

HTTP

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. HTTP functions as a combination of FTP and SMTP.

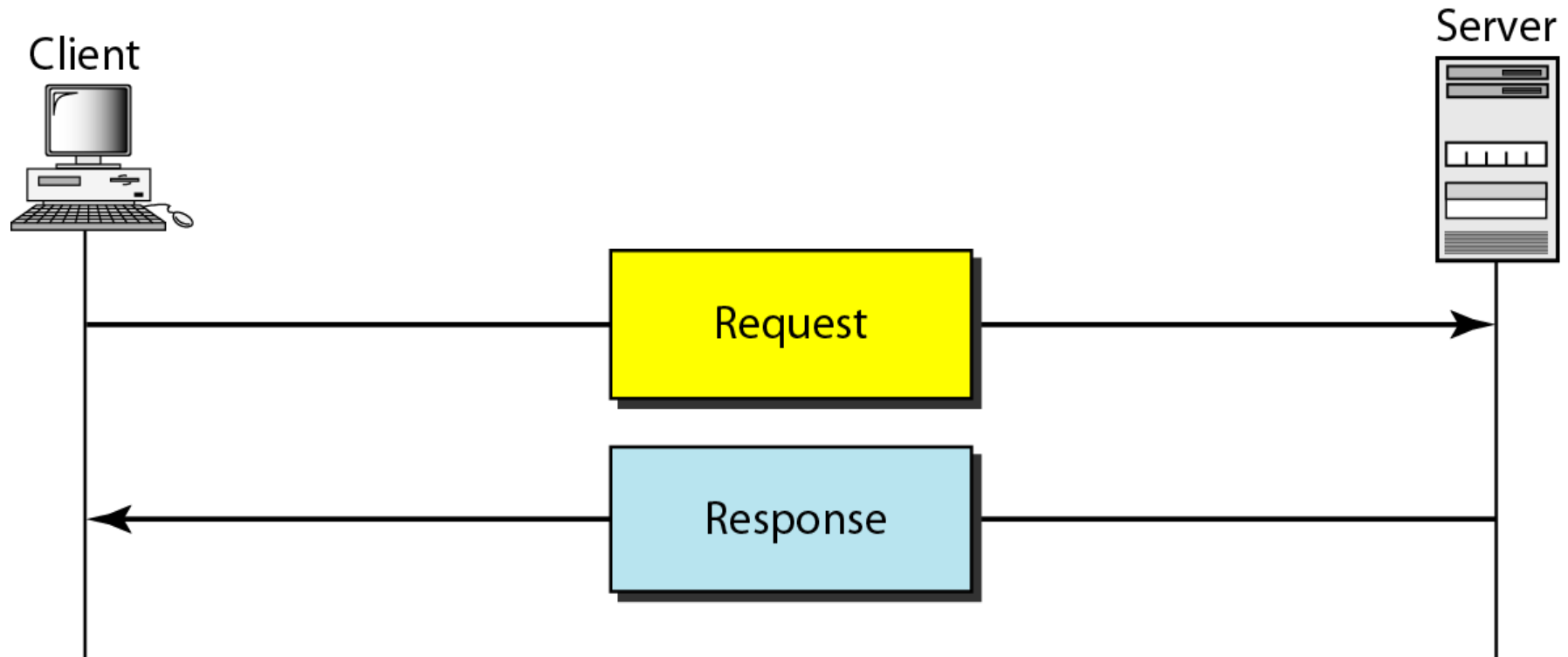
Two other protocols that you can use on the Internet are the File Transfer Protocol (FTP) and the Telnet Protocol



