



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





Goal

Understand the SPECCHIO system:

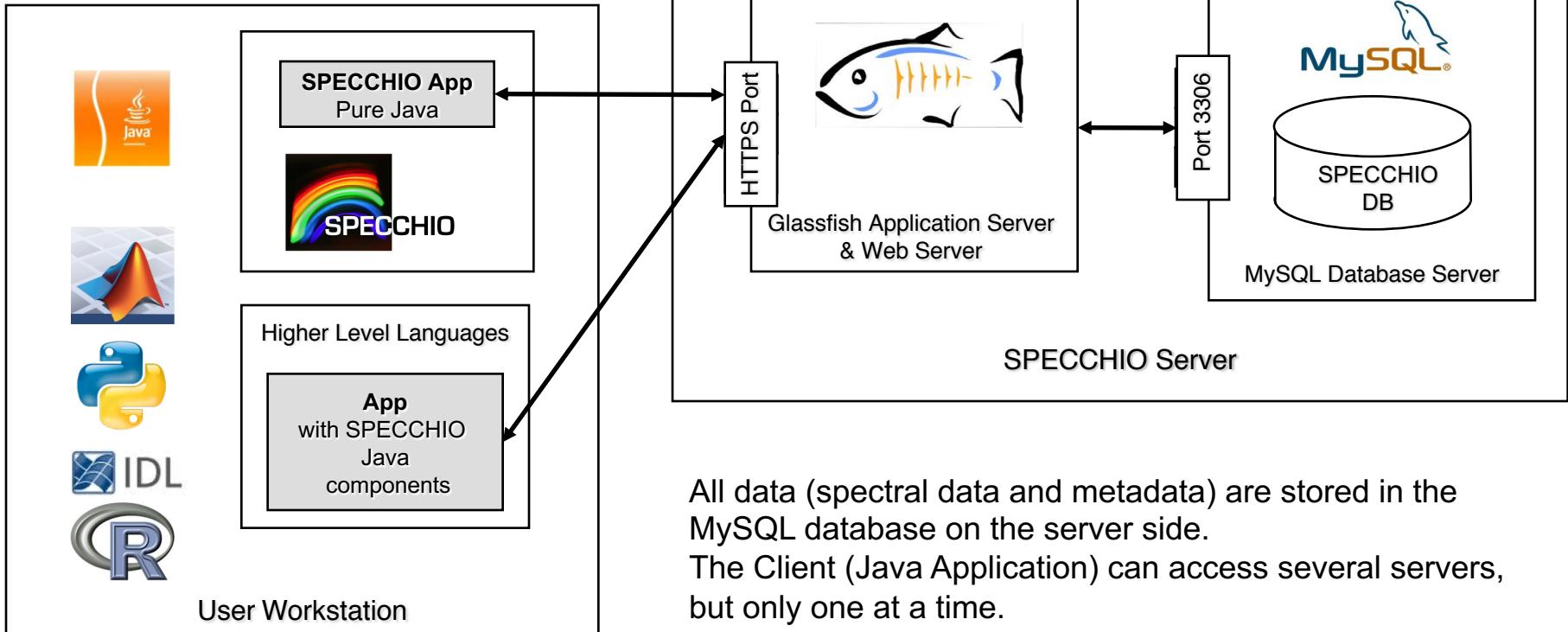
- Client-server architecture
- Data storage concepts
- Space factory concepts
- Access from other languages
- SPECCHIO API
- Updating your SPECCHIO system



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

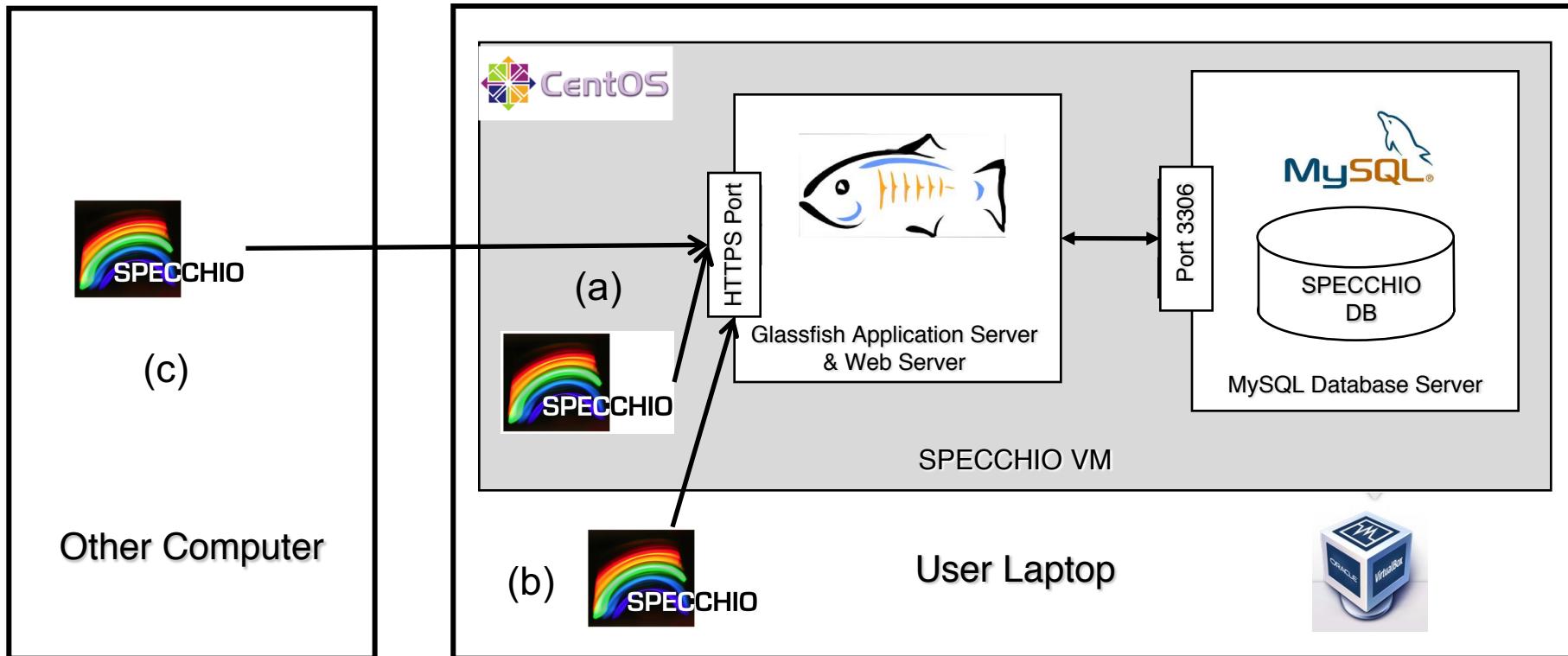
Architecture



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.

