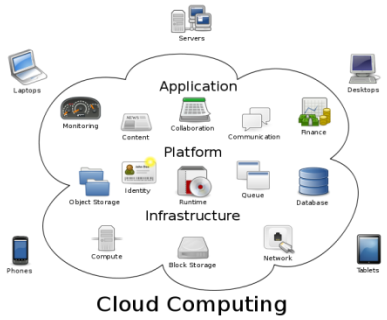
Lecture 5 Architectural Type and Pattern

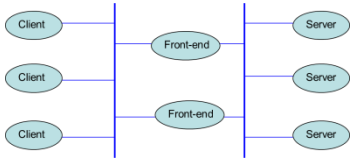
- SA Types
- SA Patterns

主要架构类型

Software Model Layers



Application System — System Pattern



Software Architecture — Architecture Pattern

Object/Class — Component Pattern (design pattern)

123-9475775 84775.99 …
01010101 0xFFFFEBAAA …

Data Structure — Data Model

主要架构类型

Various Patterns for SA Layer Abstraction

System Pattern: presents the macroscopic system pattern for software applications, such as batch processing system, Web service architecture, SOA, cloud computing architecture, mobile application system, etc.

Architecture Pattern: the architectural patterns used to design and build the software applications/systems, such as filter-pipe, C/S, B/S, MVC, broker, three-tier arch, etc.

Component Pattern (design pattern): the design reuse of class/module that carries certain type of structure and behavior

Data Model: the abstraction of data records through data structure design that can be used for further processing and manipulation

主要架构类型

Types of Software Architecture

- Data Flow Architecture
  - batch processing
  - filter/pipe
- Data Centered Architecture
  - repository design
  - blackboard architecture
- Hierarchy Architecture
  - layered architecture
  - main/subroutine
- Interaction Oriented Architecture
  - MVC (Model-View-Control)
  - PAC (Presentation-Abstraction-Control)

主要架构类型

Software Architecture Types (cont'd)

- Object Oriented Architecture
  - OOAD
  - UML model
- Component- Based Architecture
  - framework approach
- Distributed Architecture
  - client/server,
  - peer-to-peer
  - multi-tier
- Service-Oriented Architecture
  - integration with business process

主要架构模式

Architectural Patterns

- Client-Server (C/S)
- Browser-Server (B/S)
- MVC (Model-View-Control)
- PAC (Presentation-Abstraction-Control)
- Pipe and Filter
- Remote Procedure Call (RPC)
- Layered Architecture
- Distributed System
- Object-Oriented Architecture
- Service-Oriented Architecture
- Multi-tier Architecture

