

## Abstract

The HOBIT [1] initiative is funded by the Helmholtz Gesellschaft. It is dedicated to form the core of a network linking bioinformatic centers together. It shall be understood as an initial organisational and technological platform for interconnection of bioinformatics activities.

The Bielefeld University Bioinformatics Server [2] supports Internetbased collaborative research and education in bioinformatics and takes part in HOBIT.

## Helmholtz Open Bioinformatics Technology (HOBIT)

The aim of the network is to concatenate applications and resources in a uniform way, providing an efficient communication tier for bioinformatics resource access. There are eleven members within HOBIT - universities and institutes located all over Germany.

Started in 2003 the members decided to use the new WebService technology as base for the connection of different bioinformatical tools. Most of them exist before as stand alone programs or with an HTML Interface for remote usage like tools on BiBiServ. None of them were accessible as WebService before.

Creating WebServices was the first step, which has to be done for members of HOBIT.

In the first phase all members of HOBIT implemented their tools as WebService.

In the second phase, which is the current state, is to combine these WebServices to build up a network of bioinformatical tools.

## The Bielefeld University Bioinformatics Server (BiBiServ)

BiBiServ was established in 1996 and provides software tools, together with a repository of educational material.

All tools are available online for free, both to academic and commercial users. Some of the software tools are distributed also or only for local installation. In the latter case commercial users may have to pay a license fee.

Currently, 29 software tools are available. These include tools from different areas such as Genome Comparison, Alignments, Primer Design, RNA Structures and Evolutionary Relationships.

Tools present a HTML interface, which is filled by the user and starts the program on the server machine.

## WebServices

Providing access using WebServices makes it possible to access the tools from other programs automatically and compute results remotely without technical knowledge of the program.

The underlying protocol SOAP [4] is defined by the W3C [5] and uses *http* [6] as transport protocol. Communication to and from WebServices is performed in XML.

A WebService is described in a so called WSDL document (WebService Description Language [3]), it includes all required information for connecting a WebService tool, like the location of the WebService and the interface description. Distributing these WSDL documents makes it possible to connect WebService tools using SOAP-Implementations, which are available for all current programming languages like Java, Perl, PHP, C and C++.

## XML Schema

XML Schema Definition [7] (short: XSD) is a language by the W3C to describe structures of XML documents.

Using such a grammar makes it easy to combine different Web services. Output of one WebService can simply be pushed to another one, without performing any conversion.

## Asynchrony

A WebService contains two sides: Client and server. In general a WebService is a synchronous data exchange service of two points connected via a network. This means after a short time (less than the *http* timeout, which normally is 300 seconds) an answer of the WebService server is required.

Bioinformatic programs run usually much longer than five minutes. To avoid problems with timeouts, HOBIT recommends a technique called *Request and reply with polling*.

