ArchReviewOldManifesto

Original Manifesto for the DSpace Architecture Review

For the proposed DSpace architecture review and development roadmap we are seeking a balanced approach to addressing the system's current limitations and problems. We need to avoid 'from scratch' redevelopment of any part of DSpace unless absolutely necessary, and instead look for an evolutionary approach that leverages the aspects of the system that work well wherever possible. This manifesto aims to define the principles derived from this approach to serve as a touchstone when considering DSpace architectural issues. The developers of XML found it useful to define 10 simple goals to guide them; this is a similar approach.

Reference is made below to the application/UI layer and the back-end layer. In the current architecture, the application/UI layer is the top-most layer, and the back-end is equivalent to the usiness Logic and epository/Storage layers. Functionality resides at all three layers, but variation in that functionality tends to decrease as you move down the stack.

#Back-end separately deployable from UIs and End-User Applications

In the current layering of the application there is a 'back-end' of code concerned with persistent storage, metadata handling, basic repository query, and most aspects of ongoing digital collection curation. The DSpace platform should separate end-user applications/interface from the core in such a way that the former can support rapid development and redeployment without requiring redeployment to the more stable back end, while retaining efficiency and scalability of the whole system.

#DSpace Should Support Innovation in UIs and End-User Applications

The vast majority of the innovation and customization in DSpace has been in the user interface and application layers. The system architecture should make these customizations and adaptations as easy to implement as possible without affecting the repository back-end.

#DSpace Should Promote Stability in the back-end

System administrators increasingly want to use DSpace to provide high availability, robust, reliable, stable repository services.

#DSpace Should Seek a balance between Standard Network Protocols and Efficient APIs

DSpace applications/UIs should interoperate with the back-end through a minimal set of open standard, language independent network protocols, to the extent possible without adversely affecting efficiency (performance) or scalability in the system. Adopting open standard protocols will promote innovation in the application area without programming language restrictions, but this must be balanced with system performance that can sometimes be adversely affected by such standards.

#Limited extensibility of the back-end, promote extension in applications

Extension of the functionality should be possible, but through limited, defined points (e.g. Observers and Interceptors). y promoting extension in 'user space' the stability of the core is minimally compromised.

#Application scope is not definable or bounded

DSpace cannot support all the variable and emerging definitions and innovations in the repository space in a single interface application. DSpace should seek to provide out-of-the-box applications for the most common use cases (e.g. an OA preprints app, a general content archive) and modular support for the easy construction of new applications.

#DSpace releases should be minimally disruptive

The architecture should reinforce good behavior in making changes/customizations/improvements to future releases of the system, so that upgrades are minimally disruptive for current adopters. For example, customizations that require extensive changes to the database should be avoided where possible.