主要架构模式

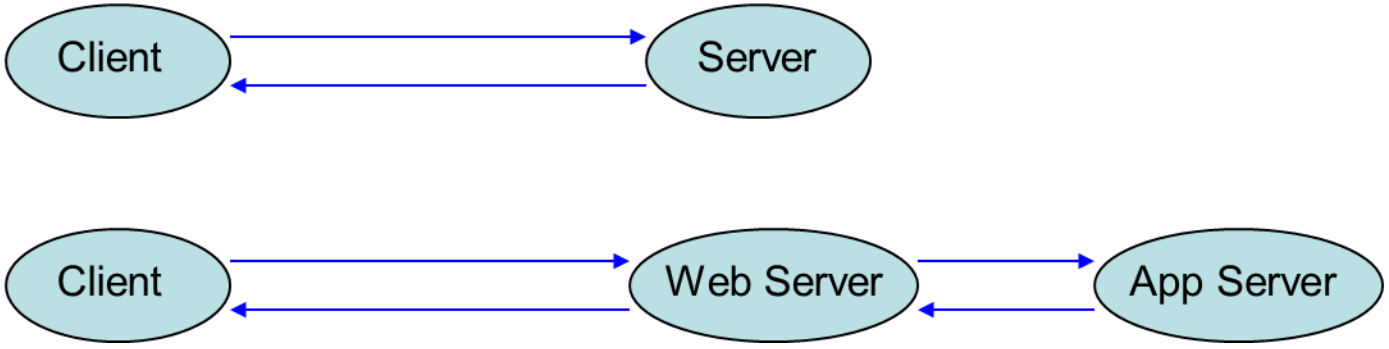
Client-Server 客户端-服务器

- scalability issue at server

Three-tier Architecture 三重架构

- split functions

  server -> web server + application server



Pros of C/S Architecture

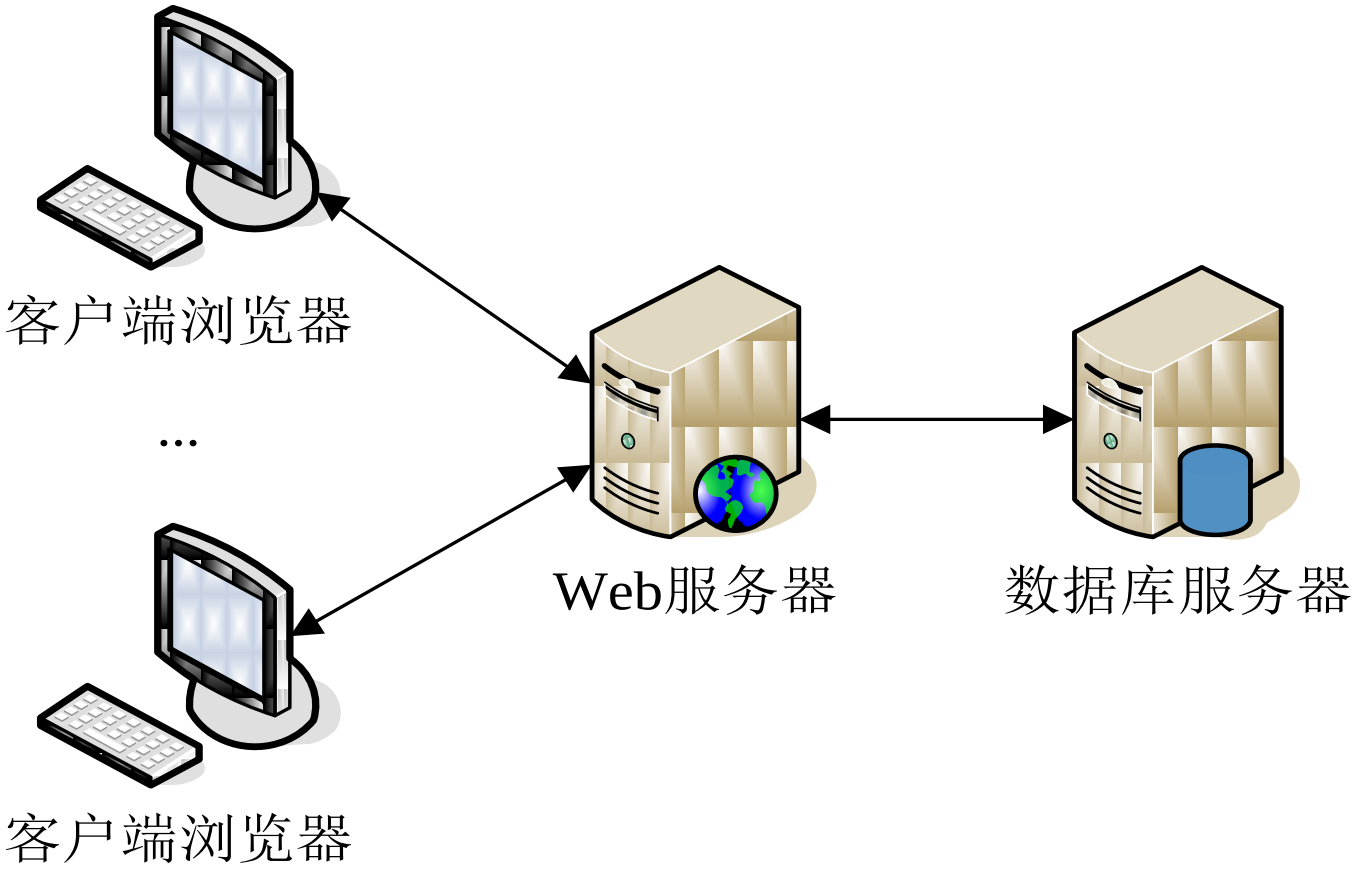
- Client program and Server application are running on separate machines, which is good to data distribution
- Architecture transparency – Client doesn't need to know where Server is located
- Client program and Server application may run on distinct platforms, with the focus on various tasks
- Client program is independent to and geographically separate from Server application, which makes the deployment of system is considerably flexible

Cons of C/S Architecture

- Since a certain amount of functions is built on the Client side, the Client program tends to grow and become large, makes it hard to main and upgrade
- The Clients are connected to the centralized Server through LAN network, which limits the system scalability due to the network bandwidth
- The standard, platform and technology used for the Client program are heterogeneous, which adds to the development and maintenance cost

主要架构类型

Browser-Server (B/S) Architecture



Pros of B/S Architecture

- The Client is a simple browser (kind of thin client) program, which is easy to use and maintain
- Based on the standard HTTP protocol， it is easy to build a cross-platform s/w system. The Server is totally separated from the Client
- Development focus is on the Server application, i.e. at the backend, which greatly reduces the development and maintenance cost
- Since Clients are connected to the Server through the Internet (WAN), its scalability is greatly improved

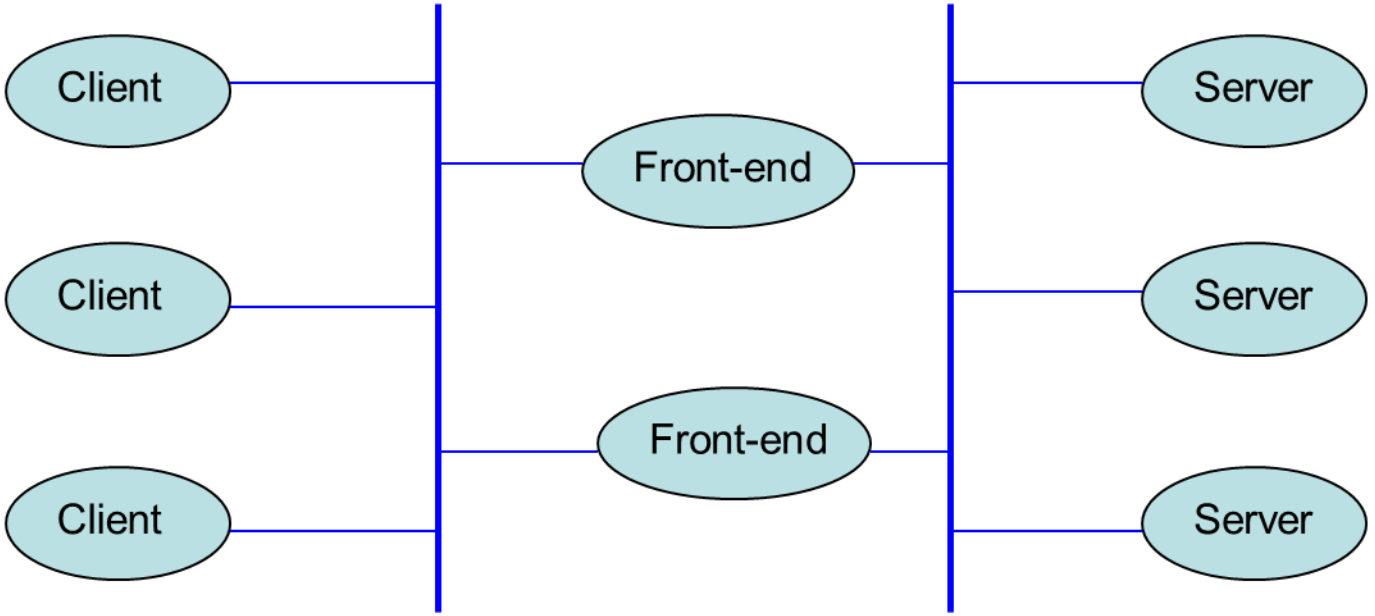
主要架构类型

Cons of B/S Architecture

- The majority of work is placed on the Server end, which makes the web server a single point of failure (SPOF)
- The interaction and data exchange between Client and Server is completed via Internet, with a limited transmission speed, so it's not quite appropriate for the Online Transaction Processing (OLTP) system
- Since the system is exposed to the Internet world, its security is an issue