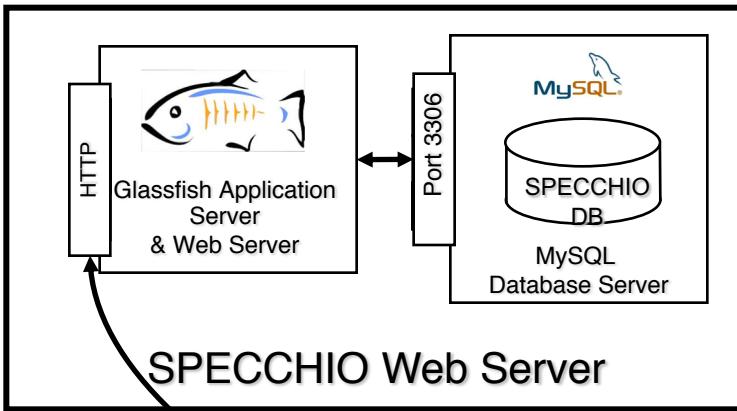
SPECCHIO VM



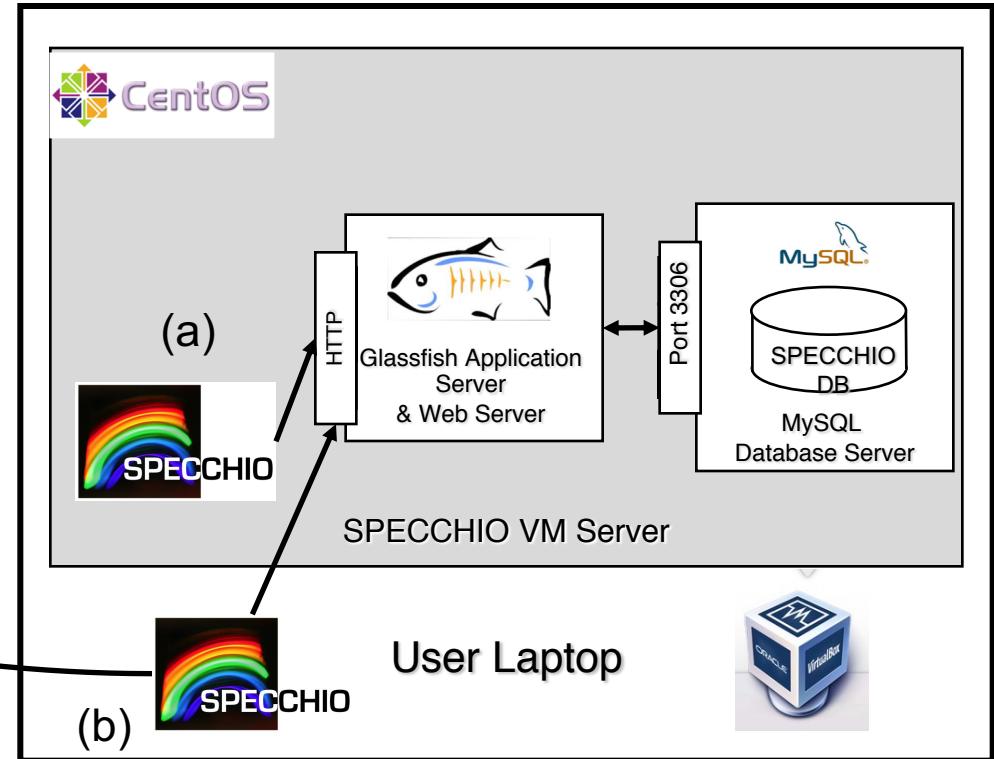
This course uses the SPECCHIO Virtual Machine (VM) basing on the Oracle VirtualBox. The VM runs CentOS 6.4. The SPECCHIO server running inside the VM can be accessed by clients from: (a) inside the VM, (b) the host computer (i.e. the computer running the VM), and (c) other computers that have network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

SPECCHIO Web Hosting versus Local Hosting via VM

Web Hosting



Local Hosting



Local hosting pro's:

