Generalizing B-Fabric towards an Infrastructure for Collaborative Research in Switzerland

Deliverable No D1:

Specification for Ad-hoc Coupling of External Data Stores

July 2009

Document Information

Project	
Project Title	Generalizing B-Fabric towards an Infrastructure for Collaborative Research in Switzerland
Project Start	01.06.2009
Project Sponsor	SWITCH (AAA/SWITCH Project)
Project Number	UZH.5

Document	
Title	Specification for Ad-hoc Coupling of External Data Stores
Date	31.7.2009
Author(s)	Fuat Akal, Dieter Joho, Can Türker
Total Number of Pages	11
File Name	D1.docx
Key words	Ad-hoc coupling, external data store, data provider, workflow, application

Table of Contents

Generalizing B-Fabric towards an Infrastructure for Collaborative Research in Switzerland 1

1 Introduction 5

2 Current Implementation: Static Coupling of Data Stores 6

2.1 Data Provider 7

2.2 Workflows 7

3 Target: Ad-hoc Coupling of Data Stores with B-Fabric 8

3.1 Data Provider 9

3.2 Workflows 10

3.3 Application 10

4 Implementation Tasks 10

5 Conclusion 11

Summary

In its current implementation, B-Fabric allows the coupling of external data stores via a static configuration using XML files. The coupling of an external data store consequently requires a complete rebuild of the system. This document sketches the plan to implement a more general concept that allows an ad-hoc coupling and running of any external data source with B-Fabric.

1 Introduction

In today's laboratories several different instruments, robots, and applications generate a huge amount of data. Whereas almost every instrument manufacturer has its own concept of data placement, there is often lack of a center-wide concept. It is difficult to keep track of what data is produced and where it is stored. This is mainly a problem when introducing a safe backup strategy or when scientists would like to process data that was produced before. Therefore, it is vital to have a solution for center-wide data management.

It is not only the heterogeneity of the data generators and stores that makes it hard to install a center-wide data management solution. It is also the fact that the instruments, applications and technologies evolve extremely fast. To have the best and the latest laboratory equipment often means to change the manufacturer and the technology. As a consequence there is a constant need to add and remove instrument connectors from the data management tool. Just providing data connectors for the instruments of a certain manufacturer is not sufficient.

In a standard laboratory, there are typically many different instruments that generate data. Some of them write the data to a local disc and some are able to write to an NFS share on the network etc. There are also applications that come as a black box and are not configurable about where to place their data. A few instruments come with a fully fledged user and permission management system and some need to be run under administrator privileges. That is, there is no common instrument handling procedure.

Shortly, new applications and instruments are installed at life sciences data processing centers very often. This means that there is a need for a highly evolving data management system. It would be the best if the instrument specialists themselves were able to integrate their instruments by simply changing the configuration of the data management system during runtime.

The previous B-Fabric solution was originally designed for an environment where evolution dynamics is not so high – at least at this project phase it was not part of the main goals. Early 2005, when started with the implementation of B-Fabric, the concept of a data provider was invented and realized. A data provider is a configurable Java component that allows coupling an external data store with B-Fabric. The disadvantage of the implemented approach lies in the static configuration of the data provider, i.e., first the

configuration is fixed before compile-time and then this configuration is compiled into corresponding Java objects. Consequently, coupling a new data store with B-Fabric requires 1) writing and including a configuration file and 2) complete rebuild of the system.

2 Current Implementation: Static Coupling of Data Stores

There are two main components in the data store coupling schema of B-Fabric as shown in Figure 1:

- *Data Provider* and
- *Workflow*.

The data provider provides an interface between the data store and B-Fabric. In order to couple a new data store with B-Fabric, the system deployer has to choose the appropriate data provider for the data store and configure it. Then, the system deployer has to choose an appropriate workflow, which handles the accessing and processing of the import data.

