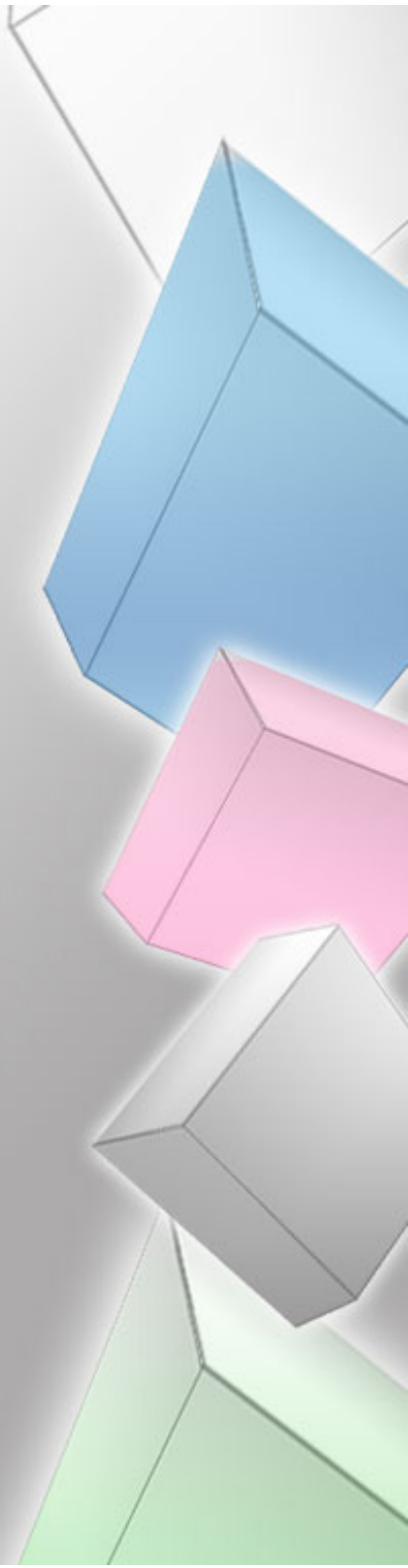
Advanced Topics in Operating Systems

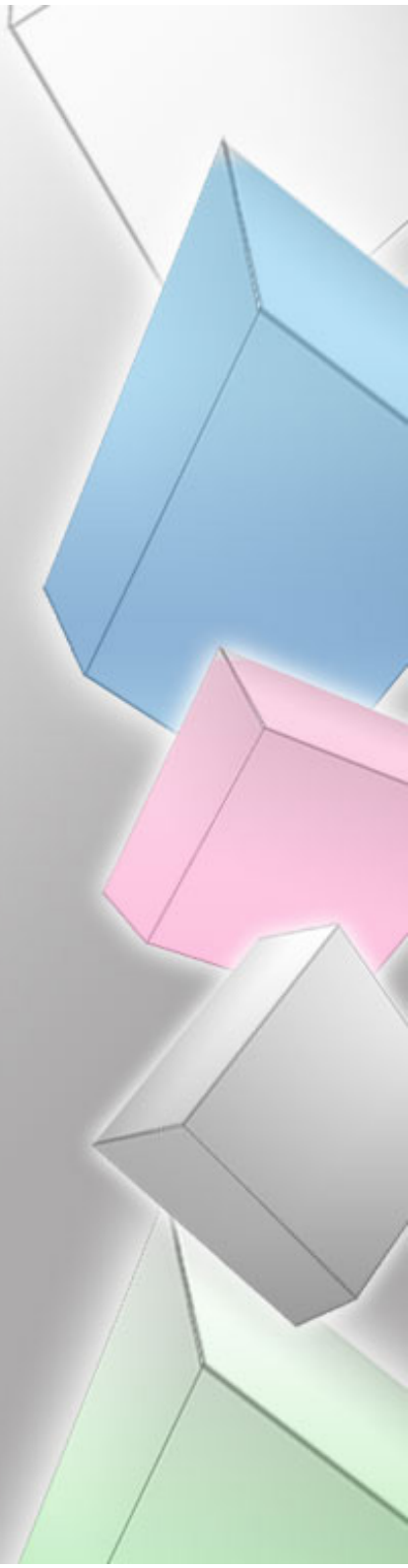
MSc in Computer Science
UNYT-UoG

Dr. Marenglen Biba
8-9-10 January 2010



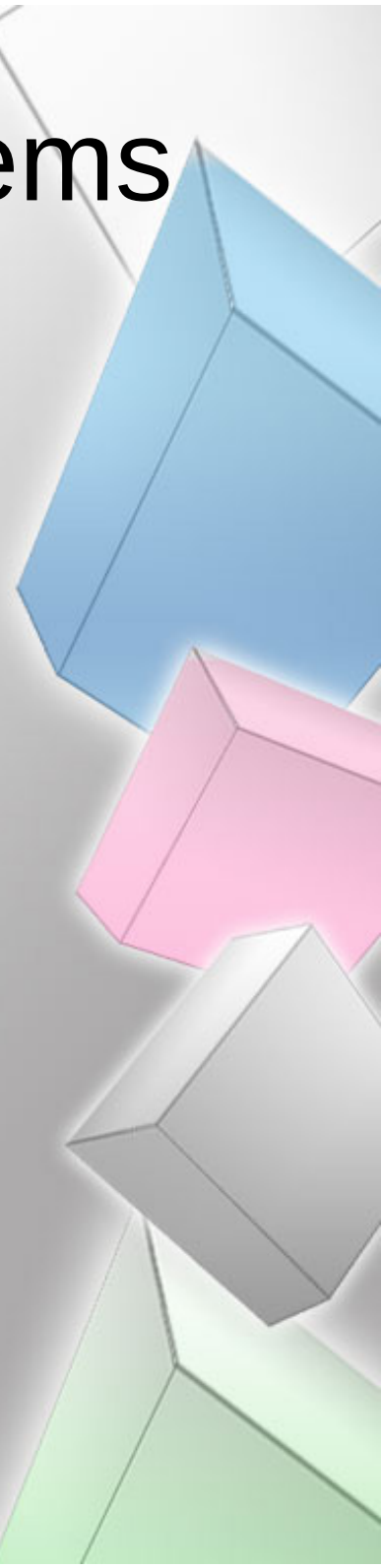
Lesson 12

01: Introduction
02: Architectures
03: Processes
04: Communication
05: Naming
06: Synchronization
07: Consistency & Replication
08: Fault Tolerance
09: Security
10: Distributed Object-Based Systems
11: Distributed File Systems
12: Distributed Web-Based Systems
13: Distributed Coordination-Based Systems



Distributed Web-Based Systems

- Architectures
- Processes
- Communication
- Naming
- Synchronization
- Consistency and Replication
- Fault Tolerance
- Security



Traditional Web-Based Systems

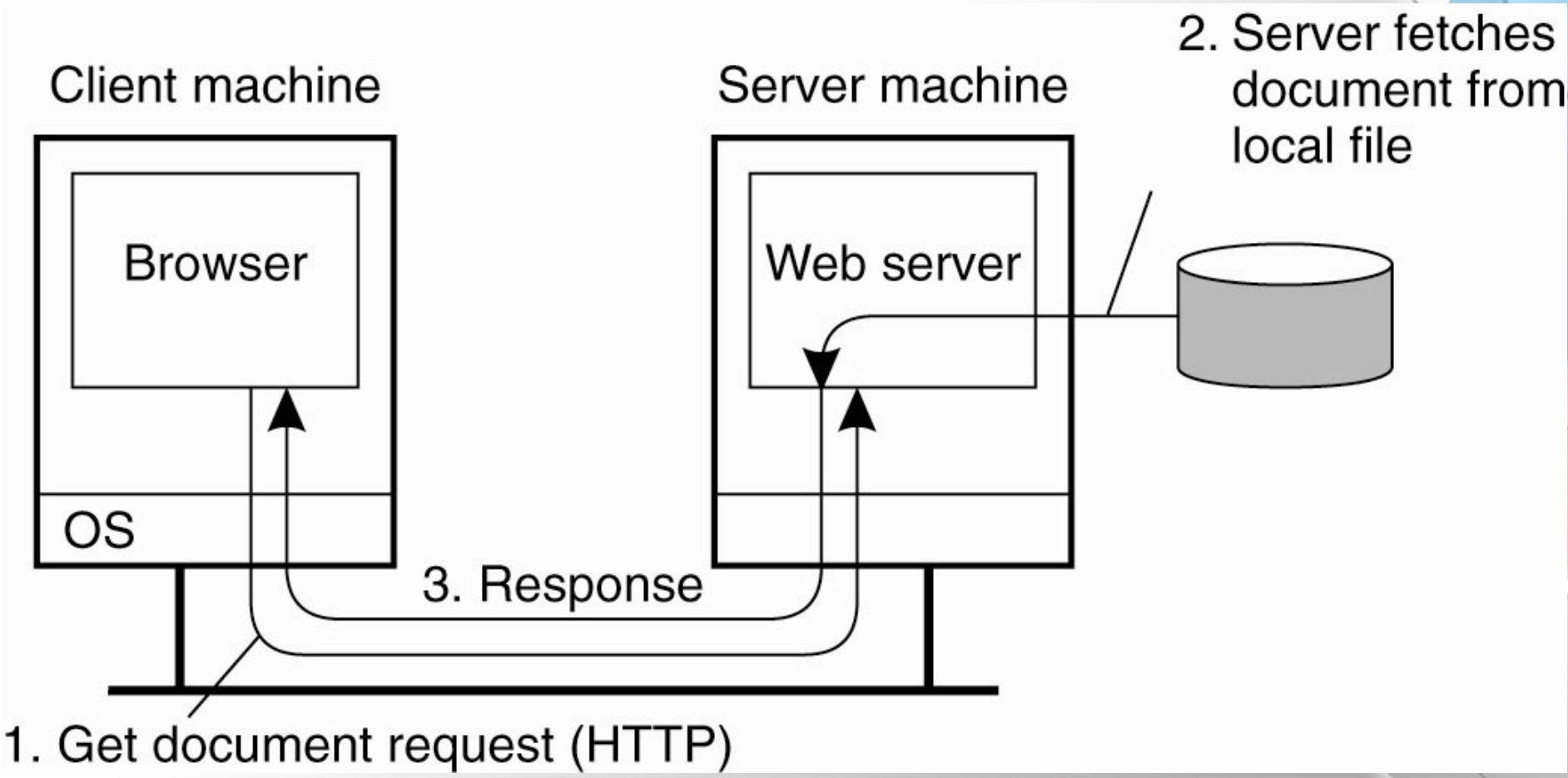


Figure 12-1. The overall organization of a traditional Web site. See the Web as a huge distributed system where services rather than just documents are being offered.

Web Documents

Type	Subtype	Description
Text	Plain	Unformatted text
	HTML	Text including HTML markup commands
	XML	Text including XML markup commands
Image	GIF	Still image in GIF format
	JPEG	Still image in JPEG format
Audio	Basic	Audio, 8-bit PCM sampled at 8000 Hz
	Tone	A specific audible tone
Video	MPEG	Movie in MPEG format
	Pointer	Representation of a pointer device for presentations
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in Postscript
	PDF	A printable document in PDF
Multipart	Mixed	Independent parts in the specified order
	Parallel	Parts must be viewed simultaneously

Figure 12-2. Six top-level Multipurpose Internet Mail Exchange (MIME) types and some common subtypes.

Multitiered Architectures
CGI: Common Gateway Interface

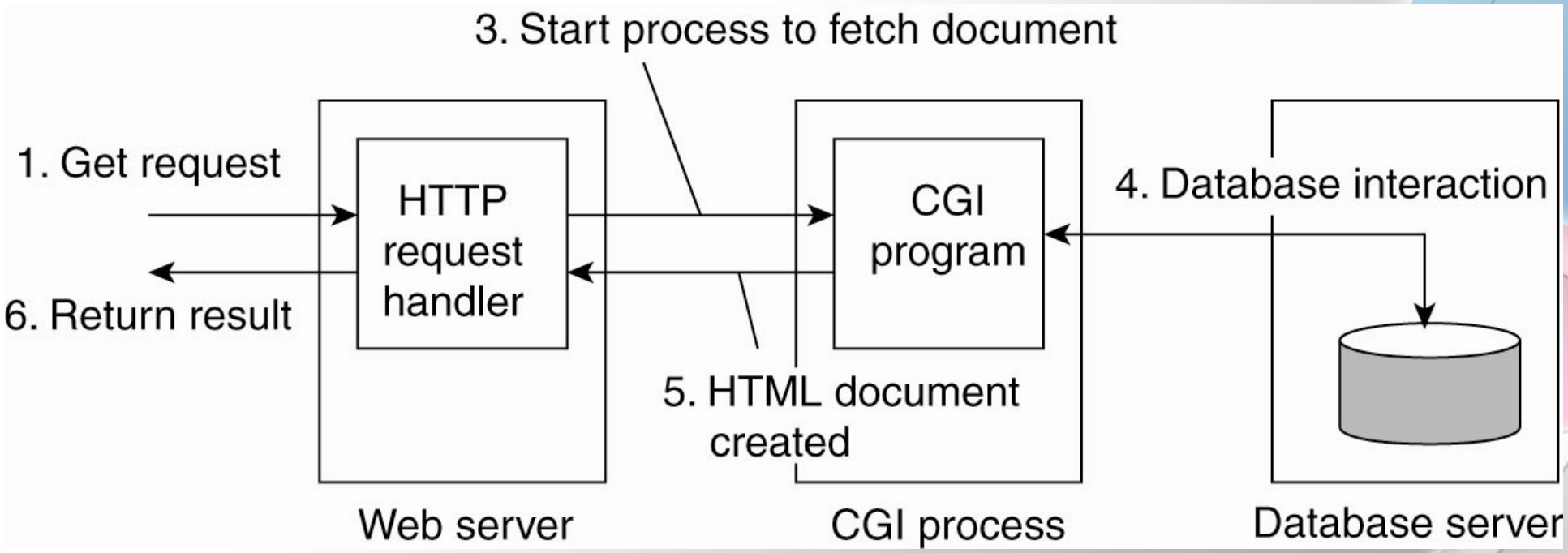


Figure 12-3. The principle of using server-side CGI programs.