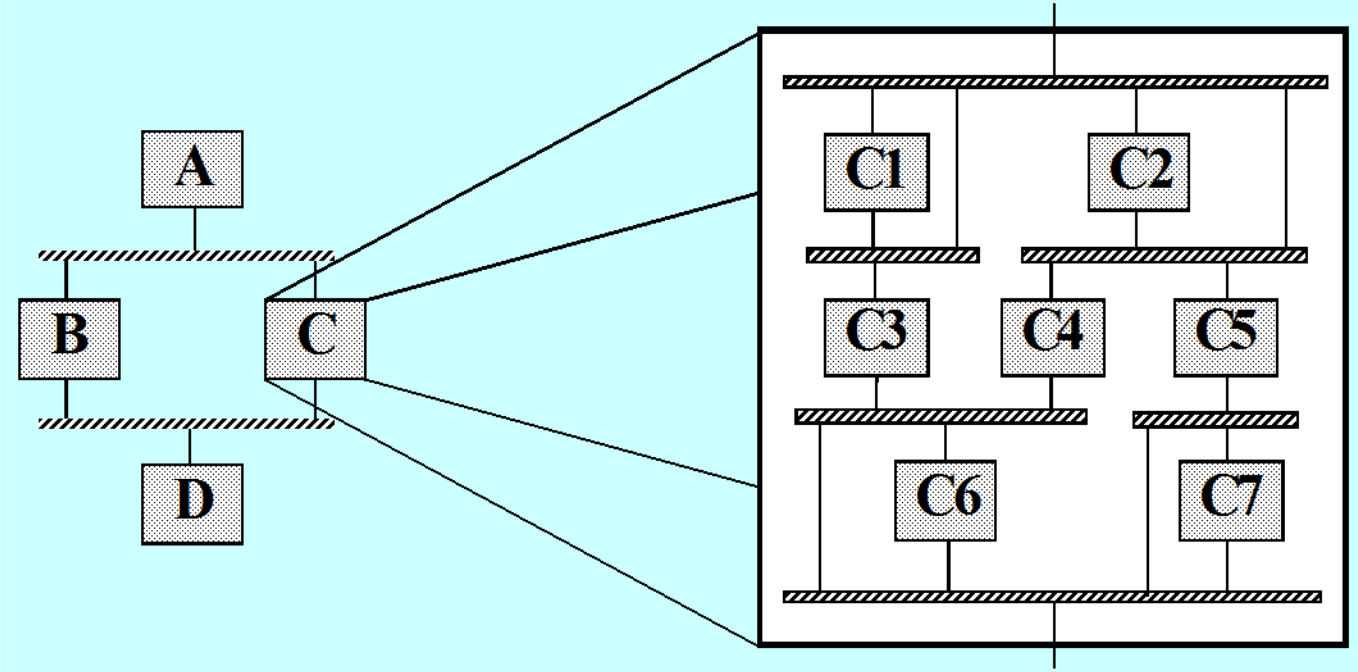
An Improvement to B/S Architecture

Multi-tier Architecture 多重架构



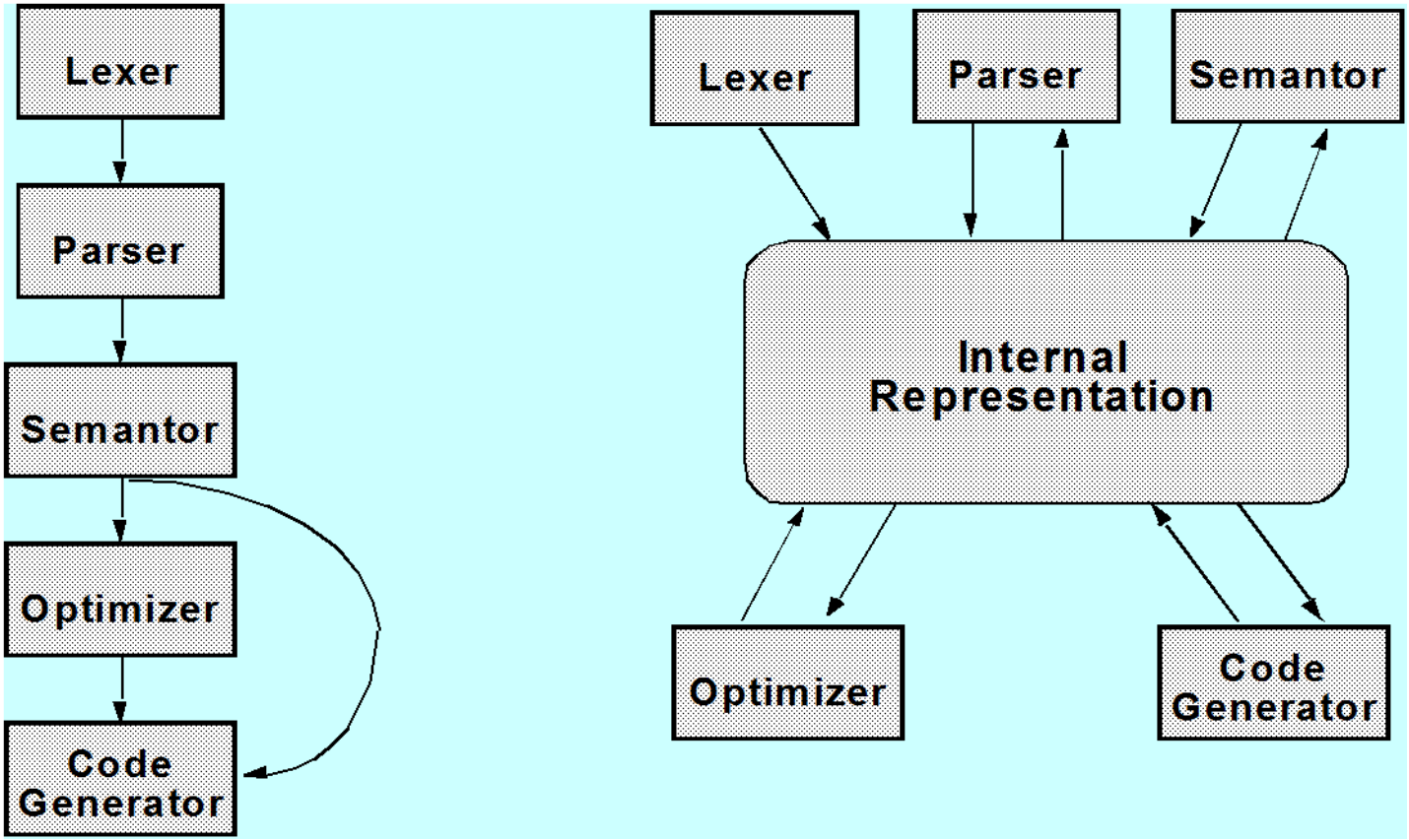
主要架构类型

