



University of
Zurich^{UZH}

Department of Geography

UNIVERSITY OF
WOLLONGONG

RSL
measurements | products | policy

The Spectral Information System SPECCHIO V3.3

Andy Hueni¹

¹ Remote Sensing Laboratories, University of Zurich, Switzerland



cost
EUROPEAN COOPERATION
IN SCIENCE AND TECHNOLOGY



a+ scnat
swiss academy of sciences



Goal

Understand the SPECCHIO system:

- Client-server architecture
- Data storage concepts
- Space factory concepts
- Access from other languages
- SPECCHIO API
- Updating your SPECCHIO system

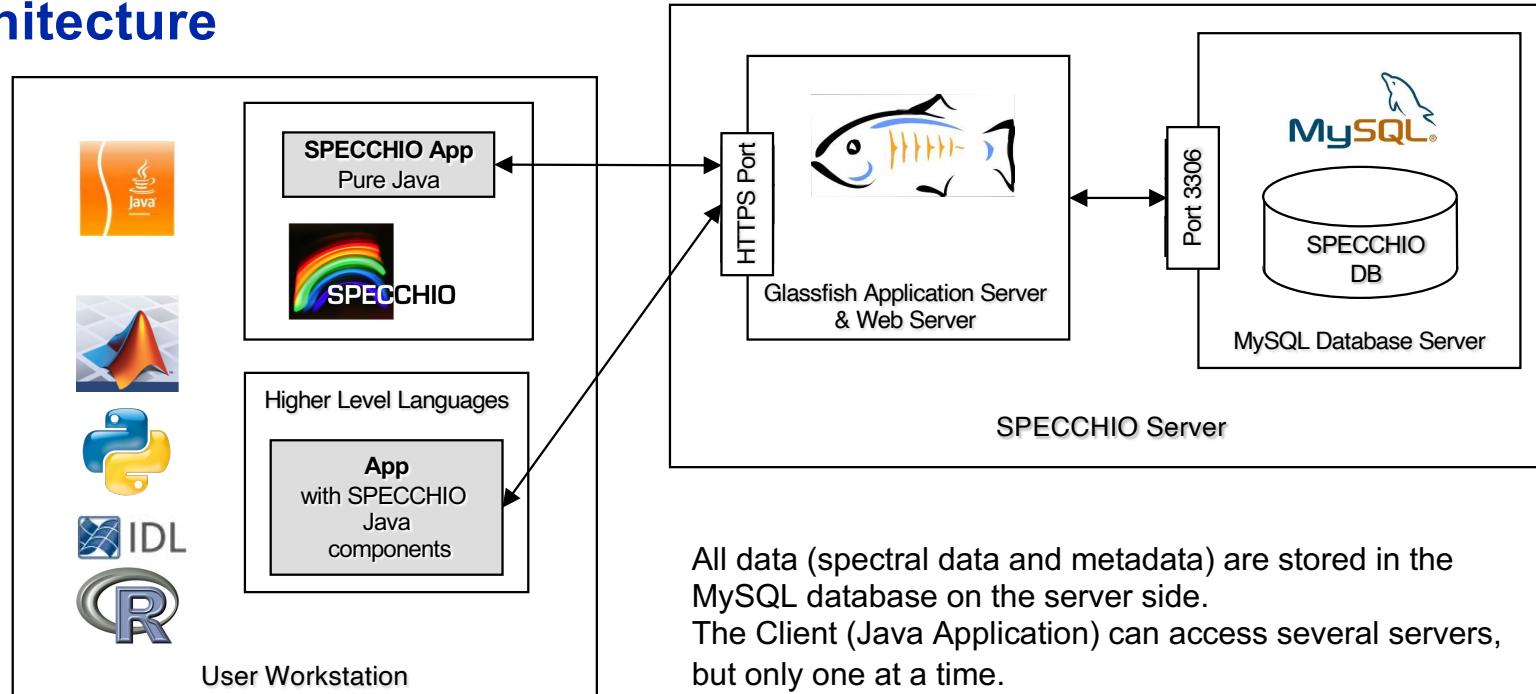


Agenda

- SPECCHIO Architecture
- SPECCHIO VM
- Data storage:
 - Spectral vectors
 - Centre wavelengths definitions
 - Metadata: non-redundant storage, categories and data types
 - Data hierarchies
- Access from Matlab/R and other languages via Java Bridge
- SPECCHIO API
- SPECCHIO code repository
- Updating SPECCHIO binaries
- Updating the SPECCHIO database
- Coding tutorial: where to go from here



Architecture



SPECCHIO uses a client-server architecture (RESTful). Data between client and server are Java objects encoded in XML. The data transfer can be encrypted (HTTPS) if required.

All data (spectral data and metadata) are stored in the MySQL database on the server side.

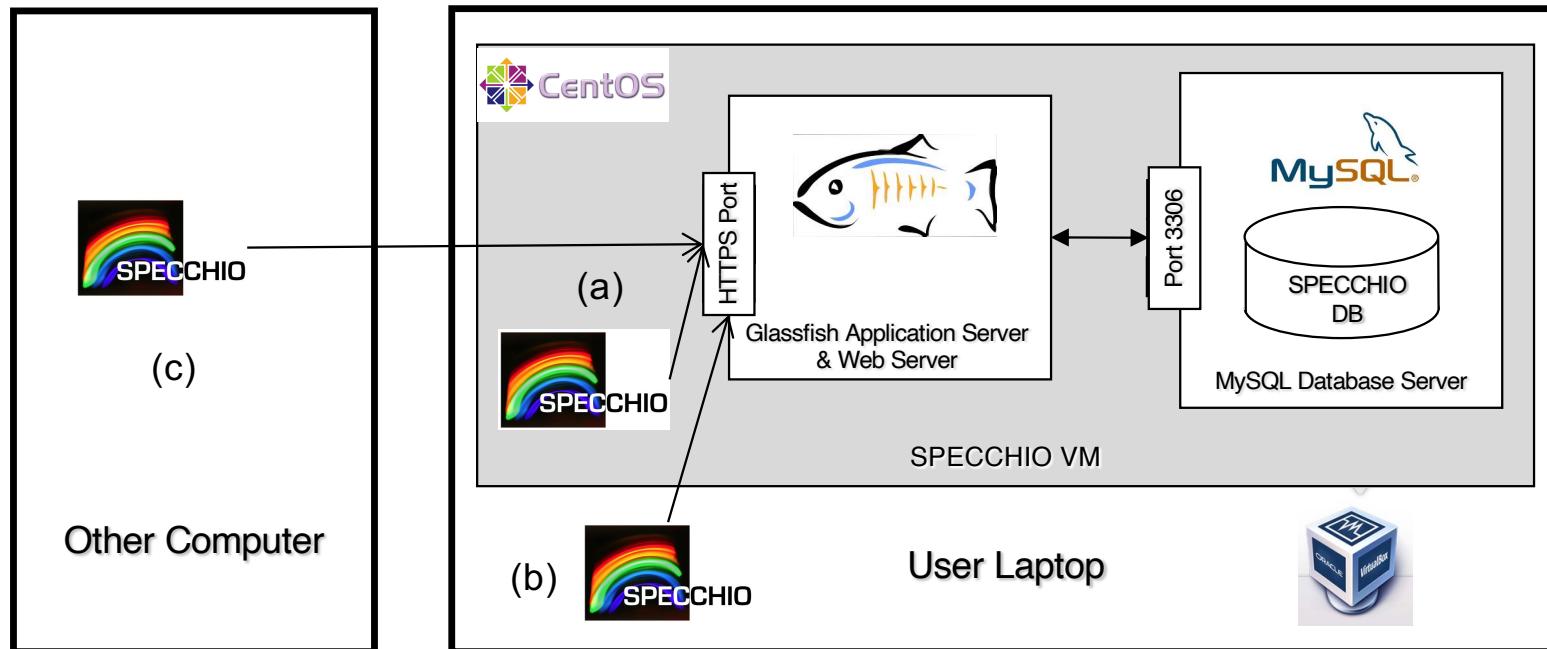
The Client (Java Application) can access several servers, but only one at a time.

