

StorageWorks™



X3T10/96-191R0

Editorial Conventions for SAM and Protocol Standards

Charles Monia
Digital Equipment Corp.
July 16, 1996

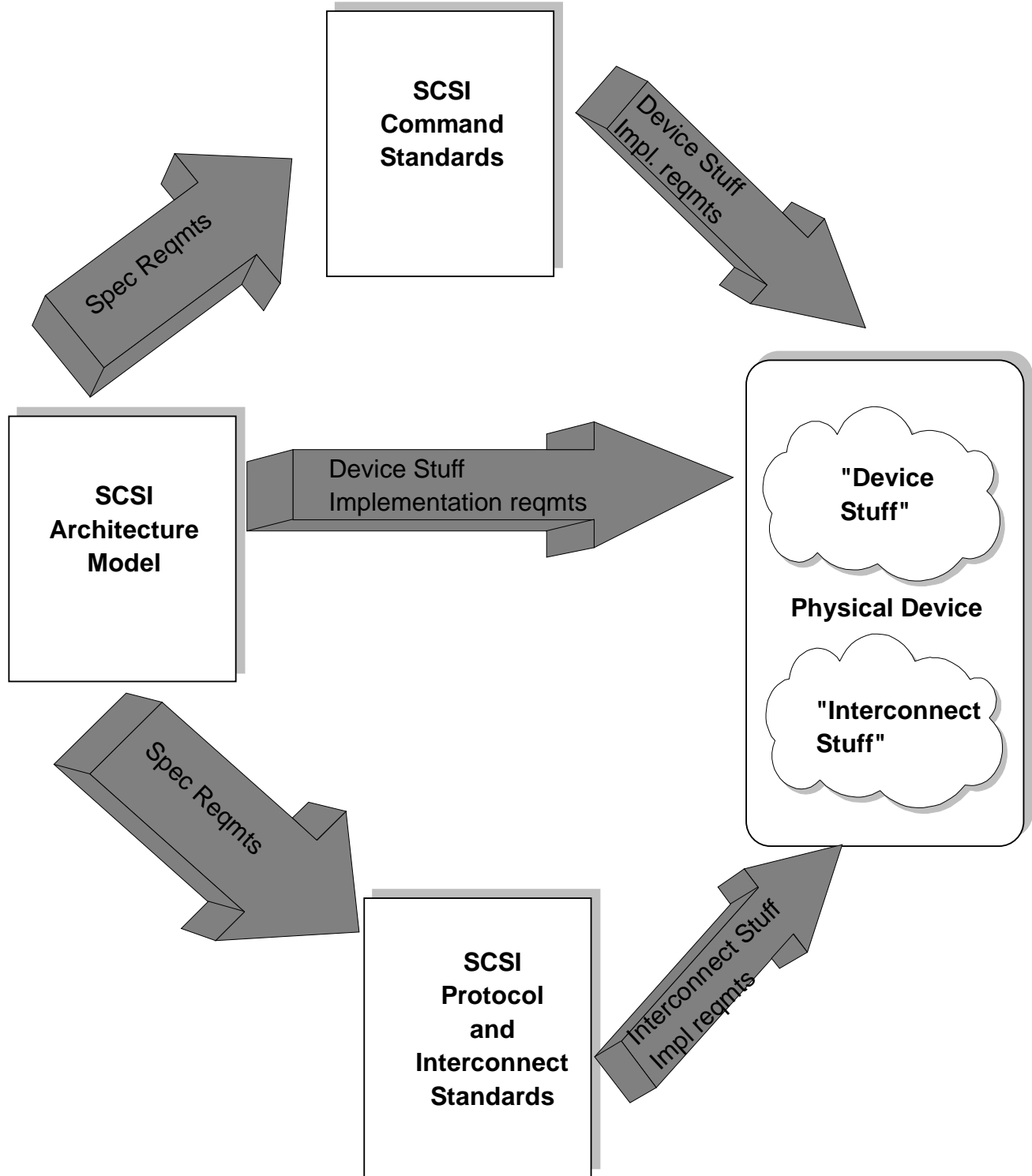
digital



- Issues:
 - Relationships between SAM and other standards are not clear.
 - Groundrules for SAM compliance are undefined.
- Approach
 - Use OSI methods to refine existing specs.
 - Define compliance in terms of: a uniform SCSI service model and protocol.



■ Requirements model





- Goal of requirements
 - Interoperability.
 - All interconnects "look the same" to host and target application layer.
 - Testability
- Specification Requirement
 - Defines specification content.
 - Test for compliance:
 - Do the words in the spec comply with the requirement?
- Implementation requirement.
 - Applies to physical implementations
 - Defines observable or measurable characteristics.
 - Defines how measurement is made.
 - Test for compliance:
 - Does the measured characteristic comply with the requirement?



