

# DSpace 2.0 Modelling DSpace Objects

## Technical Specification for DSpace+2.0 Data Model API.

This page should contain a technical drafting of the requirements around the DSpace Data Model drawn from the comparative analysis done in... [DSpace+2.0-Comparing\\_Existing\\_Technologies](#)

We agree that an anemic data model is most appropriate. That is, a data model that is an API almost entirely devoid of implementation at its base. Initially some concepts concerning the look of this API were expressed in the Architectural Review here:

[ArchReviewDataModel](#)

*I take this further and suggest that the "Domain aspect" of this model (I.E. Communities/Collections/Items/Manifestations...) should be an abstraction modeled within the data and not expressed as a Concrete set of Java Classes. This means that a DSpaceObject is just a URI used to reference a resources that may be stored in various different states within each service. For example: an Assetstore will store the Content representation of a DSpaceObject, while the MetadataService will store a Metadata representation of the object. This means that Service can evolve separate "non-dominant" data models that are more tailored to interacting with them. --Mark Diggory 15:49, 16 September 2008 (EDT)*