Distributed systems solution 2001 p891 JMB

a) RPC
call remote-proc(argument-list)
integration of comms. with strongly typed prog. language(s).
Support for flattening arguments via stub routines at sender and receiver.
Need support for naming externally invocable procedures
Had to introduce call-by-copy to replace call-by-reference
Implements Appn, Pres, & Session ISO layers. General distributed programming

b) O-O middleware:
Similar to RPC in operation but object id's are first-class values so argument passing is cleaner.
Need support for name (id) to location mapping
Both RPC and O-O may support heterogeneous component if an interface definition language & prog. lang. bindings to it are defined. For general distributed programming in a local environment.

a) & b) are both synchronous, request – response.

c) MOM is asynchronous
Sender sends a packet of bytes & proceeds. Receiver waits & either blocks or is passed a message from its queue
Message structure – maybe none, maybe fields, not typed
Reliability: MQseries is transactional by default: CICS.

d) Event-based middleware.
Asynchronous – cf – MOM with subscription + notification
Filtering may be close to or at source or at client.
Registration may be with event name only (eg early CORBA event service) or with parameter values & wild cards. (CORBA notification service). Needed for app^n areas such as active house/city – sensor-rich environment –

a) & b) & c) & d) need naming support: directory service for recipient of comm – name to address mapping.

e) Publish – subscribe
Similar to event system: register in a "topic" & are notified messages on that topic. Based on multicast comm^n. Cf "tuple space": comms = shared distributed database
eg. TIBCO. eg. stock values broadcast service, Reuters etc

In a) & b) blocked thread waits for reply
In c) d) e) a thread must be ready on a callback address to receive messages.