Server-side scripts

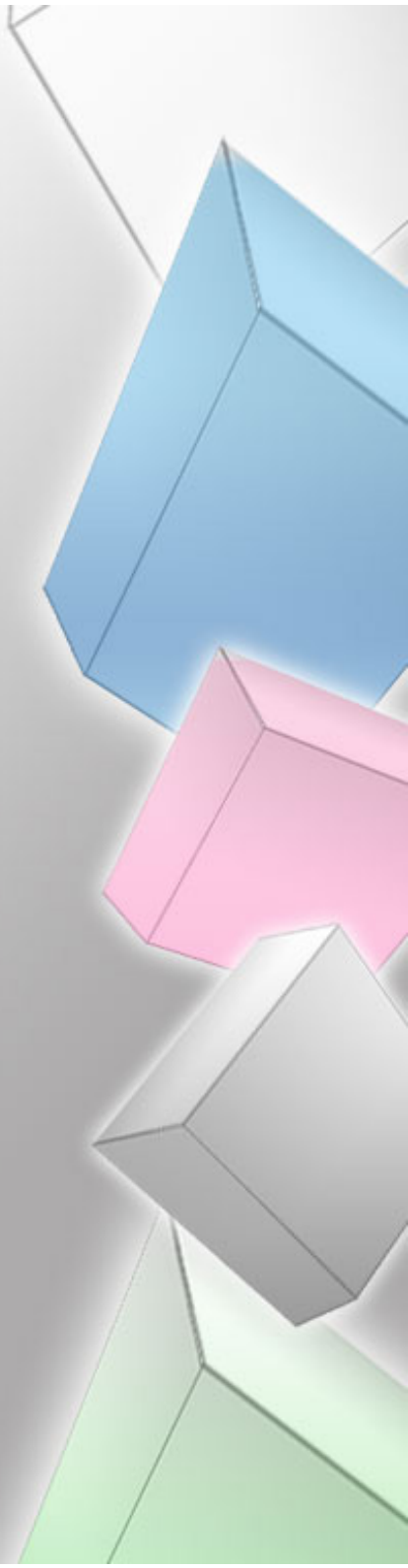
- Servers nowadays do much more than just fetching documents.
- One of the most important enhancements is that servers can also process a document before passing it to the client.
- In particular, a document may contain a server-side script, which is executed by the server when the document has been fetched locally.
- The result of executing a script is sent along with the rest of the document to the client.
- The script itself is not sent. In other words, using a server-side script changes a document by essentially replacing the script with the results of its execution.

Three-tiers and servlets

- As server-side processing of Web documents increasingly requires more flexibility, it should come as no surprise that many Web sites are now organized as a three-tiered architecture consisting of a Web server, an application server, and a database.
- Application server runs all kinds of programs that may or may not access the third tier. consisting of a database.
  - For example, a server may accept a customer's query, search its database of matching products, and then construct a clickable Web page listing the products found.
- In many cases the server is responsible for running Java programs, called servlets, that maintain things like shopping carts, implement recommendations, keep lists of favorite items, and so on.

Practical Session

- Apache Tomcat
- JSPs
- Servlets



Web Services Fundamentals

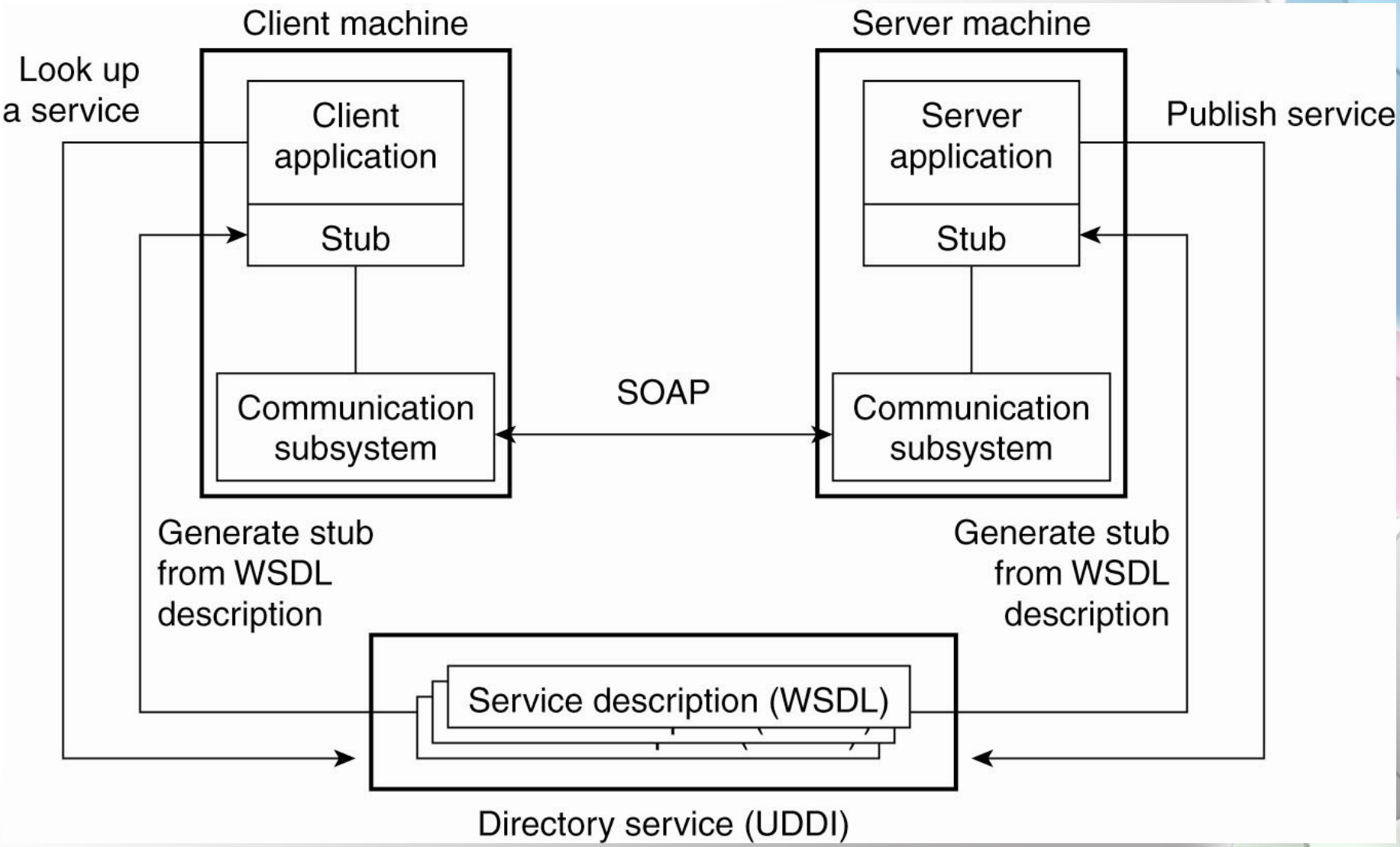


Figure 12-4. The principle of a Web service.

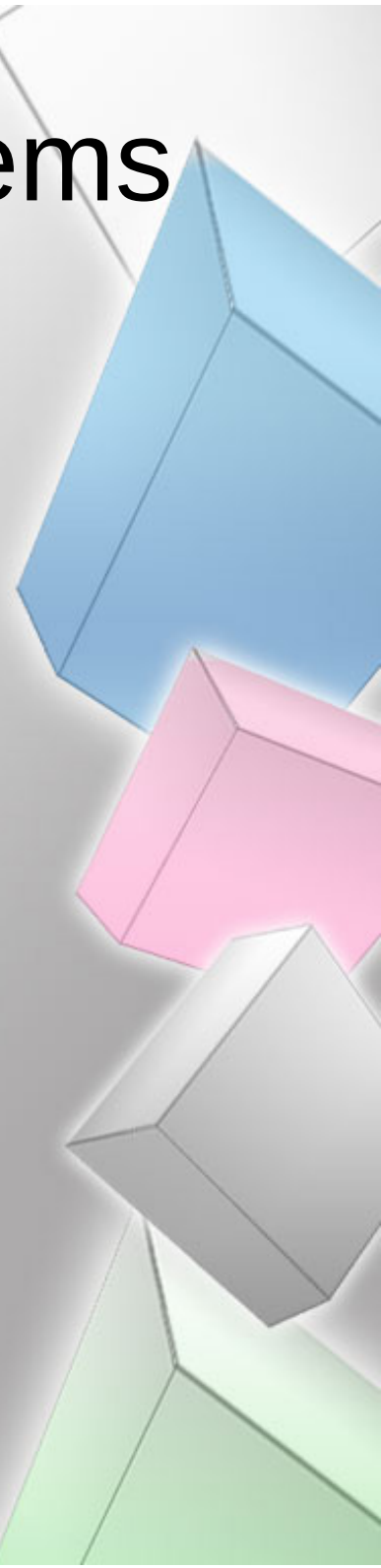
Practical Session: web services

- JAX-WS (Java API for XML Web Services)
- JAX-WS is a technology for building web services and clients that communicate using XML. JAX-WS allows developers to write message-oriented as well as RPC-oriented web services.
  - HelloService
    - Test the WS without a client
  - SimpleClient
    - Test the Web Service with the client

- Other
  - ASP.NET

Distributed Web-Based Systems

- Architectures
- Processes
- Communication
- Naming
- Synchronization
- Consistency and Replication
- Fault Tolerance
- Security



Processes – Clients (1) + Plug-ins

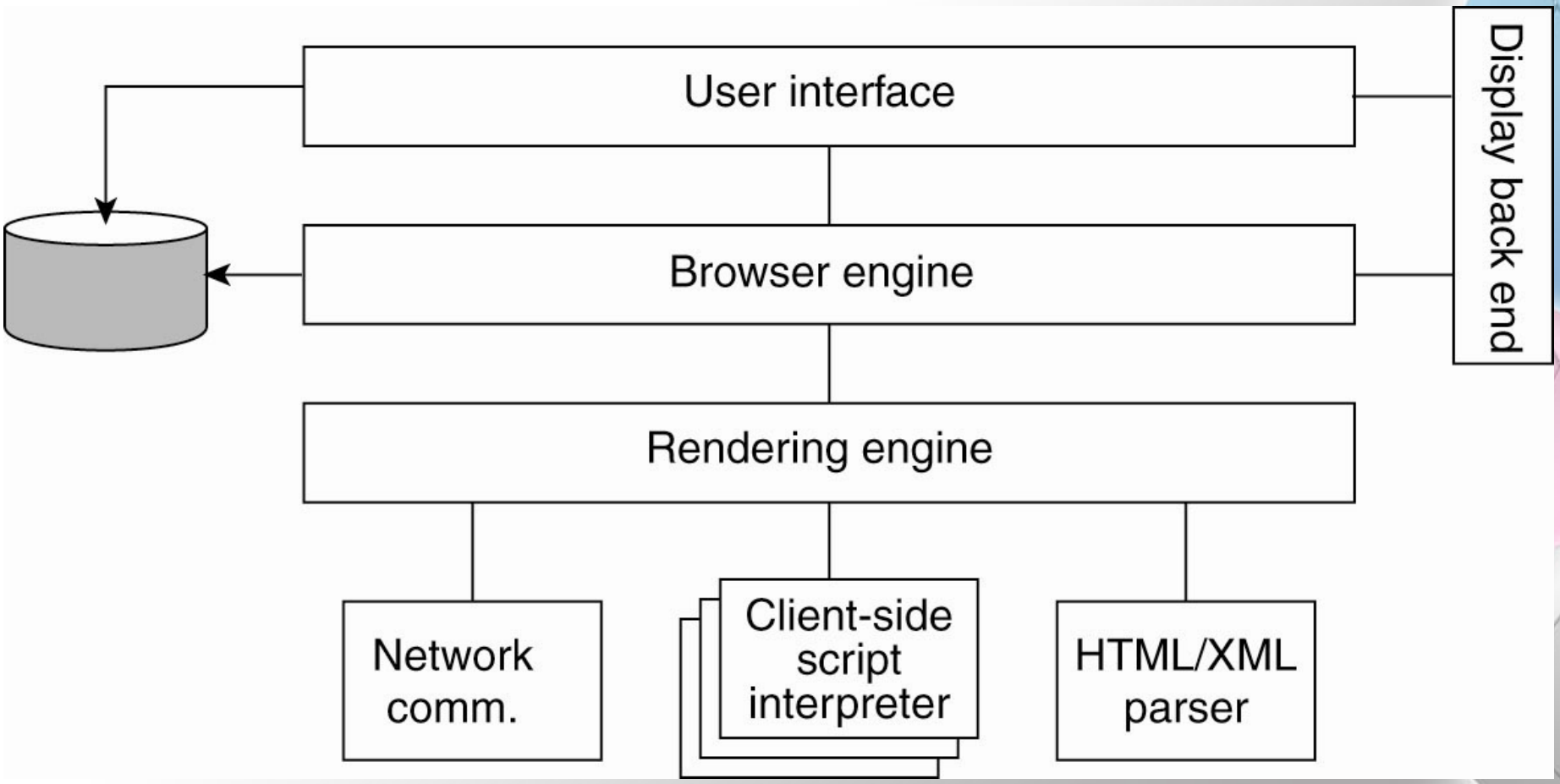


Figure 12-5. The logical components of a Web browser.

