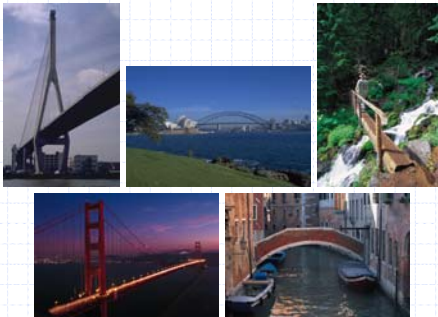


Architectural Styles

John Reekie
University of Technology, Sydney

Terms of Use: Creative Commons Attribution-ShareAlike 2.0
<http://creativecommons.org/licenses/by-sa/2.0/>

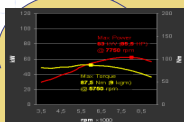
Design is recurring patterns...



Images from <http://office.microsoft.com/clipart/>

"Design" needs a vocabulary

"Longitudinally-mounted 90-degree V-twin"



"Torque"



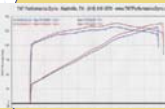
"Perfect primary balance" "Growl"



Images from <http://www.ducati.com>

Vocabulary (2)

"Transverse inline four"



"Power peak"



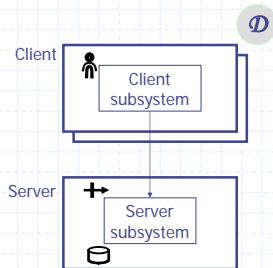
"Screamer"



Images from <http://www.tntperformanceofdyno.com>
<http://dsc-racer.net> and <http://www.motorcycle.com>

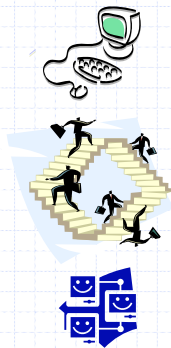
Client-server

- ◆ Notionally, simply denotes a many-client / single-server deployment
- ◆ Has many variants (discussed later)

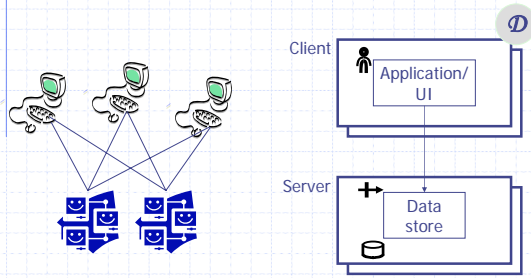


N-tier architectures

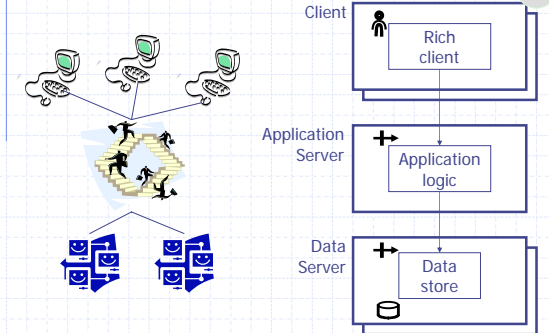
- ◆ Arose from "client-server" business information systems
- ◆ These days, an architectural style that can be applied to systems that contains clients, processing, and persistent data storage