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

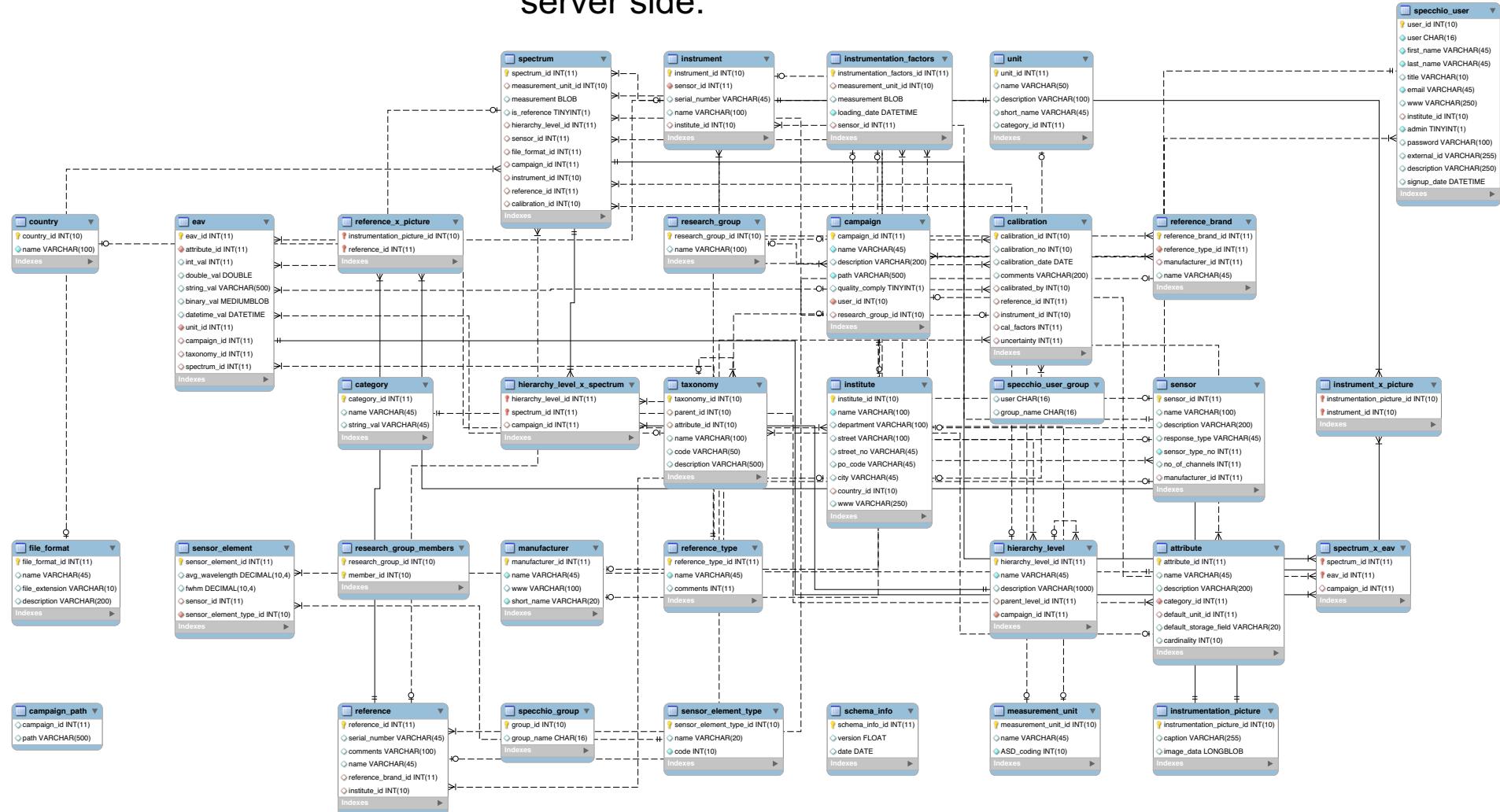
Web hosting pro's:

- 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.



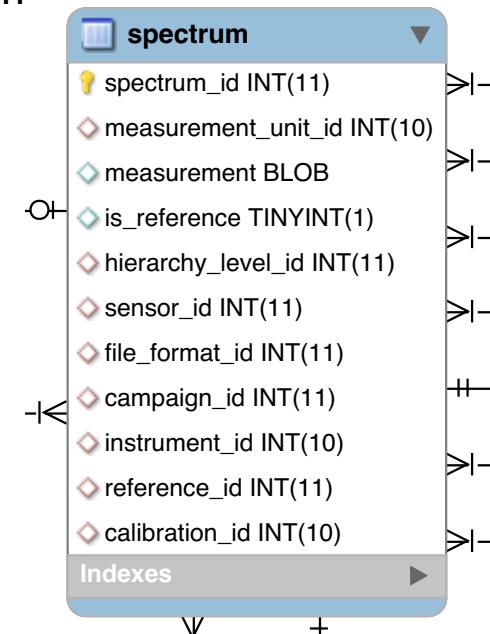


Data storage: Spectral vectors

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

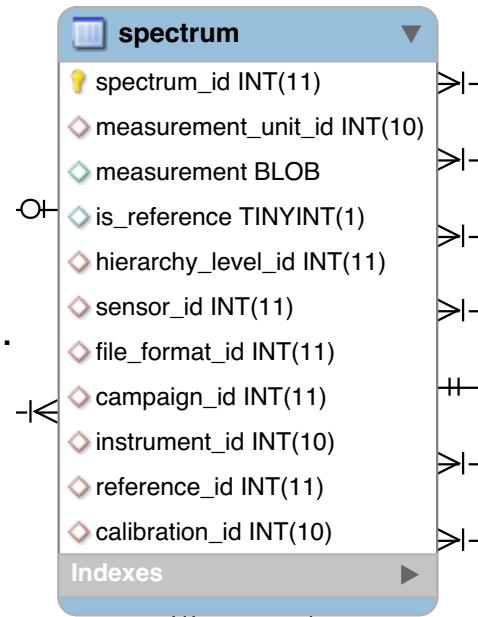
A spectrum is linked with a sensor_id, a instrument_id and and calibration_id.

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: 357

Time/Date	<input type="text" value="2005-09-12 12:09:50"/>				
Spatial Data	<table><tr><td>Latitude</td><td>Longitude</td></tr><tr><td>-40.38487999999995</td><td>175.62147</td></tr></table>	Latitude	Longitude	-40.38487999999995	175.62147
Latitude	Longitude				
-40.38487999999995	175.62147				

String File Comments Archive scans starting from soil specimen 1453

Integer Number of internal Scans

Double Leaf Area [cm²]

Spectrum Link Reference Data Link

Taxonomies (Defined Vocabularies)