## Request and reply with polling

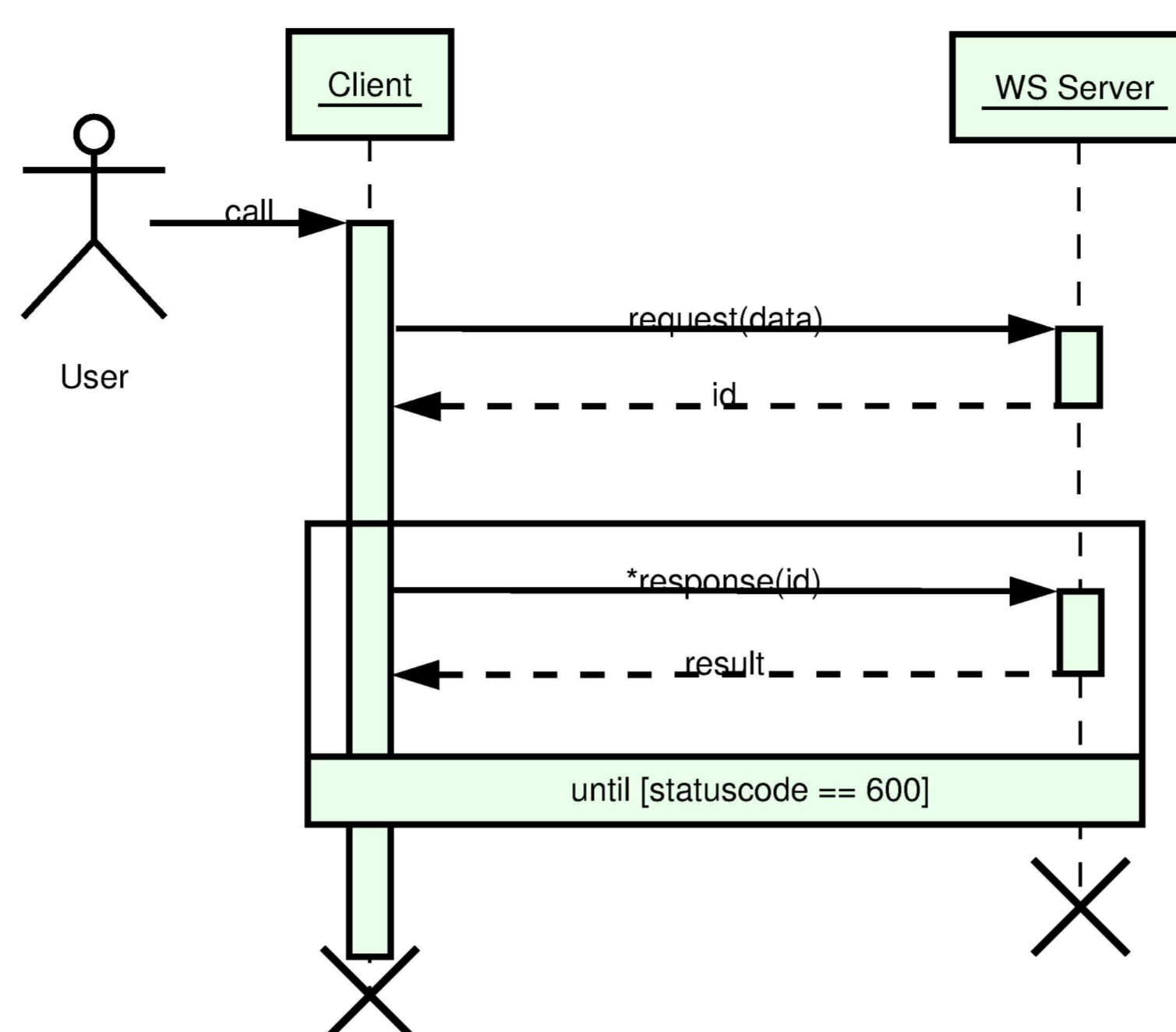


Figure 1: Asynchrony of WebServices using the *Request and reply with polling* technique

Like shown in Figure 1 the client side submits a job with the required data (program parameters and data etc.) and immediately gets an *id* after the job was started which normally takes some seconds (name convention *request*).

Afterwards the client side is able to ask for the result using this *id*. If the computation of the program is not finished, the client side gets back a statuscode with enhanced information as description. Otherwise the result of the call is returned (name convention *response*).

These statuscodes and their descriptions are defined by HOBIT to reach interoperability between WebServices, which use these statuscodes. They will be represented within the detail part of a SOAP Fault Message according to a HOBIT-specified XML Schema called *hobitStatuscode*. This can easily be evaluated by all SOAP implementations.

This polling technique completely avoids problems with timeouts. The client side can even request the results hours or days later or from another host, just with knowledge of the *id*.

## Repository

Next step for HOBIT and also the BiBiServ will be to set up a system, which provides access to information about the available WebServices, a so called UDDI Server. Members of HOBIT, or even collaboratives, could publish their WSDL document of a WebService to this server, thus users could automatically discover them.