Composite Subsystem 复合型系统



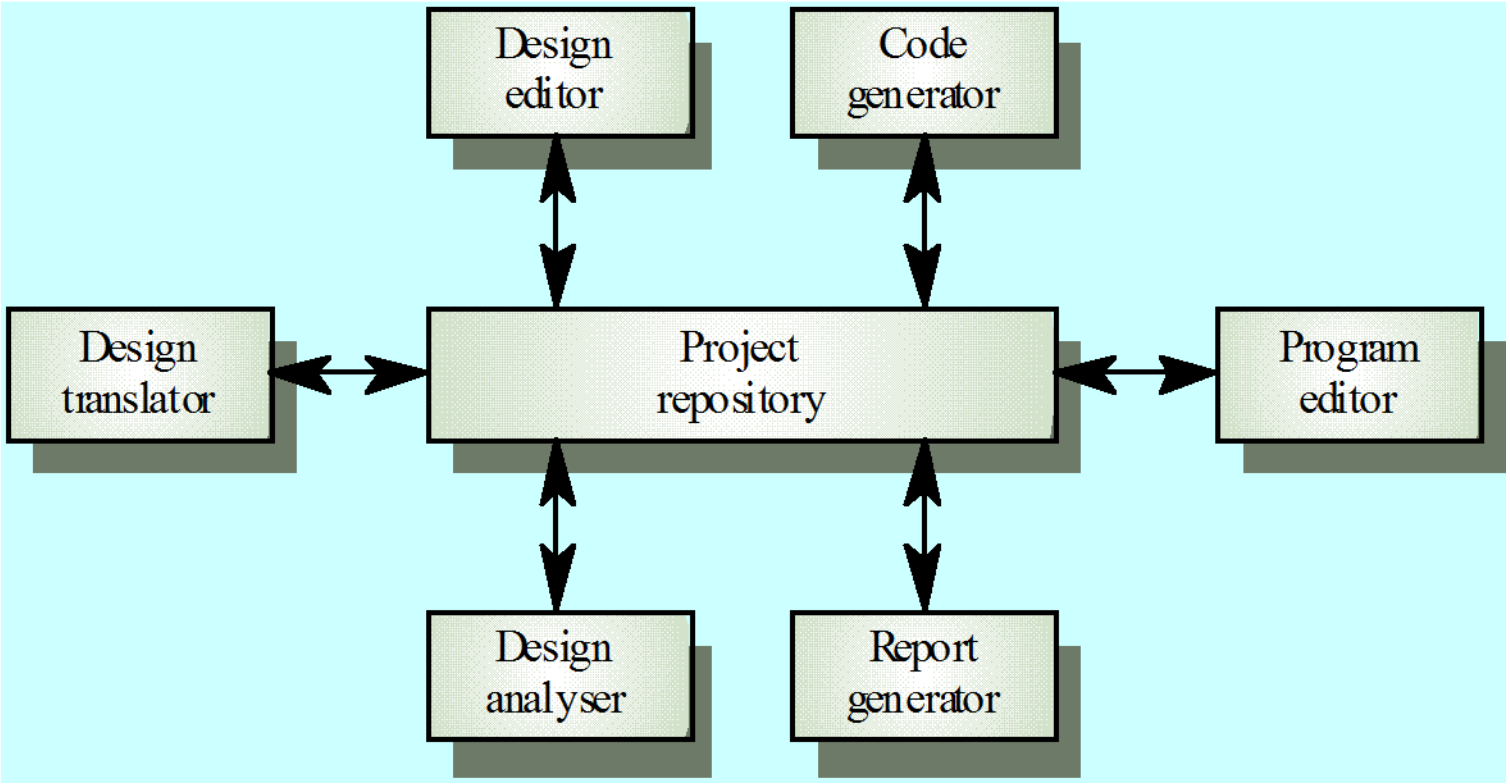
主要架构类型

Compiler Topology 编译器构型



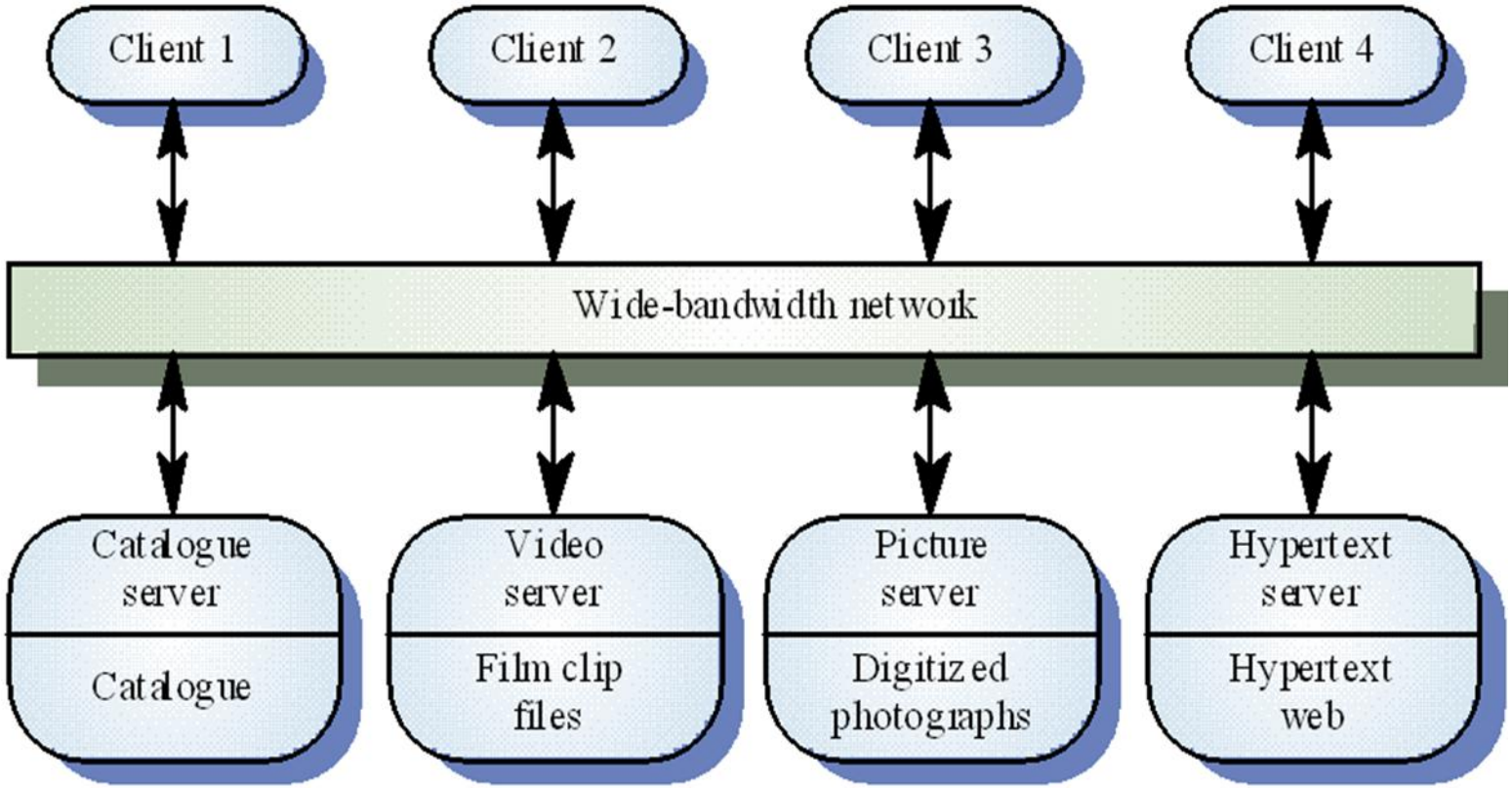
主要架构类型

Blackboard Architecture 告示板式架构



