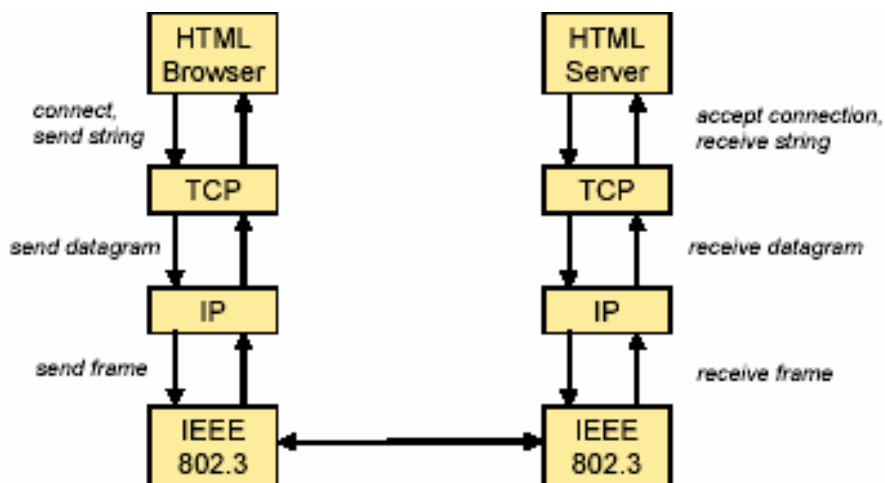
**Transport** Split messages  
Quality Of Service (QOS)  
multiplexing

**Network**  fast transmission  
routing  
accounting

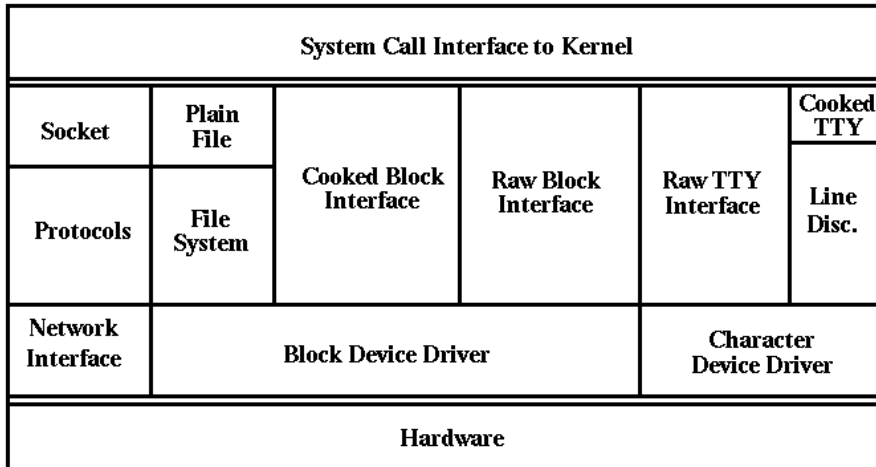
**Data Link** error-free transmission

**Physical** 

# Web Browsing



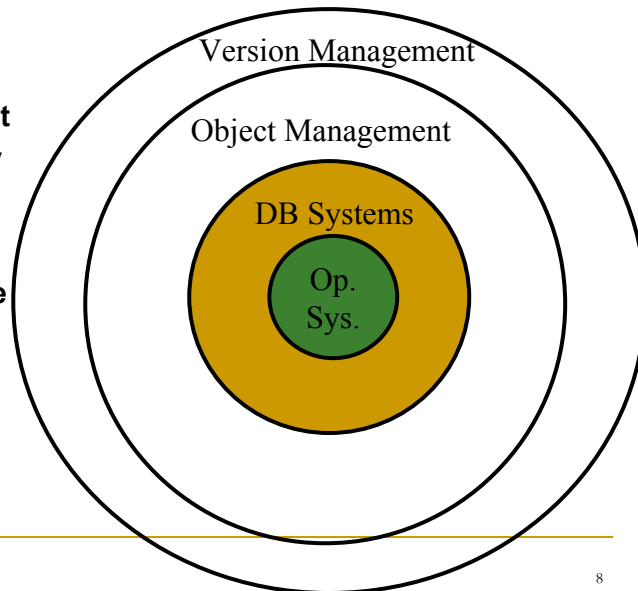
## Unix Layered Architecture



7

## Version Management

- **Advantages**
  - incremental development
  - modifiability
  - portability
- **Drawbacks**
  - performance
  - structuring can be difficult



8

## Properties of the Layers Architecture

### ■ Performance

- If we are developing layer  $k$ , we are limited to the services of layer  $k-1$
- Possible that lower layers have services that might be important to performance
- Therefore trade off advantages in separation of concerns, modifiability versus possible performance issues

9

## Properties of the Layers Architecture

### ■ Modifiability

- If layer  $k$  changed internally
  - No impact on other layers
- If services provided by layer  $k$  change
  - Impact limited to layer  $k+1$
  - I.e., only layer using layer  $k$  may have to change
- If you are considering implementing a system based on layers, need to ask
  - What are likely modifications?
  - Can these modifications be confined to the layers?
  - Do they require changes to multiple layers?

10

---

## Layered Style Advantages

- Design: based on increasing levels of abstraction as we move up through layers – partitions complex problems.
  - Enhancement: since changes to the function of one layer affects at most two other layers.
  - Reuse: since different implementations (with identical interfaces) of the same layer can be used interchangeably.
  - Standardization based on layers e.g. OSI
- 

11

---

## Layered Style Disadvantages

- Not all systems are easily structured in a layered fashion.
  - Performance requirements may force the coupling of high-level functions to their lower-level implementations.
- 

12