To connect to a SPECCHIO server a user account must exist.

New user accounts can be generated via the client if the connection details to the server are known.



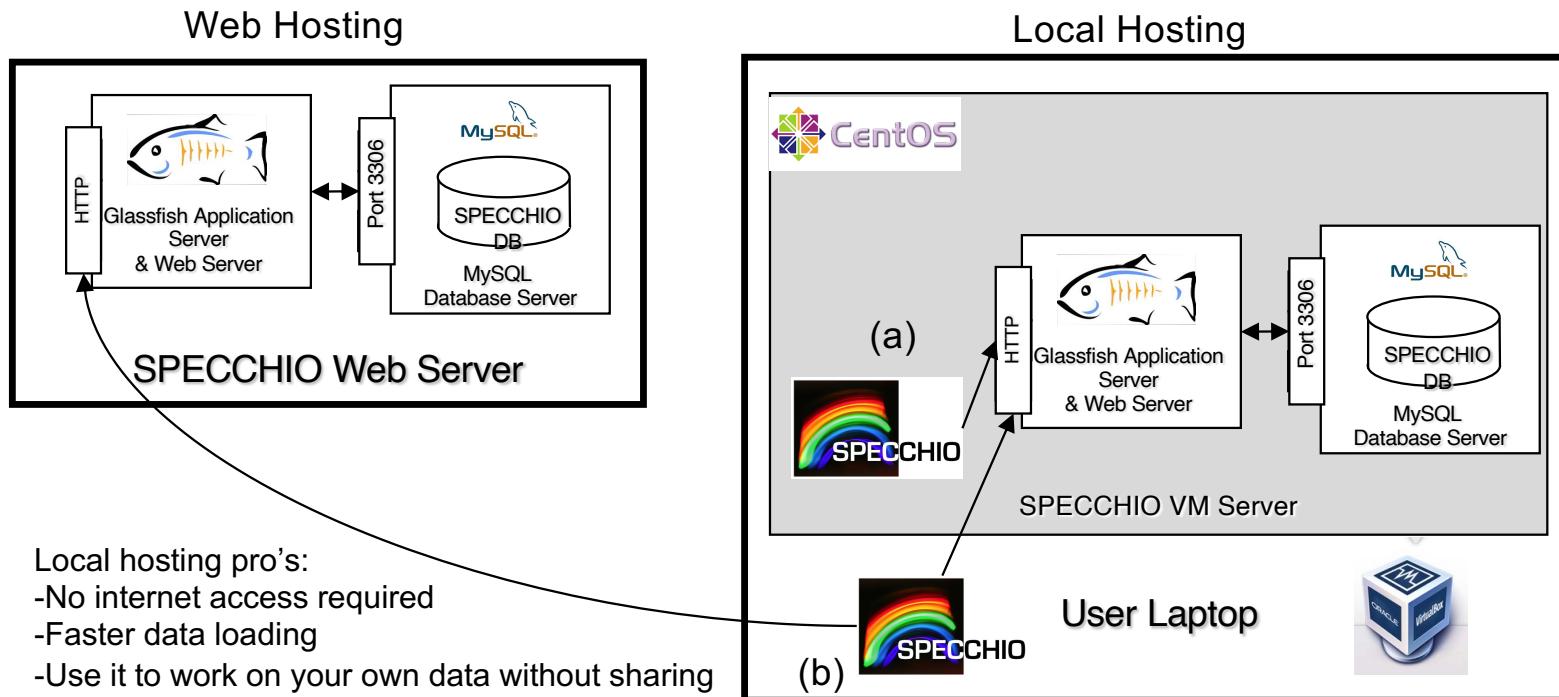
SPECCHIO VM



This course uses the SPECCHIO Virtual Machine (VM) basing on the Oracle VirtualBox. The VM runs CentOS 6.4. The SPECCHIO server running inside the VM can be accessed by clients from: (a) inside the VM, (b) the host computer (i.e. the computer running the VM), and (c) other computers that have network access to the host machine.



SPECCHIO Web Hosting versus Local Hosting via VM



Local hosting pro's:

- No internet access required
- Faster data loading
- Use it to work on your own data without sharing it