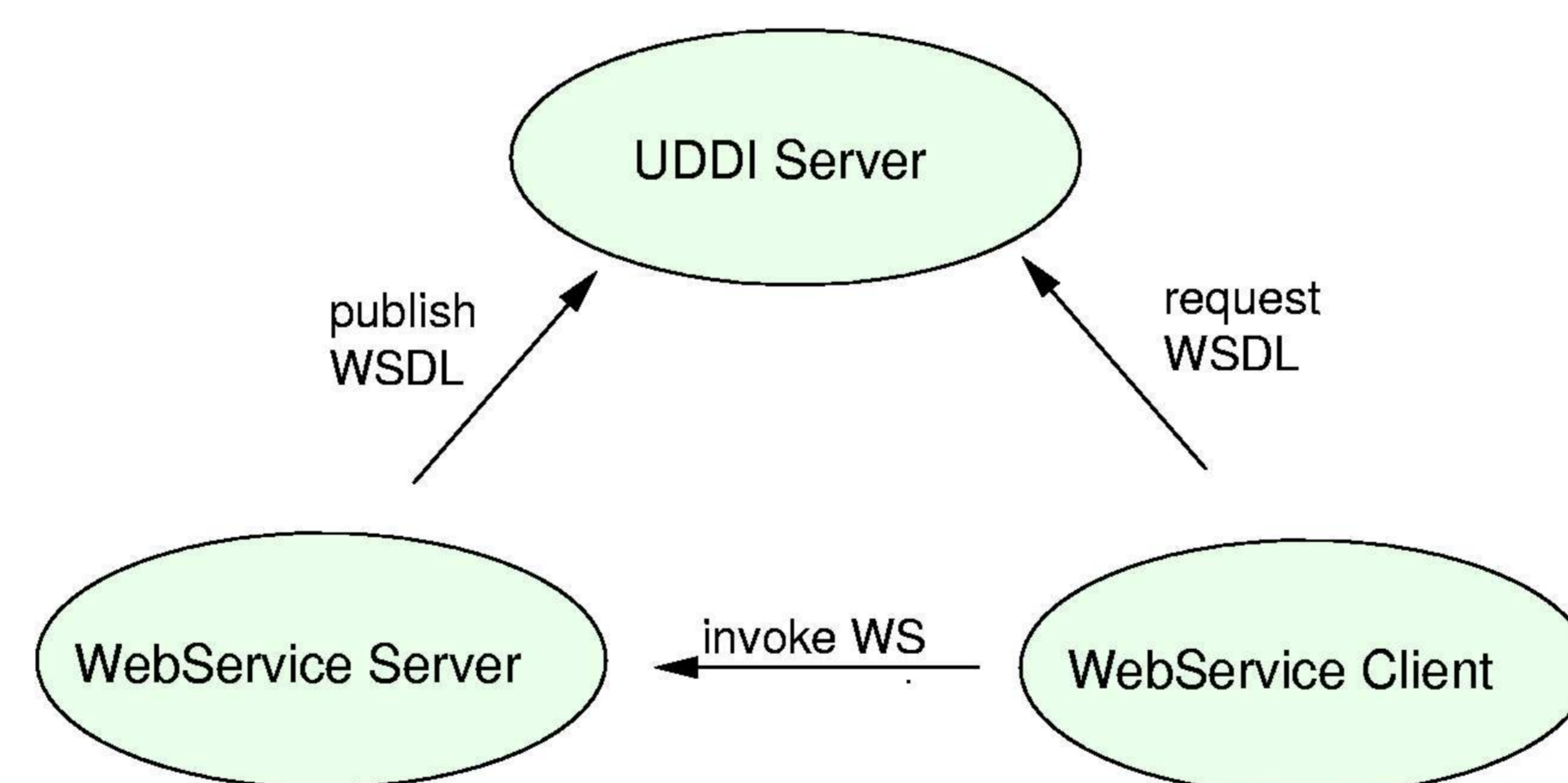


Figure 2: Basic idea of a repository for WebServices.