Processes – Clients (2)

Figure 12-6. Using a Web proxy when the browser does not speak FTP.

SQUID
http://www.squid-cache.org

The Apache Web Server

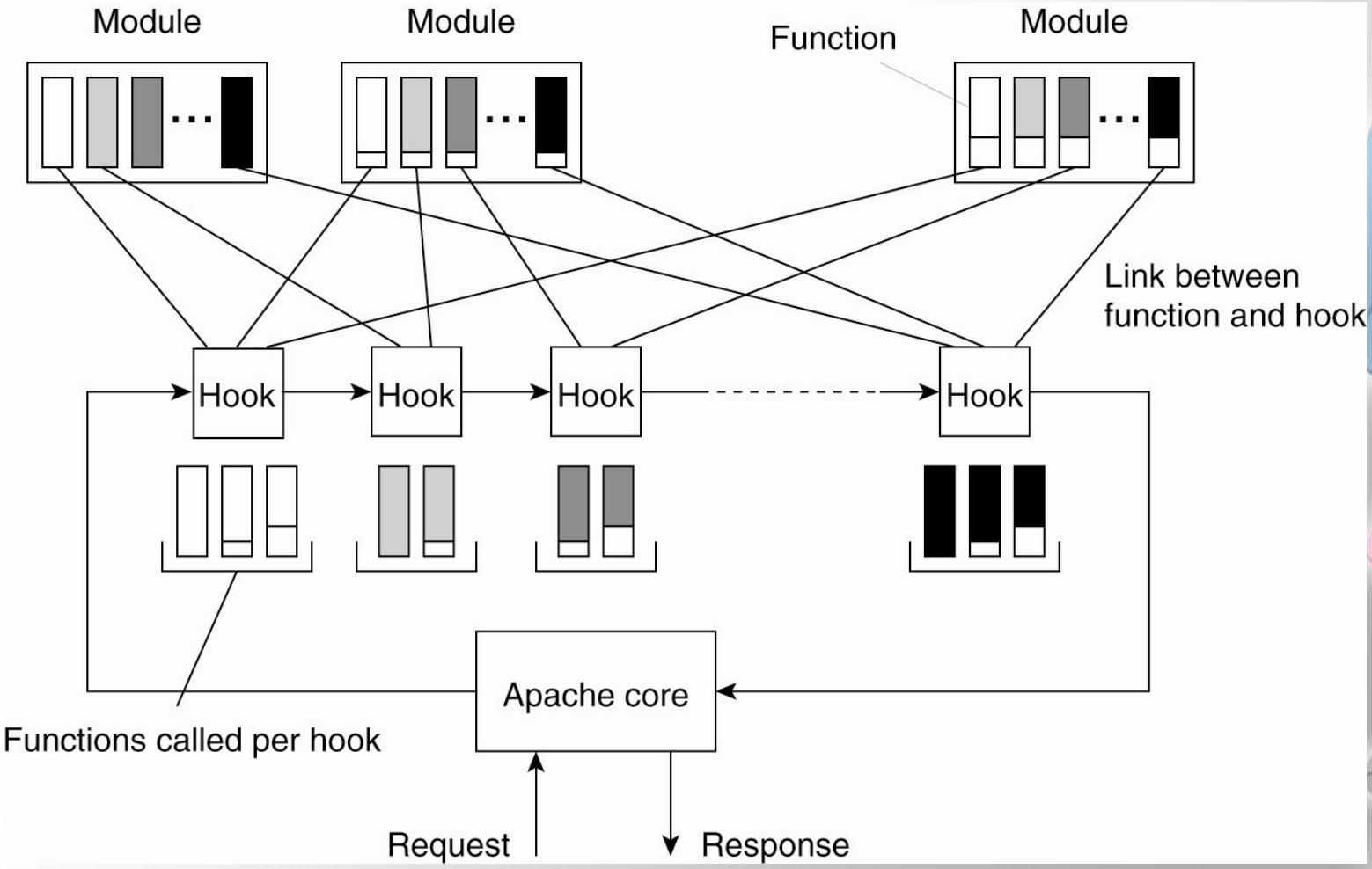


Figure 12-7. The general organization of the Apache Web server.

Web Server Clusters (1)

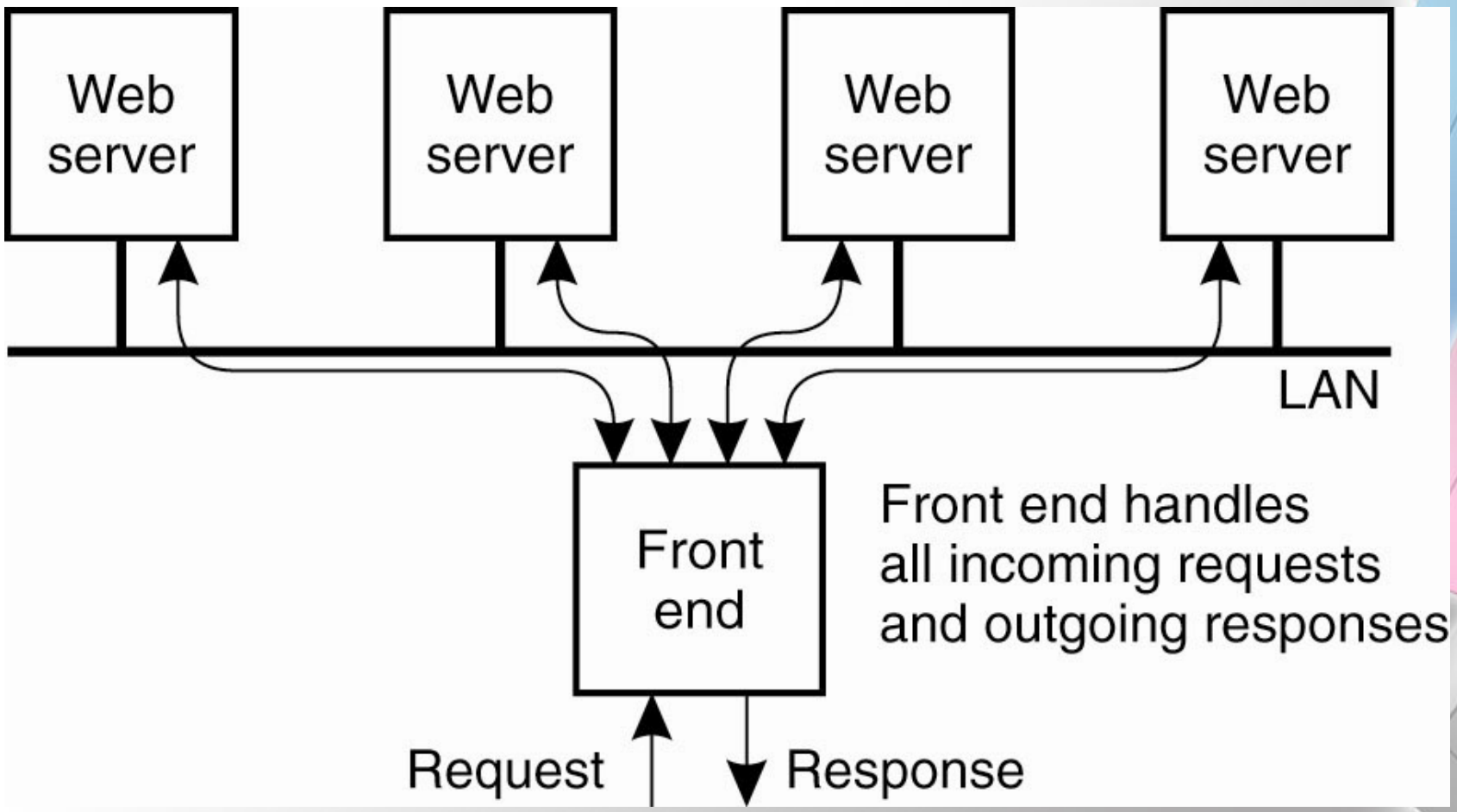


Figure 12-8. The principle of using a server cluster in combination with a front end to implement a Web service.

Web Server Clusters (2)

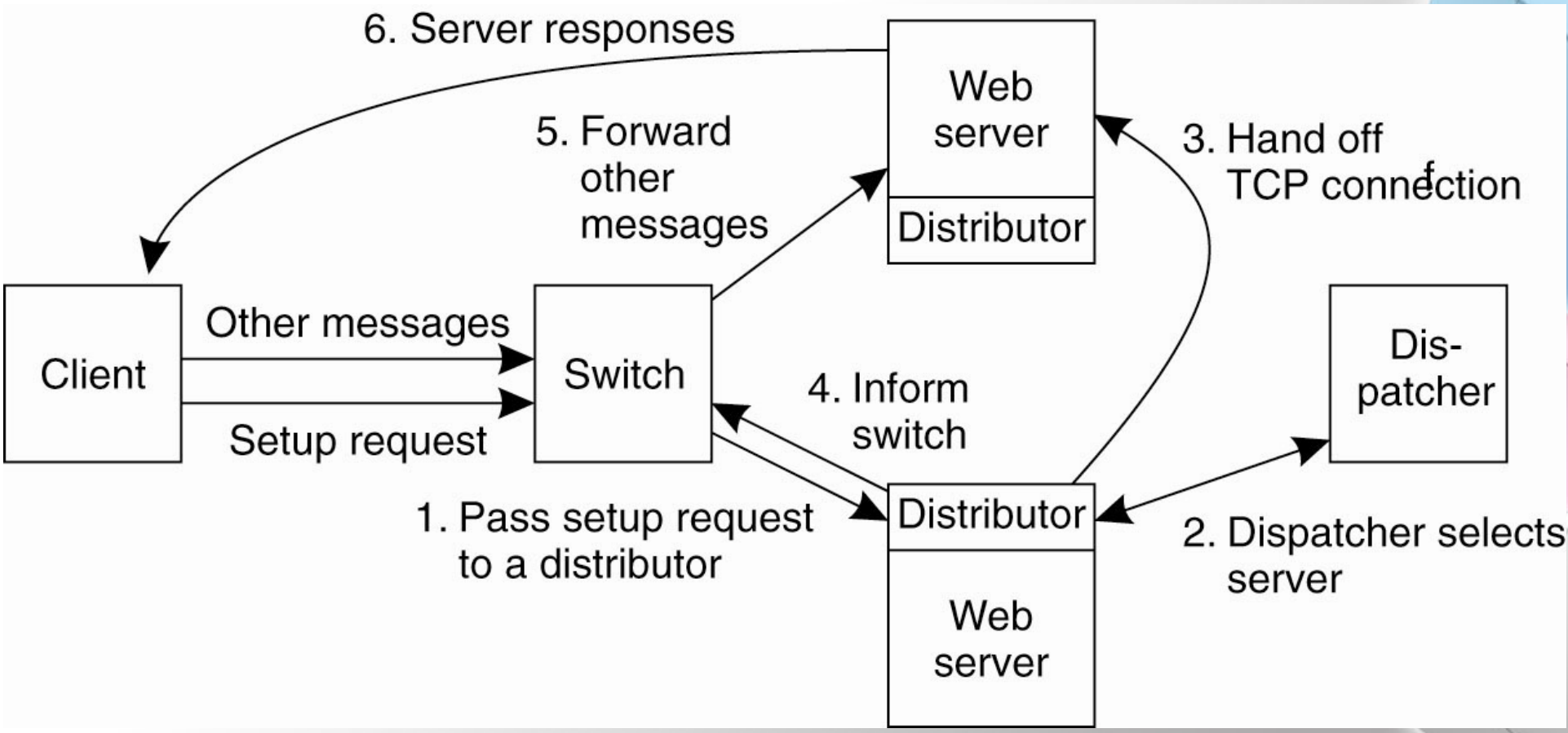
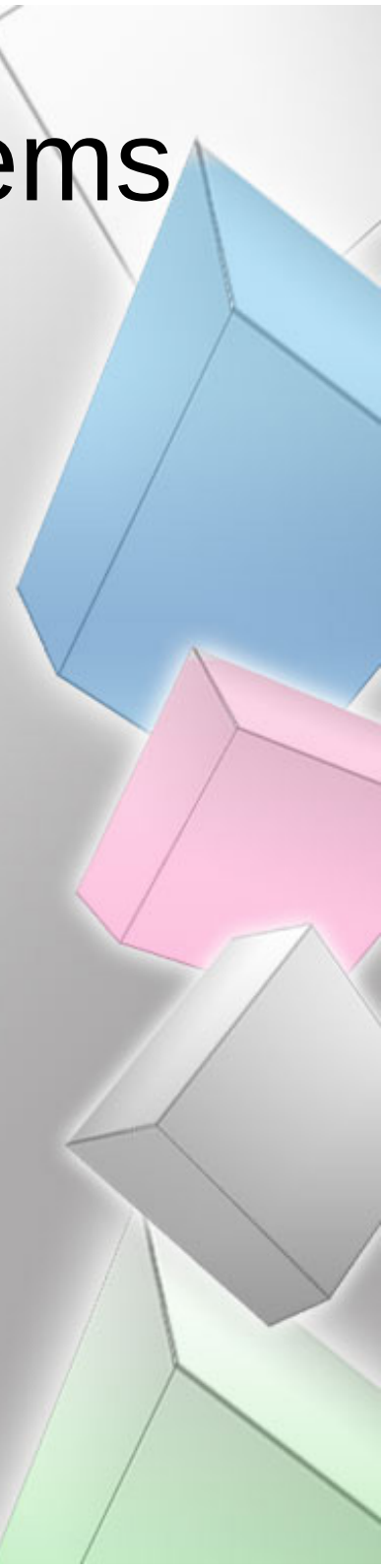


Figure 12-9. A scalable content-aware cluster of Web servers.

