If several senders are involved, the following message ordering schemes may be applied:

1. no serialization.
2. loosely-synchronous.
   There is a loosely synchronized global time which provides a consistent time ordering.
3. virtually-synchronous.
   The message order is determined by causal interdependencies among the messages. For example, a message N has been sent after another message M has been received, i.e. N is potentially dependent on M.
4. totally ordered.
   By token: before a sender can send a message, it must request the send token.
   a selected component (the coordinator) determines the order of message delivery for all recipients.

---

**Levels of Abstraction**

<table>
<thead>
<tr>
<th>Level of Abstraction</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>object space, collaborative applications</td>
</tr>
<tr>
<td></td>
<td>network services, object request broker</td>
</tr>
<tr>
<td></td>
<td>remote procedure call, remote method invocation</td>
</tr>
<tr>
<td></td>
<td>Client/Server, Peer-to-Peer</td>
</tr>
</tbody>
</table>

---

**Client-server model**

The client-server model implements a sort of handshaking principle, i.e., a client invokes a server operation, suspends operation (in most of the implementations), and resumes work once the server has fulfilled the requested service.

**Terms and definitions**

- Concepts for client-server applications
- Processing of service requests
- File service
- Time service

**Definition:** A time service provides a synchronized system-wide time for all nodes in the network.

**Name service**

**LDAP - Lightweight Directory Access Protocol**

**Failure tolerant services**
sender, receiver: pure message exchanging entities.

client, server: entities acting in some specialized protocol.

Client

Definition: A client is a process (some say, an application) that runs on a client machine and that typically initiates requests for service operations.

Potential clients are a priori unknown.

Service

Definition: A service is a piece of software that provides a well-defined set of service operations. This piece of software may run on one or multiple (server) machines.

Server

Definition: A server is a subsystem that provides a particular service to a set of a priori unknown clients. A server executes a piece of service software on a particular server machine. Obviously, a single server machine can host multiple server subsystems.

A server provides a set of operations (procedures).

1. Client interface (import interface)
   - It represents the server within the client;
   - It prepares parameters and sends the request messages to the server;
   - It prepares the interpretation of the result that is extracted from the answer message submitted by the server.

2. Server interface (export interface)
   - It represents all potential clients within the server;
   - It accepts client requests; interprets the parameters; prepares results;
   - It invokes the respective service operation;
   - It prepares and sends the answer message containing the result of the service operation.

Terms and definitions

Timing process
### Concepts for client-server applications

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>presentation execution</td>
<td>presentation database</td>
</tr>
<tr>
<td>presentation execution</td>
<td>execution</td>
</tr>
<tr>
<td>presentation execution (with local database)</td>
<td>execution (with local database)</td>
</tr>
<tr>
<td>presentation execution database</td>
<td>database</td>
</tr>
<tr>
<td>Case 1</td>
<td>Case 2</td>
</tr>
</tbody>
</table>

#### Different cases

- **Case 1**: remote data storage, access, for example, via Sun NFS.
- **Case 2**: remote presentation (for example X window system).
- **Case 3**: distributed application
  - cooperative processing among the individual components of an application.
- **Case 4**: distributed data storage
  - The information is distributed between client and server; information replication is possible.