- SAM Contains:
 - Specification requirements for SCSI-3 protocol and command standards.
 - Implementation requirements for
 - A SCSI I/O system
 - "Device Stuff"
 - How queuing works
 - Task Management functions
 - How ACA works
 - Unit Attention behavior
 - Command Status values and definitions
 - Hard reset
 - General CDB format
 - Etc.
 - Device Model -- What "device stuff" gets implemented in the physical device.
- Protocol Standards specify:
 - Implementation requirements for "interconnect stuff"
- Sam and the protocol standards define layers of functionality.

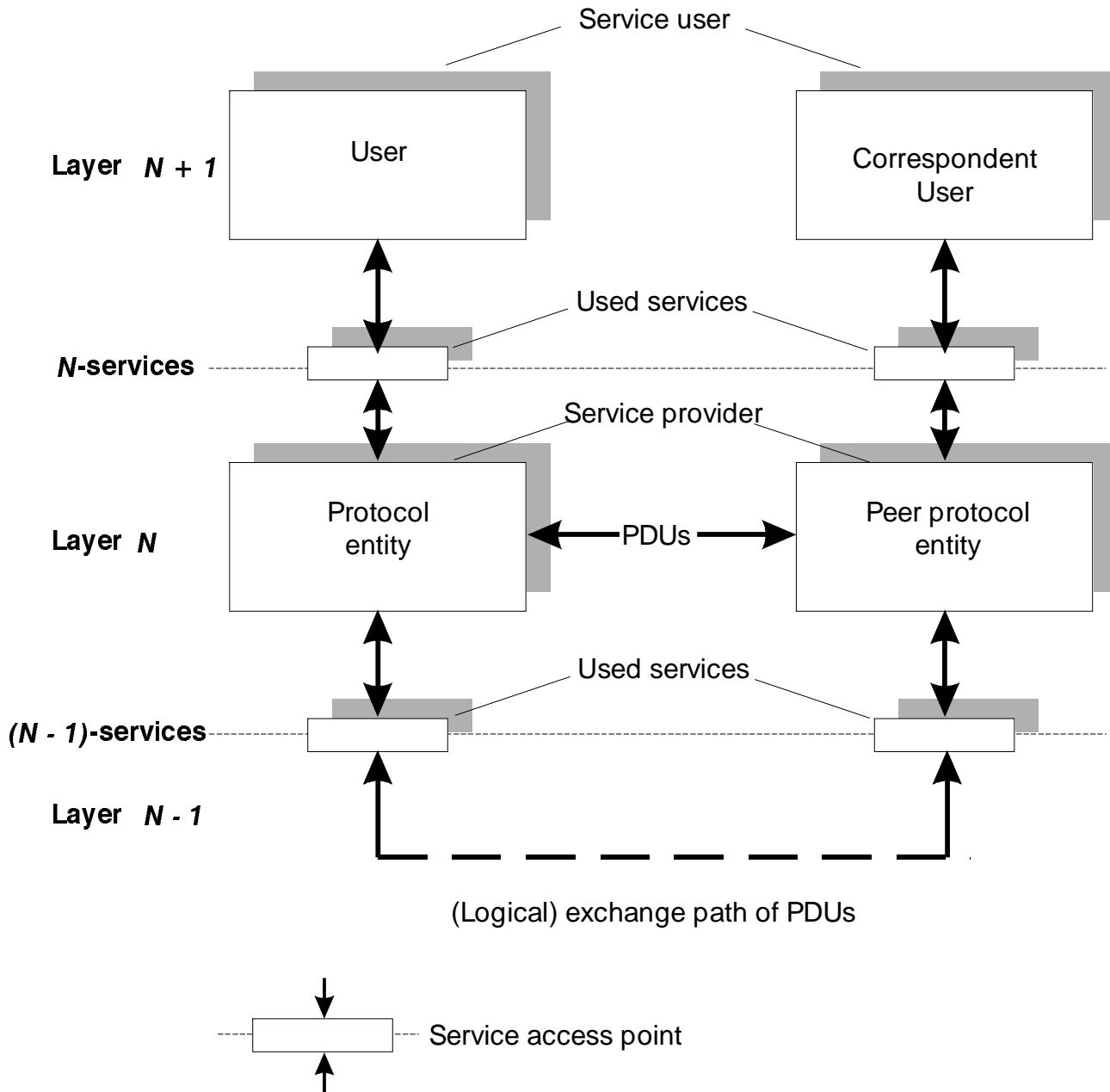




- OSI distributed system
 - Functional layers
 - Service interfaces between layers.
 - A good way to specify distributed systems with layered functions.
- Each Layer
 - Provides services to the layer above.
 - Communicates by invoking services provided by the layer below.
- Each layer is defined by:
 - The user services it provides.
 - Service Access Point: The service interface between two protocol layers.
 - Service primitives: the services to be provided by a layer
 - The protocol it implements.
 - Defined in the protocol specification document.