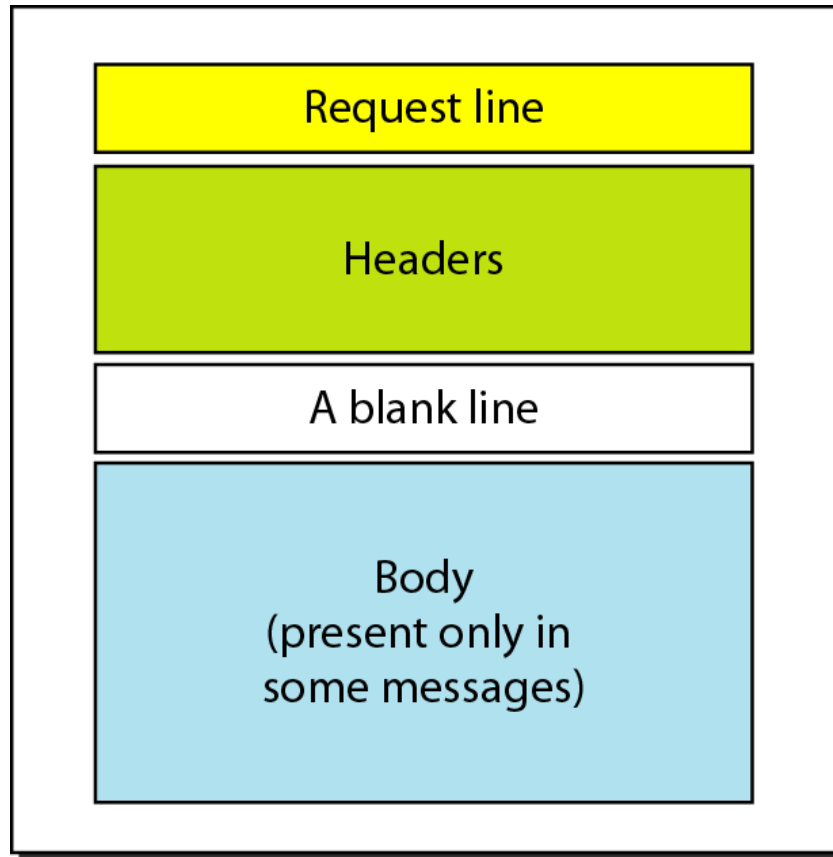
Note

HTTP uses the services of TCP on well-known port 80.