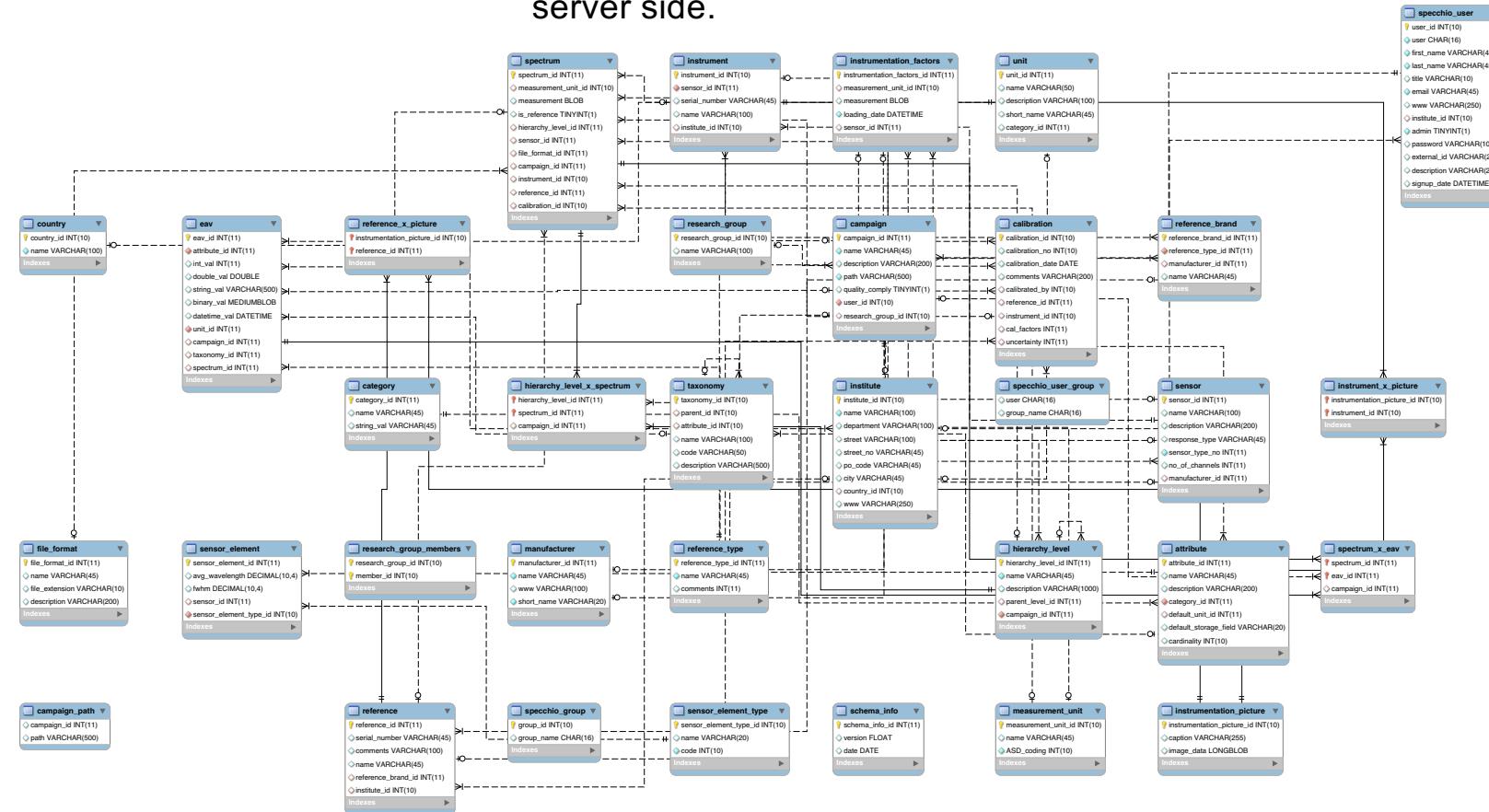
Web hosting pro's:

- Data sharing possible, changes in OPTIMISE data are live
- Easier installation: only client required



SPECCHIO Schema

A direct access to the SPECCHIO schema is not required; such knowledge is only needed when programming the server side.





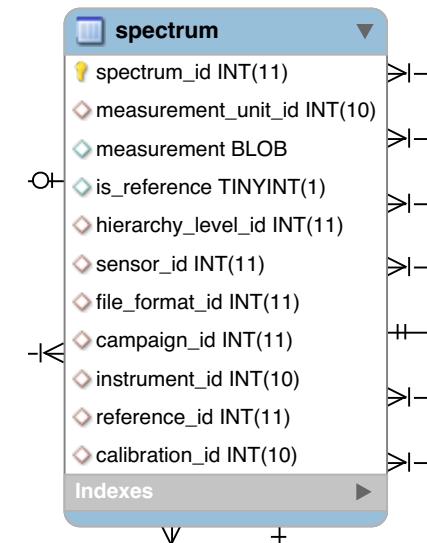
Data storage: Spectral vectors

Spectral data (vectors) are stored as BLOBs in the MySQL database.

The SPECCHIO server has functions to read and write these BLOBs and the API user is always presented with the spectrum as a Java float array.

The spectrum is the primary data, all metadata are attached to a spectrum.
Metadata without a spectrum cannot exist in the system.

A spectrum is identified by the spectrum_id





Data storage: Centre wavelengths definitions

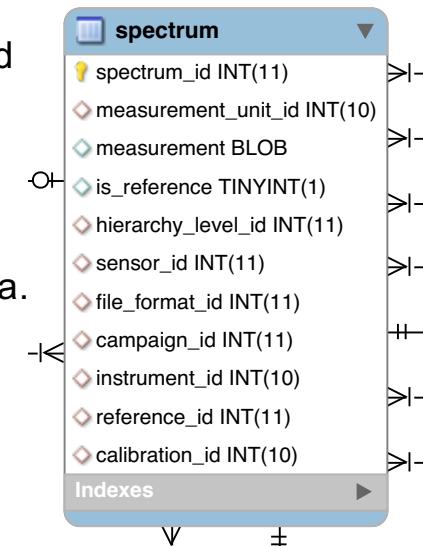
A spectrum is linked with a sensor_id, a instrument_id and and calibration_id.

sensor_id: refers to a general class of instrument, such as the ASD Fieldspec 3.
Can be used to select all spectra collected by that type of sensor.

instrument_id: refers to a physical instrument with an instrument number

calibration_id: refers to a wavelength calibration of the linked instrument.

Wavelengths calibrations are automatically entered
(if not yet existing) into the system when loading new spectra.





Data storage: Metadata

- Flexible, generic metadata storage
- Grouped by categories
- Current # of Meta Elements: >=431

Time/Date

2005-09-12 12:09:50

Spatial Data

Latitude	Longitude
-40.38487999999995	175.62147

String

File Comments Archive scans starting from soil specimen 1453

Integer

Number of internal Scans 30

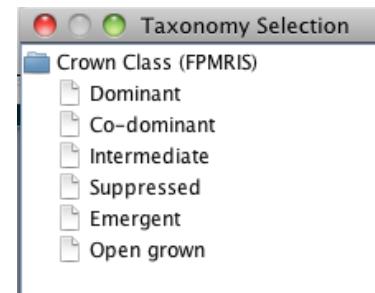
Double

Leaf Area [cm²] 367.04

Spectrum Link

Reference Data Link 251302

Taxonomies (Defined Vocabularies)

 Taxonomy Selection

Crown Class (FPMRIS)

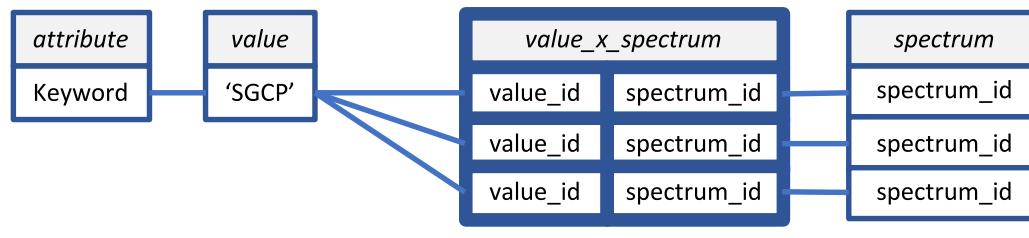
- Dominant
- Co-dominant
- Intermediate
- Suppressed
- Emergent
- Open grown

