According to their varying focuses, generic repository software packages (e.g. Fedora Commons) provide homogenous interfaces to potentially very heterogeneous information (images, PDF files, XML documents, etc). Communication with client applications in most cases uses the Hyper-text Transfer Protocol (HTTP).

Research projects which decide to use such software packages get sufficient solutions for scenarios in which they have to handle heterogeneous or unpredictable data, or data of a volatile structure.

As a development project the Person Data Repository (PDR) in its first term was confronted with the latter case, because the data types and their serialisation were subject to frequent changes and extensions. After this procedure it became clear that the PDR has to deal with large numbers of data objects, mainly in the form of small XML documents.

Without challenging the strengths of generic solutions, it is virtually unavoidable that there may occur weaknesses in use cases with specific homogenous data and well-defined types, as it was the case in the PDR project. Thus, we decided to move away from Fedora Commons and replace it with a custom-made solution, including our own module for data storage (PaDRe) and a powerful index provided by Apache Solr.

In our opinion this approach might be beneficial to other projects with relative homogeneous data too.

It enables involved software developers to control every single aspect of the actual processes of data storage, queries, and dissemination. Thus, a solution can be found which perfectly fits the project’s purposes in terms of the needed range of functions, scalability, and performance. However, there are always two sides to a coin.

The approach prevents involved software developers from simply using readily utilisable features of a generic solution. Many things have to be developed and sometimes reinvented with focus on the project’s objective. This is expensive in terms of time and money and may have an immediate effect on aspects of a project’s workflow and progress which depend on a stable and functional data storage.

Interested? Let’s talk.

Further information
fkoerner@bbaw.de
http://pdr.bbaw.de

Software components of PDR’s repository structure with possible alternatives and extensions

**pros**
- possible optimisation (e.g. by an enhanced internal structure of used DBMS, or index)
- better scalability (e.g. by integration of other DBMS, or more indices or further components)
- extendable range of features (e.g. customised interfaces for specific purposes)

**cons**
- readily useable features of a generic solution are lost
- development is expensive in terms of time and money
- trained staff with experiences in the development of distributed systems is needed