Taxonomy Selection

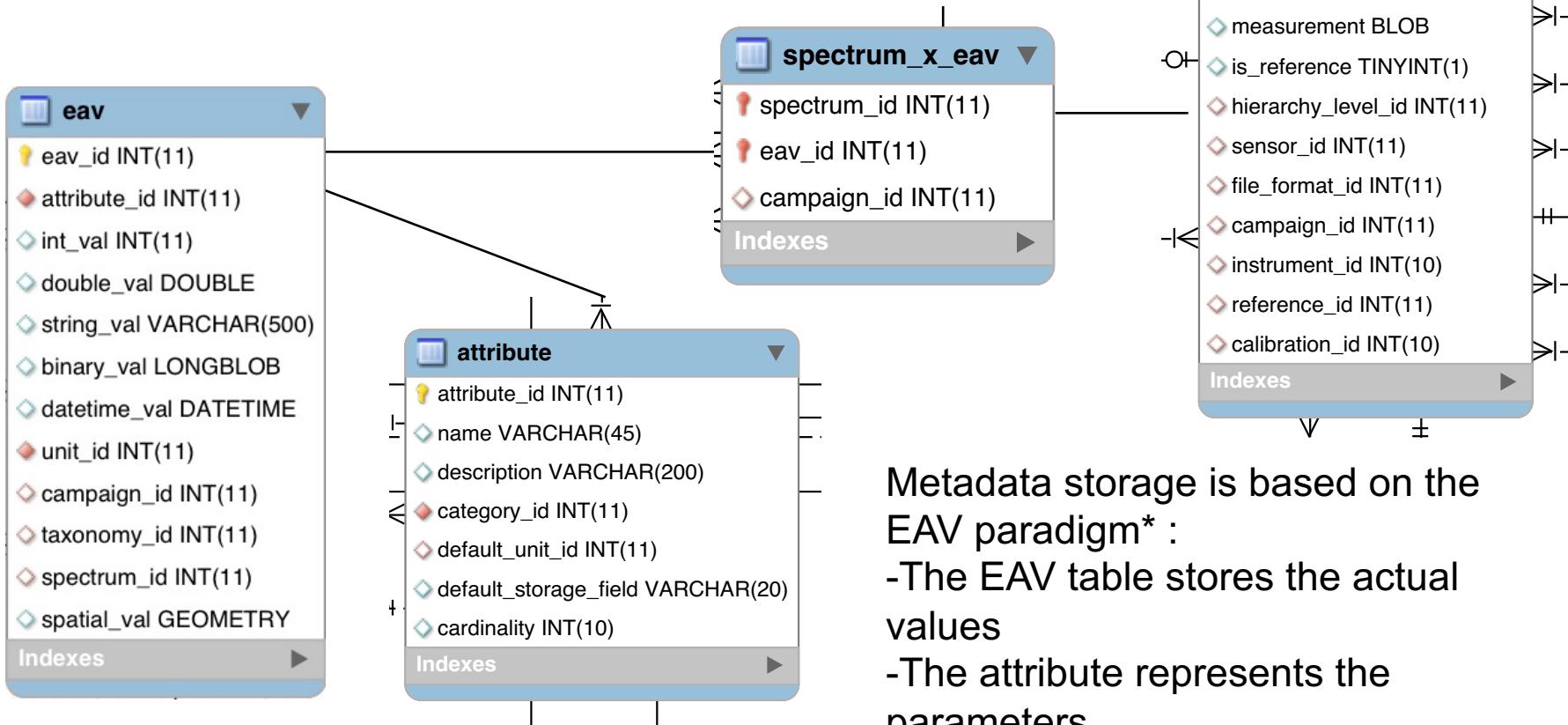
<input type="checkbox"/> Crown Class (FPMRIS)
<input type="checkbox"/> Dominant
<input type="checkbox"/> Co-dominant
<input type="checkbox"/> Intermediate
<input type="checkbox"/> Suppressed
<input type="checkbox"/> Emergent
<input type="checkbox"/> Open grown

Digital Content: Images and PDFs





Data storage: Metadata

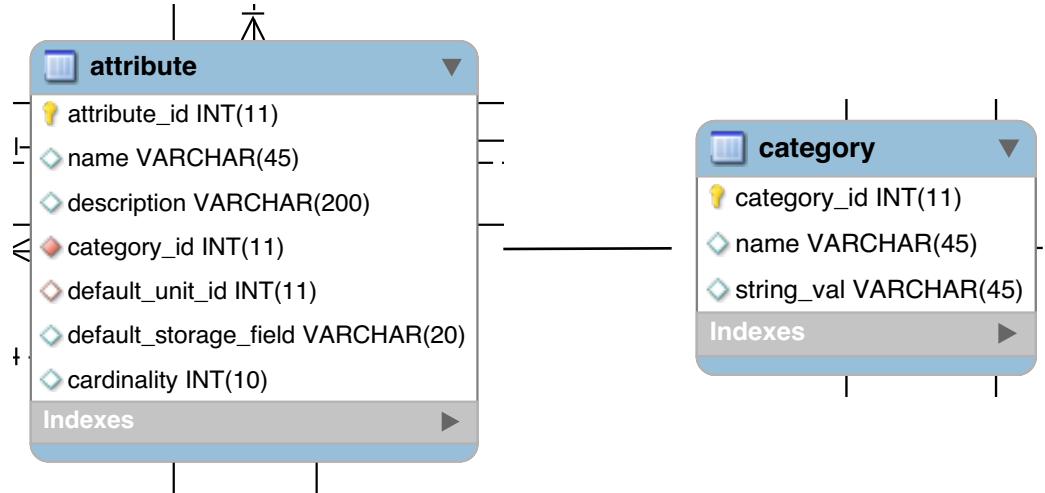


Metadata storage is based on the EAV paradigm* :

- The EAV table stores the actual values
- The attribute represents the parameters
- The spectrum – eav crosslink table allows non-redundant storage

* Nadkarni, P.; Marenco L; Chen R; Skoufos E; Shepherd G; P., M. Organization of heterogeneous scientific data using the eav/cr representation. *Journal of the American Medical Informatics Association* 1999, 6, 478-493.

Data storage: Categories



category_id	name
1	Location
2	Optics
3	General
4	Environmental Conditions
5	Instrument Settings
6	Sampling Geometry

