|  |  |  |  |
| --- | --- | --- | --- |
|  | availability | interoperability | modifiability |
| layered | The components in the layer are only responsible for the single layer. | The internal components are organized into layers, each of which performs a specific role.  (No significant impact) | Grouping services into functional layers reduces the impact of change. Most changes affect only the layer in which they're made, with few side-effects that impact other layers. This fundamentally simplifies service maintenance. |
| broker | A software may only be available on a particular computer. | Broker mainly completes the handshake between Client and Server.  There are physical and logical boundaries between processes or computers running on the same network. To make objects running on different processes or computers communicate across these boundaries, you have to deal with issues such as communication, coding, and security. | Usually, there is a need of having great flexibility, maintainability and changeability when developing applications. |
| MVC | Not suitable for applications of small or medium scale. | The MVC model allows us to use a variety of different styles of view to access the same server-side code, because multiple views can share one model.  (No significant impact) | Product design flexibility is very small, it is difficult to meet the changing needs of users.  However, these models can be applied directly to the interface because the data returned by the model does not have any format. Once the model is changed, it is easy to modify. |
| pipe-filter | The filter must be an independent entity, the state of each filter is not affected by other filters. | Not suitable for the systems who need to meet frequent user interaction. | When an object is changed, it is necessary to modify all the part of the object that calls it. |
| client-sever | Only applicable to local area network. | Can give full play to the client's ability to deal with a lot of work, which can be submitted to the server after the client processing. | This means that if you need to modify the application code, you can modify it only on the server, instead of modifying thousands of client applications.  However, when system software upgrades, each client needs to be reinstalled, the cost of maintenance and upgrade are very high. |
| P2P | Achieve the separation of file searching and file transferring. | A group of high performance central servers store directory information for all active peer-to-peer computers on the network. After the query initiates the peer to receive the response, it selects the information according to the network traffic and delay, establishes a connection with the appropriate peer, and starts the file transfer. | With the expansion of the network scale, the cost of maintaining and updating the central index server will increase dramatically, and the cost is very high. |
| service-oriented | Applying this pattern requires creating a service inventory blueprint, a list of services with associated functionality. | Service-oriented architecture makes it easier for software components to communicate and cooperate over the network, without requiring any human interaction or changes in the underlying program, so that service candidates can be redesigned before their implementation. | A service presents a simple interface to the requester that abstracts away the underlying complexity acting as a black box, Further users can also access these independent services without any knowledge of their internal implementation. However, the black box itself is hard to modify. |
| publish-subscribe | The broker may not notify the state of the system message transfer. So we don't know whether the message is successful or not. | The publisher does not need to know the number of subscribers and in which way the subscriber runs. They can run independently of each other so that you can separate the two parts without worrying about any subtle effects on the state or implementation. | This mode can reduce the degree of coupling, it is easy to modify. |
| share-data | The system can make decision by sharing data. | Since the system is isolated, the data required by the system and the data they generate cannot be communicated. In particular, the data cannot be shared by other applications. | Good modification to avoid data conflict. |
| multi-tier | The improvement of availability is limited.  (No significant impact) | Each layer can communicate and be deployed separately.  (No significant impact) | The development of the business is based on the component, so adding and deleting the new function is very convenient. Independent upgrades and changes can be made without affecting other layers. |
| map-reduce | The framework solves complex problems such as data distributed storage, job scheduling, fault tolerance.  Does not meet the needs of real-time applications. | Help with the communication between systems and machines. | Difficult to modify. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | performance | security | testability | usability |
| layered | This makes it easy to build modules with effective roles and responsibilities in the architecture, which is also easy to manage. | Each layer of the architecture is marked as closed, which means, when a request is moved between layers, it must pass through the layer below to get to the next level. This contributes to security. | Each layer can be tested independently. | It goes well with the traditional IT communication and organization structure, so most business application will choose it. |
| broker | Can use multiple CPU or a group of low-cost computers.  Add a layer of Broker message forwarding, which reduces the efficiency. | For security reasons, the parts of the software may be running in different segments.  Inherent networking complexities such as security concerns, partial failures, etc. | Due to the small coupling of components, the work of debugging and testing is also controllable.  However, it has added debugging and testing work. | It can locate the server, if there is an exception, the exception can be passed to Client. |
| MVC | The efficiency of view accessing the model is very low. Depending on the operating interface of the model, the view may require multiple calls to obtain data. Unnecessary frequent access to the data that is not changed will also damage the performance. | There may be malicious information. | Because of the strict separation of model and view, it also brings some difficulties to testing. | It will increase the complexity of the structure. |
| pipe-filter | Support concurrently executing threads. | Although there are some constraints on the input and output of the filter, the filter does not need to know the internal details of the filter that provides the data flow. | When an object is changed, it is necessary to modify all the part of the object that calls it, which increase the tasks of testing. | Any two filters can be connected as long as they meet the same constraint. |
| client-sever | Client response fast. | Data storage management function is transparent. | The testing tasks are very heavy. | Network management staff have to maintain and manage the server, as well as the client, which requires high investment and complex technical support. |
| P2P | High efficiency of resource discovery. | The paralysis of the central server can easily lead to the collapse of the whole network. Low reliability and security. | Because different users use different browsers, computer models are different, the operating system may not be the same, so in order to allow all users to access the P2P website platform, the testing is complex. | In the face of the dynamic change of the network, it has better fault tolerance, so it has good usability. |
| service-oriented | Service-oriented architecture aims to allow users to combine large chunks of functionality to form applications which are built purely from existing services and combining them in an ad hoc manner. | There was a conflict between security and easy availability. | It is a black box for its consumers. So it may have influence on testability. | Different services can be used in conjunction to provide the functionality of a large software application. |
| publish-subscribe | With the increase of the number of publishers and subscribers, message transmission increased lead to instability in the structure, easy to load when the problem. | An attacker (malicious publisher) can invade the system and tear it apart. This will lead to a malicious message being issued, the subscriber can obtain message that they could not get before. | Can easily find out whether the publication or subscribers will get the wrong information, so it is easy to test. | It needs intermediaries / agents, message specification and related rules will add complexity to the system. |
| share-data | With the increase of application system users. It is easy to cause the application system to slow down, crash, and even lead to the paralysis of the central library. | There exists risks in the security of data. With the increase of the application system and the disunity of the construction unit, the data of the central database can be confused in the case of the direct use of the central library, and even some data can be deleted. | Simple test.  (No significant impact) | It is very easy to use in data analysis. |
| multi-tier | Because many networks, computers and processes are running independently, once the system hardware and network bandwidth is poor, the performance of the whole system will be slow. | A better secure control of the whole system, we can implement different security policies for each layer, since business layer and data layer need higher security than view layer. We can put the high security layer behind the firewall protection. | Each layer can be tested independently. | In the web development field, three-tier is often used to refer to websites, commonly electronic commerce websites.  (No significant impact) |
| map-reduce | Programmers can easily complete large data parallel programming.  But if there is a conflict, it will bring additional communication overhead, reduce efficiency. | Lack of security mechanisms. | Improve test efficiency. | Not only can be used to deal with large amounts of data, but also to hide a lot of tedious details. |