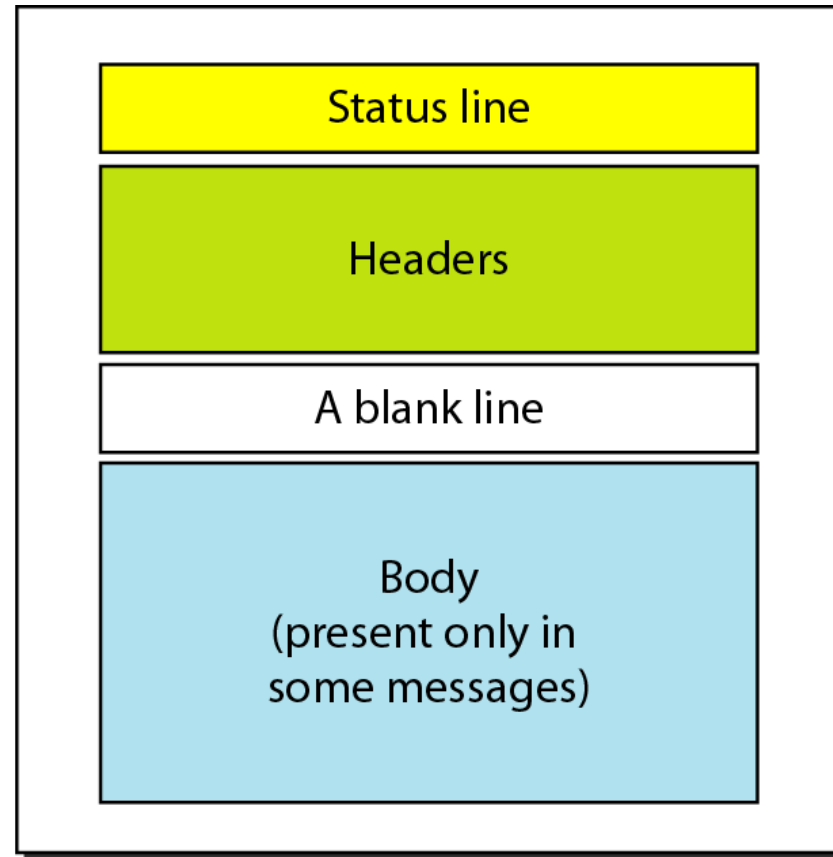
HTTP transaction



Request and Response messages

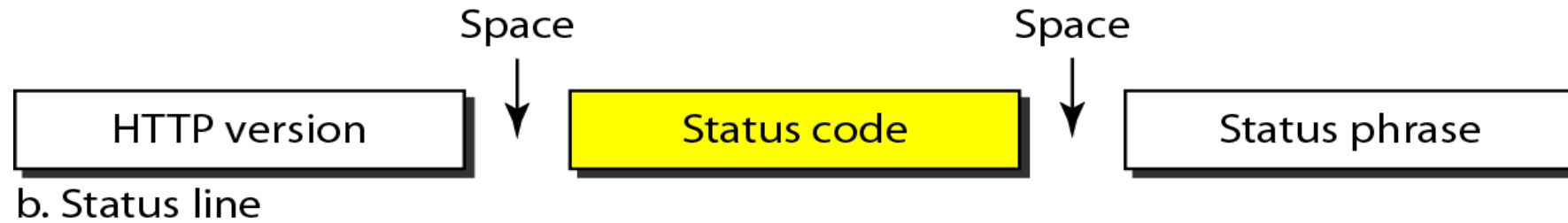
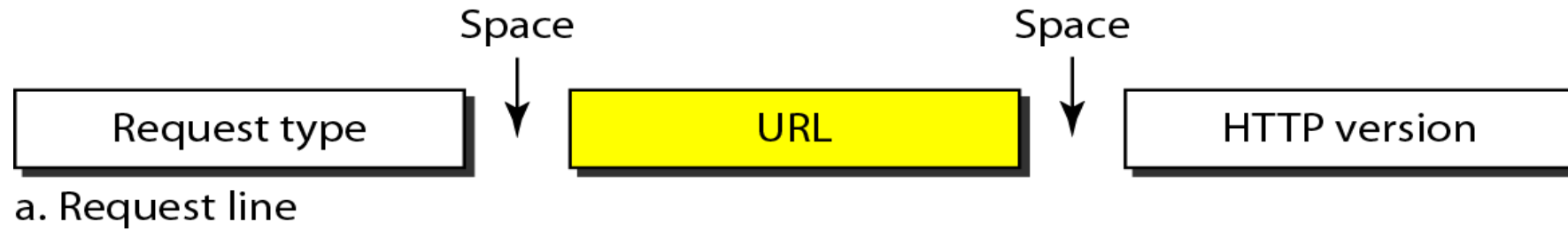


Request message



Response message

Request and status lines



Methods

<i>Method</i>	<i>Action</i>
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires about available options

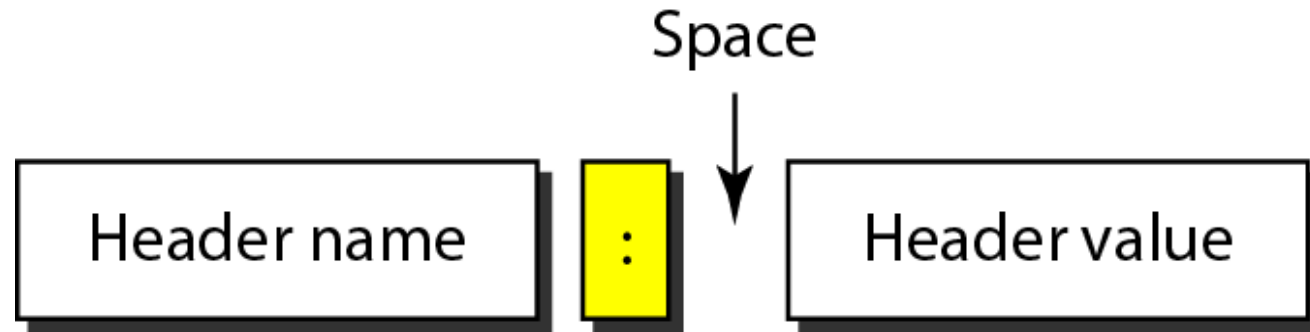
Status codes

<i>Code</i>	<i>Phrase</i>	<i>Description</i>
Informational		
100	Continue	The initial part of the request has been received, and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
Success		
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon.
204	No content	There is no content in the body.

Status codes (continued)

<i>Code</i>	<i>Phrase</i>	<i>Description</i>
Redirection		
301	Moved permanently	The requested URL is no longer used by the server.
302	Moved temporarily	The requested URL has moved temporarily.
304	Not modified	The document has not been modified.
Client Error		
400	Bad request	There is a syntax error in the request.
401	Unauthorized	The request lacks proper authorization.
403	Forbidden	Service is denied.
404	Not found	The document is not found.
405	Method not allowed	The method is not supported in this URL.
406	Not acceptable	The format requested is not acceptable.
Server Error		
500	Internal server error	There is an error, such as a crash, at the server site.
501	Not implemented	The action requested cannot be performed.
503	Service unavailable	The service is temporarily unavailable, but may be requested in the future.