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', '');
```



Listing Categories in the MySQL Workbench

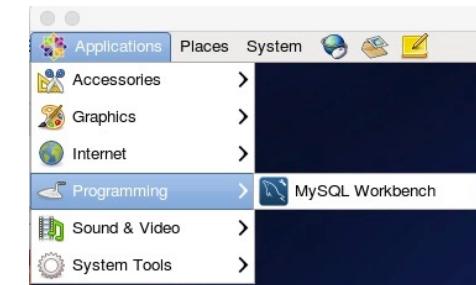
The SPECCHIO VM comes with the MySQL Workbench pre-installed. Start the Workbench, open the local instance, then browse to the specchio schema tables and over the category table bring up the context sensitive menu. Click ‘Select Rows – Limit 1000’. All SPECCHIO categories are being listed.

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

information_schema

specchio

Tables

attribute calibration campaign campaign_path category cloud_cover

Query 1 | campaign |

1 • `SELECT`

Result Grid | Filter Row

#	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.

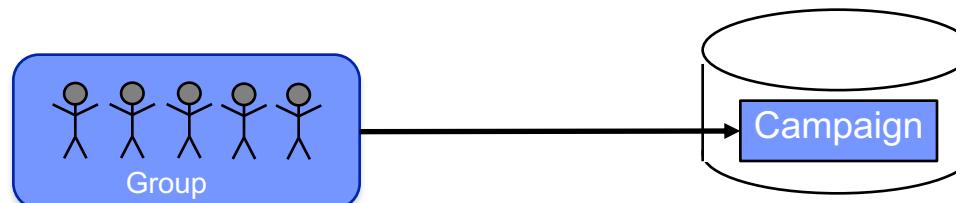
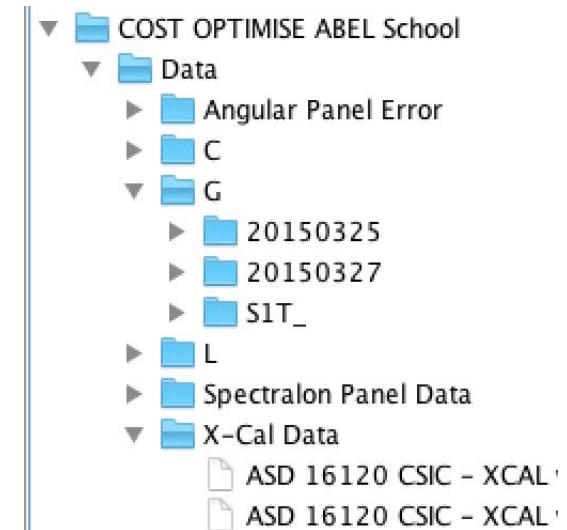
This can help the data browsing.

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.

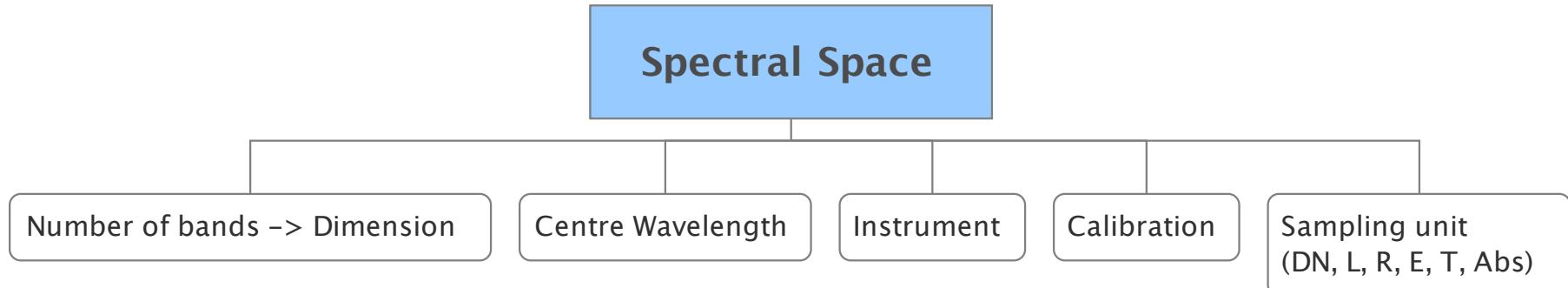




Some background on the Space Factory

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

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



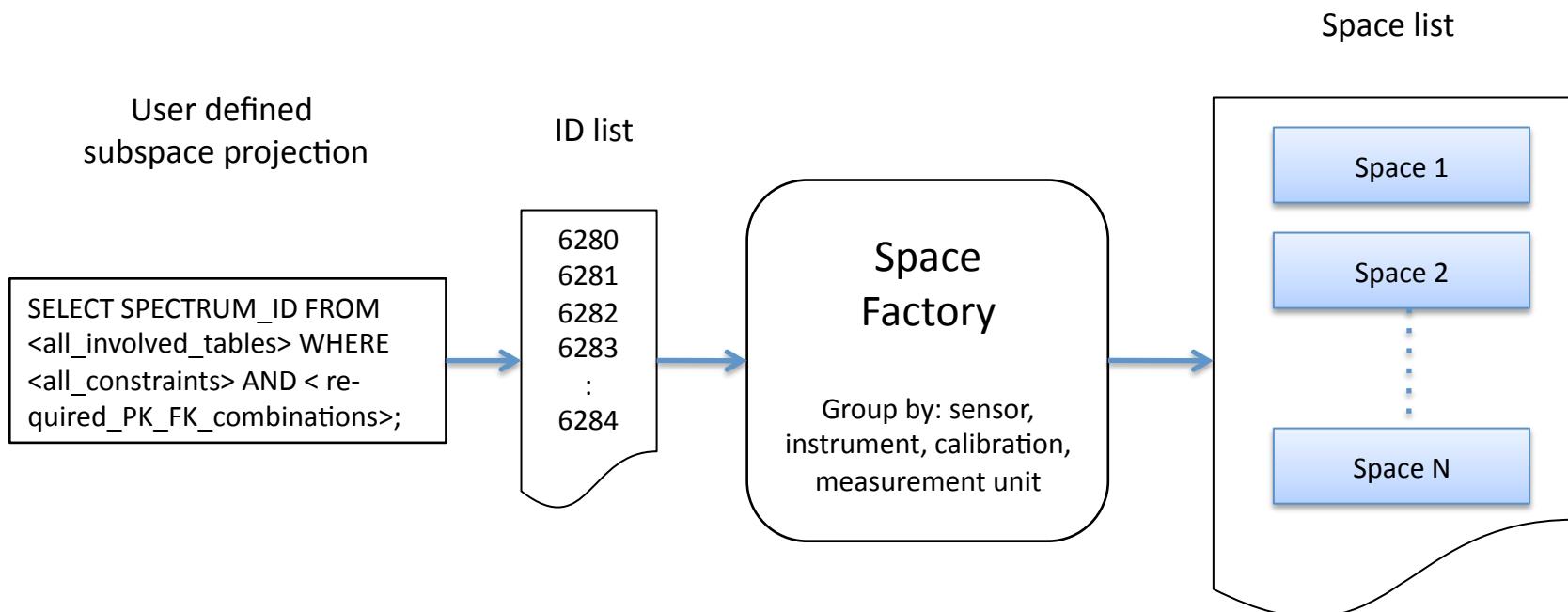
Spaces hold spectral data in matrices:

N bands	x1	y1	z1	q1
	x2	y2	z2	q2
	x3	y3	z3	q3
	x4	y4	z4	q4

M spectra →



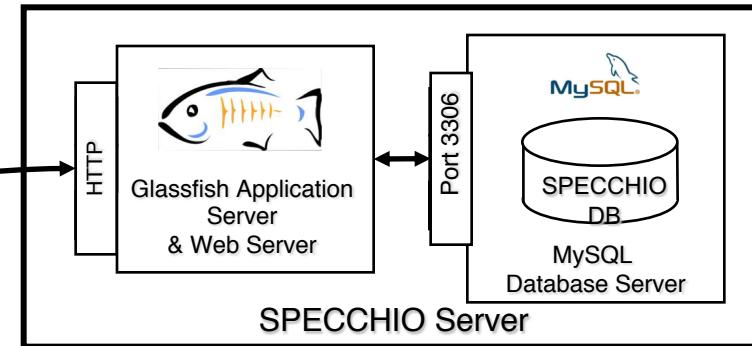
Concepts – Space Building





SPECCHIO access methods

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



The User Workstation interface is shown in three panels:

- SPECCHIO Application Programmer Interface (Java):** A Java application window showing a spectral line plot (Space no 0) with multiple colored curves representing different spectra. A Java logo icon is visible on the plot.
- Data Browser (V3):** A browser-based interface showing a hierarchical tree of datasets and a spectral plot window below it. The plot shows Radiance vs Wavelength [nm].
- User Workstation:** A desktop environment with several windows:
 - A MATLAB editor window showing a script for the SPECCHIO_Sandbox function.
 - A MATLAB plot window showing a 3D surface plot of a spectrum.
 - A terminal window showing command-line interaction with the SPECCHIO_Sandbox function.

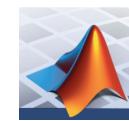