Distributed Web-Based Systems

- Architectures
- Processes
- Communication
- Naming
- Synchronization
- Consistency and Replication
- Fault Tolerance
- Security



HTTP Connections (1)

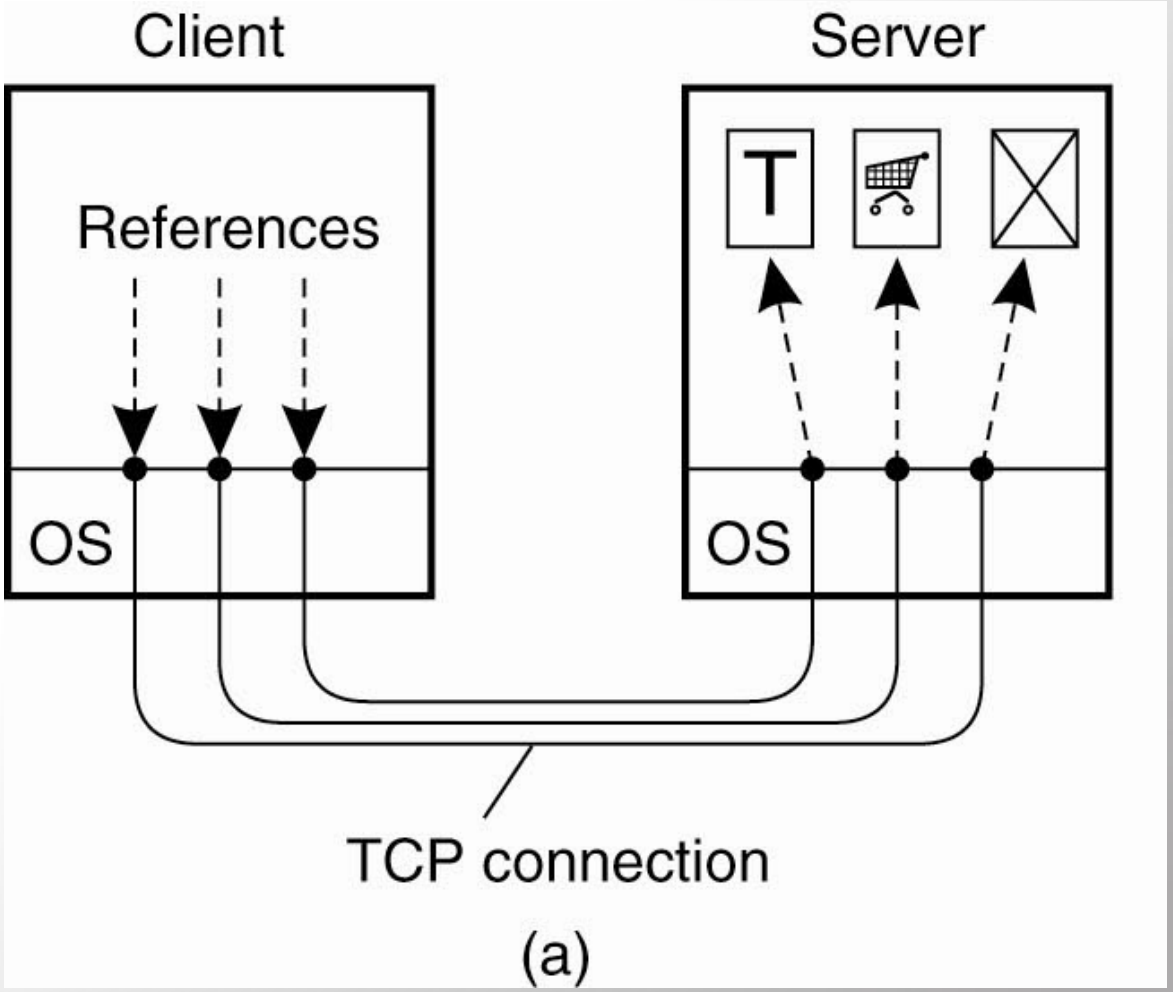
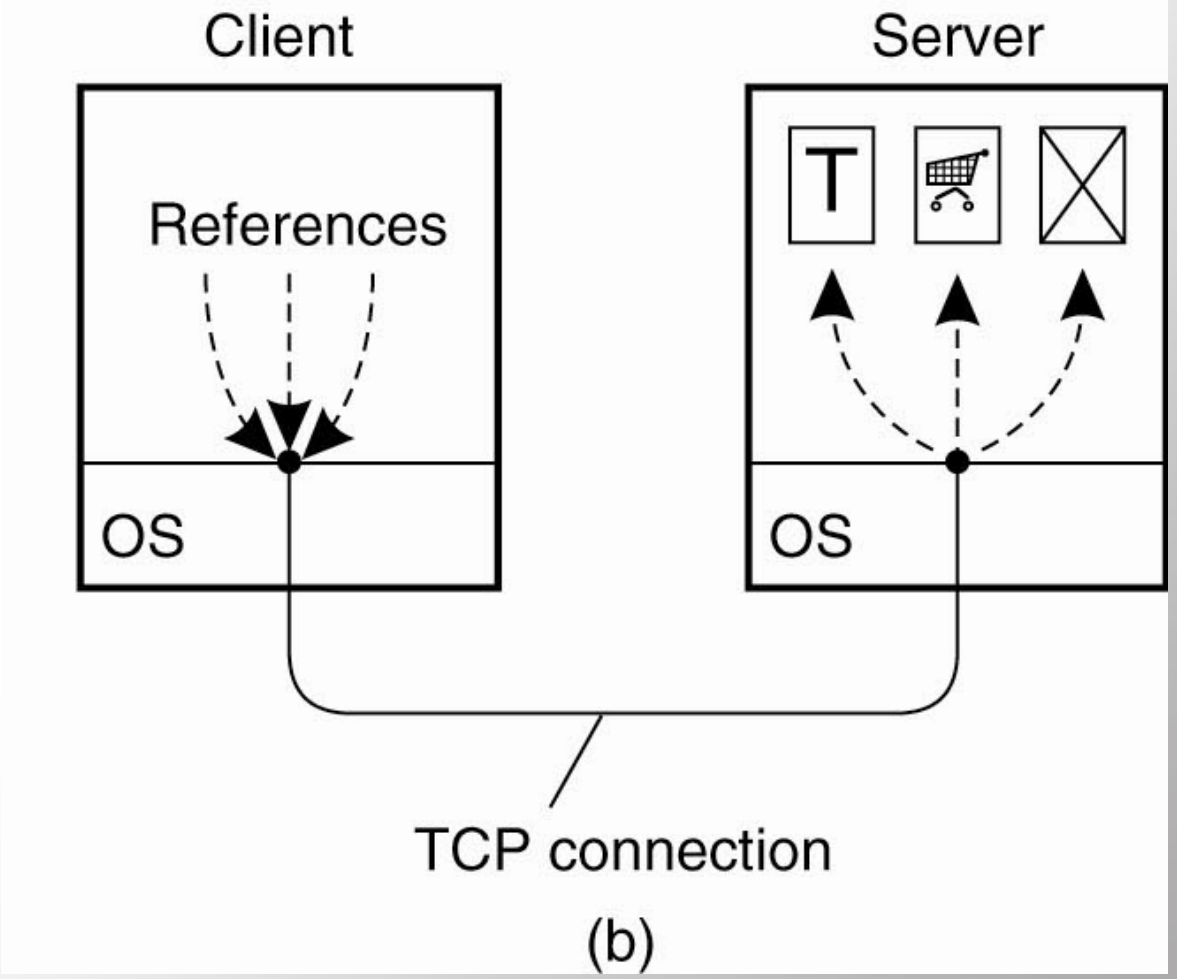


Figure 12-10. (a) Using nonpersistent connections.

HTTP Connections (2)



- Figure 12-10. (b) Using persistent connections.

HTTP Methods

Operation	Description
Head	Request to return the header of a document
Get	Request to return a document to the client
Put	Request to store a document
Post	Provide data that are to be added to a document (collection)
Delete	Request to delete a document

Figure 12-11. Operations supported by HTTP.

HTTP Messages (1)

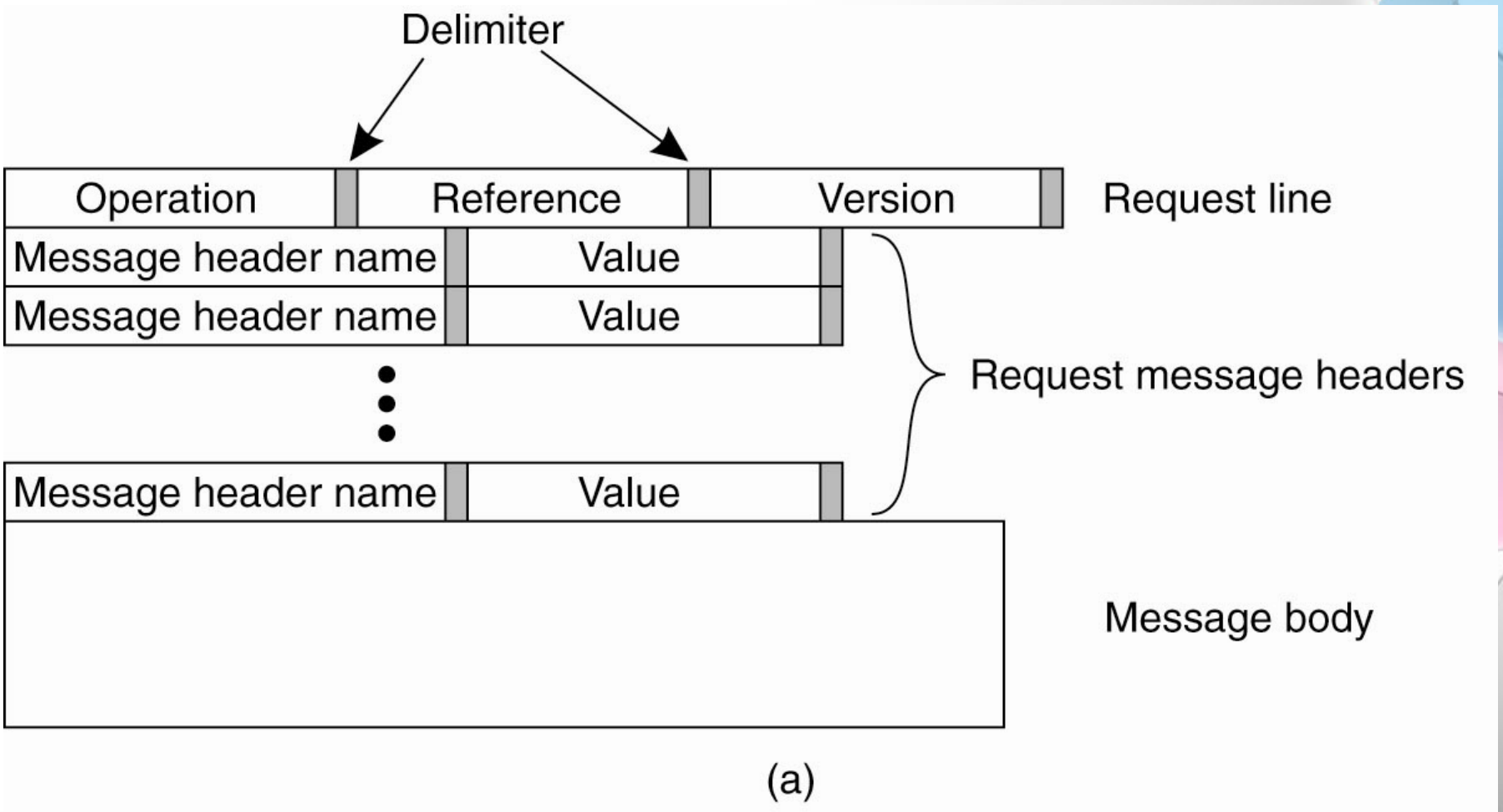


Figure 12-12. (a) HTTP request message.