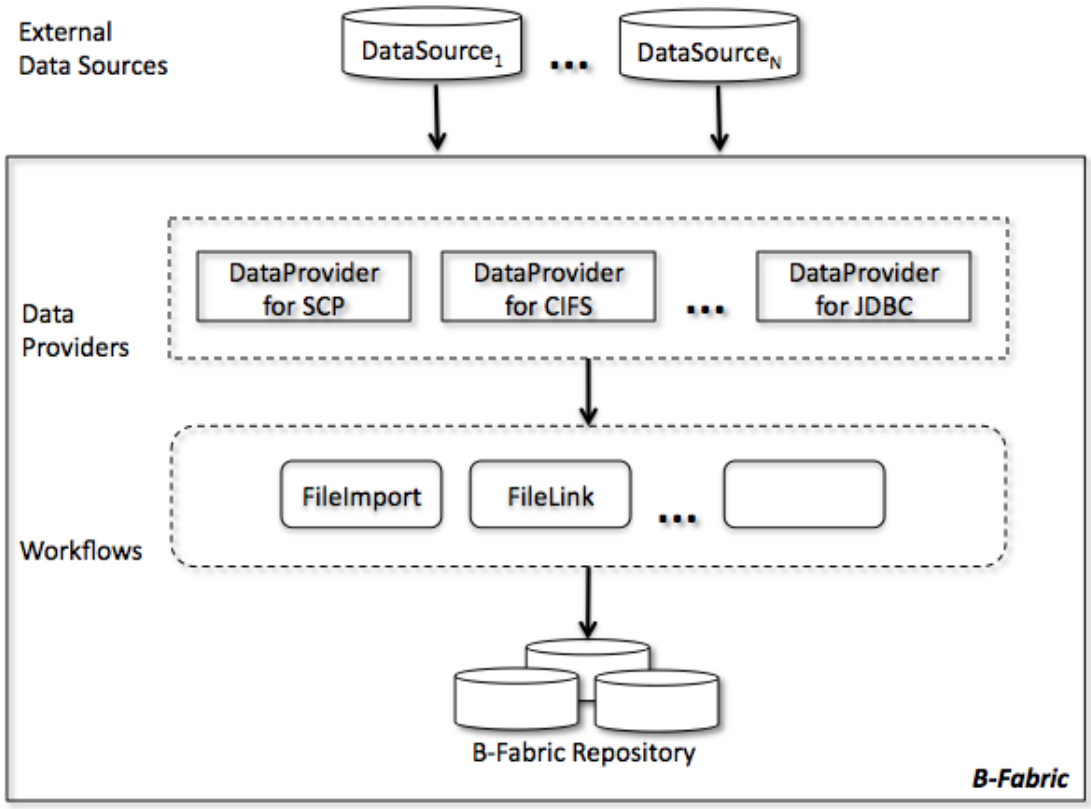

Figure 1: Existing Architecture

2.1 Data Provider

The data provider is the connector to the instrument itself. It is responsible for retrieving a list of data resources that can be imported into B-Fabric. The selected data resources are then transformed into B-Fabric data.

Data providers are instrument-specific. This means that there has to be a data provider for every single instrument that is attached to B-Fabric. The reason is that the data provider contains all the instrument-specific configurations and the workflow that is used to import and process the data.

Attributes that have to be provided by a data provider are bound to the nature of the desired instrument. For an SCP (for a Unix-based system) or a CIFS (for a Windows based system) based data provider, for instance, the data provider must among others provide the attributes below:

- **Host**: The host name or address of the data store
- **Base Path**: The base path of the data directory
- **Include Pattern**: Filter for the files to be included among the listed data resources
- **Exclude Pattern**: Filter for the files to be excluded among the listed data resources

While implementing data providers, we will focus on SCP and CIFS based solutions since our instruments at the FGCZ can be accessible with either one. However, one could think of supporting other data store types. Writing a JDBC based data provider could be used to directly import database records into B-Fabric. Maybe a SOAP-based data provider could be used to connect any web services.

2.2 Workflows

The workflow is responsible for processing the (input) data as well as for all kind of data preparation and handling procedures. Although there are many different instruments at the FGCZ, there are just a few workflows to handle them. Two main workflows are:

- **FileImport**: Copy the selected data resources into the B-Fabric repository and let the user annotate the data

- **FileLink**: Link the selected data resources into the B-Fabric repository and let the user annotate the data

The file import workflow is used when the generated data of the instrument is not located in a safe repository. The file link workflow, on the other hand, is used when the generated data is already in a safe file repository. In this case there is no need to duplicate the data.

There are a few special workflows for certain instruments. These are workflows that include steps to parse and process the data before importing it into the repository. Developing and implementing such specialized workflows could be a tricky task – which usually requires deeper computer science knowledge about workflow specification and management. In a typical research environment like the Functional Genomics Center Zurich, this effort pays only off for workflows that are stable and often used.

3 Target: Ad-hoc Coupling of Data Stores with B-Fabric

To allow fast and on the fly coupling of new data stores (instruments) with B-Fabric, it is vital to have a dynamic configurable connector. The topping of it all would even be that the connectors could be managed by the scientists and instrument specialists themselves, without involvement of a software developer.

To achieve this goal, we are going to change the system architecture by introducing a new component called "Application" (see Figure 2).

