# Section 5: Control Strategies

## Question 1

Describe why the client-server control strategy should be applied to a wider range of design problems than pipes and filters designs.

### Answer

Because interactive systems require a request – response interaction (protocol) between a client and server components. For example, between a browser or phone app and the system (web application).

The pipes and filter strategy relies on the asynchronous exchange of messages between producers and consumers. It is difficult to implement a client-server protocol in a P&F design.

## Question 2

Describe the difference between stateful and stateless services. Describe the example of each from the slides.

How does web services (which are stateless because of HTTP) implement state for each of its individual clients?

### Answer

A stateful service maintains a persistent connection with its client. Individual clients (users) start a new session with a unique instance of a service class. The service instance is not shared with other clients and so can maintain session-specific values (state) that is retained between service requests. SSH is stateful because the SSH client is connected to a shell process across a persistent connection. Each client command is processed by the same shell process, and the process is not shared with any other client.

A stateless service does not maintain a persistent connection with its clients between client requests. Individual clients share a single instance of the service class, so a stateless service cannot maintain client-specific state information. HTTP is defined as a stateless protocol i.e. the connection between client and server does not persist between client requests. So web services which rely on HTTP cannot be assured of accessing the same service instance.

HTTP servers implement the Session (HTTP Session). The session is a key-value data structure that is maintained by the HTTP server. The server creates a new Session instance and associates the instance with a unique SessionID. The ID is returned to the client and the client is expected to provide its SessionID in each request it makes to the service. Using the ID, the server can lookup a client’s session and store client-specific values such as shopping carts, etc.

## Question 3

What is the major difference between a workflow design that uses Client-Server vs. Pipes and Filters in terms of how information is processed?

### Answer

A design that uses client-server typically executes in a single thread (of execution) where the client makes a service request (method) to the server object. The service encapsulates many processing steps and decisions that makes up the workflow implemented by the method. The workflow is executed to respond to the client’s request and produces a value that is returned to the client. Examine the sequence diagram presented in the section slides to illustrate what is meant by single thread.

A design that uses pipes & filters (dataflow) will include several independent processing stages (filters) each of which independently implements a processing step in the workflow. Rather than encapsulate the workflow into a single module (function / method), the workflow’s is spread across multiple filters which cooperate with other filters to fulfill the needed workflow.

Each filter executes in an individual process / thread. Filters are connected to each other by communication “pipes” that pass messages from one filter to the next. In a P&F design, messages pass from one filter to the next in a one-way processing flow, and there is no request-response protocol.

## Question 4

Describe why, or why not, a pipes and filter design will be suitable for the implementation of a HTTP web service.

### Answer

A P&F design is not suitable for a web service. This is because HTTP is a client-server protocol. The client makes a request to a service, the request is processed by the server which generates response (result) that is returned to the requesting client. All of this occurs in a single thread meaning that the client execution is blocked while the server processes the request, and the client’s execution resumes once the server’s response is received. This is true even when the client and server execute on different machines and communicate over a network connection e.g. HTTP.

A P&F design divides the processing into discrete filters each of which execute independently of the others in the system. There is no connection with the first processing step triggered by a client’s request and the last filter in the workflow. There is no (natural) means of connecting the last processing stop to the client’s original request so the result cannot be returned to the client as a server response.

## Question 5

Describe why an architecture design based on pipes and filters will likely have better cohesion and less coupling than a design based on the client-server architecture.

### Answer

A pipes and filters architecture works best with a problem whose solution can be divided into separate, independent processing steps (filters). The design of each filter should be focused on a single data processing operation of the information it receives and information it generates. This is the definition of Cohesion in a software design.

To enhance its reusability, the filter’s design should not share state with (be coupled with) other filters in the system. This second point is the definition of loose coupling. Often each filter is implemented as a separate process, possibly on separate machines, where sharing of state is difficult and the only form of coupling possible is the messages exchanged between producer and consumer.

## Question 6

What is the difference between designs that use Pipes and Filters vs. Batch Processing in term of how information is processed?

### Answer

Both P&F and Batch designs share a common theme of processing stages, interconnected by communication channels, that process data provided by upstream processes and provide processed data to downstream processes.

However, there are these differences:

1. In a P&F design, the filters (information processors) are likely implemented as individual active processes that process incoming data (messages) in real-time. Messages are processed as they arrive (in real-time) and data flows continuously though the network of pipes and filters.
2. In a batch design, the information processors are processes scheduled to run specific times (not real-time). Each process is executed against a set of input data that has been accumulated over time in a repository (files, databases, etc.). Each process reads and processes all the information from its input, writing each result to its output. Each process processes the entire contents of its input repository before exiting. Each process runs individually (not concurrently as in P&F).

## Question 7

Briefly describe a filter’s processing loop.

What is the significance of blocking in a filter’s design?

### Answer

Each filter has a separate thread that executes a simple processing loop:

1. Read a new data item from its input pipe. The filter process will block if the filter’s input pipe is empty (contains no data) and will remain blocked until an upstream filter places data in its output pipe.
2. Process the input item producing an output item,
3. Write the output item to the filter’s output pipe.

It is the blocking of the filter’s execution on reading an empty input pipe that coordinates the overall activities of the network of filters.

# Question 8

What are two important features of a Pipes and Filter design?

## Answer

Any two of the following will do…

1. The design of the filter is based on its transformation of input into output data. Its design should not depend on the implementation of its upstream or downstream filters.
2. Filters should be simple and cohesive. Each filter should do “one thing” in the design.
3. Filters should be designed for reuse, both within the system and perhaps in different systems.
4. Filters should be replaceable. A filter’s implementation can be substituted with a different design/implementation so long as the data it accepts and produces, and its processing algorithm remains constant across all implementations.

# Question 9

Describe how the Batch Processing strategy is used to delay data processing until off-peak hours.

## Answer

During the system’s busy hour, work (data) can be collected into a batch e.g. a file or database repository. Processing of the batched data can be scheduled during off-peak hours to make better overall use of the enterprise compute resources (either on-site or in the cloud). This approach can be contrasted with a Pipes & Filters design where the arriving data is processed in real-time (as it arrives) and not accumulated in a repository.

# Question 10

Provide a brief description of ETL. What are its advantage and disadvantage?

## Answer

Extract, Transform, Load is a strategy where existing data sources can be repurposed to meet the requirements of new services / applications. An ETL extracts and aggregates information from one (and usually several) legacy enterprise databases. The extracted data is stored in a new application-specific database. The new applications access the extracted data and not the original legacy sources. The ETL approach is safer and has less impact on legacy applications than allowing new applications access the legacy data sources directly which may interfere with the legacy system’s operation.

The advantage of an ETL is that new applications can be installed that utilize existing legacy databases without the need to modify either those databases or the legacy applications that maintain the data.

The disadvantage of an ETL is that the extracted data is not real-time i.e. it reflects the state of the enterprise that existed when the ETL was performed.