HTTP Messages (2)

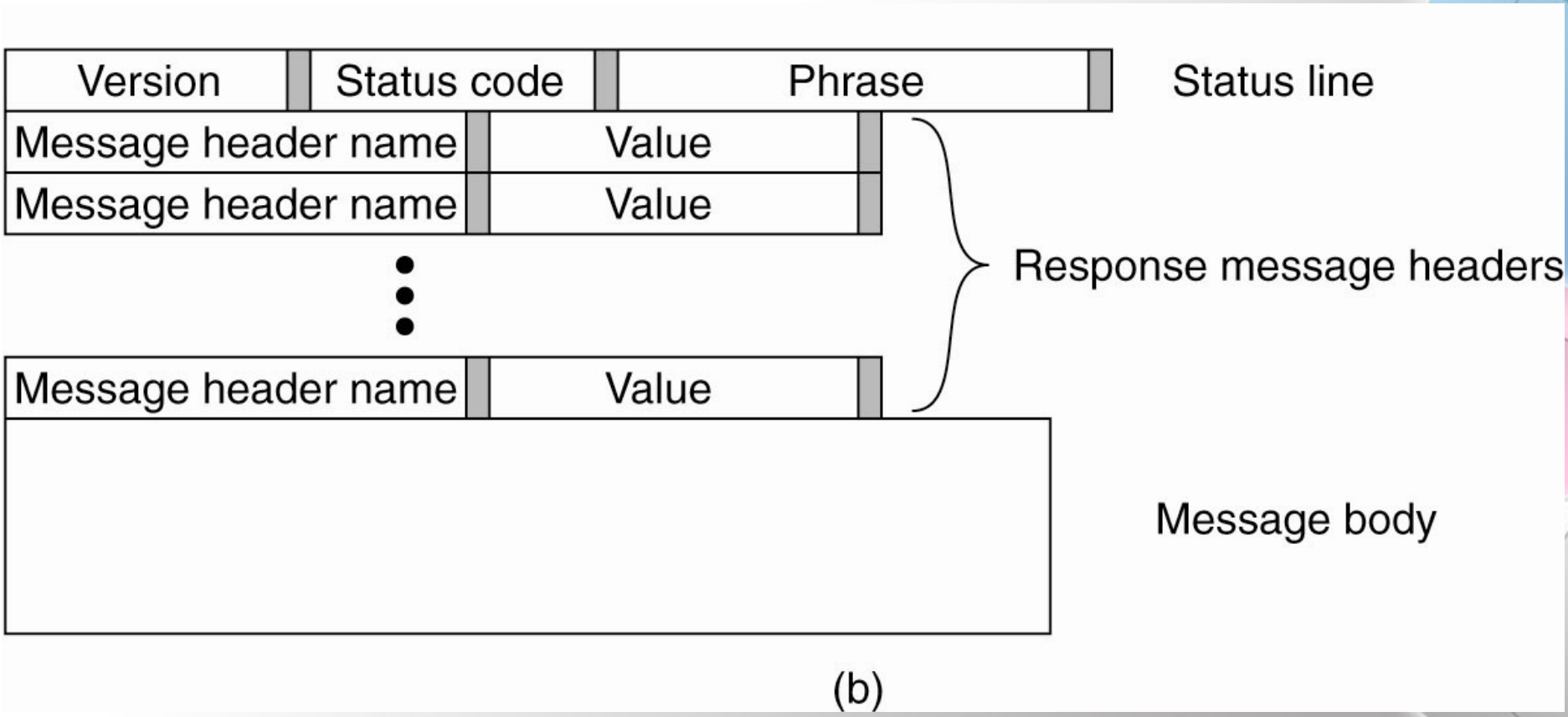


Figure 12-12. (b) HTTP response message.

HTTP Messages (3)

Header	Source	Contents
Accept	Client	The type of documents the client can handle
Accept-Charset	Client	The character sets are acceptable for the client
Accept-Encoding	Client	The document encodings the client can handle
Accept-Language	Client	The natural language the client can handle
Authorization	Client	A list of the client's credentials
WWW-Authenticate	Server	Security challenge the client should respond to
Date	Both	Date and time the message was sent
ETag	Server	The tags associated with the returned document
Expires	Server	The time for how long the response remains valid
From	Client	The client's e-mail address
Host	Client	The DNS name of the document's server

Figure 12-13. Some HTTP message headers.

HTTP Messages (4)

Header	Source	Contents
If-Match	Client	The tags the document should have
If-None-Match	Client	The tags the document should not have
If-Modified-Since	Client	Tells the server to return a document only if it has been modified since the specified time
If-Unmodified-Since	Client	Tells the server to return a document only if it has not been modified since the specified time
Last-Modified	Server	The time the returned document was last modified
Location	Server	A document reference to which the client should redirect its request
Referer	Client	Refers to client's most recently requested document
Upgrade	Both	The application protocol the sender wants to switch to
Warning	Both	Information about the status of the data in the message

Figure 12-13. Some HTTP message headers.

