

Proposal: WS-AT policy assertion ATAssertion MUST/MAY/MUST IGNORE semantics

WS-Atomic Transaction policy assertion <wsat:ATAssertion> currently specifies MUST/MAY/SHOULD NOT flow transaction semantics. There is an asymmetry in the definition of <wsat:ATAssertion> and the absence of <wsat:ATAssertion> that this writeup illustrates could result in an interop issue. This asymmetry could be addressed by changing the MUST to a SHOULD. (Monica will be sending out an alternative proposal to consider that approach to addressing the asymmetry.) This writeup concentrates on addressing the asymmetry by changing the SHOULD NOT to MUST IGNORE. The table below is derived from normative text in WS-AT Specification(link to specific version at end), section 5.2, lines 247-257, and illustrates that it is undefined if a target service MUST honor a txn context that SHOULD NOT have been flowed with the requestor's message.

Transaction 'T1' represents a transaction scope created in requestor.

WS-AT Policy Assertion	Requestor's Txn Scope	Txn Context flowed in Requestor Msg	Txn Context used in Target Service
<wsat:ATAssertion>	none	None	None
	T1	T1	T1
Absence of <wsat:ATAssertion>	none	none	None
	T1	mostly none and sometimes T1 ^[a]	Undefined <i>(infer mostly none and sometimes T1)[b]</i>
<wsat:ATAssertion wsp:Optional="true"/>	none	none	none
	T1	T1	T1

[a] SHOULD NOT flow transaction is treated as most implementations would not propagate but some may. At last OASIS WS-TX conference call, SHOULD NOT evolved to undefined during discussions.

Table below illustrates the non-deterministic nature when considering different combinations of implementations that are strict or lenient in the interpretation of SHOULD NOT flow a transaction context with requestor's message and how the target service interprets the absence of <wsat:ATAssertion>. Note that the table below is relating four different implementations of ws-at. The most lenient interpretation could decide that all variants of <wsat:ATAssertion> permit a transaction to be flowed, so there is no need to look at policy at all on either the requestor or target service side.

	Target Service (using ws-at impl C with strict SHOULD NOT)	Target Service (using ws-at impl D with lenient SHOULD NOT)
Requestor of target service (using ws-at impl A with strict SHOULD NOT)	none	none
Requestor of target service (using ws-at impl B with lenient)	none	T1

	Target Service (using ws-at impl C with strict SHOULD NOT)	Target Service (using ws-at impl D with lenient SHOULD NOT)
SHOULD NOT)		

The above uncertainty could be addressed by explicitly stating that a target service MUST IGNORE a txn context flowed in a requestor's message when there is no <wsat:ATAssertion/>. Below is a new table illustrating predictability of the txn context that the target service is executed within.

WS-AT Policy Assertion	Requestor's Txn Scope	Txn Context flowed in Requestor Msg	Txn Context used in Target Service
<wsat:ATAssertion>	none	None	None
	T1	T1	T1
Absence of <wsat:ATAssertion>	none	none	None
	T1	mostly none and sometimes T1 ^[a]	None^[b]
<wsat:ATAssertion wsp:Optional="true"/>	none	none	none
	T1	T1	T1

[a] SHOULD NOT flow transaction is treated as most implementations would not propagate but some ma

[b] Based on stating that a WS-AT implementation MUST IGNORE a txn context that was flowed and the target service did not have a <wsat:ATAssertion> policy.

Link to the document referenced (respectively):

<http://www.oasis-open.org/committees/download.php/17325/wstx-wsat-1.1-spec-cd-01.pdf>

(public: [1]

<http://www.oasis-open.org/committees/download.php/17325/wstx-wsat-1.1-spec-cd-01.pdf>)