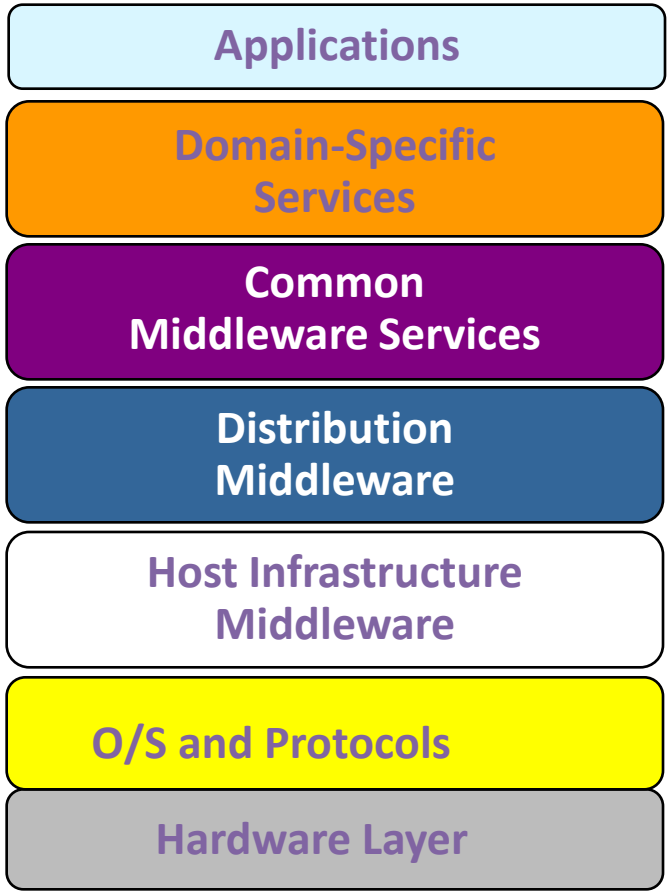
主要架构类型

Invocation Architecture 通道式架构

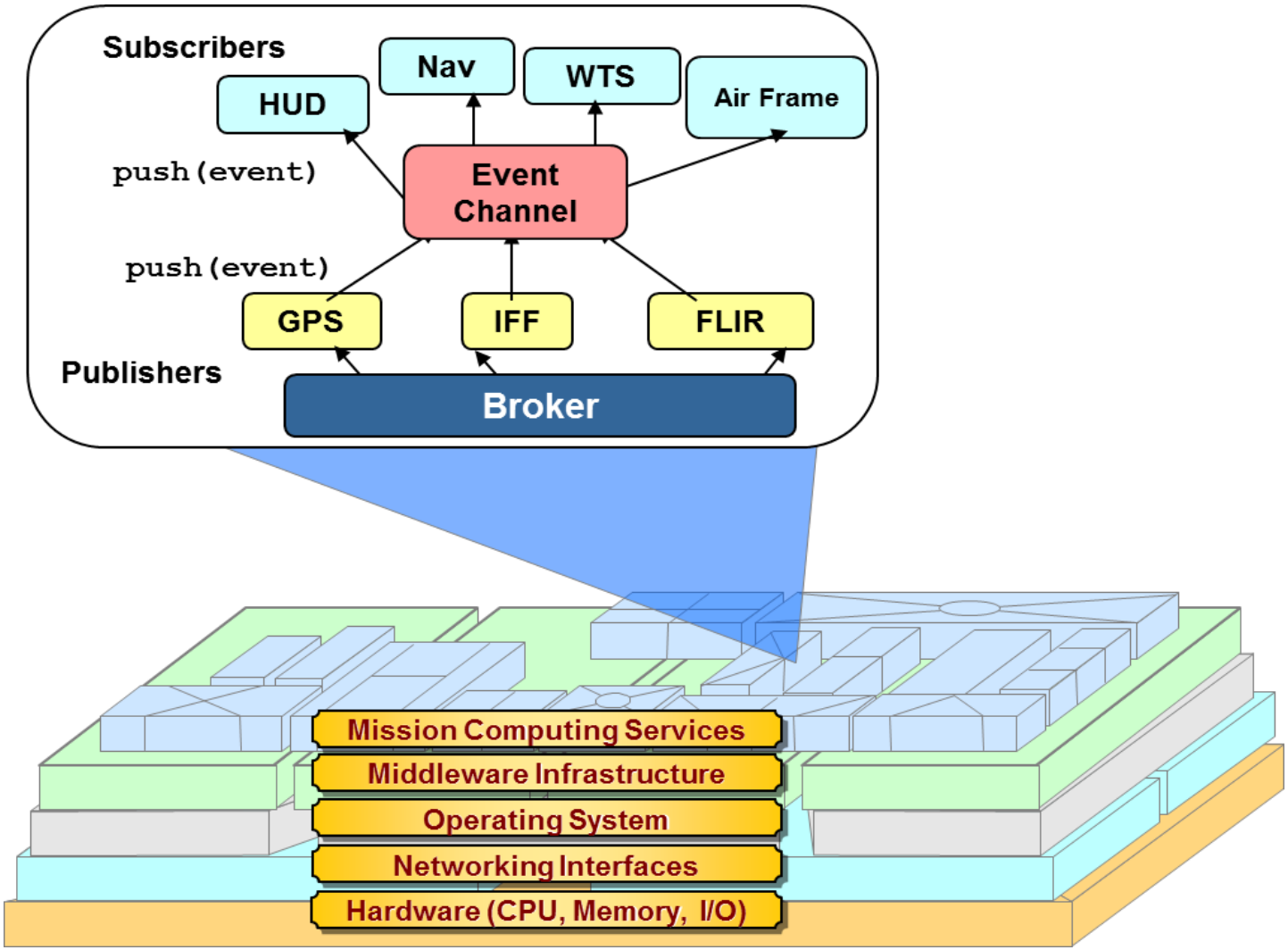


主要架构类型

Structural Model – layered architecture