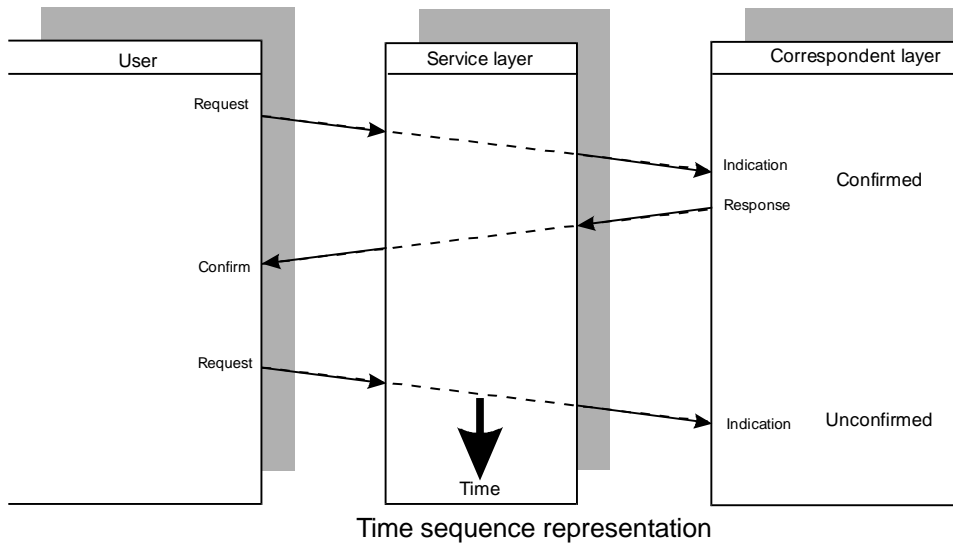
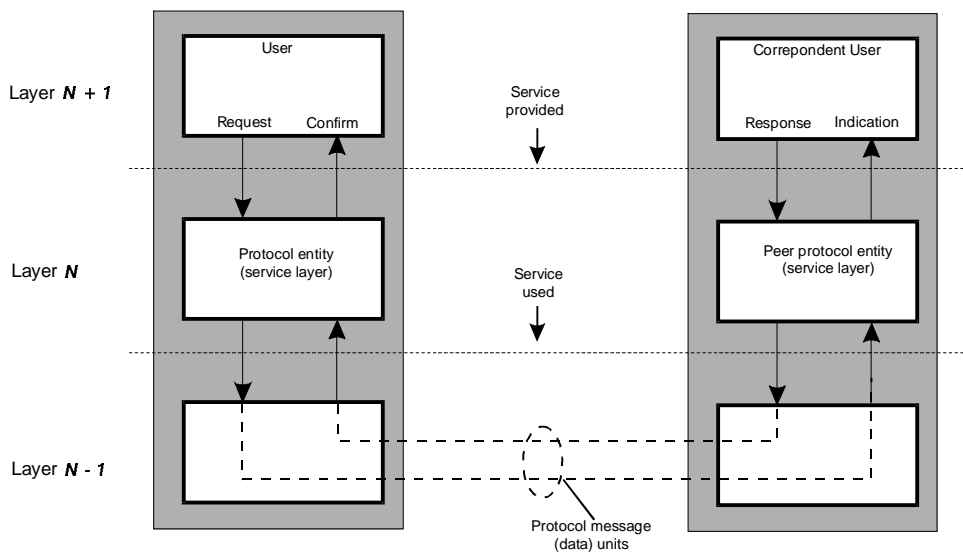