Header format



General headers

<i>Header</i>	<i>Description</i>
Cache-control	Specifies information about caching
Connection	Shows whether the connection should be closed or not
Date	Shows the current date
MIME-version	Shows the MIME version used
Upgrade	Specifies the preferred communication protocol

Request headers

<i>Header</i>	<i>Description</i>
Accept	Shows the medium format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
From	Shows the e-mail address of the user
Host	Shows the host and port number of the server
If-modified-since	Sends the document if newer than specified date
If-match	Sends the document only if it matches given tag
If-non-match	Sends the document only if it does not match given tag
If-range	Sends only the portion of the document that is missing
If-unmodified-since	Sends the document if not changed since specified date
Referrer	Specifies the URL of the linked document
User-agent	Identifies the client program

Response headers

<i>Header</i>	<i>Description</i>
Accept-range	Shows if server accepts the range requested by client
Age	Shows the age of the document
Public	Shows the supported list of methods
Retry-after	Specifies the date after which the server is available
Server	Shows the server name and version number

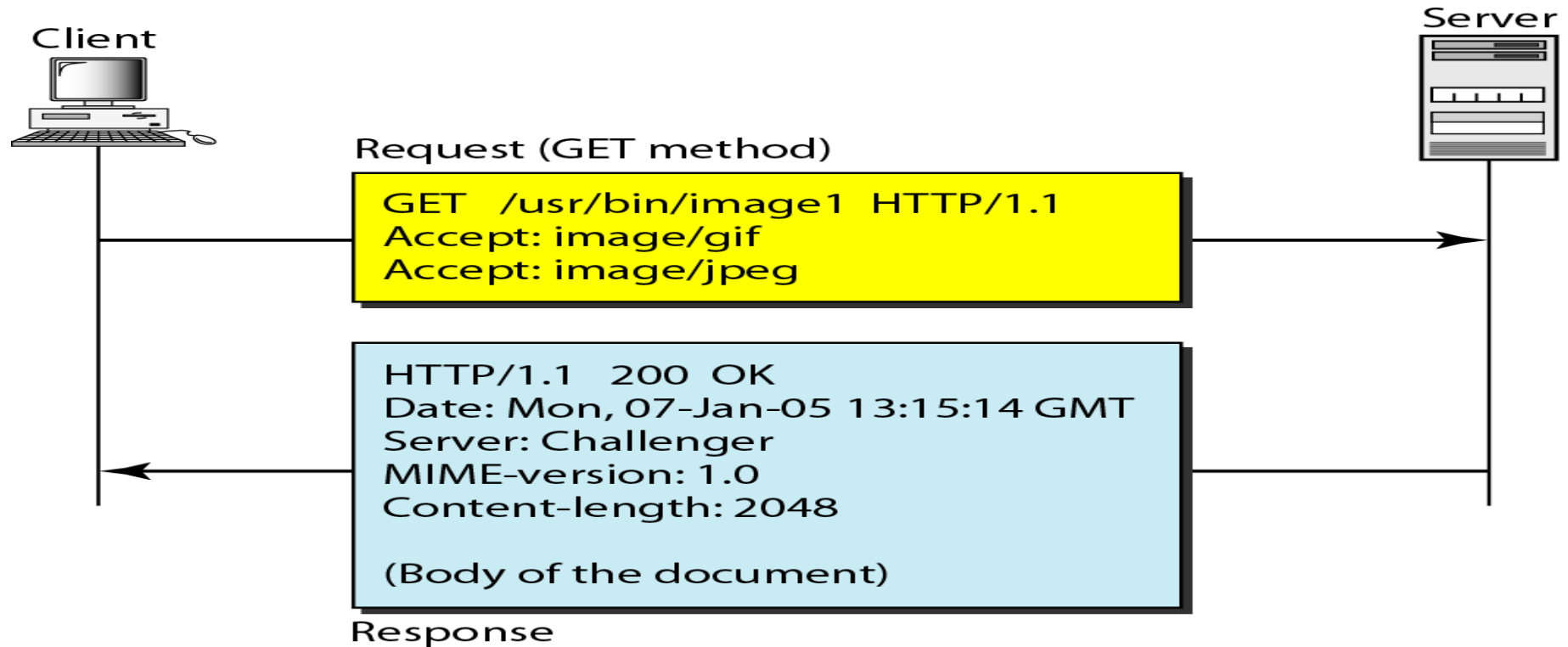
Entity headers

<i>Header</i>	<i>Description</i>
Allow	Lists valid methods that can be used with a URL
Content-encoding	Specifies the encoding scheme
Content-language	Specifies the language
Content-length	Shows the length of the document
Content-range	Specifies the range of the document
Content-type	Specifies the medium type
Etag	Gives an entity tag
Expires	Gives the date and time when contents may change
Last-modified	Gives the date and time of the last change
Location	Specifies the location of the created or moved document