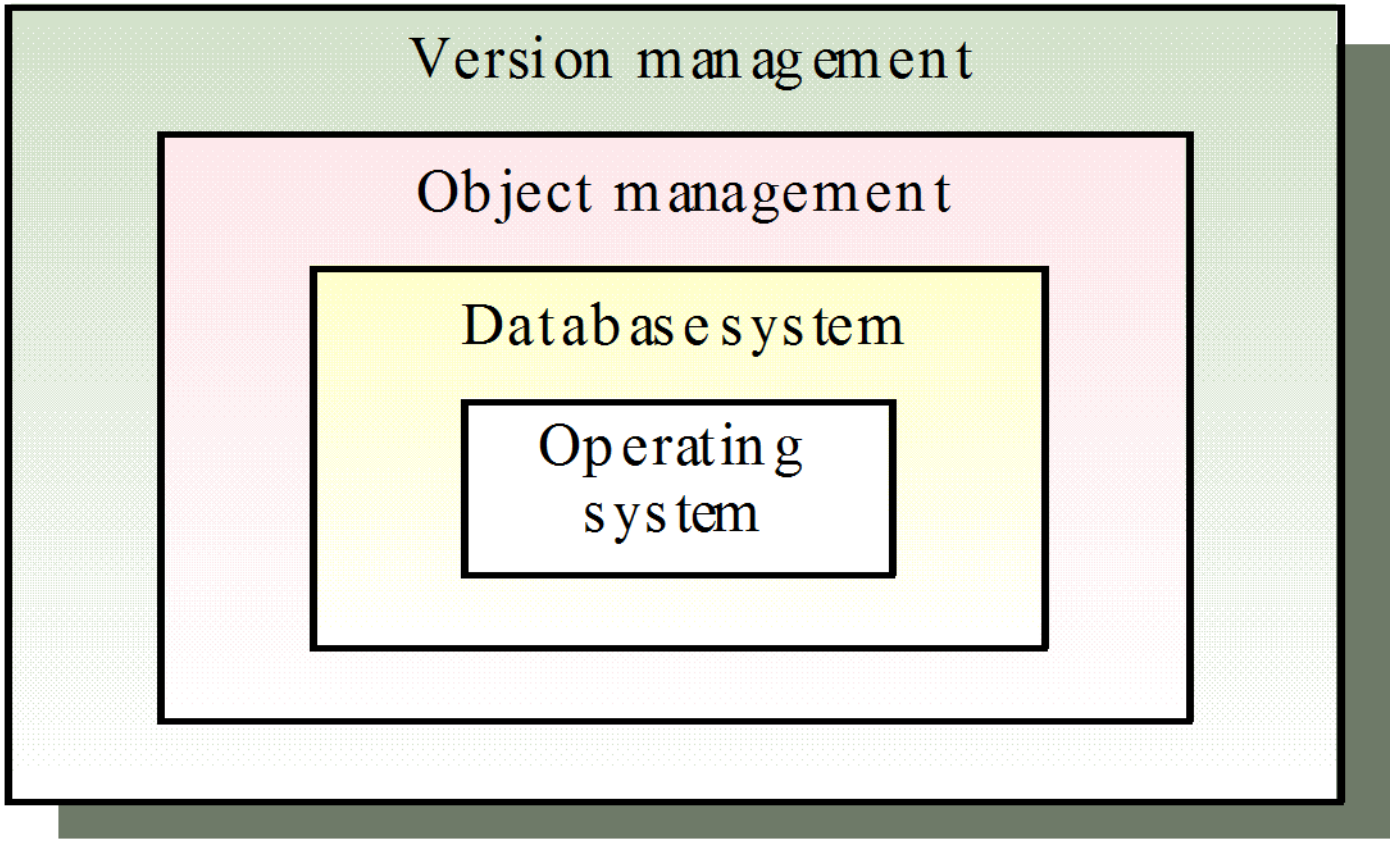


主要架构类型

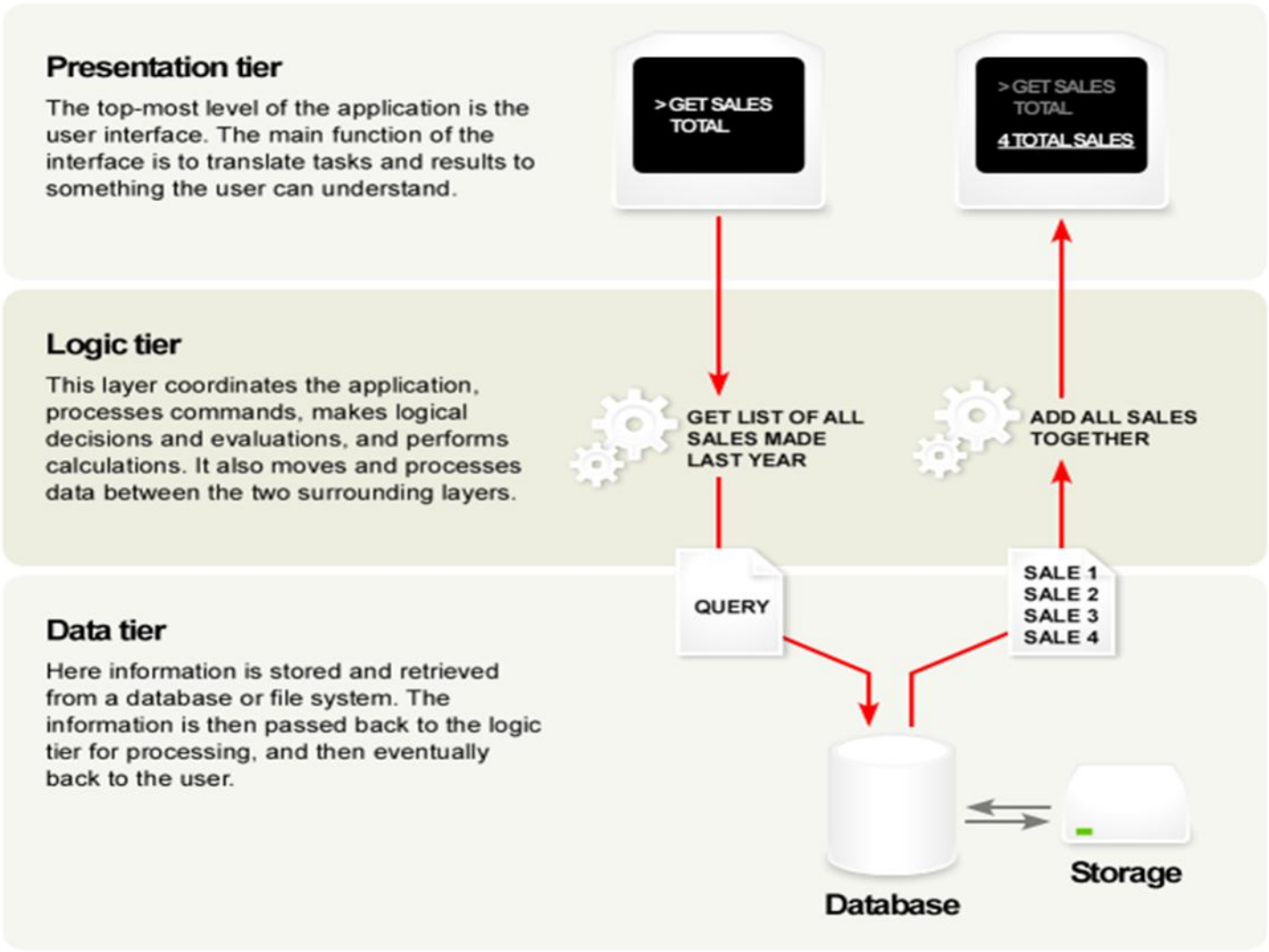

Subscribers
Nav
WTS
Air Frame
HUD
push (event)
Event Channel
push (event)
GPS
IFF
FLIR
Publishers
Broker
Mission Computing Services
Middleware Infrastructure
Operating System
Networking Interfaces
Hardware (CPU, Memory, I/O)

