All Fields marked with * are mandatory.

Change Request #:	**254**
Assigned OGC Document #:	**12-150**
Name:	*Panagiotis (Peter) A. Vretanos
Organization:	*CubeWerx Inc.
Email:	*pvretano@techie.com
Document Name/Version:	*Web Feature Service 2.0 Interface Standard (also ISO 19142) / 2.0
OGC Project Document:	*09-025r1

If this is a revision of a previous submission and you have a Change Request Number, then check here: ☐

Enter the CR number here:

Enter the Revsion Number that you are revising here:

Title:	* [WFS 2.0] Decouple the query model from the presenation model
Source:	*Eric.Boisvert@RNCan-NRCan.gc.ca
Work item code:	
Category:	* C (Functional modification of feature)
Reason for change:	*

1- expose a simpler query model (or a collection of simple query models adapted for specific use cases), which can have hidden side effect. A great example in geology is named age semantic (Devonian is part of Paleozoic, if someone filters on Palezoic, Devonian should match without being explicitly listed and this logic can be hidden in the "view")
2- reuse fes:Filter syntax, which provide a broader filtering capability (including spatial filtering)
3- Lower the requirements to comply to domain models (like GeoSciML). Serializing an arbitrary model into a common model is much easier that mapping a request to an arbitrary model. Having a community to comply to a series of simple view can be done using a bunch of XSLT
4- can handle tricky query that can't be expressed in Filter (my favorite is the boolean scoping problem: the problem that arises when AND and OR operator are used to filter a complex feature inner components where its cardinality is 0..* or 1..*. For example, we might want to extract all wells that have a till interval where the

	top of that interval is above 25m (assuming top measurements go downward from the surface). MATERIAL = â��Tillâ�� AND top < 25 Unfortunately, this expression is ambiguous because it might mean either â��select all well that have any interval above 25 and any interval that are tillâ�� while we only want wells that have intervals that satisfies both conditions. You might either get a well that has one interval that is till and the another interval that is above 25 (but is sand for example). The problem is related to the fact the scope of the boolean operation is the well, not the interval. Actually, the result might depend on the implementation details of the service. "matchaction" does not solve this)
Summary of change:	* Proposal #1: The current StoredQuery syntax limit the parameters comparaison to equality (=) Adapted from an example you provided us in 2009 - I could not find an example of StoredQuery in 09-025r1 (definitively need one) <wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:ogc="http://ww.opengis.org/ogc/2.0" xmlns:gml="http://www.opengis.net/gm1/3.2" xmlns:xsi="http://ww.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://ww.opengis.net/wfs/2.0 ../../wfs.xsd" resolve="all" resolveDepth="2" service="WFS" version="2.0.0"> <wfs:StoredQuery id="urn:CubeWerx:StoredQueries:GeologyByAge"> <wfs:Parameter name="age">245.6</wfs:Parameter> </wfs:Parameter> </wfs:StoredQuery> </wfs:GetFeature> This should also be valid <?xml version="1.0"?> <wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:ogc="http://ww.opengis.org/ogc/2.0" xmlns:gml="http://www.opengis.net/gm1/3.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:fes="http://ww.opengis.org/fes/2.0" resolve="all" resolveDepth="2" service="WFS" version="2.0.0" xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/2.0/wfs.xsd"> <wfs:StoredQuery id="urn:CubeWerx:StoredQueries:GeologyByAge"> <fes:Filter> <fes:PropertyIsGreaterThan> <fes:ValueReference>age</fes:ValueReference> <fes:Literal>245.3</fes:Literal> </fes:PropertyIsGreaterThan> </fes:Filter> </wfs:StoredQuery> </wfs:GetFeature> In the latter example, the "StoredQuery" acts more like a view, because it exposes a simple list of queriable parameters but it returns a potentially more complex feature. . It also means that parameters must be optional, so

	<wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:ogc="http://ww.opengis.org/ogc/2.0" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:fes="http://ww.opengis.org/fes/2.0" resolve="all" resolveDepth="2" service="WFS" version="2.0.0" xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/2.0/wfs.xsd"> <wfs:StoredQuery id="urn:CubeWerx:StoredQueries:GeologyByAge"/> </wfs:GetFeature> Returns everything (with respect to maxFeatures and server limitation) Proposal #2: > The way you are describing it, it sounds more like you want a separation between presentables and queryables ... is my impression correct? Correct > If yes, wouldn't a better change request be to put an option on the DescribeFeatureType to be able to ask for the presentable schema(s) or the queryable schema(s)? Right now they are the same thing but they don't need to be. Just thinking out loud ... If this ambiguity can be resolved at the GetFeature operation, it might indeed be a better solution. Actually, this is just what I look for. A disconnected query model and response model. When I first heard of StoredProcedure, this was my hope.
Consequences if not approved:	
Clauses affected:	* 9, 11
Additional Documents affected:	
Supporting Documentation:	
Comments:	
Status:	Assigned
Assigned To:	WFS/FES SWG
Disposition:	Referred and Posted