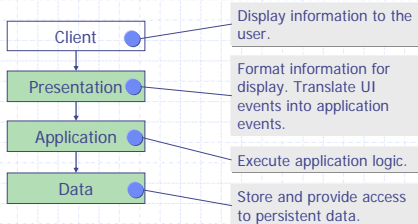
"Traditional" 2-tier



"Traditional" 3-tier

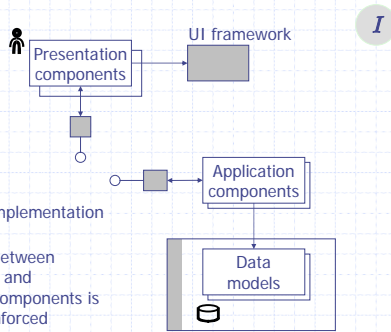


The "logical 3-tier" architecture



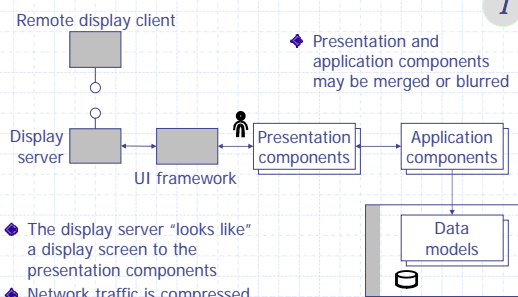
- ◆ The conceptual distinction between the tiers is important
- ◆ Really only an "N-tier" architecture if carried through into the implementation architecture

Rich-client 3-tier



- ◆ This is the implementation architecture
- ◆ Separation between presentation and application components is physically enforced

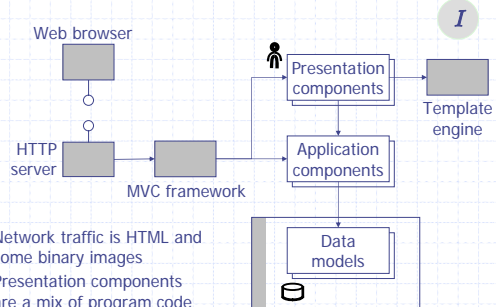
Thin-client 3-tier



- ◆ Presentation and application components may be merged or blurred

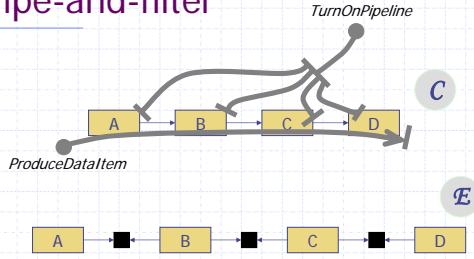
- ◆ The display server "looks like" a display screen to the presentation components
- ◆ Network traffic is compressed incremental bitmaps

Web-client 3-tier



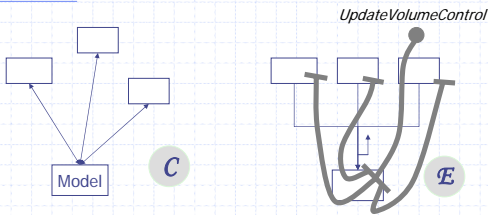
- ◆ Network traffic is HTML and some binary images
- ◆ Presentation components are a mix of program code and HTML template files

Pipe-and-filter



- ◆ Data processed by “filters”
- ◆ All data flows through “pipes”
- ◆ Many variations in the realtime systems module

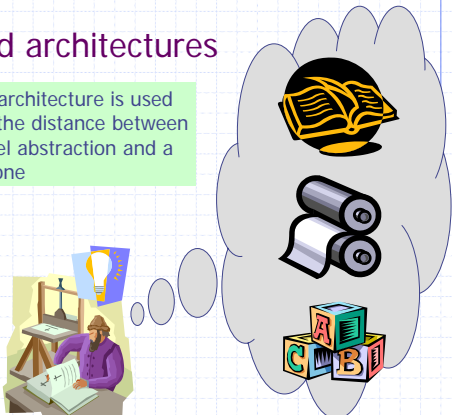
Notification architectures



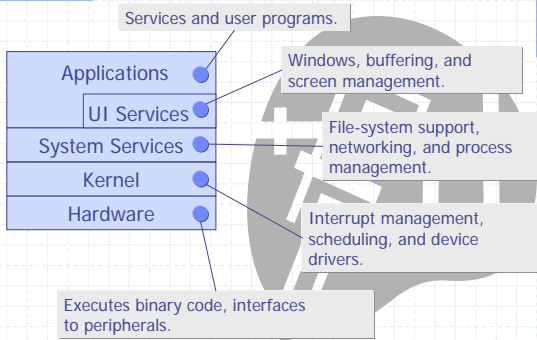
- ◆ Interested components “subscribe” to events, and are updated when they occur
- ◆ Can be used both within a process, and at the process level (as shown here)
- ◆ Example: change volume control in Windows