Digital Content: Images and PDFs

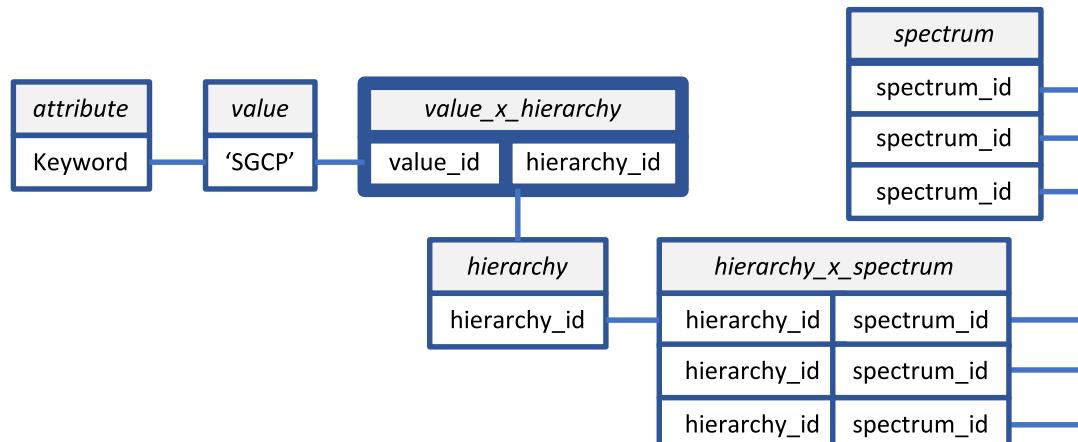




Data storage: Metadata



(a)

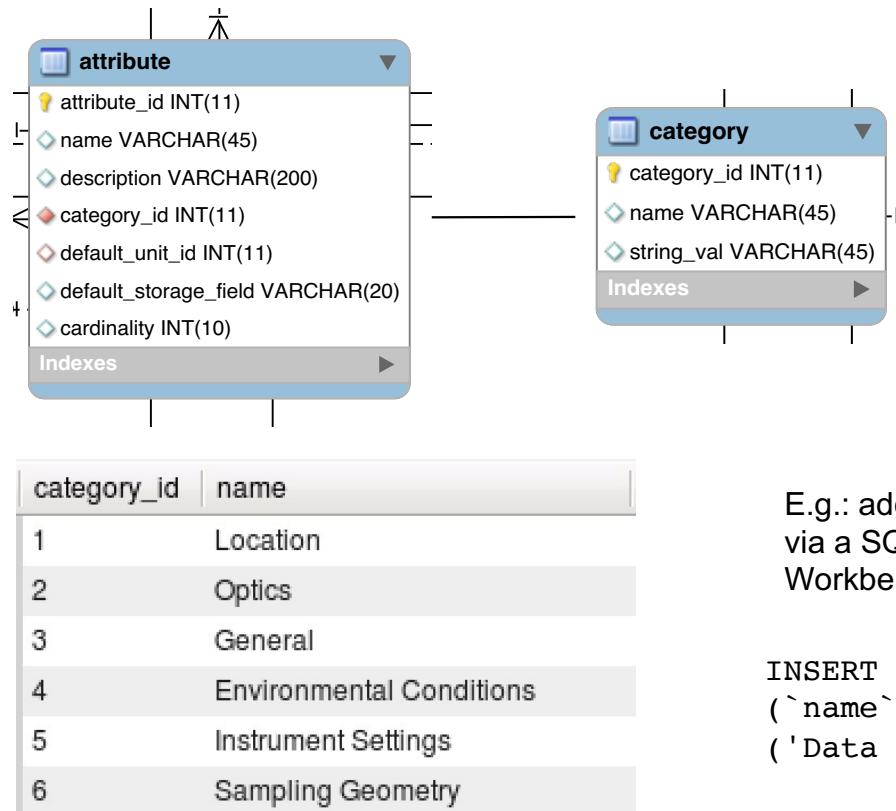


Metadata storage is based on the EAV paradigm* :

- The EAV table stores the actual values
- The attribute represents the parameters
- The spectrum – eav and spectrum-hierarchy crosslink tables allow non-redundant storage and efficient data entries



Data storage: Categories



Attributes are grouped by categories.
New categories can simply be added by inserting a new entry into the category table.

E.g.: adding a new category for data link via a SQL insert statement in the MySQL Workbench:

```
INSERT INTO `specchio`.`category`(`name`, `string_val`) VALUES ('Data Links', ''');
```



List Categories in the MySQL Workbench

The SPECCHIO VM comes with the MySQL Workbench pre-installed. Start the Workbench, open the local instance, then browse to the specchio schema tables and over the category table bring up the context sensitive menu. Click ‘Select Rows – Limit 1000’. All SPECCHIO categories are being listed.

Query 1 | campaign | attribute | category |

1 • `SELECT * FROM specchio.category;`

Result Grid | Filter Rows: | Edit: | Export/Imp

#	category_id	name	string_val
1	1	Location	
2	2	Optics	
3	3	General	
4	4	Environmental Conditions	
5	5	Instrument Settings	
6	6	Sampling Geometry	
7	7	Instrument	
8	8	Names	
9	9	Campaign Details	
10	10	Instrumentation	
11	11	Keywords	
12	12	Vegetation Biophysical Variables	



Management | Schemas

SCHEMAS

Filter objects

specchio

Tables

attribute calibration campaign campaign_path category cloud_cover

Result Grid | Filter Rows

#	attribute_id	name
1	1	Ambien
2	2	Air Pres
3	3	Wind Str

Select Rows - Limit 1000



Data storage: Adding new metadata parameters

Due to the EAV paradigm, a new metadata parameter can just be added into the system by defining a new entry in the attribute table. This is done by an SQL insert statement in the MySQL Workbench. After definition and restarting the server the new attributes are immediately available in the SPECCHIO client.

Example: Adding a new field for 'Instrument Temperature', to be added to the 'Instrument Settings' category.

```
INSERT INTO `specchio`.`attribute`(`name`, `category_id`, `default_storage_field`,  
`description`) VALUES ('Instrument Temperature', (select category_id from  
`specchio`.category where name = 'Instrument Settings'), 'double_val', 'Internal  
temperature of the instrument');
```



Data storage: Data hierarchies and Campaigns

Spectral data can be organised in a hierarchical structure in the database.