On the right side of the User Workstation panel, there are logos for R and Python.



SPECCHIO access methods: Which to choose



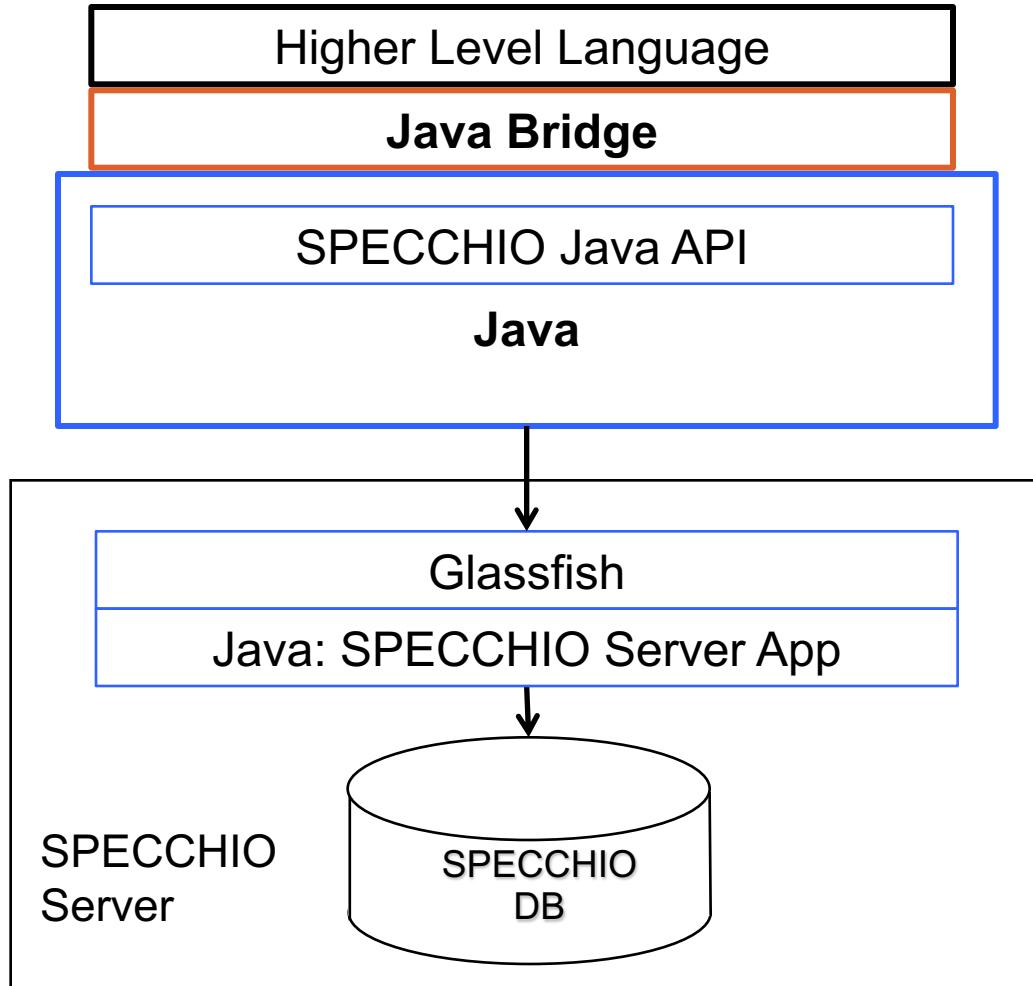
- User friendly interface
- Limited for actual science
- File loading of supported files is super easy
- For free
- Runs on all platforms
- Installing requires no admin rights



- **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

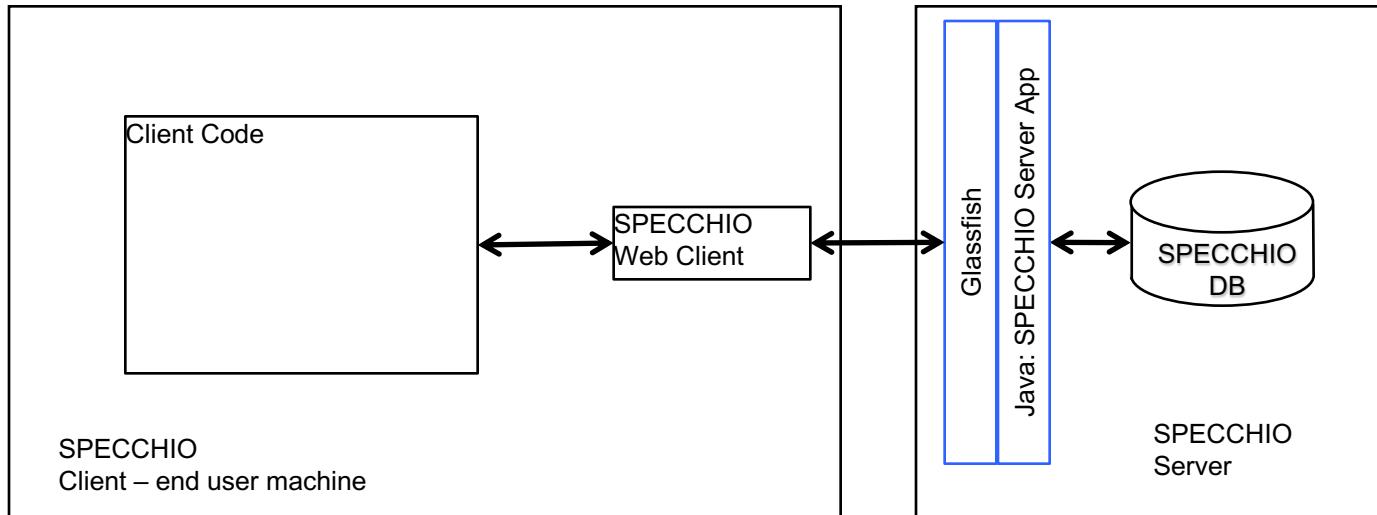
SPECCHIO Application Programmer Interface (API)

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

The SPECCHIO API documentation can be found online:
<https://specchio.ch/javadoc/>

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.job

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

void

Method and Description

[clearMetaparameterRedundancyList\(\)](#)

Clears the known metaparameter list held by the server for this user

[connect\(\)](#)

Connect to the SPECCHIO web application.

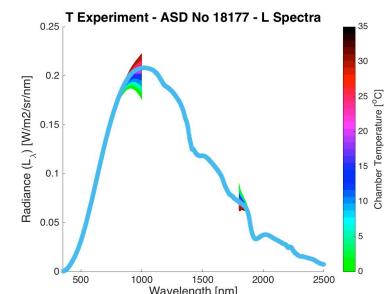
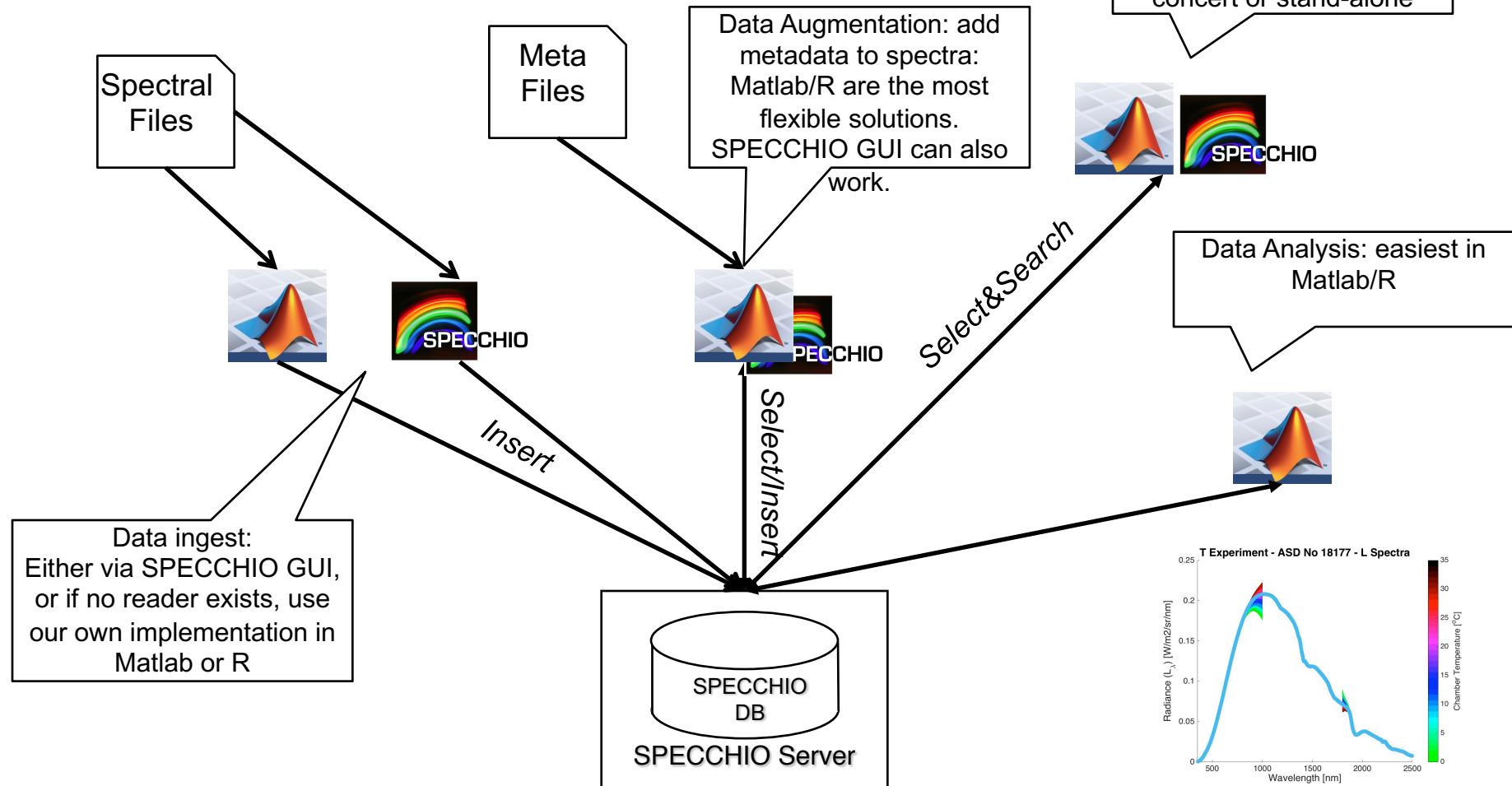
[copySpectrum\(int spectrum_id, int target_hierarchy_id\)](#)

Copy a spectrum to a specified hierarchy.

int



General Processing Workflow





SPECCHIO code repository

All SPECCHIO code (client and server) is open source and can be found on GitHub:
<https://github.com/ahueni/SPECCHIO>

The screenshot shows the GitHub repository page for 'ahueni / SPECCHIO'. The top navigation bar includes links for 'This repository', 'Search', 'Pull requests', 'Issues', 'Gist', and user icons. Below the header, the repository name 'ahueni / SPECCHIO' is displayed, along with statistics: 785 commits, 1 branch, 25 releases, and 4 contributors. A summary table lists recent commits across various files and folders. The commits are as follows:

File/Folder	Commit Message	Date
conf	Bug fixes in schema for MySQL 5.7	14 days ago
doc	V3.2.1.6	13 days ago
pkg	Bug fix for a metadata insert issue (e.g. in UTC computation)	2 days ago
src	Moving away from Campaign Factory abstraction.	7 days ago
web	Fixed the file sizes on the web page.	3 years ago
LICENCE.html	Added licensing information.	4 years ago
README	Add link to DC10 Project GitHub page.	2 years ago



Updating SPECCHIO binaries

New binaries are available via Gitub: https://github.com/ahueni/SPECCHIO/tree/SPECCHIO_Master/pkg