## BiBiWS - A library for async. WebServices

<http://bibiserv.techfak.uni-bielefeld.de/bibiws/>

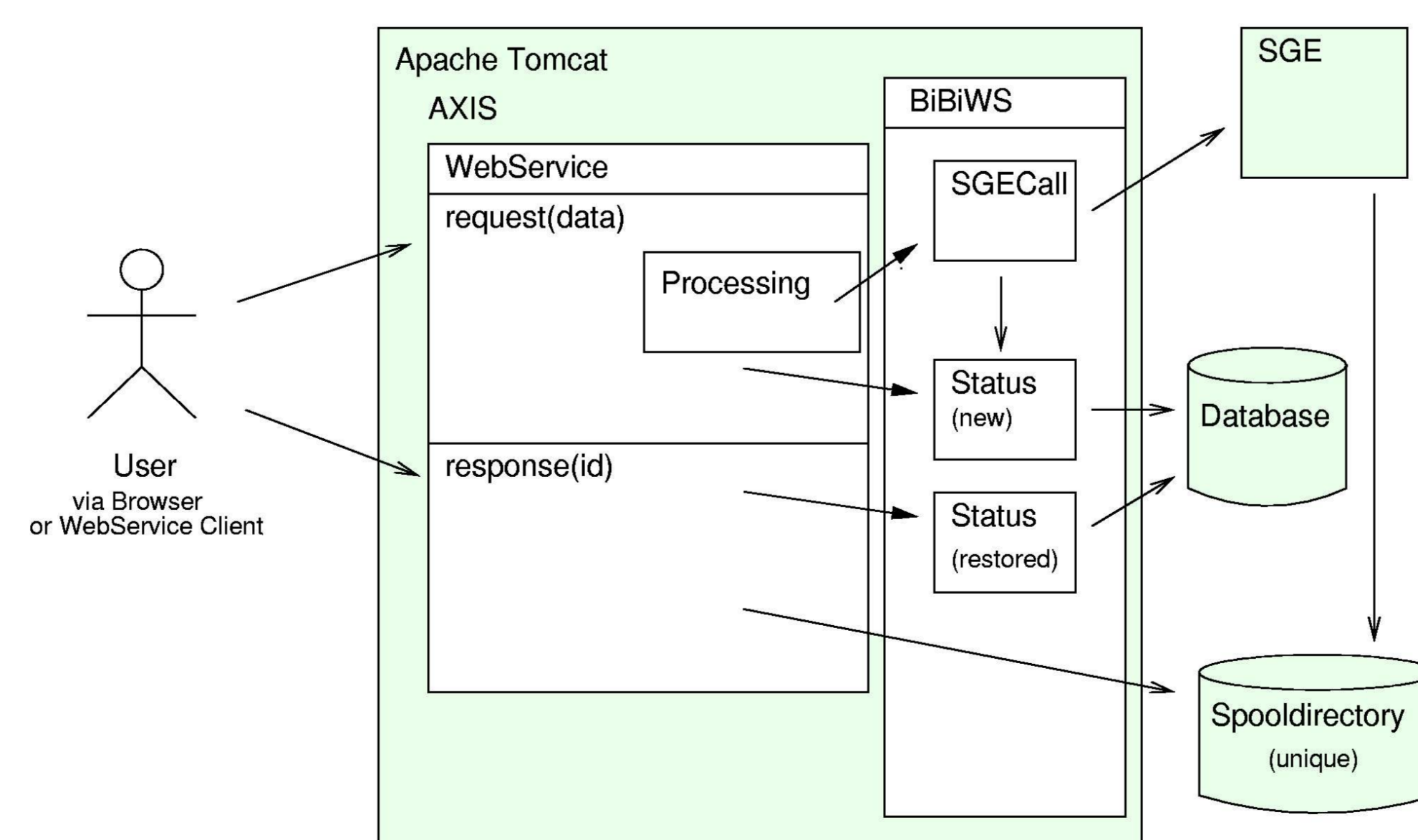


Figure 3: supporting the developer on asynchronous WebServices.

BiBiWS [7] was developed as the diploma thesis of Henning Mersch. It is a library for developing and publishing asynchronous WebServices at the BiBiServ.

