

### 5.3.2 QName and XML Information Set representation

Web Services Policy language allows Assertion Authors to invent their own XML dialects to represent policy assertions. The policy language relies only on the policy assertion XML element QName. This QName is unique and identifies the behavior represented by a policy assertion. Assertion Authors have the option to represent an assertion parameter as a child element (by leveraging natural XML nesting) or an attribute of an assertion. The general guidelines on when to use XML elements versus attributes apply is:

- **For attributes:** Use a unique QName to identify a distinct behavior
- **For XML outline (child element):** Provide an XML outline (plus an XML schema document) to specify the syntax of an assertion.

Deleted: .

Formatted: Highlight

Formatted: Bullets and Numbering

Deleted: and p

=====

## 5.4 Comparison of Nested and Parameterized Assertions

There are two different ways to provide additional **qualifying** information in an assertion beyond its type. We cover these two cases below followed by a comparison of these approaches targeting when to use either of the approach.

=====

### 5.4.1 Assertions with Parameters

Policy assertion parameters are the opaque payload of an assertion. Parameters carry additional useful information for engaging the behavior described by an assertion and are preserved through policy processing such as **normalization**, merge and policy intersection. Requesters may use policy intersection to select a compatible policy alternative for an interaction. Assertion parameters do not affect the outcome of policy intersection unless the assertion specifies domain specific processing for policy intersection.

In the XML representation of a policy assertion, the child elements and attributes of the assertion excluding child elements and attributes from the policy language namespace name are the assertion parameters.

Deleted: ¶

Formatted: Font: (Default) Verdana

### 5.4.2 Nested Assertions

...[Additional information could be needed in the description to explain the relationship between the nested assertion semantics and the parent assertion.](#)

...**Best practice 12: Declare Nested Assertions**

Deleted: e

Deleted: might

If there is a nested policy expression, an assertion description should declare it and enumerate the nested policy assertions that are allowed.

.....If the assertion authors want to delegate the processing to the framework, utilizing nesting should be considered. Otherwise, domain specific comparison algorithms may need to be devised and be delegated to the specific domain handlers that are not visible to the WS-Policy framework. However, **the requirement to use** domain specific intersection processing **could reduce interoperability and could increase** the burden to implement an assertion.

.....

=====

Formatted: Highlight

Formatted: Highlight

Deleted: s

Deleted: s