Interface identity
TODO: The most common way of identifying an interface is to name it. In some cases, merely naming the interface is not enough and the version of the interface must be specified as well.

Resources
TODO: The heart of the interface documentation is the set of resources it provides. Resources are typically operations, but can be messaging endpoints, shared memory or data streams. For each resource, provide the syntax, semantics and error handling. The syntax is the signature of the resource. The semantics is a description of what happens when the resource is used, along with pre- and post-conditions and usage restrictions. Error handling lists exceptions and error codes that can occur when the operation is executed.

Data types and constants
TODO: describe new data types created for the data passed to or returned by resources in the interface. Describe also any constants defined in the interface to hold commonly-used values that make programming against the interface more convenient.

Error handling
TODO: Describe error conditions that can be raised by the resources on the interface. This section has the error handling description that is common to two or more resources. Error handling is also described in the Resources section separately for each resource.

Variability
TODO: If the interface allows the element to be configured in any way, then these configuration parameters and how they affect the semantics of the interactions in the interface must be documented here.

Quality attribute characteristics
TODO: document what quality attribute characteristics, such as performance or reliability, the interface makes known to its users.

Rationale and design issues
TODO: record the reasons behind the design of the interface.

Usage guide
TODO: describe step-by-step one or more examples of using the interface.