

# Web services for LOVD and Mutalyzer

**Jeroen F. J. Laros**

**Department of Human Genetics**

**Center for Human and Clinical Genetics**

**LUMC**

Strictly speaking, a web service has the following properties:

- Machine-to-machine interaction over a network.
- The interface is described in a machine-processable format.
- Communication via SOAP messages.

In practice: A (web) application programming interface.

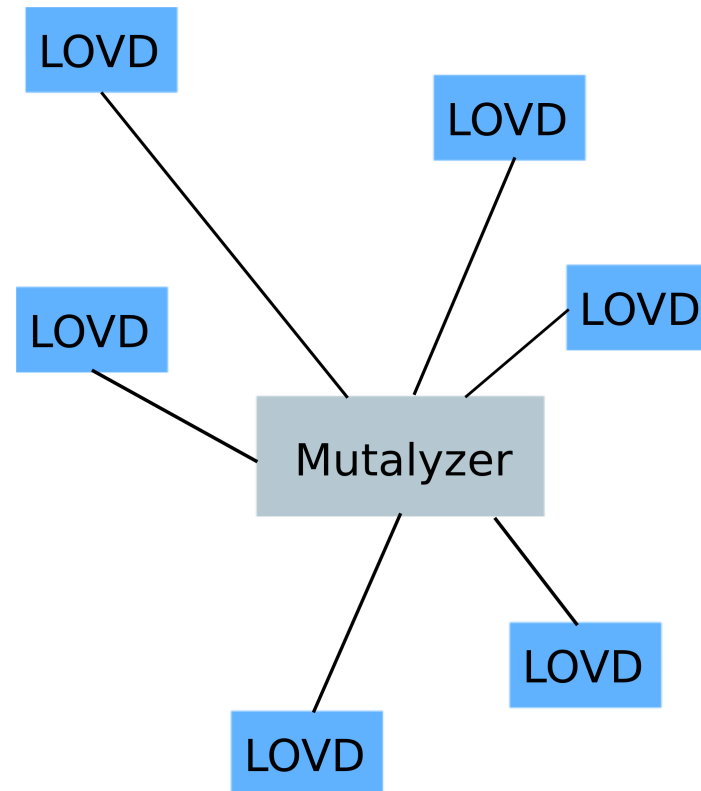
## Pros:

- No code duplication.
- No data duplication.
- Little or no client side configuration needed.
- Specialised servers.

## Cons:

- Dependency on network.
- Danger of a single point of failure.

As an example, we describe the connection between Mutalyzer and LOVD.



The server running Mutalyzer will run the web service.

The clients will be LOVD servers.

Mutalyzer is an LSDB curational tool.

- Variant nomenclature checker applying Human Genome Variation Society guidelines.

Current server running:

- Version 1.0.4.
- Recent stable Linux and Python versions.
- Subversion.
- Trac bug tracking system.

We are currently working on a new version.

Complete rewrite of the code:

- Improved performance.
- Restructured web interface.

Features:

- Context-free HGVS nomenclature parser.
- GenBank record parser (Biopython).
- Implementation of current HGVS guidelines.

## Planned features:

- Mutalyzer database check.
  - Enable automatic retrieval and submission to LSDBs.
  
- Implementation of:
  - Extended HGVS nomenclature guidelines.
  - LRG parsing.
  - Automatic variant descriptions.
  - Sequence Ontology descriptions.

## Technical details of the server.

- Lots of dependencies, most of them not present on normal web servers.
  - Part of the UCSC genome browser database.
  - Python / biopython / pyparsing.
  - ...
- Due to modular design, capable of many tasks.
- Capable of many requests.
- One installation available.



Technical details of the client.

- Designed to work on most common web servers.
- Depends on MySQL and PHP.
- Designed for genes (coordinates are coding sequence oriented).
- Many independent installations ( $\pm 80$ ).

Chromosome oriented coordinates are needed for:

- Querying purposes (e.g. search all LOVD's).
- Visualisation of mutations (e.g. all deletions in the UCSC genome browser).

LOVD needs a coordinate conversion system:

- For mapping on a chromosome, we need to convert the coding sequence orientated coordinates to chromosome coordinates.
- Mutalyzer is capable of doing this.
- Installing Mutalyzer locally is not an option.

⇒ Make a web service.

The steps involved in this mapping are:

- LOVD receives a new entry from a curator or submitter.
- LOVD sends the necessary data to Mutalyzer:
  - Accession number of the used transcript (NM number).
  - The variant (in HGVS notation).
- Mutalyzer processes the variant, extracts the positions.
- Mutalyzer returns the data needed by LOVD:
  - Corrected gene-orientated coordinates.
  - Genomic coordinates.
  - The variant type.

Results:

With these newly computed (and stored) columns, a number of new things can be done by LOVD:

- Make custom tracks for use in genome browsers.
- A more robust search for all deletions / SNPs / insertions, ...
- Search for mutations within a certain range.
- Make a web service for LOVD itself (e.g. search all LOVD's for genes in a range).
- ...

LOVD acts as a web service itself.

- Listing of all genes in the database.
- Searching on the gene symbol (full match only).
- Showing only one specific gene entry.
- Listing of all variant entries in a certain gene.
- Searching on the DNA position (full match only).
- Searching on the DNA field.
- Searching on the DBID field.
- Showing only one specific variant entry (internal ID only).

## Acknowledgements

Peter Taschner  
Johan den Dunnen  
Ivo Fokkema  
Gerard Schaafsma  
Jacopo Celli