Resource sharing • Sharing of hardware and software resources.

Openness • Use of equipment and software from different vendors.

Concurrency • Concurrent processing to enhance performance.

Scalability • Increased throughput by adding new resources.

Fault tolerance • The ability to continue in operation after a fault has occurred.

Transparency •

For security and inter-operability reasons, most distributed computing has been implemented at the enterprise level.

Local standards, management and operational processes apply.

Newer models of distributed computing have been designed to support inter-organisational computing where different nodes are located in different organisations.

Distributed systems support resource sharing, openness, concurrency, scalability, fault tolerance and transparency.

Client-server architectures involve services being delivered by servers to programs operating on clients.

User interface software always runs on the client and data management on the server. Application functionality may be on the client or the server.

In a distributed object architecture, there is no distinction between clients and servers.

Distributed object systems require middleware to handle object communications and to add and remove system objects.

The CORBA standards are a set of middleware standards that support distributed object architectures.

Peer to peer architectures are decentralised architectures where there is no distinction between clients and servers.

Service-oriented systems are created by linking software services provided by different service suppliers.