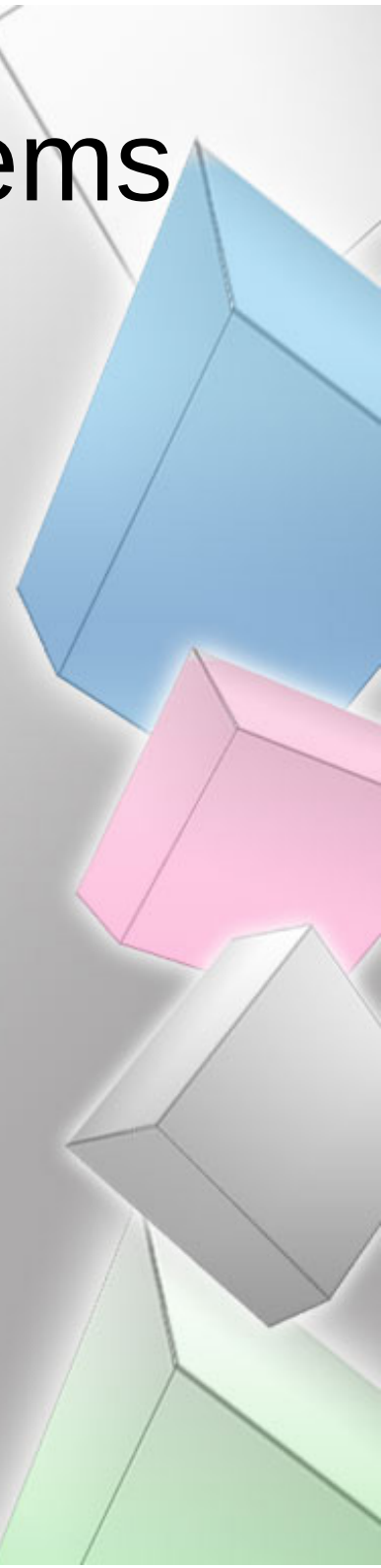
Simple Object Access Protocol

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <env:Header>
        <n:alertcontrol xmlns:n="http://example.org/alertcontrol">
            <n:priority>1</n:priority>
            <n:expires>2001-06-22T14:00:00-05:00</n:expires>
        </n:alertcontrol>
    </env:Header>
    <env:Body>
        <m:alert xmlns:m="http://example.org/alert">
            <m:msg>Pick up Mary at school at 2pm</m:msg>
        </m:alert>
    </env:Body>
</env:Envelope>
```

Figure 12-14. An example of an XML-based SOAP message.

Distributed Web-Based Systems

- Architectures
- Processes
- Communication
- Naming
- Synchronization
- Consistency and Replication
- Fault Tolerance
- Security



Naming (1)

Scheme	Host name	Pathname
http ://	www.cs.vu.nl	/home/steen/mbox

(a)

Scheme	Host name	Port	Pathname
http ://	www.cs.vu.nl :	80	/home/steen/mbox

(b)

Scheme	Host name	Port	Pathname
http ://	130.37.24.11 :	80	/home/steen/mbox

(c)

- Figure 12-15. Often-used structures for URLs. (a) Using only a DNS name. (b) Combining a DNS name with a port number. (c) Combining an IP address with a port number.

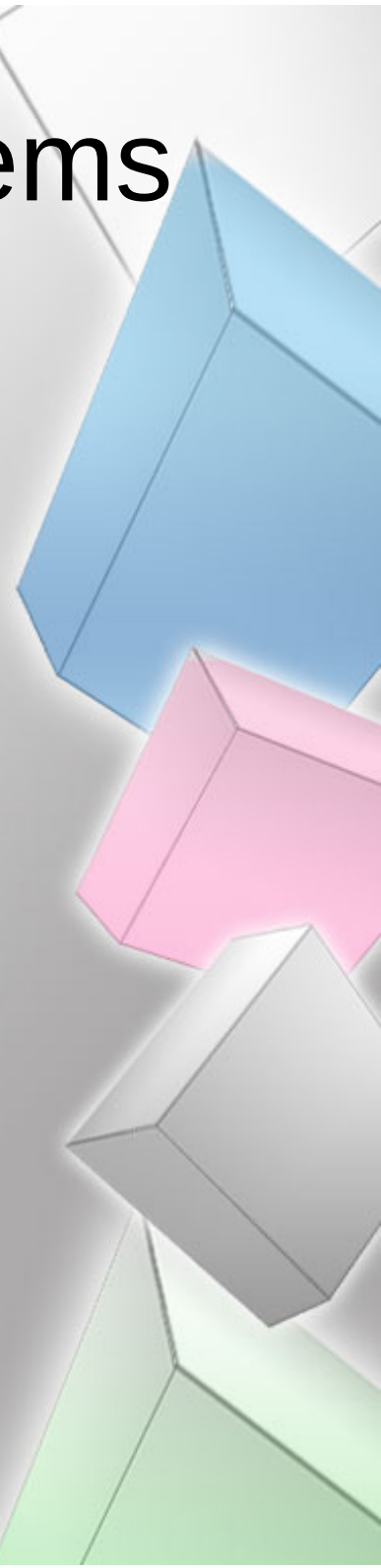
Naming (2)

Name	Used for	Example
http	HTTP	http://www.cs.vu.nl:80/globe
mailto	E-mail	mailto:steen@cs.vu.nl
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:/edu/book/work/chp/11/11
data	Inline data	data:text/plain;charset=iso-8859-7,%e1%e2%e3
telnet	Remote login	telnet://flits.cs.vu.nl
tel	Telephone	tel:+31201234567
modem	Modem	modem:+31201234567;type=v32

Figure 12-16. Examples of URIs.

Distributed Web-Based Systems

- Architectures
- Processes
- Communication
- Naming
- Synchronization
- Consistency and Replication
- Fault Tolerance
- Security

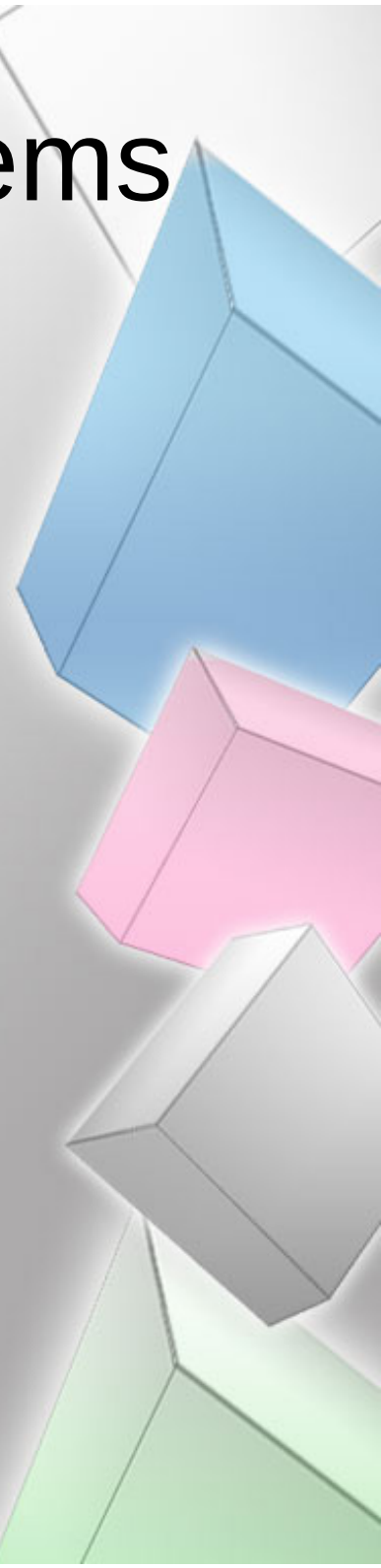


Synchronization

- Distributed authoring of Web documents is handled through a separate protocol, **Web Distributed Authoring and Versioning (WebDAV).**
  - Provides a simple means to lock a shared document, and to create, delete, copy, and move documents from remote Web servers.