The screenshot shows the GitHub repository page for `ahueni / SPECCHIO`. The repository has 3 forks and 1 star. The code branch is `SPECCHIO_Master`. The commit history shows several updates, including a bug fix for metadata insert issues and updates to client and database definitions. The repository also contains installation packages like `specchio-client-MacOSX-installer.jar` and `specchio-webapp.war`.

File	Description	Last Commit
<code>specchio-client-MacOSX-installer.jar</code>	V3.2.1.6	13 days ago
<code>specchio-client-installer.jar</code>	V3.2.1.6	13 days ago
<code>specchio-client.zip</code>	V3.2.1.6	13 days ago
<code>specchio-database-definition.zip</code>	Updated package to include V3.2.0 and upgrade scripts to 3.2.2	2 months ago
<code>specchio-webapp.war</code>	Bug fix for a metadata insert issue (e.g. in UTC computation)	2 days ago

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/ahueni/SPECCHIO/tree/SPECCHIO_Master/conf/mysql

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

Branch: SPECCHIO_Master ▾	SPECCHIO / conf / mysql /	Create new file	Upload files	Find file	History
	ahueni Bug fixes in schema for MySQL 5.7			Latest commit 4da85fb 14 days ago	
	..				
	SPECCHIO_V3.0.2.sql	Merge commit 'a1de1503c954dcc18205a2ffda5dd55aaffb2ebf' into SPECCHIO...		a year ago	
	SPECCHIO_V3.1.3.sql	Added version number		2 years ago	
	SPECCHIO_V3.2.0.sql	Bug fixes in schema for MySQL 5.7		14 days ago	
	SPECCHIO_V3.2.0_Beta.sql	Added missing triggers to database dump		2 years ago	
	ands_attributes_definition.sql	Fixed an error when building the FOR code taxonomy.		3 years ago	
	sdb_admin_creation.sql	Removed insert of sdb_admin into specchio_user_group as already		2 years ago	
	specchio_database_upgrade_V2.x....	Changes to upgrade the online version of SPECCHIO		14 days ago	
	specchio_database_upgrade_V3.0....	Added schema info		2 years ago	
	specchio_database_upgrade_V3.0....	Changes to upgrade the SPECCHIO online database and avoid conflicts w...		14 days ago	
	specchio_database_upgrade_V3.1....	script renames and various upgrades		2 years ago	
	specchio_database_upgrade_V3.2....	Bug fix of schema version number to floating point.		14 days ago	