■ OSI layers



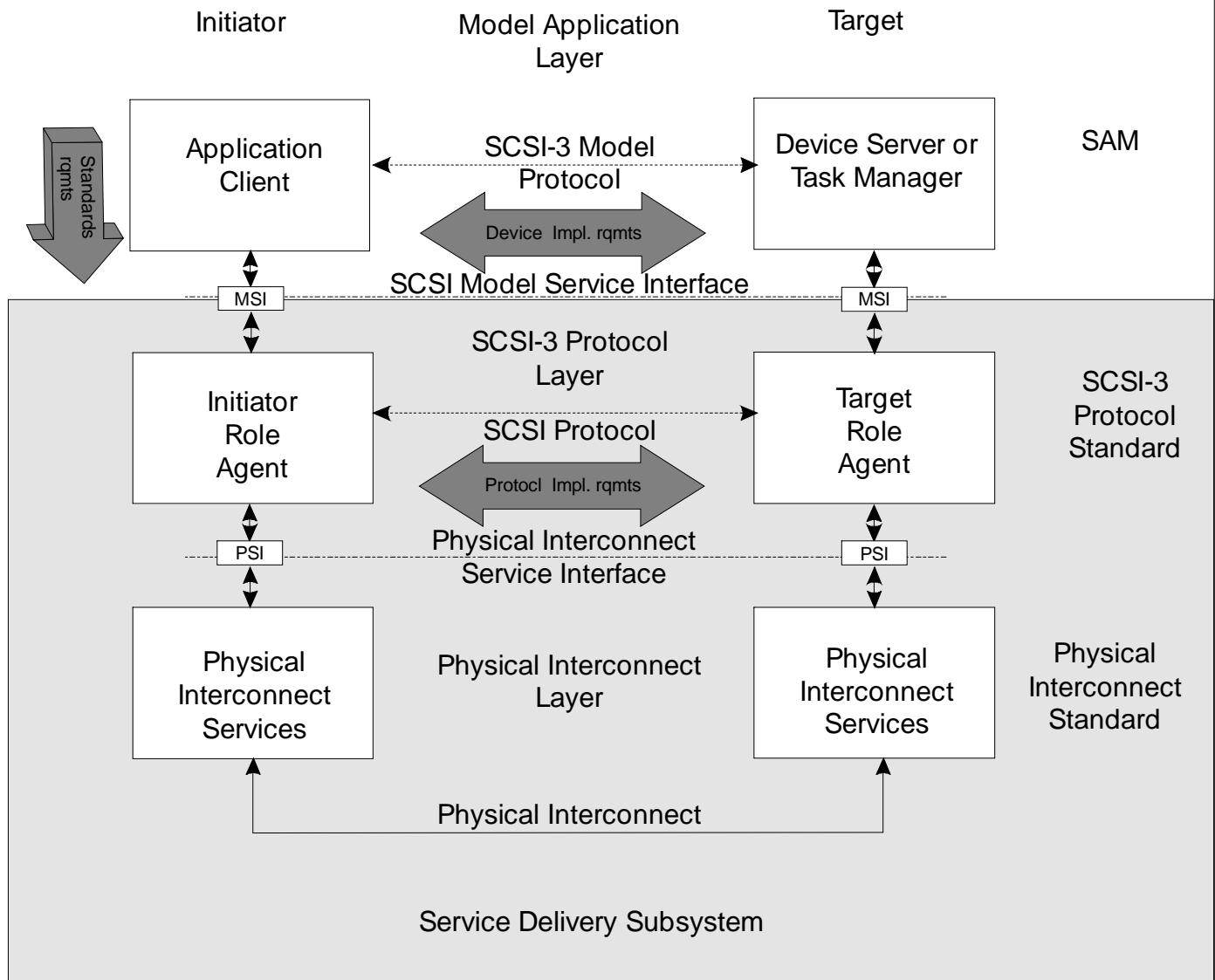


- OSI Service Primitives
 - Request
 - Indication
 - Response
 - Confirmation
- OSI Service Types
 - Confirmed
 - Unconfirmed





■ SCSI Layered Model



- MSI -- Model Service Interface,
- SCSI-3 Model protocol
- Provided to complete the description of behavior and integrate the architecture with the protocol standards.