Layered architectures

A layered architecture is used to “span” the distance between a high-level abstraction and a low-level one



Operating systems, for example



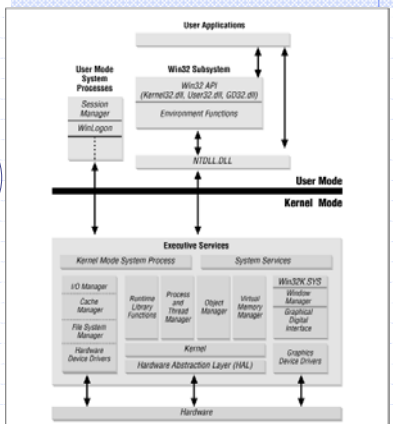
Example

What are the responsibilities, interfaces, and data models in this architecture?

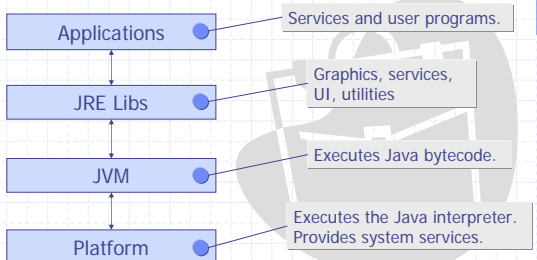


From Chapter 9 of *Win32 API Programming with Visual Basic*, by Steven Roman

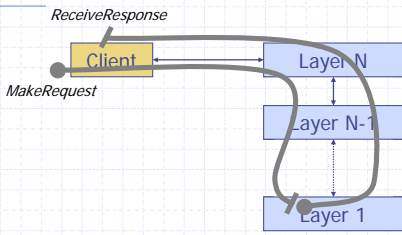
<http://www.microsoft.com/technet/prodtechnol/ntwrkstn/evaluate/featfunc/winarch.mspx>



Virtual machines, for example

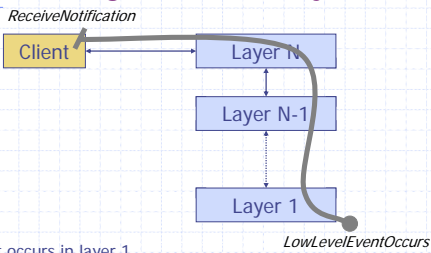


Client issues a request



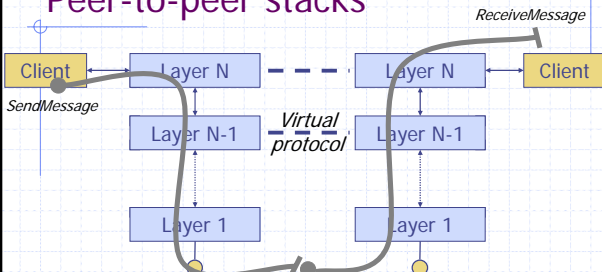
- ◆ A client generates a request to layer N
- ◆ The request travels down to layer 1 (typically), and a response returns
- ◆ Class exercise: what happens in the JVM when a client issues a request to draw a window on the screen?

An event originates in Layer 1



- ◆ An event occurs in layer 1
- ◆ Notification of the event travels up to layer N (typically) and then to the client
- ◆ Class exercise: what happens in the JVM when a user clicks the mouse button?

Peer-to-peer stacks



- ◆ Clients communicate via complementary "stacks"
- ◆ All layers except Layer 1 communicate using a virtual protocol
- ◆ Class exercise: explain how a packet gets from one client to the other.