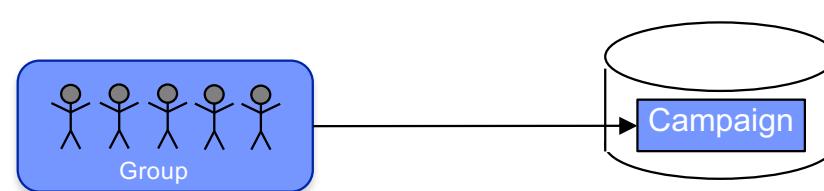
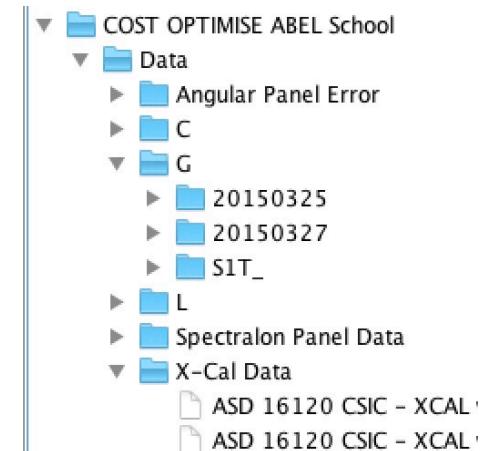
This can help the data browsing.

Data structures can either be replicated from the hard disc where the original spectral files are stored (automatically done during data ingestion) or they can be generated programmatically.

Data are organised by 'Campaigns'.

Campaigns have a defined owner – the user that generates the campaign has full rights.

Campaigns can be shared with other users, giving full access to other existing users within the same database.

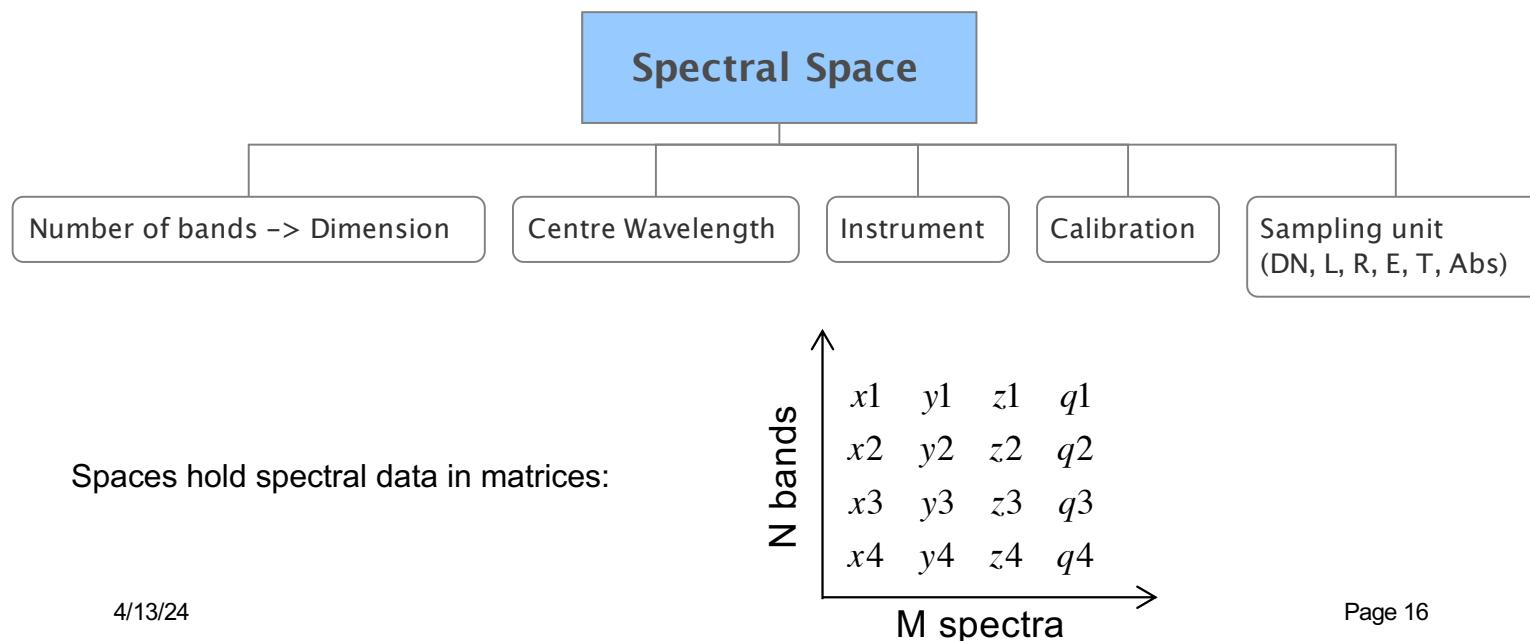




Some background on the Space Factory

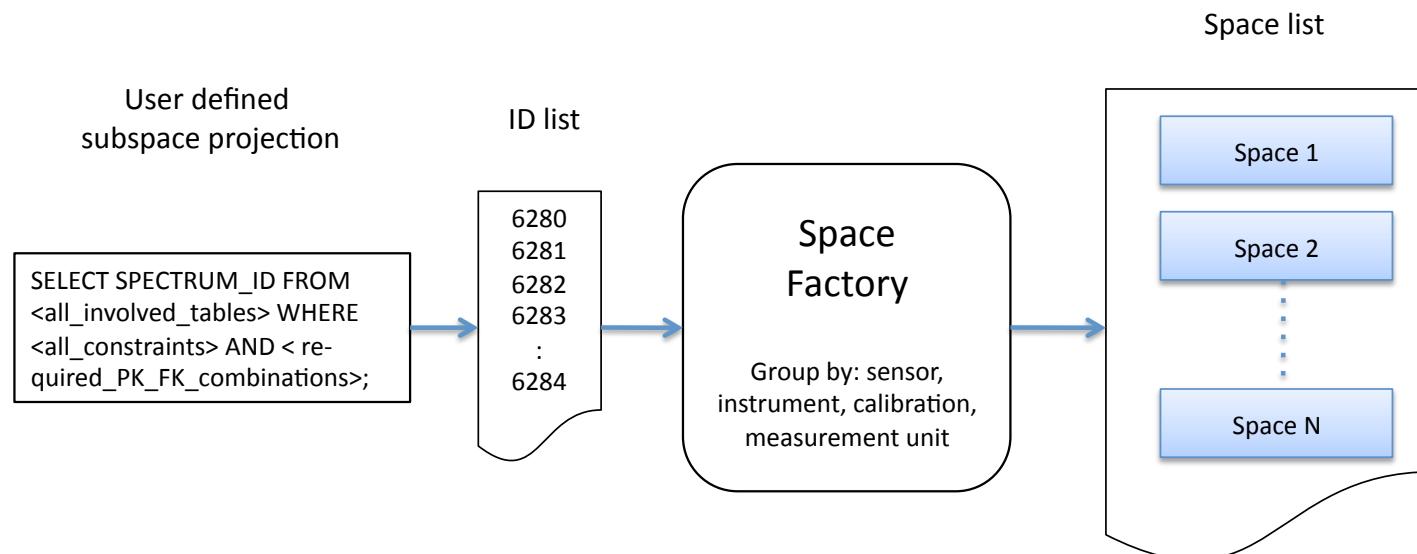
Approach to deal with different instruments in the database when processing spectral data.