- To synchronize concurrent access to a shared document, WebDAV supports a simple **locking mechanism**. There are two types of write locks.
  - An exclusive write lock can be assigned to a single client, and will prevent any other client from modifying the shared document while it is locked.
  - A shared write lock, which allows multiple clients to simultaneously update the document.
- Because locking takes place at the granularity of an entire document, shared write locks are convenient when clients modify different parts of the same document.

Distributed Web-Based Systems

- Architectures
- Processes
- Communication
- Naming
- Synchronization
- Consistency and Replication
- Fault Tolerance
- Security



Web Proxy Caching

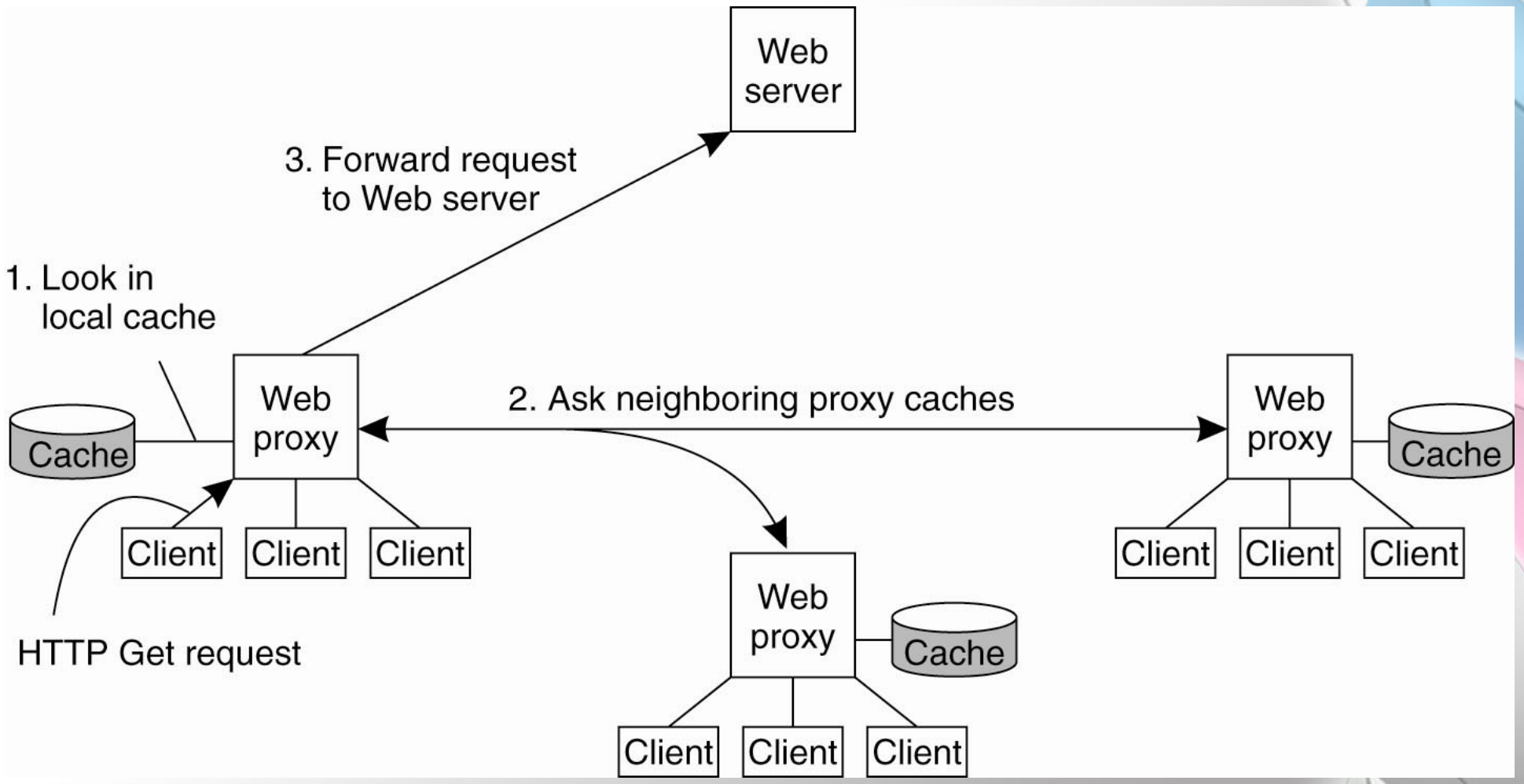


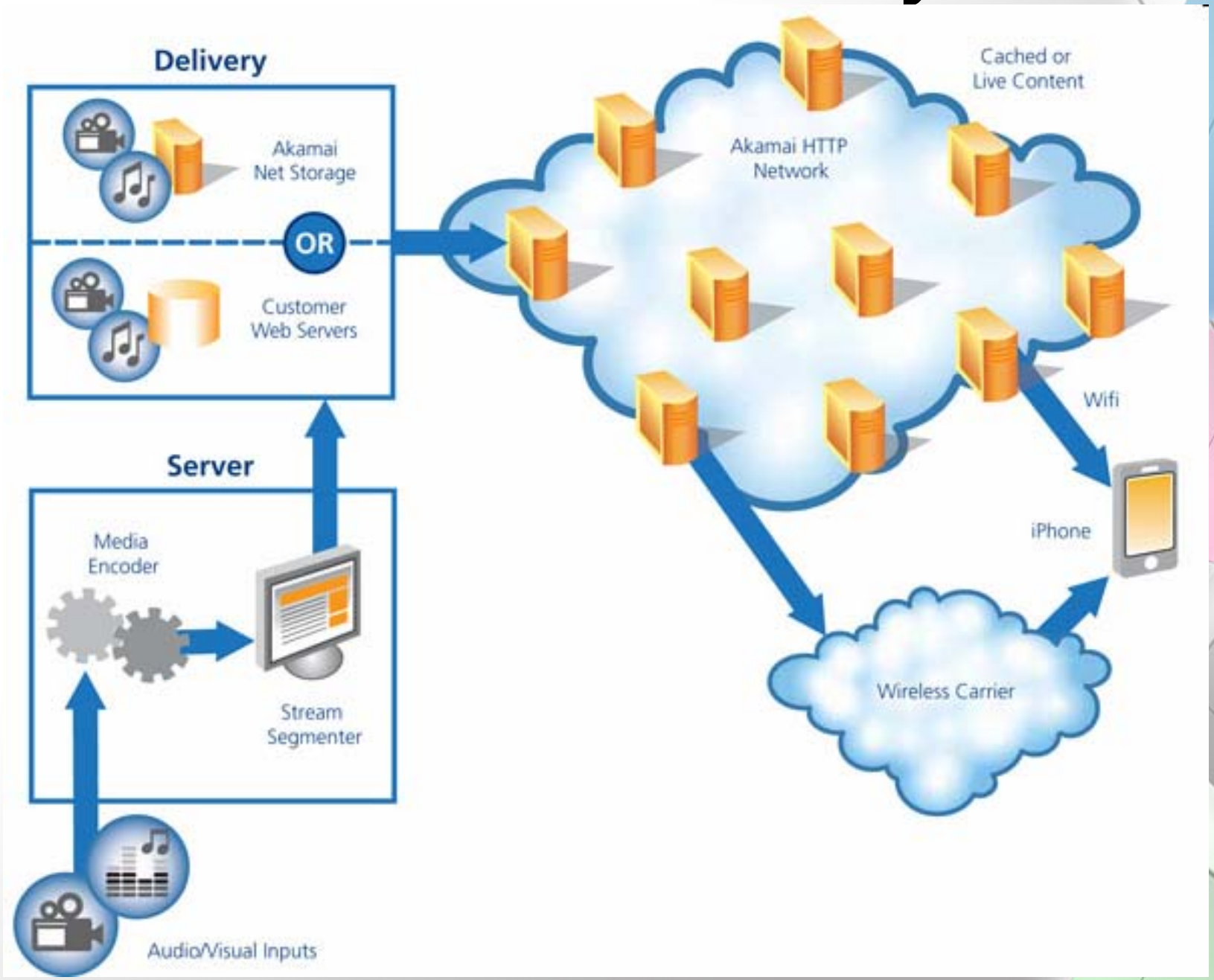
Figure 12-17. The principle of cooperative caching.

Dimensions: what are we talking about?

- Ten years since its creation, distributed computing platform Akamai now delivers 20 per cent of the world's internet traffic by request.
- Whether you've heard of Akamai or not, you've already used their products. Akamai owns and operates the world's largest distributed computing platform. On 40,000 servers in 70 countries, it transparently mirrors content from its customer's servers. Sometimes this content includes media, and sometimes a customer's whole site. (Dec 4, 2008)

- AKAMAI: founded by Prof. Tom Leighton and his student Daniel Lewin in 1998 at M.I.T.

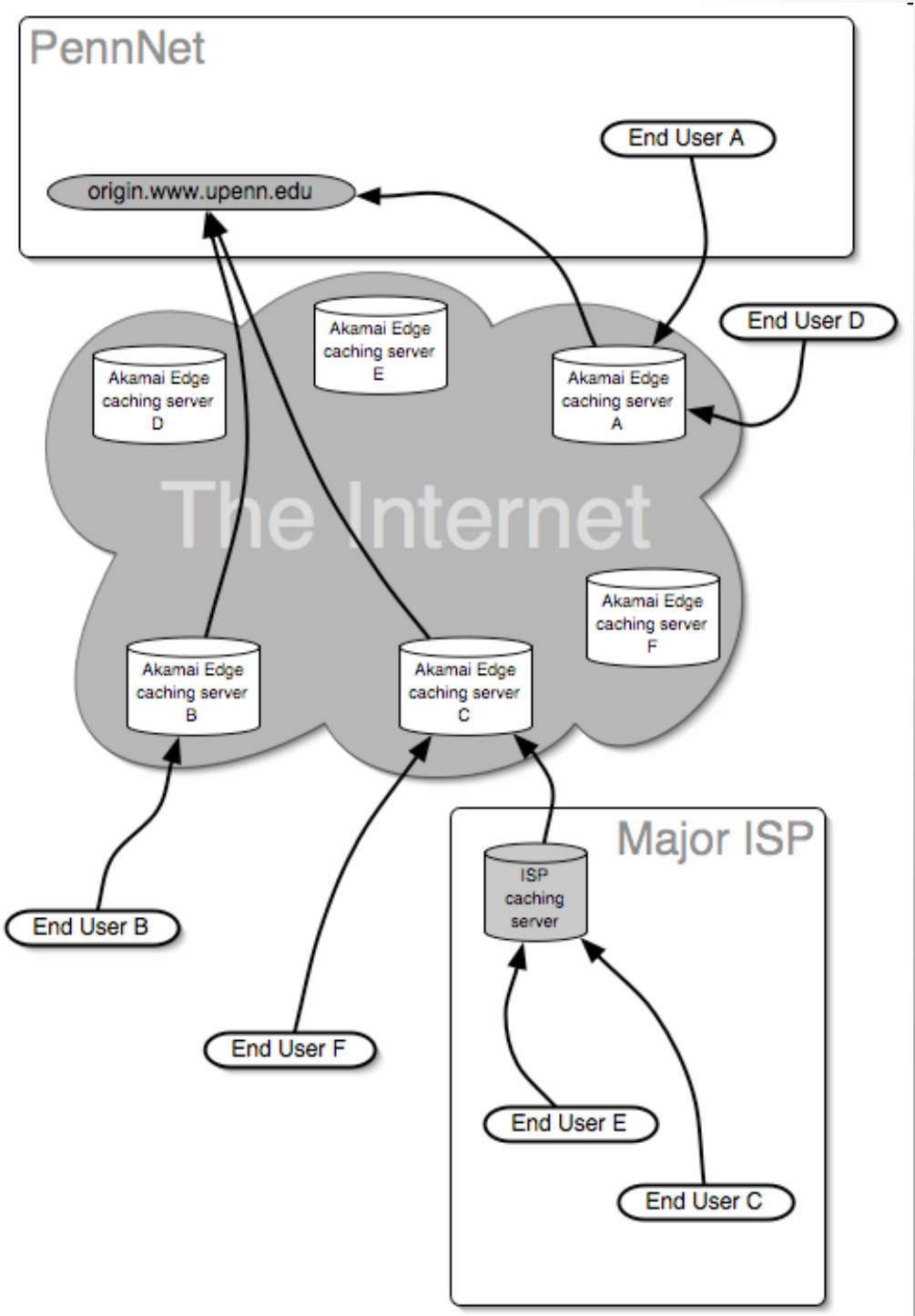
AKAMAI: Delivery



Replicated web application

Today Akamai handles tens of billions of daily Web interactions for companies like Audi, NBC, and Fujitsu, and organizations like the U.S. Department of Defense and NASDAQ -- powering brand new business models that serve the changing online economy. (From AKAMAI Website)

AKAMAI Caching



Replication for Web Hosting Systems

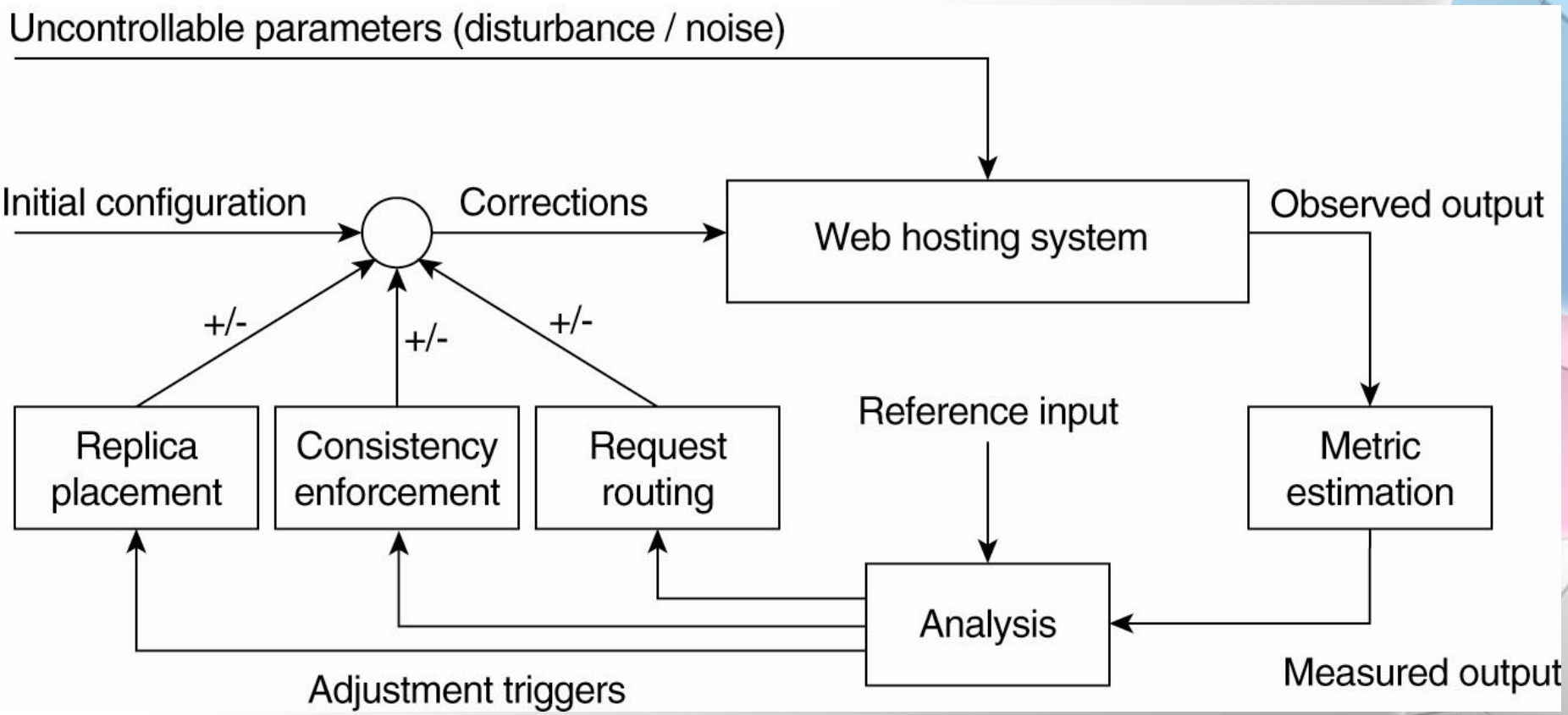


Figure 12-18. The general organization of a CDN as a feedback-control system (adapted from Sivasubramanian et al., 2004b).