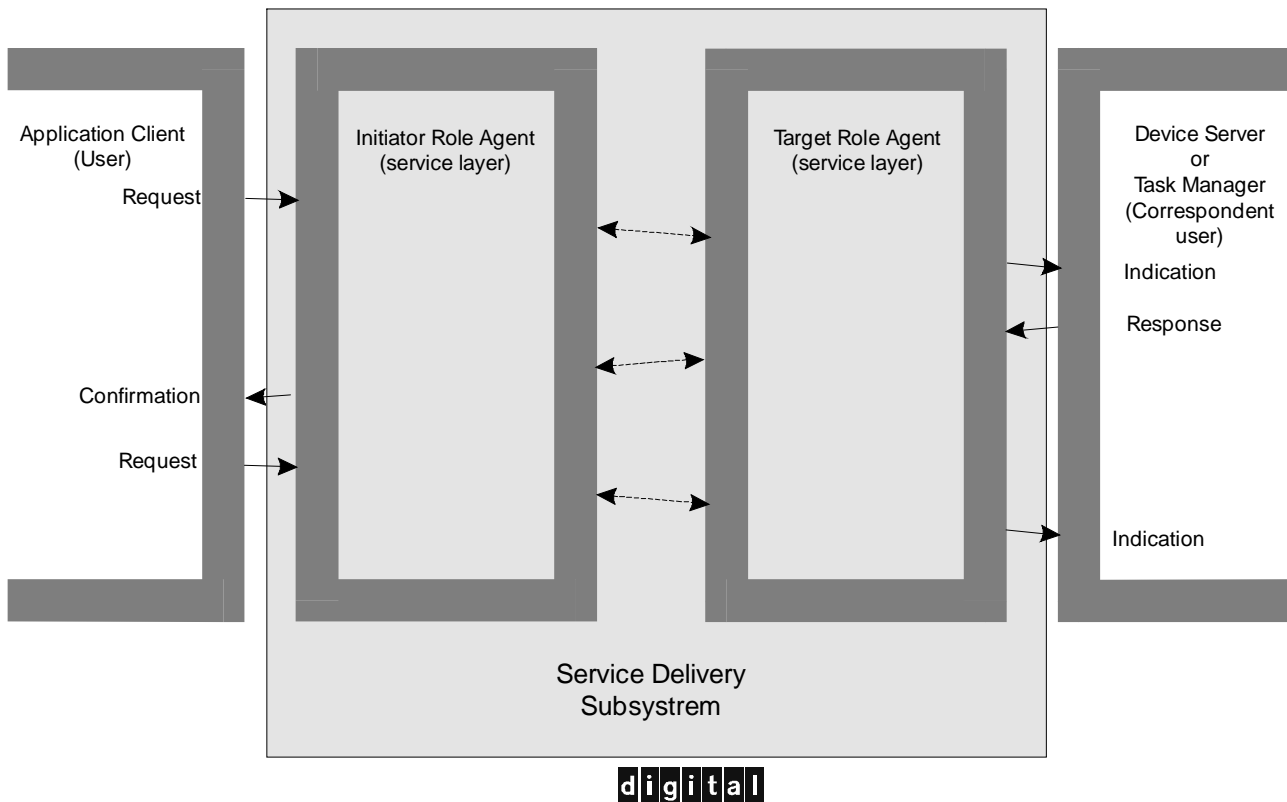
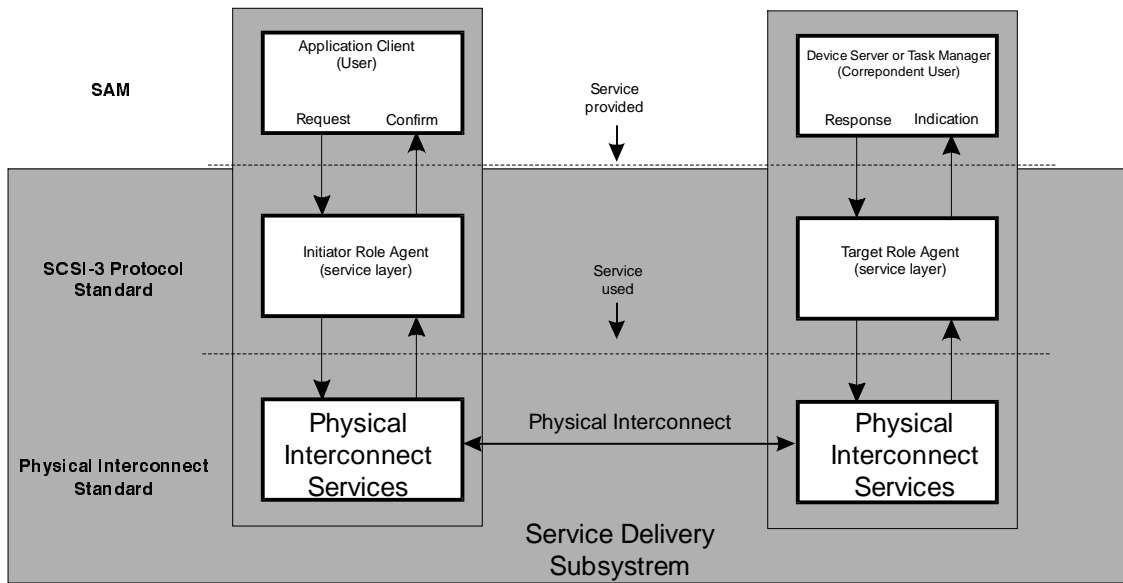


- Model Application Layer
 - Model service primitives.
 - Model Protocol.
 - Hook for specifying device-level implementation requirements
 - **Defined to complete the description of behavior only.**
 - Represents a generic protocol implementation
 - Not intended to be implemented.
- Specification requirement
 - Each protocol standard shall define the protocol-specific mapping of the model services.
- A SCSI protocol standard conforms to SAM if:
 - It correctly maps the model service primitives.
 - Complies with other SAM specification requirements.