	specchio_database_upgrade_V3.2....	New taxonomies, new attributes and boolean support		14 days ago	



Coding tutorials: where to go from here

University of Zurich^{UZH}
Department of Geography

RSL measurements | products | policy

Matlab accessing SPECCHIO

Andy Hueni¹

¹Remote Sensing Laboratories, University of Zurich, Switzerland

SPECCHIO Programming Course
Zurich 2017

17/04/17 OPTI-MISE EU COST RSL Page 1

University of Zurich^{UZH}
Department of Geography

RSL measurements | products | policy

R accessing SPECCHIO

Andy Hueni¹

¹Remote Sensing Laboratories, University of Zurich, Switzerland

SPECCHIO Programming Course
Zurich 2017

17/04/17 OPTI-MISE EU COST RSL Page 1

University of Zurich^{UZH}
Department of Geography

COST European Cooperation in Science and Technology
 RSL measurements | products | policy

Python accessing SPECCHIO

Andy Hueni¹

¹Remote Sensing Laboratories, University of Zurich, Switzerland

OPTIMISE SPECCHIO Programming Course

17/04/17 OPTIMISE Innovative Optical Tools for Proximal Sensing of Ecophysiological Processes 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

Follow

COST European Cooperation in Science and Technology
 COST is supported by the EU Framework Programmes Horizon 2020

SPECCHIO Programming Course - Promoted by [COST OPTIMISE Workshop WG1](#)

SPECCHIO Programming Course Material is provided via the links below.

Downloads:

[Presentation: The Spectral Information System SPECCHIO - Architecture and Concepts](#)

[Presentation: Matlab accessing SPECCHIO](#)

[Presentation: R accessing SPECCHIO](#)

[Presentation: Python_accessing_SPECCHIO](#)

[Input File: spectra.csv](#)

[Input File: metadata.csv](#)

Download the programming tutorials from the SPECCHIO web page:

<https://specchio.ch/programming-course/>



Next Steps: For you to decide*

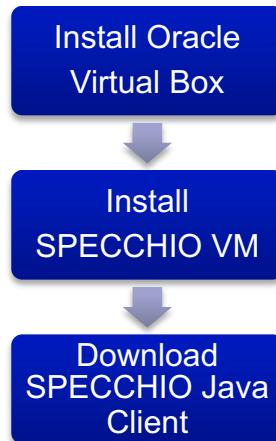
1

Web Hosting



2

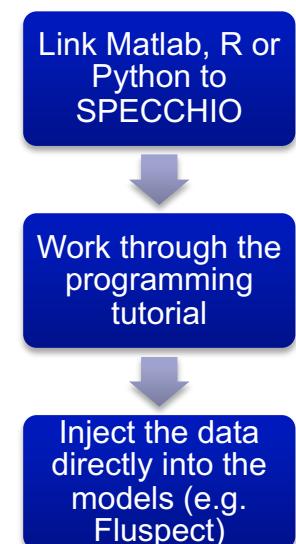
Local Hosting



GUI Access



Programmatic Access⁺



* 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

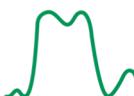
https://twitter.com/SPECCHIO_DB 



Australian Government

Department of Industry
Innovation, Science, Research
and Tertiary Education



OPTI  ISE

