

Middleware Domain Best Practices:

- Best Practice 1.** A middleware layer will be used to connect the business logic tier to the data access tier (or back-end databases). This layer can be the database access components or transaction processing monitor (TPM) components.
- Best Practice 2.** DOIT developed SOA is currently developed using .NET and Java developed Web Services.
- Best Practice 3.** It is recommended that all return values and parameters into a web service method should be XML formatted data with corresponding XSD Schema for validation purposes.
- Best Practice 4.** A good application design should include definitions of all data types as required in the SDM.
- Best Practice 5.** Application data types should be defined first and then the web method parameters and return values should be based on the application data definitions.
- Best Practice 6.** Security for MQ should consist of securing the channels, queues, processes connections, name lists, and commands. Only those parties that are authorized should be able to access them. This is to protect the confidentiality, integrity, and availability of the messages and ensure that unauthorized access, unauthorized modification, and unauthorized deletion do not occur.
- Best Practice 7.** Messaging should be XML with a defined schema used for validation.
- Best Practice 8.** Sensitive data should be encrypted while at rest within a queue.
- Best Practice 9.** Queue Capacity Management should be implemented to assure that the queue depths can handle the expected volume. Monitoring and scripting should be implemented to avoid delays and other issues that can arise.
- Best Practice 10.** Tools and or scripting should be in place to be proactive in ensuring that the messages / data flow can be tested so that all components can be isolated for troubleshooting purposes.
- Best Practice 11.** The State of CT MQ naming conventions should be followed for any messaging products.
- Best Practice 12.** Applications must also follow the MQ application development standards. All applications should have error handling code written the application and sent to a configurable logging file. The ability to alter for more robust features should be built for troubleshooting purposes. This can include sending outbound messaging to alert technicians when errors occur to be more proactive and reactive.