This example retrieves a document. We use the GET method to retrieve an image with the path /usr/bin/image1. The request line shows the method (GET), the URL, and the HTTP version (1.1). The header has two lines that show that the client can accept images in the GIF or JPEG format. The request does not have a body. The response message contains the status line and four lines of header. The header lines define the date, server, MIME version, and length of the document. The body of the document follows the header

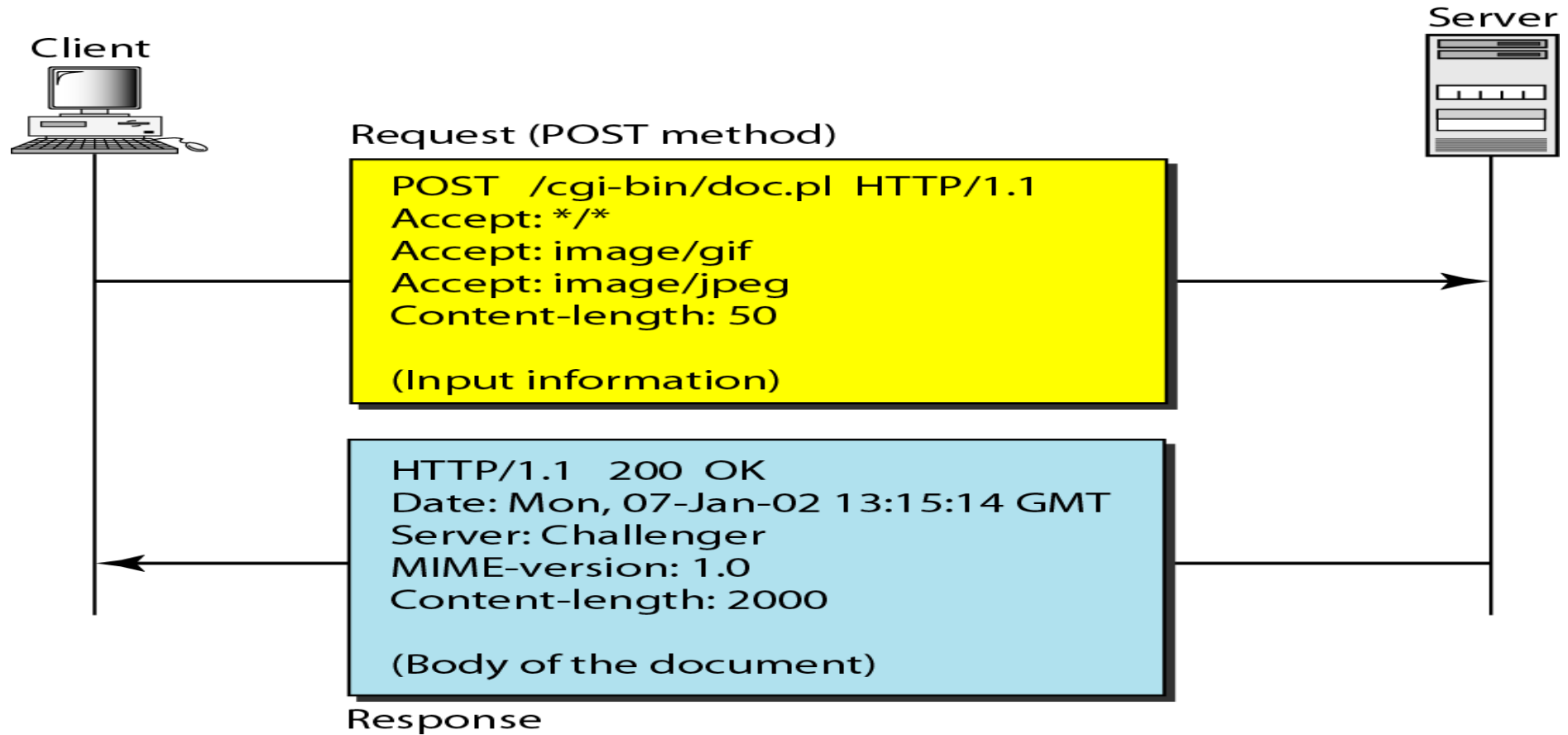
Example shown as Figure





In this example, the client wants to send data to the server. We use the POST method. The request line shows the method (POST), URL, and HTTP version (1.1). There are four lines of headers. The request body contains the input information. The response message contains the status line and four lines of headers. The created document, which is a CGI document, is included as the body.

