

ReqId	Category	Level	Description	Action
D-CR-017	Transaction	SHOULD	A choreography SHOULD enable an agreement or agreement(s) that provides the business context of the choreography definition.	Candidate for deletion; Reclassify
D-CR-052	Management and Provisioning	MUST	It MUST be possible to manage choreographies and their relationships and the messages exchanged between them.	Clarification
D-CR-009	Exception Handling	MUST	A choreography MUST provide the ability to transition to a distinct state when a timeout occurs.	Clarification
D-CR-040	Composition	MUST	It MUST be possible to describe parallel composition of services.	Clarification
D-CR-041	Composition	MUST	It MUST be possible to model events that are strictly related in time, as well as those that are unrelated in time - i.e. parallelism or partial ordering.	Clarification
D-CR-059	Transaction	MUST	It MUST be possible to describe negotiation between Web Services participating in a choreography, where the result of a transaction may depend on repeated iterations of an ask/answer cycle.	Clarification (of use case)
D-CR-032	Exception Handling	MUST	It MUST be possible to manage exceptions between choreographies, and to include exception-handling information in dependency management.	Clarification on propagation
D-CR-030	Exception Handling	MUST	A choreography MUST be able to distinguish error and regular state transitions, and to describe unhandled exceptions.	Clarification on unhandled exceptions
D-CR-027	Exception Handling	MUST	Error/fault handling and compensation features MUST to be able to be expressed in the choreography definition language.	Clarification("Do we need to distinguish")
D-CR-067	Exception Handling	SHOULD	A choreography definition language SHOULD support exceptions whereby a given service component could be sent at any point in the transaction.	Clarification(We don't understand any of these)
D-CR-036	Composition	SHOULD	Choreographies SHOULD be able to call other choreographies in a hierarchical fashion.	Clarification: Dynamic whereas 34 is static design time? But seem to be related - agreed needs clarification.
D-CR-042	Composition	MAY	A choreography MAY have run time changes that allow the behavior of the actual choreography to vary based on state.	Clarification: Not understood - major clarification required
D-CR-018	Exception Handling	SHOULD	A choreography SHOULD define a standardized way to recover from catastrophic failure.	Clarification; Leave til later
D-CR-026	Exception Handling	MUST	A choreography MUST support robust exception handling.	Clarify in conjunction with CR-005
D-CR-054	Composition Management and Provisioning	SHOULD	A choreography SHOULD express the composition of participant Web Services into a new service.	Delete
D-CR-013	Management and Provisioning	MUST	A choreography MUST provide support for a specific observer role.	Delete; Implementation
D-CR-020	Management and Provisioning	MUST	It MUST be possible and practicable to store instances of use of choreographies in a repository, and to be able to search for such instances and to retrieve them.	Delete; implementation feature not part of the language
D-CR-033	Exception Handling	MUST	A choreography MUST provide support for defining behavior of the system, when valid error/exception messages arrive for choreography instances after their completion (or before their initiation).	Deletion
D-CR-065	Exception Handling	MUST	A choreography definition language MUST support the following exception: message format error - the components of the message were not validly constructed.	Deletion candidate
D-CR-066	Exception Handling	MUST	A choreography definition language MUST support the following exception: message content structure error.	Deletion candidate
D-CR-068	Exception Handling	MUST	Exception types MUST be limited to the ones that will be used in the choreography definition language.	Deletion candidate
D-CR-070	Exception Handling	MUST	A choreography definition language MUST support the following exception: transmission error - message not sent.	Deletion candidate
D-CR-029	Messaging	MUST	A choreography MUST support some standard taxonomy of messages, such as a business messages, generic error reporting messages and acknowledgement messages.	Deletion candidate
D-CR-051	Messaging	MUST	A choreography definition language MUST provide a construct that describes the sending of a single message.	Deletion candidate
D-CR-058	Transaction	SHOULD	It MUST be possible for a choreography to modify its behavior based on its operational context.	Deletion candidate
D-CR-039	Messaging	MUST	It MUST be possible to describe a sequence of communications among choreographies.	Deletion candidate; Clarification (among chor) and duplicate
D-CR-043	Exception Handling	MUST	It MUST be possible to differentiate errors (unknown and fatal) and exceptions (known and potentially recoverable) in the context of choreography.	Deletion; Further Discussion
D-CR-061	Exception Handling	MUST	A choreography MUST provide the ability to transition to a predefined state when an exception occurs.	Duplicate
D-CR-060	Messaging	MUST	It MUST be possible to define a choreography without having to specify the contents of the messages being used.	Duplicate