Spaces are N-dimensional, where N=number of spectral bands





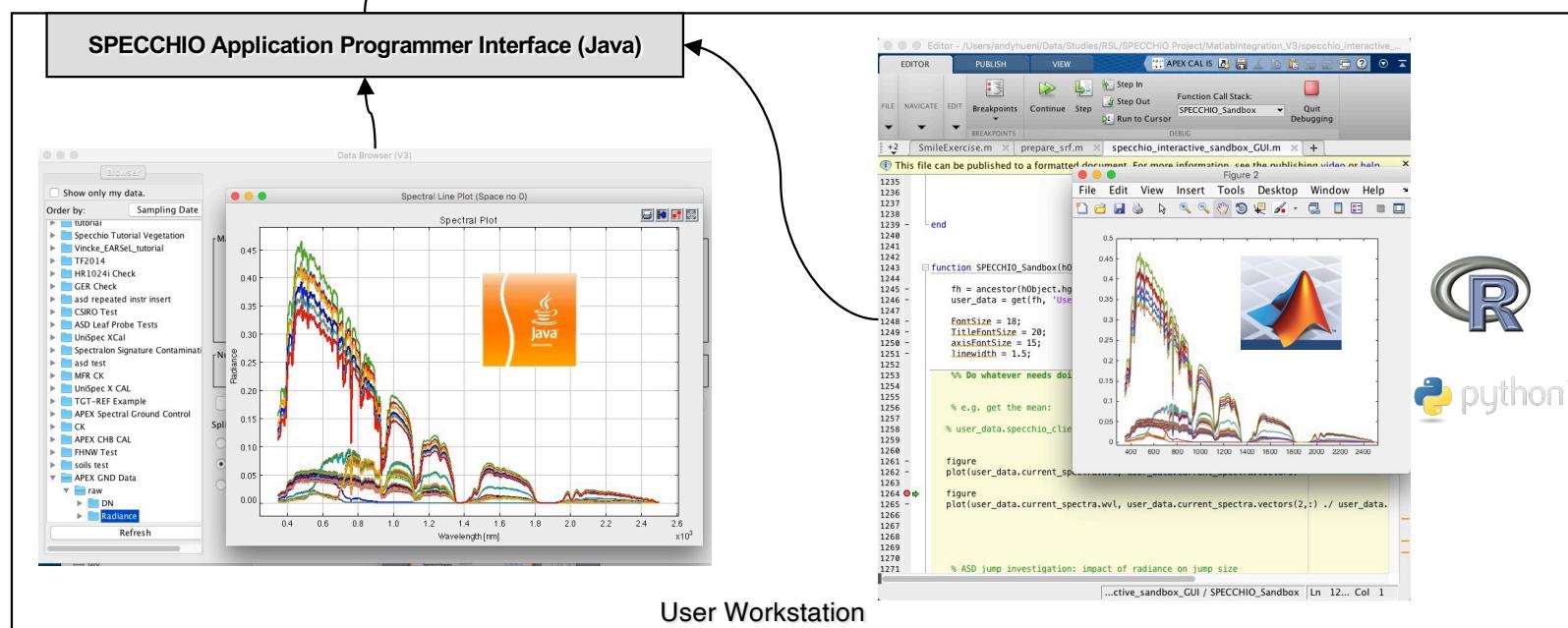
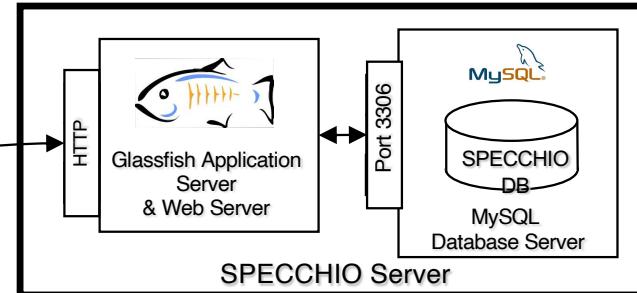
Concepts – Space Building





SPECCHIO access methods

Unified access: either via a Java GUI or via Matlab, R, et al

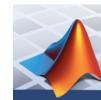




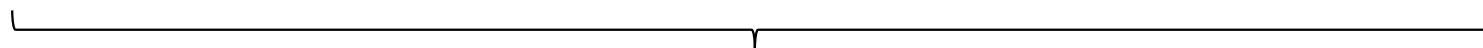
SPECCHIO access methods: Which to choose



- User friendly interface
- Limited for actual science
- File loading of supported files is super easy
- For free
- Runs on all platforms
- Installing requires no admin rights



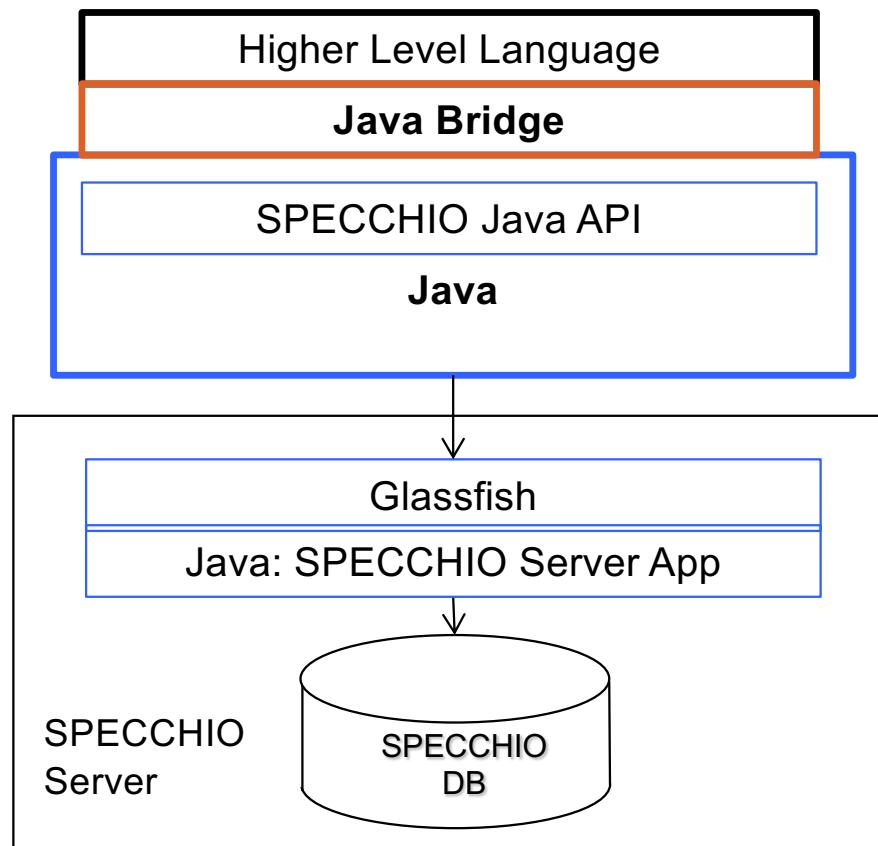
- **THE tool for science while basing on a database!**
- You need to know a programming language
- Supports operations currently not available in the Java GUI
- Write your whole pre-processing and analysis for a data source that can provide a live feed
- Depending on the chosen language, you can write your own GUI interfaces
- Connect SPECCHIO to other tools, like Fluspect