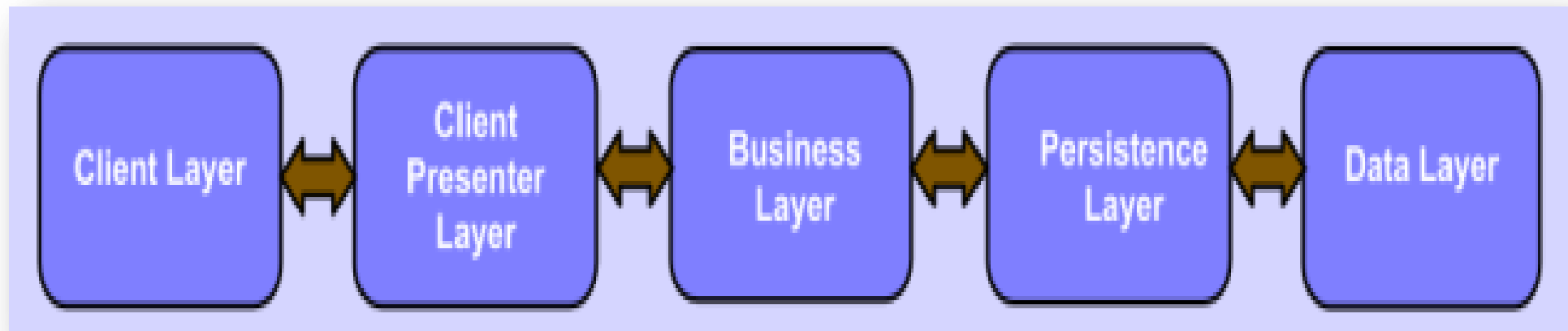
Example shown as Figure



N- Tier Architecture

is an industry-proved software architecture model

suitable to support enterprise-level client/server applications by resolving issues like scalability, security, fault tolerance etc



Layers

Client layer: this layer is involved with users directly. There may be several different types of clients coexisting, such as WPF, Window form, HTML web page and etc

Client presenter layer: contains the presentation logic needed by clients, such as ASP .NET MVC in IIS web server. Also it adapts different clients to the business layer

Business layer: handles and encapsulates all of business domains and logics; also called domain layer

Persistence layer: handles the read/write of the business data to the data layer, also called data access layer (DAL)

Data layer: the external data source, such as a database

Differences between Tier and Layer

***Tier** – physical deployment computer*

***Layer** – logical component group mainly by functionality; used for software development purpose*

***Note:** Layer and tier may or may not exactly match each other.*

- ✓ *Each layer may run in an individual tier.*
 - ✓ *Multiple layers may also be able to run in one tier.*
 - ✓ *A layer may also be able to run in multiple tiers.*
-

Three layers

***Presentation layer** - a layer that users can access directly, such as desktop UI, web page and etc. Also called client*

***Application layer** - this layer encapsulates the business logic (such as business rules and data validation), domain concept, data access logic and etc. Also called middle layer*

***Data layer** - the external data source to store the application data*

1, 2, 3 or More Tier Architecture

1-Tier:

- ✓ *all above layers can only run in one computer*
 - ✓ *In order to achieve 1-Tier, we need to use the embedded database system, which cannot run in an individual process*
-

1, 2, 3 or More Tier Architecture (continued)

2-Tier:

- ✓ *either presentation layer and application layer can only run in one computer*
 - ✓ *or application layer and data layer can only run in one computer*
 - ✓ *The whole application cannot run in more than 2 computers*
-

1, 2, 3 or More Tier Architecture (continued)

3-Tier:

- ✓ *the simplest case of N-Tier architecture*
 - ✓ *all above three layers are able to run in three separate computers*
 - ✓ *Practically, these three layers can also be deployed in one computer (3-Tier architecture, but deployed as 1-Tier)*
-

Thanks...