D-CR-048	Transaction	MUST	It MUST be possible to describe a choreography in terms of its messaging behavior.	Duplicate
D-CR-024	Composition	MUST	It MUST be possible to dynamically determine the participants in a choreography at runtime.	Duplicate with 003; Leave in duplicate? About composition of Choreographies - clarify heirarchy in this req
D-CR-034	Composition	MUST	Choreographies MUST be composable into a hierarchy.	Duplication (KL: important requirement)
D-CR-006	Composition	MUST	It MUST be possible to describe choreographies as a composition of other choreographies.	Duplication (KL: important requirement)
D-CR-031	Exception Handling	MUST	A choreography MUST support an exception condition for Delivery Failure - The message was sent but was (probably) not received.	Duplication candidate
D-CR-069	Exception Handling	MUST	It MUST be possible to model different states for termination of the choreography (e.g., failure and success).	Duplication candidate

D-CR-011	Messaging	SHOULD	A choreography SHOULD describe exchanges of information that change the state of the process. It MUST be possible to validate a choreography definition for correct behavior at the time it is in operation.	Duplication(merge with 007)
D-CR-050	Testing and Validation	MUST	An implementation of a process that is following a choreography MUST be able to verify that the choreography is being followed correctly as specified in the choreography definition.	Duplication: Wrong way around also a duplicate
D-CR-008	Testing and Validation	MUST	It MUST be possible to model message flows that repeat, based on information within the messages (for instance, the contract negotiation protocol).	Duplication; Wordsmithing ("needs to mention viewpoint")
D-CR-035	Messaging	MUST	It MUST be possible to describe conditional behavior for a choreography.	Further discussion
D-CR-038	Transaction Management and Provisioning	MUST	It MUST be possible to query the state of a choreography.	Further Discussion; Reclassify ("Composition"); Wordsmith (What not how)
D-CR-004	Transaction	MUST	It MUST be possible to define multi-party interaction.	Further discussion; contentious
D-CR-057	Ease of Use	MUST	A choreography MUST be uniquely named.	Reclassify
D-CR-012	Transaction Management and Provisioning	MUST	It MUST be possible to describe a behavior recursively, where not all interactions are presented.	Reclassify - semantics
D-CR-037	Exception Handling	SHOULD	If a process detects that a choreography is not being followed correctly, then the process SHOULD be able to use the choreography definition to identify exactly what went wrong.	Reclassify ("Composition"); Wordsmith (What not how)
D-CR-056	Management and Provisioning	MUST	A choreography MUST provide a global model for presenting its interactions from the point of view of all the parties, and not from the point of view of just one party.	Reclassify (Composition)
D-CR-064	Composition	MUST	It MUST be possible to define a new choreography by "extending" an existing one.	Reclassify to Management and mild word smithing
D-CR-016	Composition	MUST	It MUST be possible to dynamically add sub-choreographies to a "running" choreography.	Reclassify("Charter Constraint")
D-CR-047	Testing and Validation	MUST	definition for correct behavior at the time it is designed.	Reclassify("Reuse")
D-CR-062	Messaging	SHOULD	It SHOULD be possible to define a choreography that uses a callback mechanism.	Remit to a V2
D-CR-049	Messaging	MUST	A choreography MUST be independent of message formats.	Remit to a V2
D-CR-055	Interfaces	MUST	A choreography MUST enable information hiding.	Wordsmith ("Callback mechanism")
D-CR-002	Messaging	SHOULD	A choreography SHOULD express the types of messages a participant may send, and the types of messages/responses the participant should anticipate receiving from the other participant(s) (including time-outs) based on the apparent state of the exchange.	Wordsmith (Chor vs CDL)
D-CR-044	Messaging	SHOULD	A choreography definition language MUST NOT be restricted to any single implementation.	Wordsmith (Information hiding vs Segmentation)
D-CR-007	Interoperability Management and Provisioning	MUST	It MUST be possible to observe the state of a choreography as it occurs in time.	Wordsmith (Must, apparent state of the exchange)
D-CR-023	Exception Handling	MUST	A choreography MUST provide exception handling that allows propagation of errors.	Wordsmithing
D-CR-053	Charter Constraint	SHOULD	A choreography SHOULD not be bound to any specific implementation.	Wordsmithing
D-CR-005	Interoperability	MAY	A choreography MAY provide an extensible binding mechanism such that choreographies could be bound to diverse technologies.	Wordsmithing "Implementation"
D-CR-010	Charter Constraint	MUST	All specified choreography descriptions MUST be compatible with WSDL 1.2.	Wordsmithing needed (binding mechanism, diverse technology)
D-CR-001	Interfaces	MUST	associated with it to enable its behavior to be explained.	
D-CR-015	Composition	MAY	A choreography MAY have run time changes that allow the actual participants to vary.	
D-CR-003	Composition	MUST	It MUST be possible to make a choreography C2 dependent on another choreography C1 such that you can only create a new instance of C2 after a related instance of C1 has been completed.	
D-CR-019				