Metric Estimation

- Latency metrics
  - Time for fetching a document
- Spatial metrics
  - Distance in terms of network-level routing hops
- Network usage metrics
  - Consumed bandwidth
- Consistency metrics
  - To what extent the replica is deviating from its master copy
- Financial metrics
  - How well a CDN is doing in terms of ROI (return of investment)

Adaptation Triggering

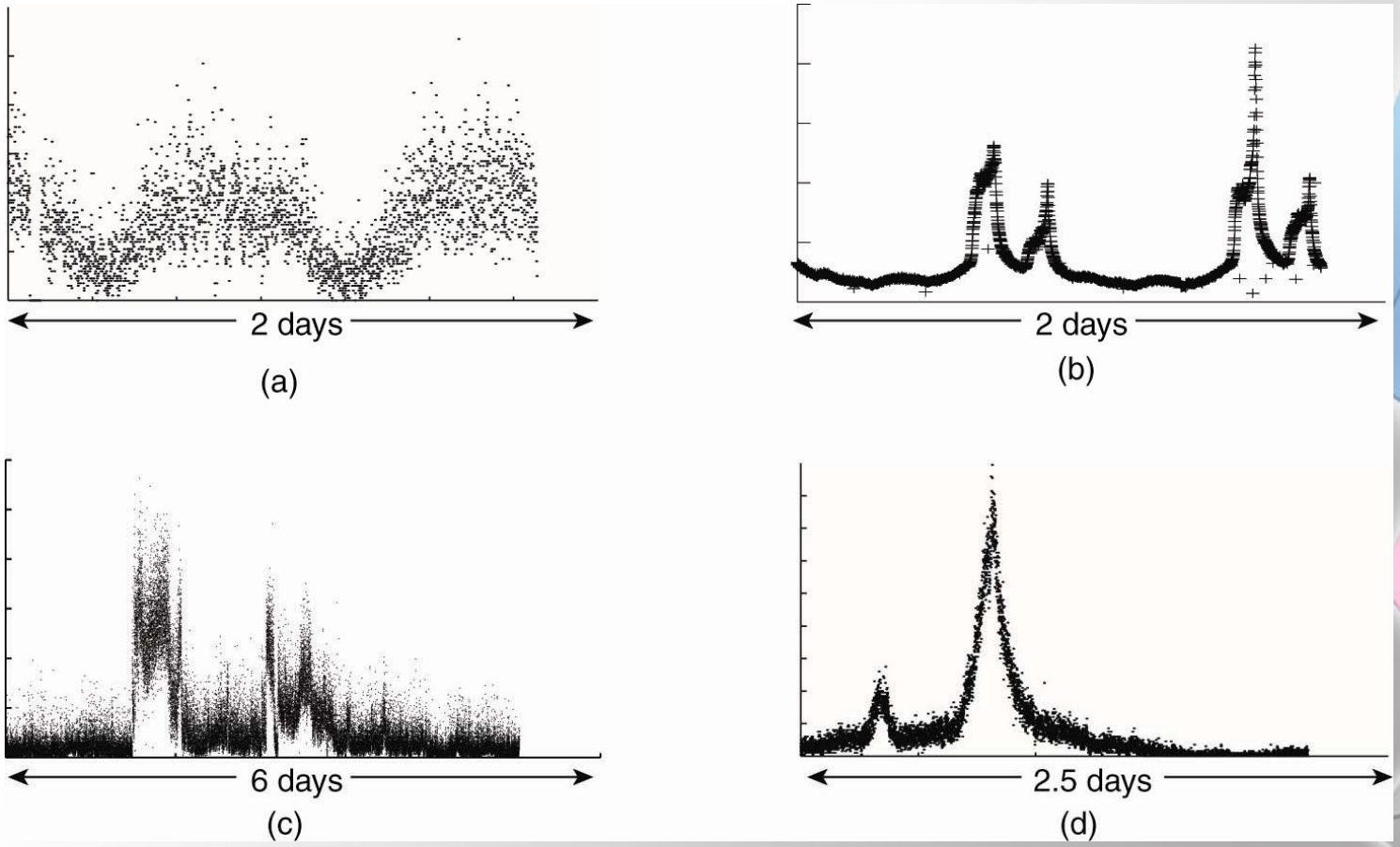


Figure 12-19. One normal and three different access patterns reflecting flash-crowd behavior (adapted from Baryshnikov et al., 2005).

Embedded documents

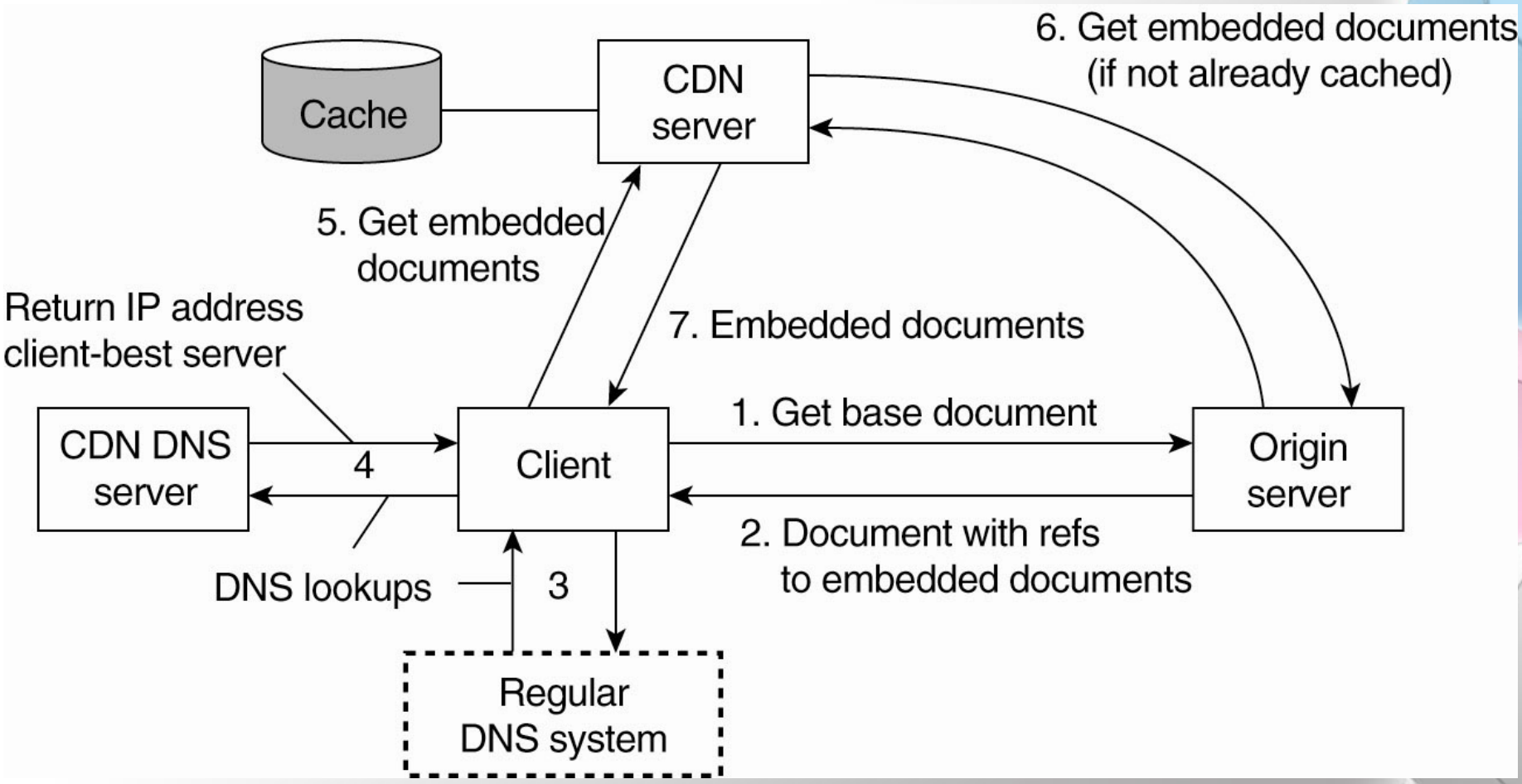


Figure 12-20. The principal working of the Akamai CDN.

Replication of Web Applications

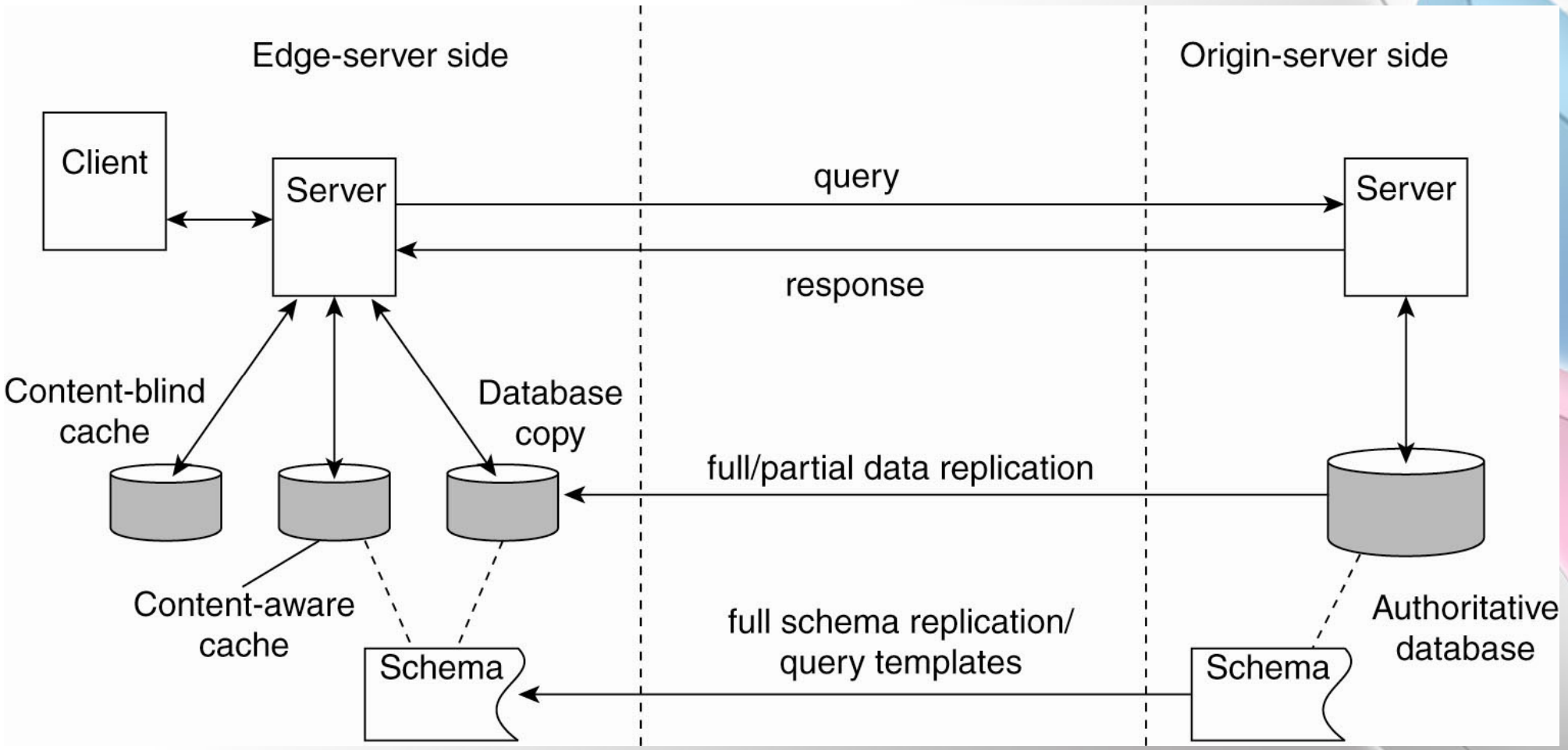
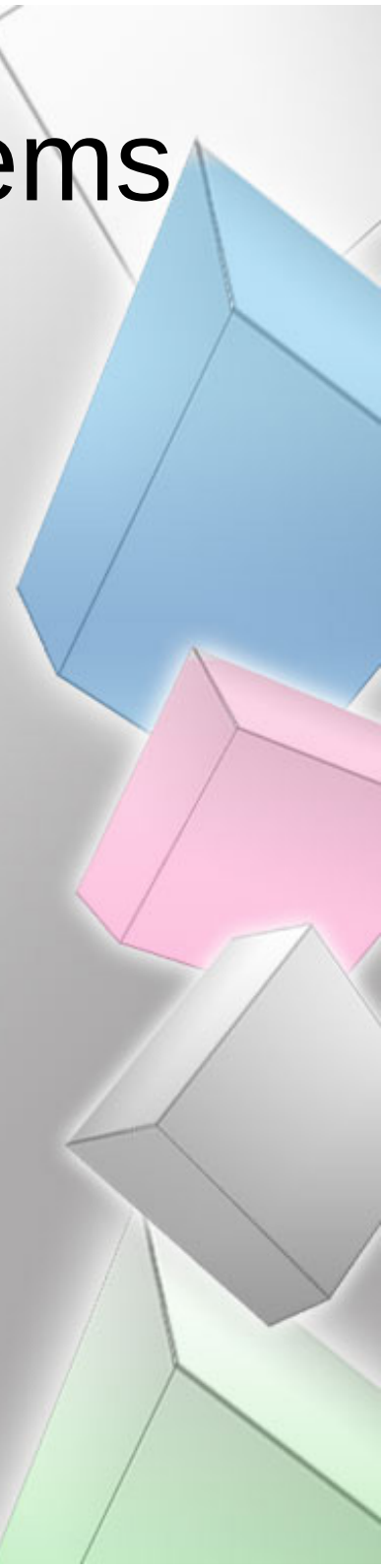


Figure 12-21. Alternatives for caching and replication with Web applications.

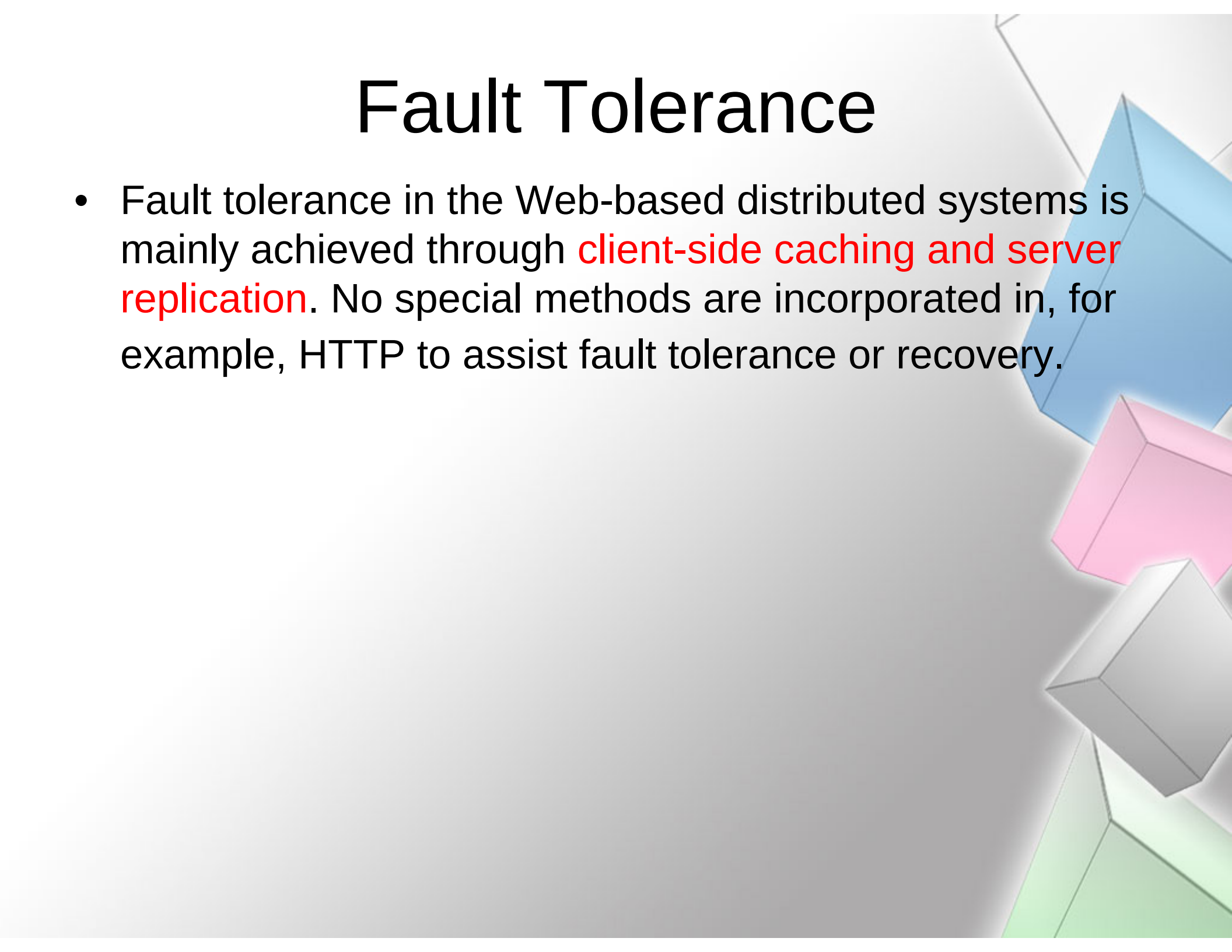
Distributed Web-Based Systems

- Architectures
- Processes
- Communication
- Naming
- Synchronization
- Consistency and Replication
- Fault Tolerance
- Security



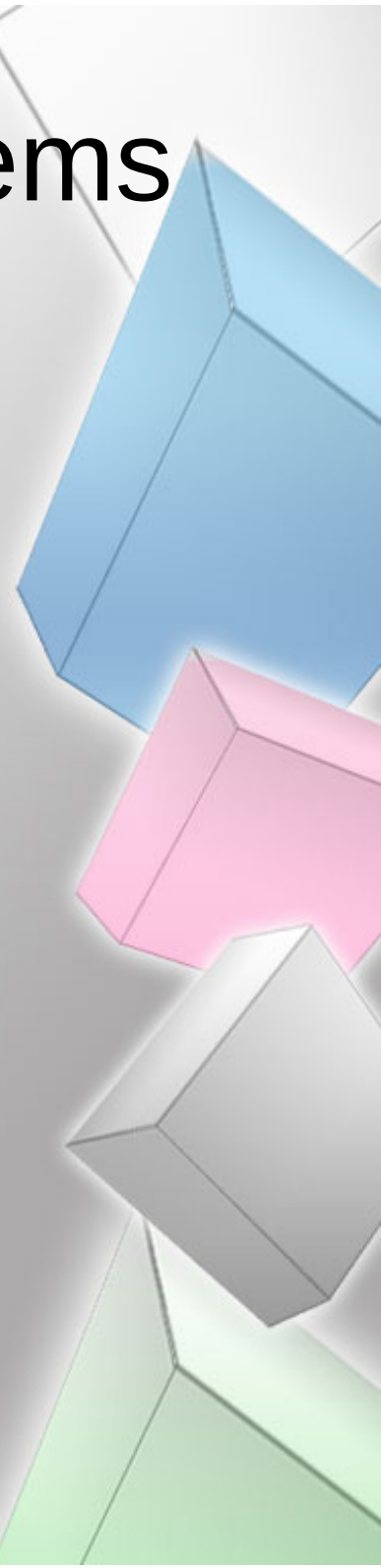
Fault Tolerance

- Fault tolerance in the Web-based distributed systems is mainly achieved through client-side caching and server replication. No special methods are incorporated in, for example, HTTP to assist fault tolerance or recovery.



Distributed Web-Based Systems

- Architectures
- Processes
- Communication
- Naming
- Synchronization
- Consistency and Replication
- Fault Tolerance
- Security



Security (1)

The predominant approach for setting up a secure channel between a client and a server is to use Secure Socket Layer (SSL). An update of SSL has been formally laid down and is now referred to as TLS protocol.

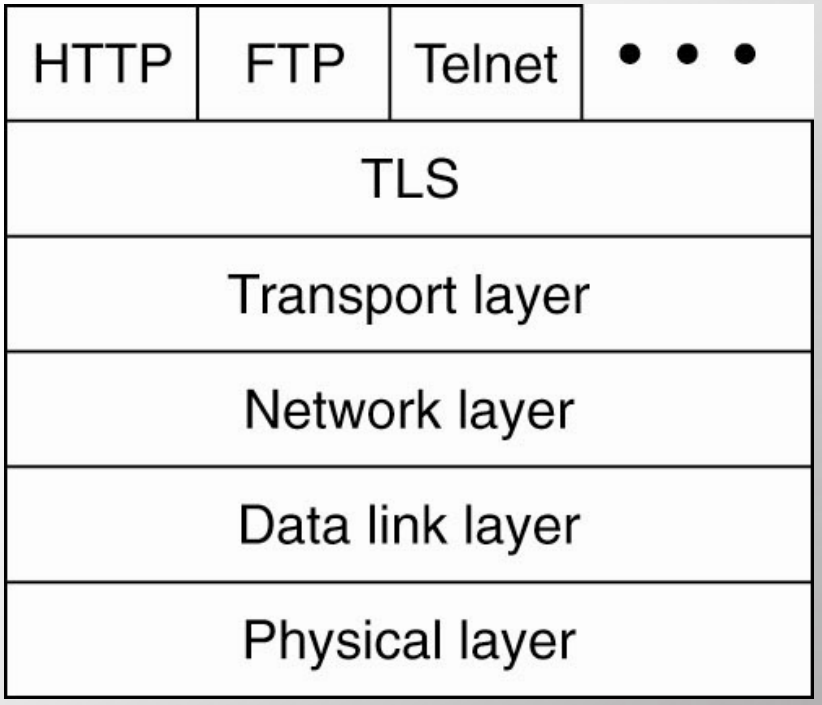


Figure 12-22. The position of Transport Layer Security (TLS) in the Internet protocol stack.

Security (2)

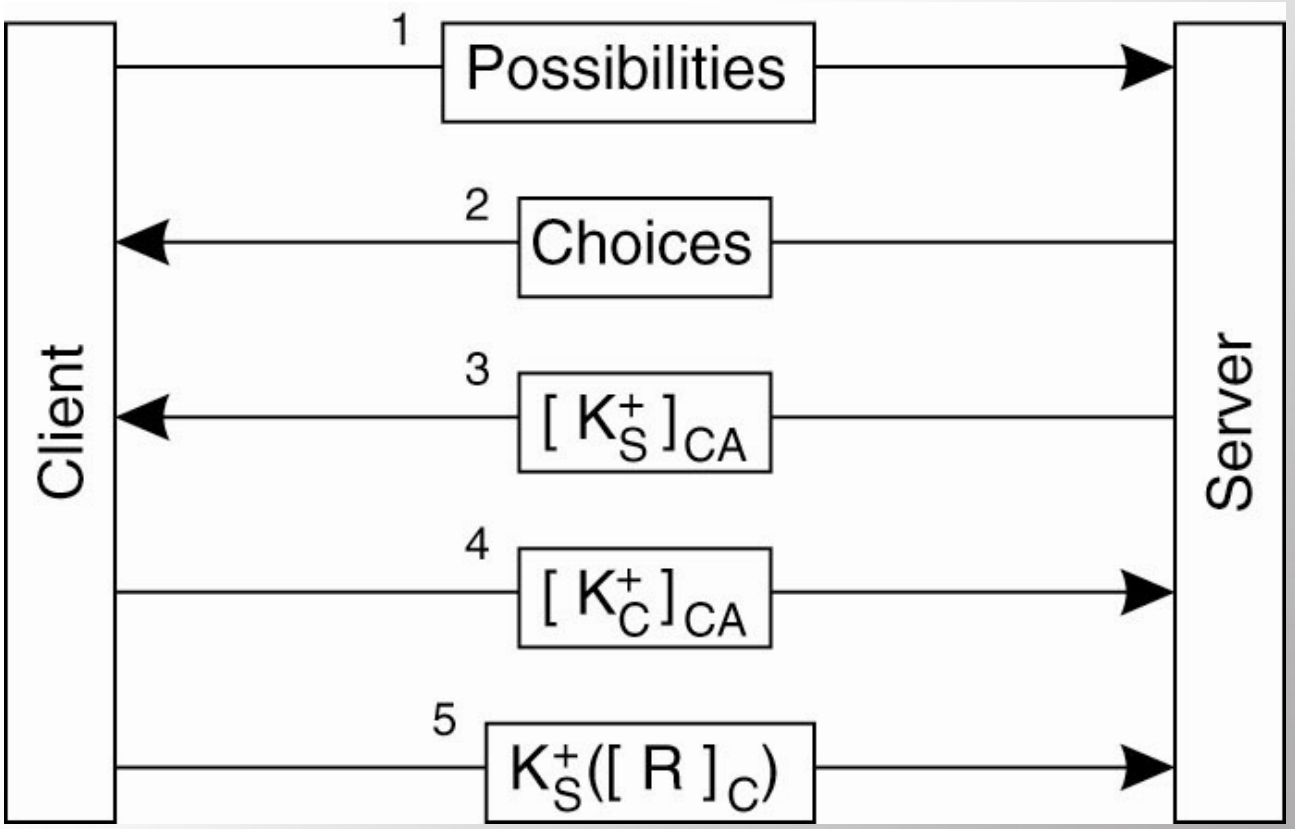


Figure 12-23. TLS with mutual authentication.

End of Lesson 12

- Readings
  - Distributed Systems, Chapter 12.