■ SCSI Service Model

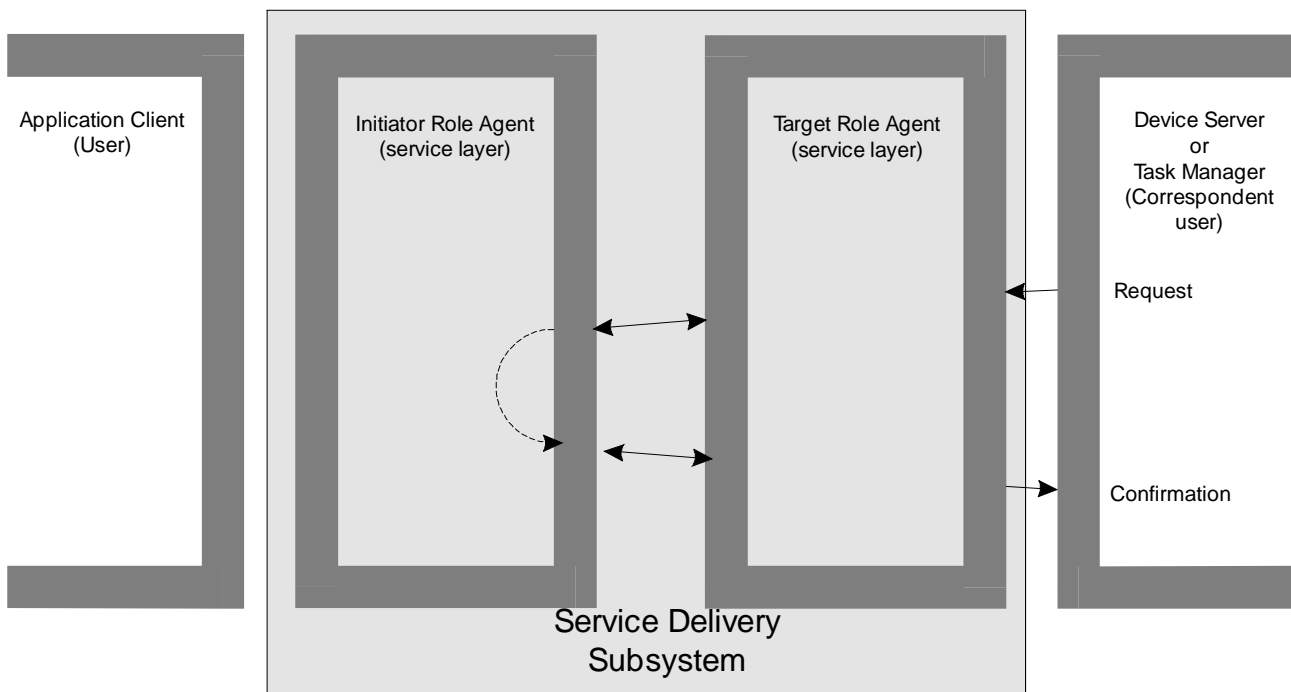
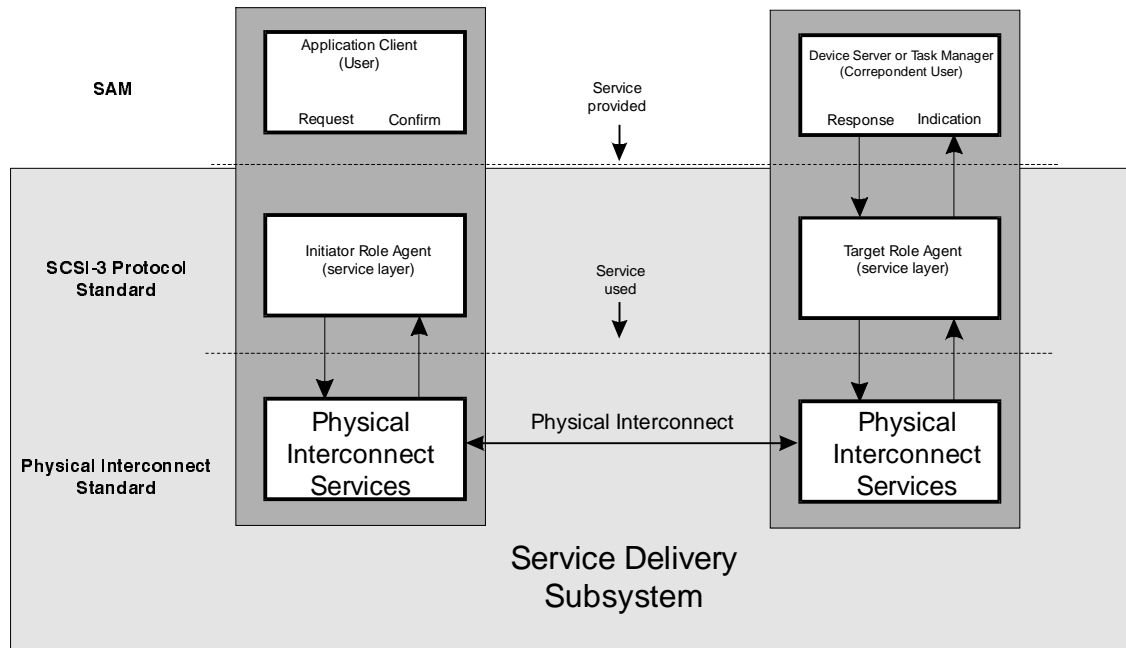
- Based on SIP, SPI
- 4-step Confirmed
- Unconfirmed





