**Layered：**

**benefits:**

Make the **development** easier.

**Testability** can be enhanced because each layer can be tested independently.

**penalties:**

Reduce the **performance**.

Reduce the **usability**.

Reduce the **scalability**( each layer is internally coupled, code adjustment is difficult).

Reduce the **configurability**.( Environmental configuration is difficult).

**Pipes and Filters:**

**benefits:**

Improve system **reusability** and support software reuse. It is important to provide data suitable for transfer between two filters, any two filters can be connected.

Improve **maintainability**, system maintenance and enhanced system performance. New filters can be added to existing systems; old ones can be replaced with improved filters.

Improve **performance**.Introducing concurrency.

**penalties:**

**Performance** is affected when dealing with interactions

**Blackboard：**

**benefits：**

Improve modifiability and **maintainability**

**Reusable** knowledge source

Improve **stability** and **robustness**

**penalties：**

Reduce **testability**

Make **development** difficult

Reduce **performance**

**Broker：**

**benefits:**

Improve **scalability**. Since the Server is registered to the Broker, the Server can be dynamically added, deleted, and changed.

Improve **reusability**.

**penalties：**

Reduce some **performance**. Added a layer of Broker message forwarding, which has reduced efficiency.

Testability requires attention with increased testing work.

**MVC:**

**benefits:**

Reduce coupling and increase **modifiability** and **scalability.**Because the model is separate from the controller and view, it is easy to change the application's data layer and business rules.

Improve **reusability.**MVC mode allows access to the same server-side code using a variety of different styles of views.

Improve the **configurability**.

Improve **maintainability**.Separating the view layer and the business logic layer also makes the WEB application easier to maintain and modify.

**penalties:**

In some cases, it will reduce **reusability**.Views and controllers are also closely related components. Views have no controllers, their applications are very limited, and vice versa, which hinders their independent reuse.

Depending on the model's operational interface, the view may need to be called multiple times to get enough display data. Unnecessary frequent access to unchanging data will also impair operational **performance**.

**C/S:**

**benefits:**

Improve **maintenance** simplicity

Improve **security**. All data is stored on the server, and the server has better control over access

**penalties:**

Lack of **robustness**.On the client server, if an important server fails, the client's request cannot be met.

**P2P:**

**benefits:**

In some cases, P2P systems can achieve direct data transfer, thereby improving **performance**.

Improve **robustness**

**penalties:**

Reduce **security**. P2P system cannot identify user identity due to environmental restrictions

Small P2P systems may not to consistently achieve quality goals such as **perfomance** and **availability.**

**SOA:**

**benefits:**

Enhanced **security, reliability, flexibility**

Low **coupling**

Due to the loose coupling between service providers and service users, and the adoption of open standard interfaces, it has **maintainability** and **usability**.

**penalties:**

Typically increases latency and has anegative effect on **scalability** and **predictability** of message delivery time.

**Publish-subscribe:**

**benefits:**

Pub/sub provides the opportunity for better **[scalability](https://en.wikipedia.org/wiki/Scalability" \o "Scalability)**than traditional client-server, through parallel operation, message caching, tree-based or network-based routing, etc.

Publishers are [loosely coupled](https://en.wikipedia.org/wiki/Loose_coupling" \o "Loose coupling) to subscribers, and need not even know of their existence.

**penalties:**

Typically increases latency and has a negative effect on **scalability** and **predictability** of message delivery time.

**PAC:**

**benefits:**

Completely isolate presentation and abstraction based on mvc and improve the **performance**.

Improve **modifiability、 scalability、reusability、configurability and maintainability**

**penalties:**

In some cases, it will reduce reusability.Presentation and controllers are also closely related components. Presentation have no controllers, their applications are very limited, and vice versa, which hinders their independent **reuse**.

Depending on the model's operational interface, the view may need to be called multiple times to get enough display data. Unnecessary frequent access to unchanging data will also impair operational **performance**.