Seamlessly combined them for the best of both worlds!



Access from Matlab/R and other languages via Java Bridge



Communication with the SPECCHIO server is done via the SPECCHIO API. Only requirement is a Java Bridge that can link Java with the higher level language.

Examples:

- Matlab: Java support is built in
- R: Java support via rJava
- IDL: Java support apparently built in
- Python: using the JPyte library



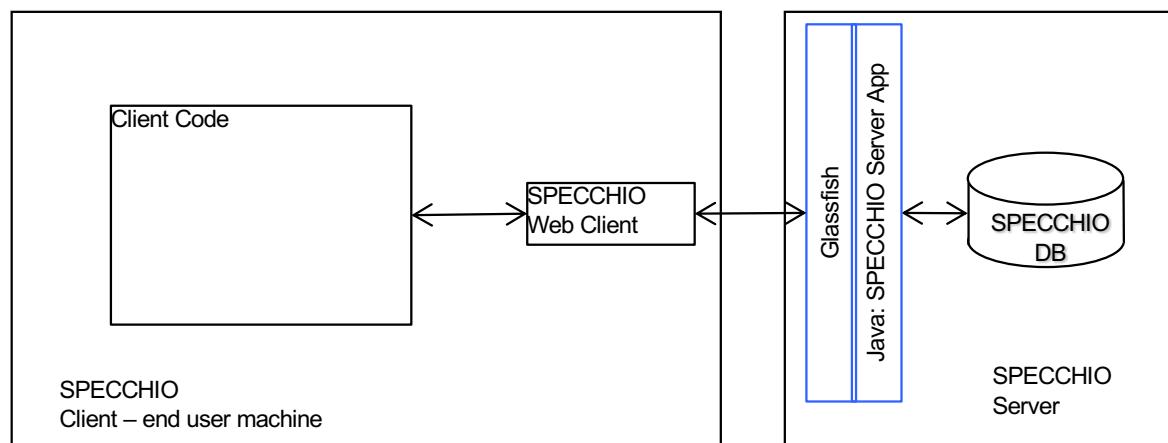
SPECCHIO Application Programmer Interface (API)

The SPECCHIO API handles all communication between the client and the server, such as:

- Connect to server
- Get information about attributes, instruments, calibrations
- Insert spectral data and metadata
- Select and read spectral data and metadata
- Delete spectral data and metadata

All SPECCHIO Java classes used on the client side can be used when programming via the Java Bridge.

An instance of the SPECCHIO Web Client handles all communication with the server.





SPECCHIO API

au.and.org.researchdata
ch.specchio.client
ch.specchio.constants
ch.specchio.db_import_export
ch.specchio.eav_db
ch.specchio.explorers
ch.specchio.factories
ch.specchio.file.reader.calibration
ch.specchio.file.reader.campaign
ch.specchio.file.reader.spectrum
ch.specchio.file.reader.spectrum.hd5
ch.specchio.file.writer
ch.specchio.gui
ch.specchio.interfaces
ch.specchio.iohub

ch.specchio.client

Interfaces

SPECCHIOClient
SPECCHIOServerDescriptor

Classes

SPECCHIOClientCache
SPECCHIOClientFactory
SPECCHIODatabaseDescriptor
SPECCHIOWebAppDescriptor
SPECCHIOWebClient
SPECCHIOWebClientFilter

Exceptions

SPECCHIOClientException
SPECCHIODatabaseClientException
SPECCHIOWebClientException

Overview Package Class Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

ch.specchio.client

Class SPECCHIOWebClient

java.lang.Object
ch.specchio.client.SPECCHIOWebClient

All Implemented Interfaces:

SPECCHIOClient

```
public class SPECCHIOWebClient
extends java.lang.Object
implements SPECCHIOClient
```

SPECCHIO web client.

Method Summary

Methods

Modifier and Type	Method and Description
void	clearMetaparameterRedundancyList() Clears the known metaparameter list held by the server for this user
void	connect() Connect to the SPECCHIO web application.
int	copySpectrum(int spectrum_id, int target_hierarchy_id) Copy a spectrum to a specified hierarchy.



Updating SPECCHIO binaries

New binaries are available via Jenkins:

<https://jenkins.specchio.ch/job/SPECCHIO/>

New binaries are available via the SPECCHIO page:

<https://specchio.ch/downloads/>

Latest release

Downloads

- specchio-installer_MacOS.jar
- specchio-installer.jar
- specchio-centos-7.6-vbox-x86_64.ova
- specchio-client.zip
- specchio-webapp.zip
- javadoc.zip

Jenkins

Dashboard > SPECCHIO >

Back to Dashboard

Status

Changes

Full Stage View

Github

Open Blue Ocean

Pipeline SPECCHIO

Last Successful Artifacts

Artifact	Size	Action
javadoc.zip	3.20 MB	view
specchio-client.zip	37.02 MB	view
specchio-installer.jar	77.94 MB	view
specchio-installer_MacOS.jar	85.70 MB	view
specchio-webapp.zip	15.10 MB	view

Recent Changes

See the SPECCHIO VM manual on how to update the server software.
Client updates are super easy using the installation packages.



Updating the SPECCHIO database

Upgrades to the SPECCHIO database are provided as MySQL scripts on GitHub:

<https://github.com/SPECCHIODB/SPECCHIO/tree/master/conf/mysql>

To run them use the MySQL Workbench within the SPECCHIO VM.