It implements the presented *Request and reply with polling* technique and supports as much functionality to a developer as possible.

## Features

BiBiWS is designed to run the implementing WebServices on a Servletcontainer, like the *Apache Jakarta Tomcat Server* [8]. The main features of BiBiWS in short are:

- The *ids* used by the *Request and reply with polling* technique are generated and managed within a PostgreSQL database.
- Spool directories for calculated results and temporary data are created and managed for each submission separately.
- Jobs are submitted via a *Sun Grid Engine* [9] to a computing grid, so there is a load balancing over all connected computing hosts. This way, BiBiWS can provide a high amount of distributed compute resources, which can be used through both interactive web and WebService interfaces.
- For getting started, *templates* exists, which are the point to start developing the different WebServices. They present the developer an working example WebService, so he or she only has to modify for their own requirements.
- All generated data is automatically cleaned up after three days.

## HTML-Interface

Current BiBiServ tools, which do not base on BiBiWS, present the user an HTML form as interface for easy usage. BiBiWS also supports this.

The web server offers an HTML interface which can be accessed via a webbrowser. Internally the jobs are submitted to the WebServices server which processes the requests .

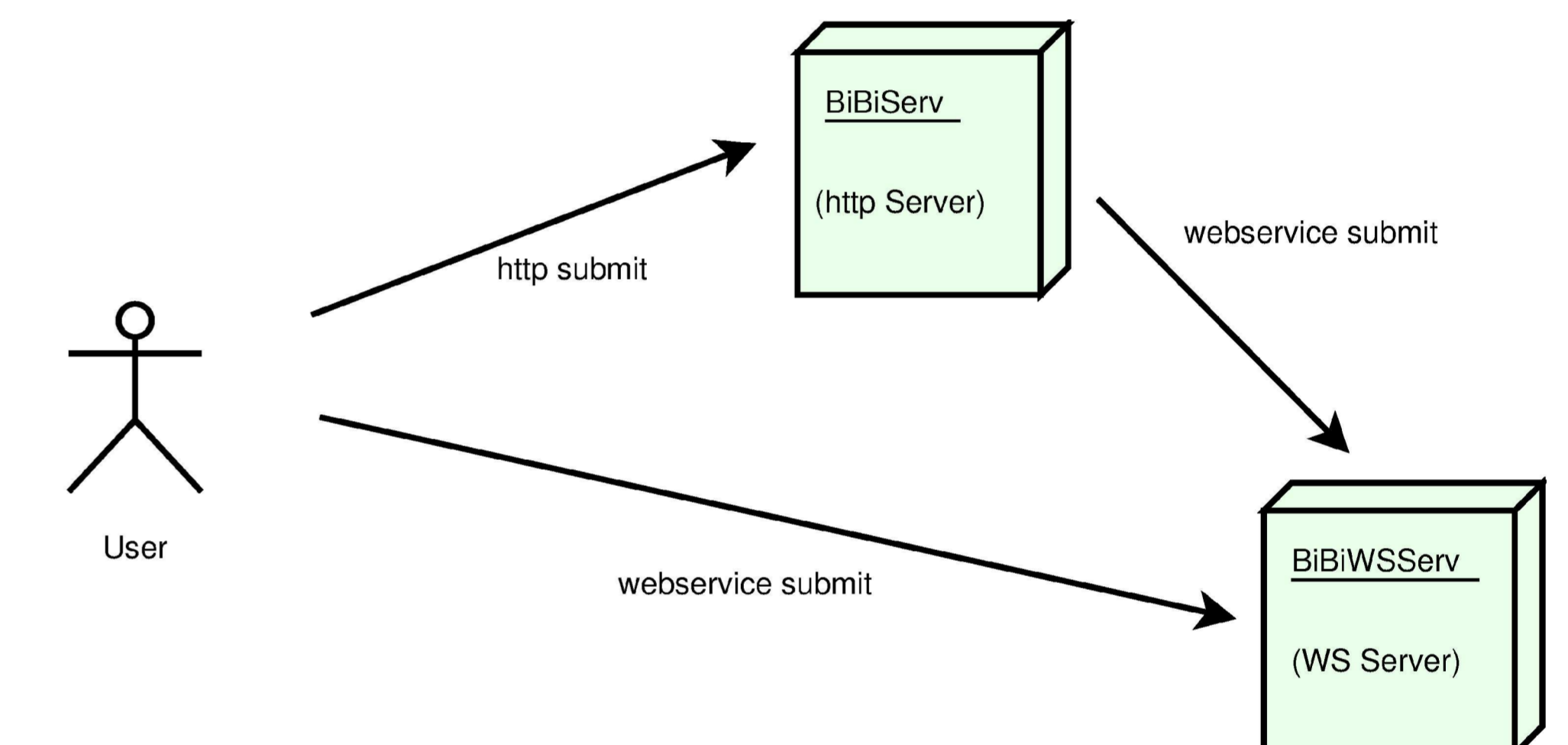


Figure 4: Submitting jobs to a BiBiServ tool based on BiBiWS.

## Current BiBiServ WebServices

<http://bibiserv.techfak.uni-bielefeld.de/wsdl/>

Currently BiBiServ offers four tools based on WebServices:

- **E2G** [10] efficiently maps large EST and cDNA data sets to genomic DNA. The server hosts huge EST databases of a few GB in size in indexed data structure.
- **REPuter** [11] provides state of the art software solutions to compute (and visualize) repeats in whole genomes or chromosomes.
