



Introduction To **MandateIP® Architecture**

MandateIP®, the latest generation of Mandate®, is a highly scaleable, distributed, realtime, “transaction based” software platform. This platform was created for, and is ideally suited for implementation of distribution and order fulfillment centers. In this particular context, a “software platform” is a layer sitting on top of a base operating system that provides a consistent structure and support for specific applications as well as providing common functionality that is required for in most distribution operations.

“Transaction based” refers to how information is conveyed between system elements. Transaction based systems primarily use “messages” to communicate between system elements or modules. The converse of a “transaction based” system is one that primarily uses either databases or other data structures that are shared between system elements or modules to convey information. Transaction based systems are inherently “event driven” where a transaction is created to signify and convey information pertaining to an event. Transactions may initiate an update to a database, the actuation of some piece of equipment, update a user screen or initiate other transactions. Transaction based systems provide an elemental feature of being able to be “distributed” across multiple computing elements. This distributed characteristic provides for scalability where computing elements to handle the transaction load may be provided to meet the system requirements. MandateIP® is a “realtime” platform in as much as transactions are executed as the associated event occurs.

The MandateIP® architecture originates from technologies initially developed in the 1970’s and the implementation of the architecture has been continuously improved over the ensuing years to meet today’s demands of stability, performance, and greater functional requirements.

MandateIP® based systems are comprised of numerous independent software modules or “transaction handlers” that are called MandateIP® “servants”. “Servants” are “operationally defined” by the functions they are to perform. A single “servant” module may support or handle a number of transactions. Typical systems may contain hundreds of servants with each servant being a small module providing a limited set of services.

The heart of the MandateIP® platform is a process (or a program) called the “message dispatcher”. A copy of the message dispatcher runs on every computing element (MandateIP® node) in a system. The message dispatchers perform three basic functions:

- 1) Receive messages from servant modules and route those messages to the responsible servant
- 2) Spawn or bring into execution any servant module that is required to perform a service (execute a message or transaction)
- 3) Spool (queue) messages for servants that are not currently able to process a new message.

Referring to figure 1, servant processes (modules) communicate only with their associated message dispatcher. Any requests for a service are dispatched to the associated servant process by the message dispatcher and any requests for a service by a servant process are delivered to the message dispatcher for routing to the proper servant. It should be noted that at times, a servant process may request a service that is provided by the requesting process. That request (transaction) is routed just as any other request for an external service.

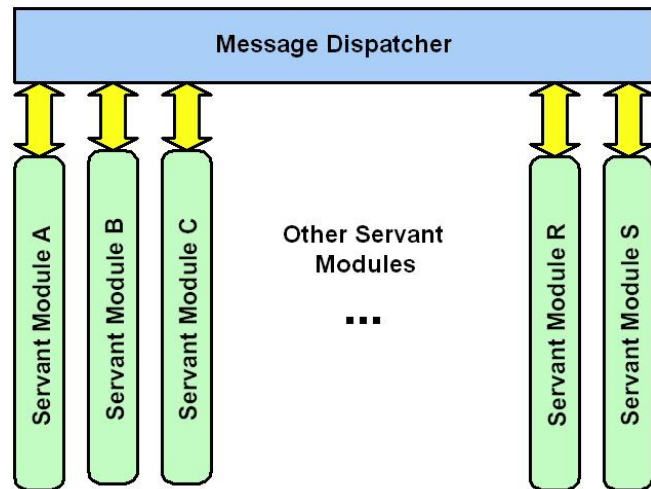


Figure 1

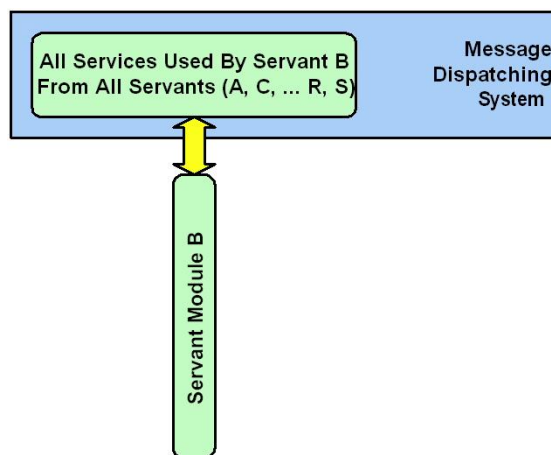


Figure 2

Servant processes register the “services” they provide with their own message dispatcher. Upon receipt of service registration information, a message dispatcher makes that information available to all other dispatchers in the system. The other dispatchers know how to route messages they receive to the appropriate service provider (servant module). This is illustrated in figure 3.

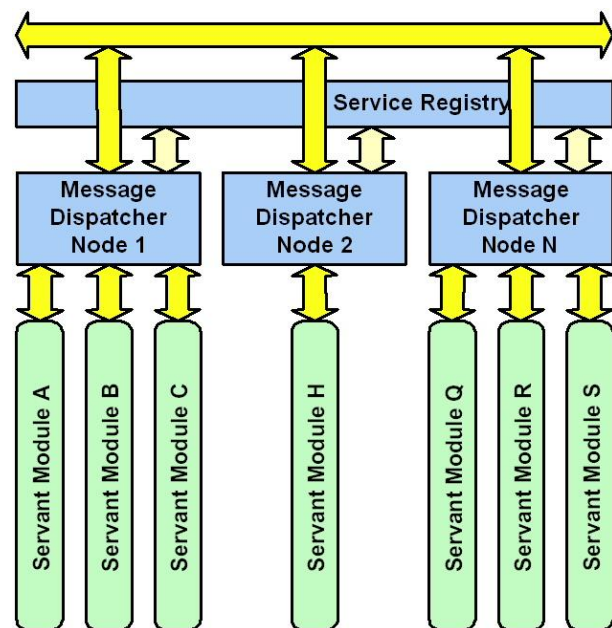


Figure 3

The MandateIP® platform provides a stable and reliable base to construct simple, modular, independent, and easy to understand servant processes. This platform proves itself daily in providing uninterrupted service for delivery of billions of dollars of product.