The screenshot shows a Mac OS X desktop environment. On the left, a Finder sidebar displays a 'Files' section with a 'master' dropdown and a 'Go to file' search bar. Below this, a 'conf/mysql' folder is expanded, showing several MySQL upgrade scripts: 'GA_specchio_database_upgra...', 'SPECCHIO_V3.0.2.sql', 'SPECCHIO_V3.1.3.sql', 'SPECCHIO_V3.2.0.sql', and 'specchio_database_upgrade_V3.3.0.sql'. On the right, a web browser window is open to the GitHub URL 'https://github.com/SPECCHIODB/SPECCHIO/tree/master/conf/mysql/'. The page lists six database upgrade scripts with their commit messages and timestamps:

Commit	Message	Time
specchio_database_upgrade_V3.3.2_V3....	Remove default unit_id from Surface Saleh Roughn...	5 years ago
specchio_database_upgrade_V3.3.3_V3....	New metadata attributes for plant probe support an...	5 years ago
specchio_database_upgrade_V3.3.4_V3....	New attributes and categories to support CAL/VAL, ...	last year
specchio_database_upgrade_V3.3.5_V3....	New structures to support Matrix storage via UJMP ...	8 months ago
specchio_database_upgrade_V3.3.6_V3....	Updates for SVC metadata and for AOT as new unit.	3 months ago
specchio_database_upgrade_uncertain...	Updates for users and uncertainty	last year



Coding tutorials: where to go from here



University of Zurich™
Department of Geography

UNIVERSITY OF ZURICH™
Department of Geography

Matlab accessing SPECCHIO

Andy Hueni¹

¹Remote Sensing Laboratories, University of Zurich, Switzerland

SPECCHIO Programming Course
Zurich 2017



OPTI-MISE COST RSL Page 1



University of Zurich™
Department of Geography

UNIVERSITY OF ZURICH™
Department of Geography

R accessing SPECCHIO

Andy Hueni¹

¹Remote Sensing Laboratories, University of Zurich, Switzerland

SPECCHIO Programming Course
Zurich 2017



OPTI-MISE COST RSL Page 1



University of Zurich™
Department of Geography

UNIVERSITY OF ZURICH™
Department of Geography

Python accessing SPECCHIO

Andy Hueni¹

¹Remote Sensing Laboratories, University of Zurich, Switzerland

OPTIMISE SPECCHIO Programming Course



OPTIMISE COST RSL Page 1



Home
News (14th Feb, 2017)
FAQs
Guestbook
Available Spectral Campaigns (V2.2)
SPECCHIO Web Interface (V3)
Create database account (Only V2.2)
User Guides (11th Jun, 2015)
Download application (03rd Apr, 2017)
Download SPECCHIO VM (04th Apr, 2017)
SPECCHIO V3.2.1.6 API Programming Course (17th Apr, 2017)
Tutorial data (14th Feb, 2017)
Publications & Referencing
Screenshots
SPECCHIO Project @ RSL
Other spectral databases

[Presentation: Matlab accessing SPECCHIO](#) 

[Presentation: R accessing SPECCHIO](#) 

[Presentation: Python accessing SPECCHIO](#) 

[Input File: spectra.csv](#)

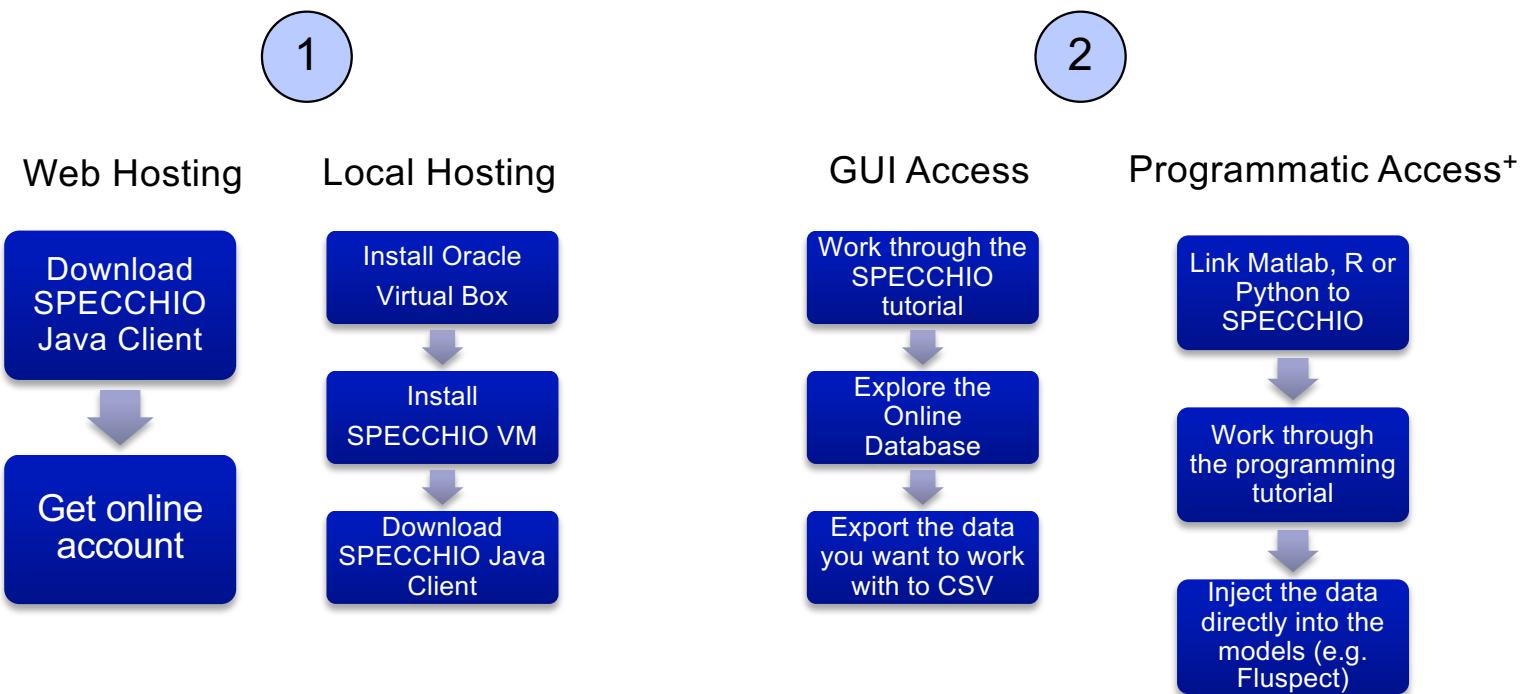
[Input File: metadata.csv](#)

[Follow](#)

Download the programming tutorials from the SPECCHIO web page:
http://specchio.ch/programming_course.php



Next Steps: For you to decide*



* Any combination of (1) and (2) will work.

+ Matlab, R or Python must be preinstalled



Thank you for your attention!

For more information on the current version of SPECCHIO see: www.specchio.ch

https://twitter.com/SPECCHIO_DB 