主要架构类型

Layered Architecture 层次架构



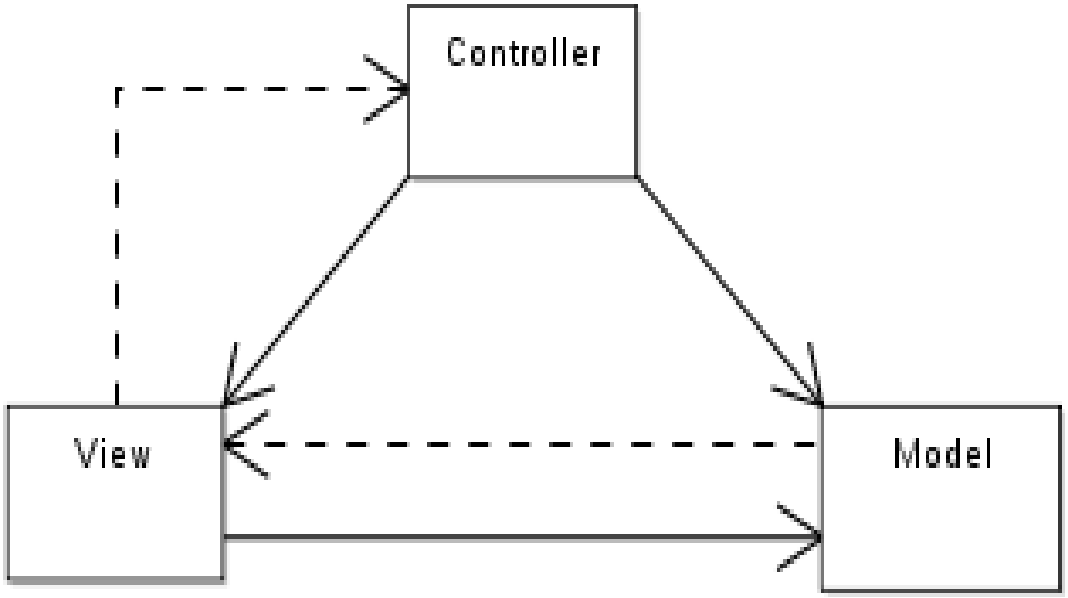
主要架构类型


Presentation tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.
>GET SALES TOTAL
>GET SALES TOTAL
4 TOTAL SALES
Logic tier
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.
GET LIST OF ALL SALES MADE LAST YEAR
ADD ALL SALES TOGETHER
Data tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.
QUERY
SALE 1
SALE 2
SALE 3
SALE 4
Database
Storage

主要架构模式

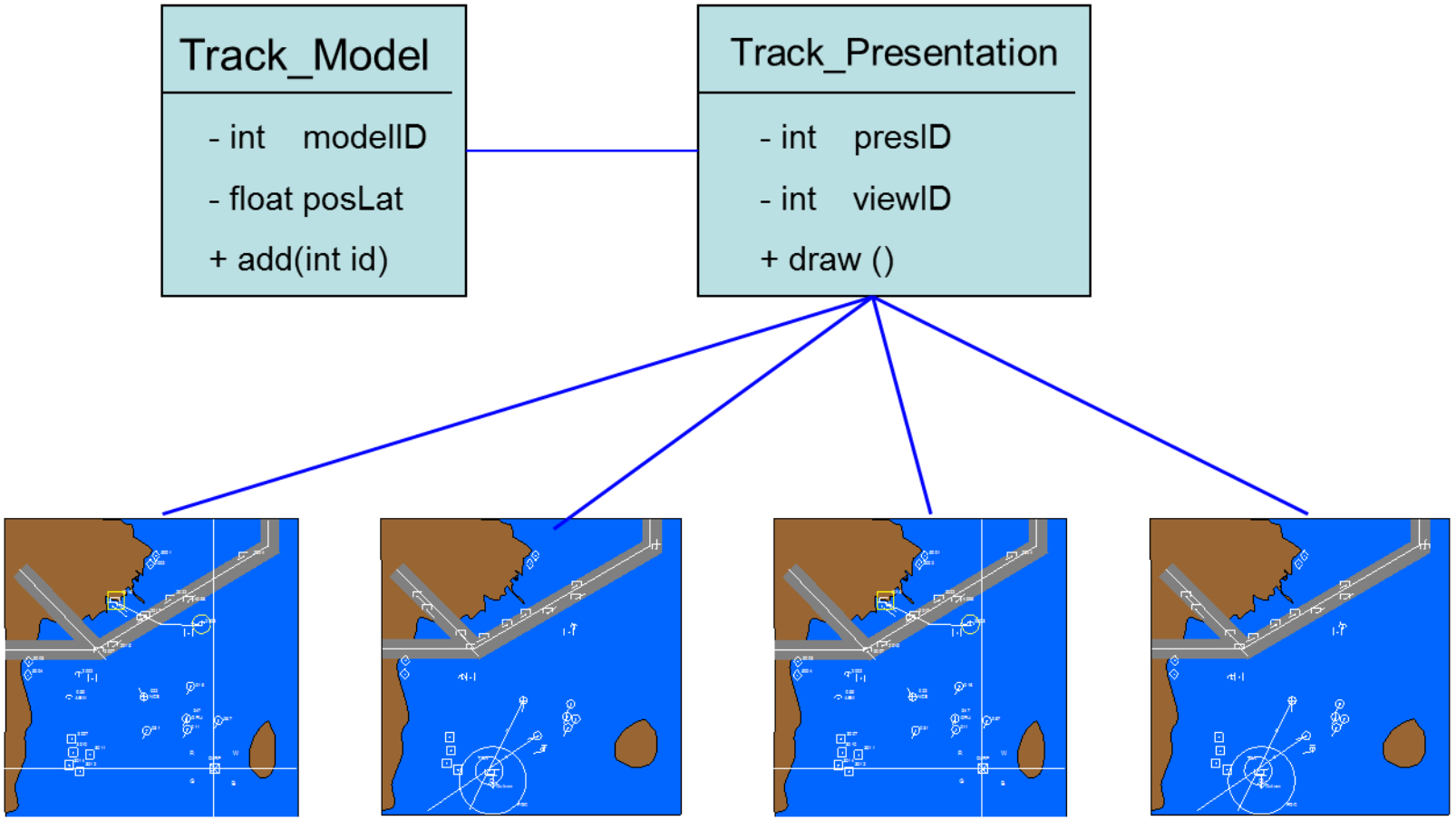
MVC (Model-View-Control) Pattern

- Java Swing
- GTK++
- MFC
- ASP .Net
- Adobe Flex



主要架构模式

PAC (Presentation-Abstraction-Control)



End of Lecture

Thanks!