■ SCSI Service Model

– Two-step, confirmed

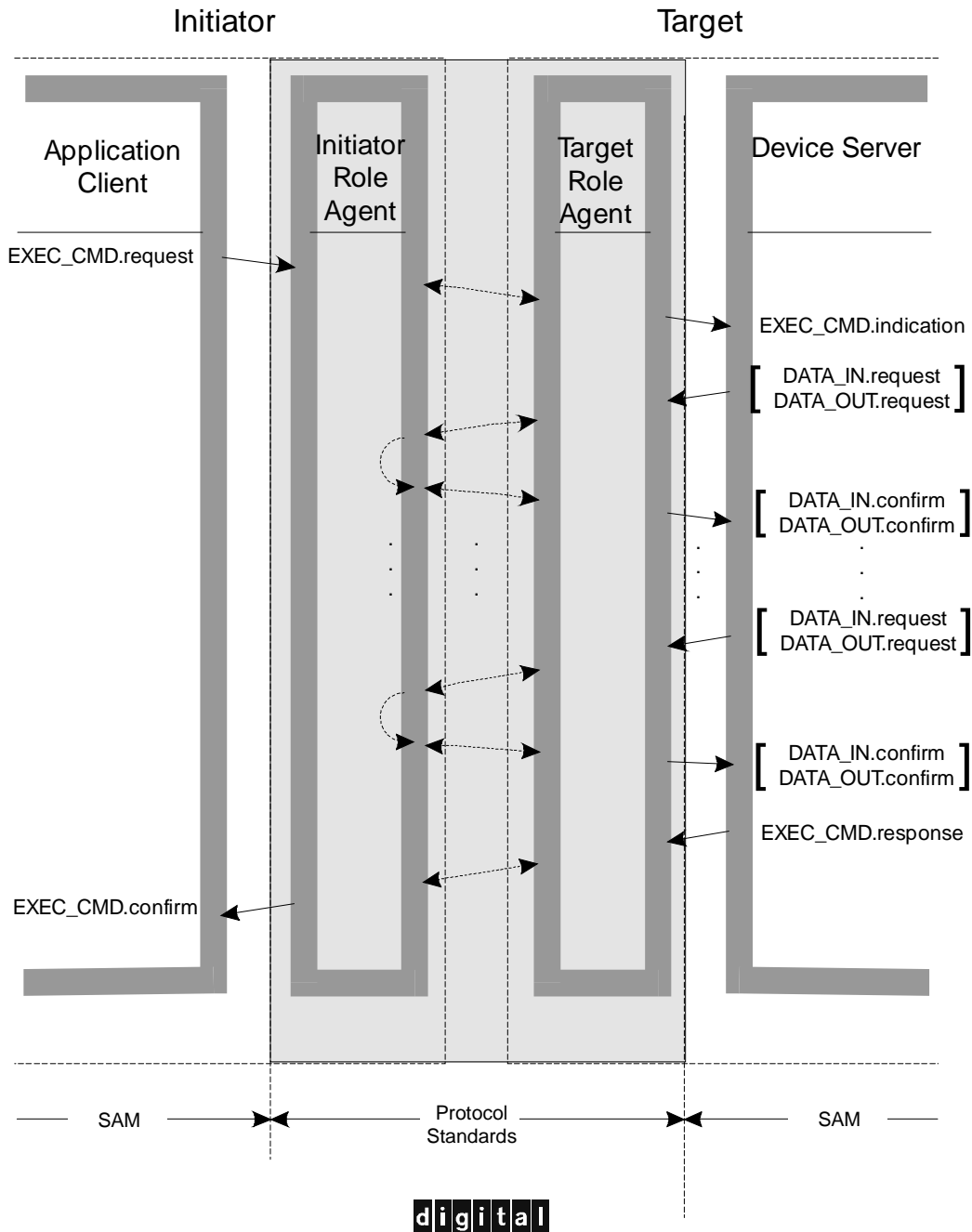




- Service Primitives
 - Request -- *nnnnn.request* (params)
 - Indication -- *nnnnn.indication* (params)
 - Response -- *nnnnn.response* (params)
 - Confirmation -- *nnnnn.confirm* (params)
- Each protocol standard defines the protocol-specific mapping of service primitives and parameters.