- **SplitsTree** [12] uses the split decomposition method to analyze and visualize distance data, especially extracted from biological sequences.
- **RNAHybrid** [13] is for finding the minimum free energy hybridisation of a long and a short RNA. The hybridisation is performed in a kind of domain mode, i.e. the short sequence is hybridised to the best fitting part of the long one.

We are currently working on integrating more tools as WebServices, which will be finished shortly.

You can have a up-to-date look at

<http://bibiserv.techfak.uni-bielefeld.de/wsdl>

## References

- [1] HOBIT (Helmholtz Open Bioinformatics Technology) <<http://hobit.sourceforge.net>>
- [2] Technische Fakultät der Universität Bielefeld: The Bielefeld University Bioinformatics Server (BiBiServ) <<http://bibiserv.techfak.uni-bielefeld.de/>>
- [3] W3C: Web Services Description Language (WSDL) 1.1 <<http://www.w3.org/TR/wsdl>>
- [4] W3C: SOAP Version 1.2 <<http://www.w3.org/TR/soap12-part1/>>
- [5] The World Wide Web Consortium <<http://www.w3.org>>
- [6] W3C: Basic http <<http://www.w3.org/Protocols/HTTP/HTTP2.html>>
- [7] Henning Mersch: BiBiWS - Ein Framework für die Entwicklung asynchroner WebServices auf dem BiBiServ <<http://bibiserv.techfak.uni-bielefeld.de/bibiws/>>
- [8] The Apache Software Foundation: Apache Jakarta Tomcat <<http://jakarta.apache.org/tomcat/>>
- [9] Sun Microsystems: Grid Engine <<http://gridengine.sunsource.net/>>
- [10] J. Krüger, A. Sczyrba, S. Kurtz, R. Giegerich: e2g - An Interactive Web-Based Server for Efficiently Mapping large EST and cDNA sets to Genomic Sequences. Nucleic Acids Research, 2004, Vol.32, Web Server issue W301-304
- [11] S. Kurtz, J. V. Choudhuri, E. Ohlebusch, C. Schlieiermacher, J. Stoye, R. Giegerich: REPuter - The Manifold Applications of Repeat Analysis on a Genomic Scale. Nucleic Acids Research, 29(22):4633-4642, 2001
- [12] D. H. Huson: Analyzing and visualizing evolutionary data, Bioinformatics, 14(1): 68-73 (1998)
- [13] M. Rehmsmeier, P. Steffen, M. Höchsmann, R. Giegerich: Fast and effective prediction of microRNA/target duplexes. RNA, 10:1507-1517, 2004.