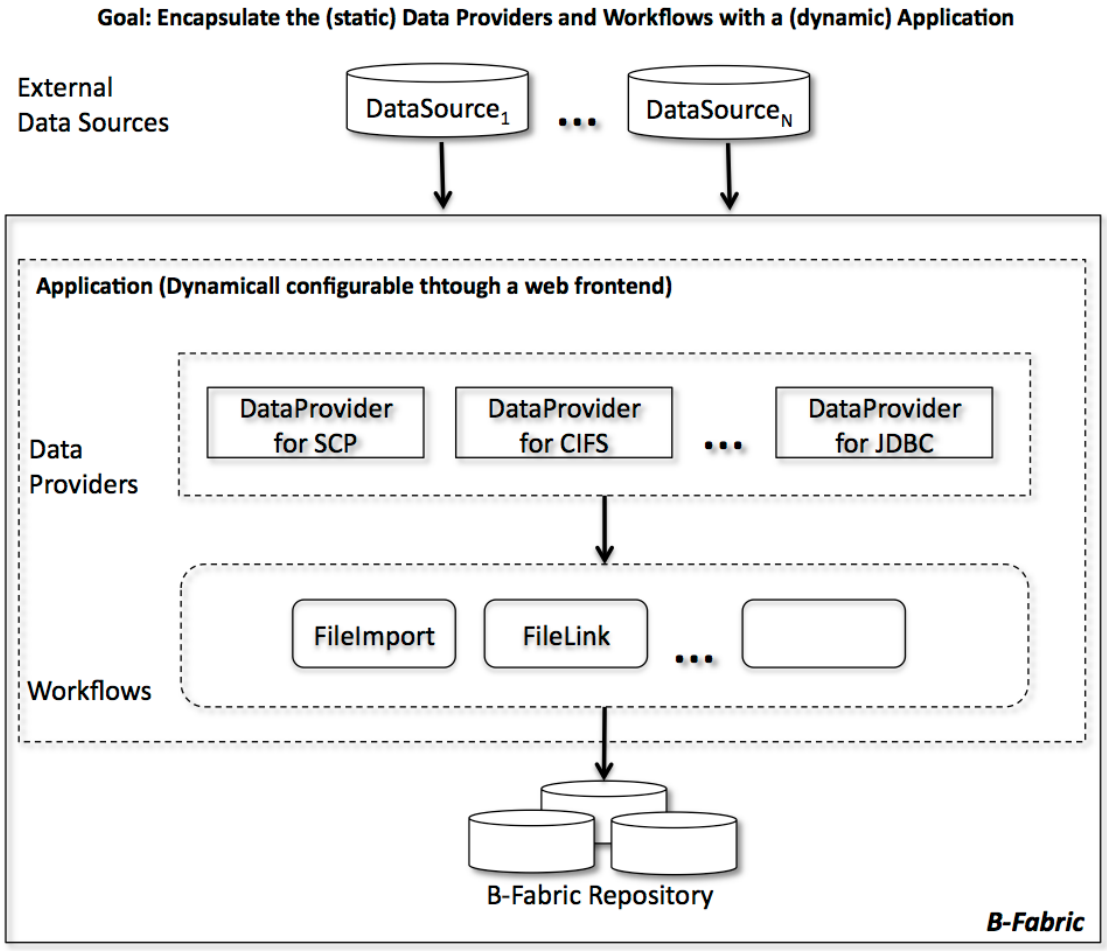

Figure 2: New Architecture

An application provides an abstraction level that allows separating the more static and center-specific implementation from a dynamic and instrument-specific configuration.

3.1 Data Provider

The data provider is reduced to a component that contains just general functionality to connect to any data store (instrument). This means that there is a data provider for every single data transfer protocol that is used (SCP, CIFS). This component does not contain any data store-specific settings (data store location, path to data, etc.) anymore. The data provider is the part of the application that has to be implemented by a software developer.

3.2 Workflows

Whereas workflows are tightly coupled with the data provider in the current B-Fabric system, it will be loosely-coupled with the "Application" component in the new system.

3.3 Application

The application is a component that will be introduced in the new system. It will contain all the data store-specific configurations and settings, such as the location of the data store and its resources, the data provider used to transfer the data resources and the workflow to handle and process the imported data.

This additional level of abstraction will allow the instrument specialists and biologists to add and change data store connectors to B-Fabric on the fly, without programming knowledge, rebuild or even restart of the B-Fabric system.

4 Implementation Tasks

The realization of the application concept among others requires:

- Introduce a new Java class Application to hold the configuration information which was maintained before in XML files.
- Ensure persistency of Application objects (implement object-relational Java-SQL mapping)
- Adapt the process to build B-Fabric.
- Provide proper Cocoon Forms and Java scripts to add, edit, delete applications
- Revise the import data selector screen. Ad-hoc read and list all available data import applications according to the technologies.
- Provide proper Cocoon Forms and Java scripts to run the applications.

- Implement a concept for ad-hoc inclusion of run buttons on the Cocoon screens which contain objects that might be used as input for that application.

- Provide a screen for listing and managing all available applications.

Since the workflows for importing and linking data from an external data store are already implemented, they can be reused by the application concept.

5 Conclusion

The ad-hoc data store coupling of B-Fabric is a simple solution to a difficult problem. It is simple as it should take few minutes to add a new data store to B-Fabric on run time by selecting the appropriate data provider and workflow. All the complexity is wrapped into the "Data Provider" and the "Workflow". The concept of an application is introduced as means to ad-hoc glue a data provider with a workflow.

An instrument specialist (with basic bio-informatics knowledge) should be able to maintain such connector to their instrument himself. The configuration of such a connector (or, an application in another word) will usually take only a few minutes. In particular, it will not require any programming work. The configuration can be done via a simple web interface.

Another effect of the generalization sketched in this document is that there is no need to redeploy B-Fabric anymore, for instance, if an instrument is upgraded. The connector can be adapted to the changes immediately so that there is no danger of losing important data.