■ SCSI-3 Command Service Model





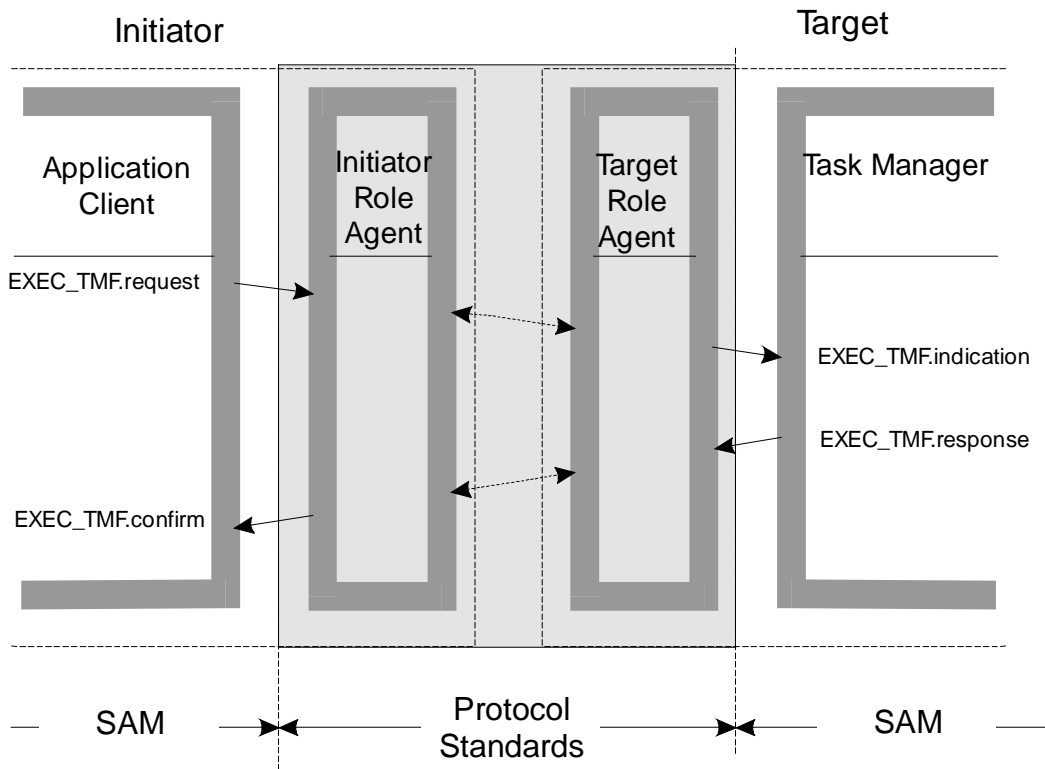
- Command Service primitives
 - Command execution
 - Exec_cmd.request (Task Address, CDB, [Task Attribute], [Data-out buffer], [Command Byte Count], [Autosense Request] ||)
 - Exec_cmd.indication (Task Identifier, [Task Attribute], CDB, [Autosense Request] ||)
 - Exec_cmd.response (Task Identifier, [Sense Data], Status, Service Response ||)
 - Exec_cmd.confirm (Task Address, [Data_In Buffer], [Sense Data])



- **Command Service Primitives (con't)**
 - Data transfer services (invoked by logical unit).
 - Two-step, confirmed service
 - Data movement controlled by logical unit.
 - Transfer data from target to initiator
 - Data_in.request(Task Identifier, Device Server Buffer, Application Client Buffer offset, Request Byte Count ||)
 - Data_in.confirm (Task Identifier ||)
 - Transfer data from initiator to target
 - Data_out.request(Task Identifier, Device Server Buffer, Application Client Buffer Offset, Request Byte Count ||)
 - Data_out.confirm(Task Identifier||)



■ SCSI-3 Task Management Service Model





- Task management services
 - Exec_tmr.request (Object Address, Function Identifier ||)
 - Exec_tmr.indication (Object Identifier, Function Identifier)
 - Exec_tmr.response (Object Identifier, Service Response||)
 - Exec_tmr.confirm (Object Address, Service Response||)