Application Programming Interface (API) for the HITRANonline web service. Current state and goals

Roman V. Kochanov[1,2], Christian Hill[1], Jonas S. Wilzewski[1],
Iouli E. Gordon[1], and Laurence S. Rothman[1]

[1]Harvard-Smithsonian Center for Astrophysics
Atomic and Molecular Physics Division
60 Garden St, Cambridge MA 02138, USA

[2]Laboratory of Quantum Mechanics of Molecules and Radiative Processes,
Tomsk State University,
Prospekt Lenina, 36, Tomsk 634050, Russia

What is API?

- API, an abbreviation of *application program interface*, is a set of routines, protocols, and tools for building software applications. The API specifies how software components should interact and be used in a program code.
- **A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together.**

Why is it a powerful tool?

- Functionality
  - Normally leads to faster development
  - Takes a burden of parsing of the “strict” format
- Flexibility
  - Customizable format of the data representation
  - Possibility to extend functionality
- Interaction with other software libraries
  - Huge number of libraries (Python, Fortran, C++ ...)

What is HITRANonline API?

- Source code library written in Python
- The code is distributed as open source
- Main goal is providing a tool to work with the HITRAN data locally

Purpose of HITRANonline API

- Retrieve and use of HITRANonline data (line lists, cross sections, molecule and isotopologue info...) from a program written in Python
- Reduce the load of HITRANonline web service (calculation of lineshapes and cross sections is resource-demanding!)
- Give more flexibility in data filtering
- Provide a possibility to work with HITRAN data offline
- Make use of machines with parallel architecture (multicore cpus, clusters...)

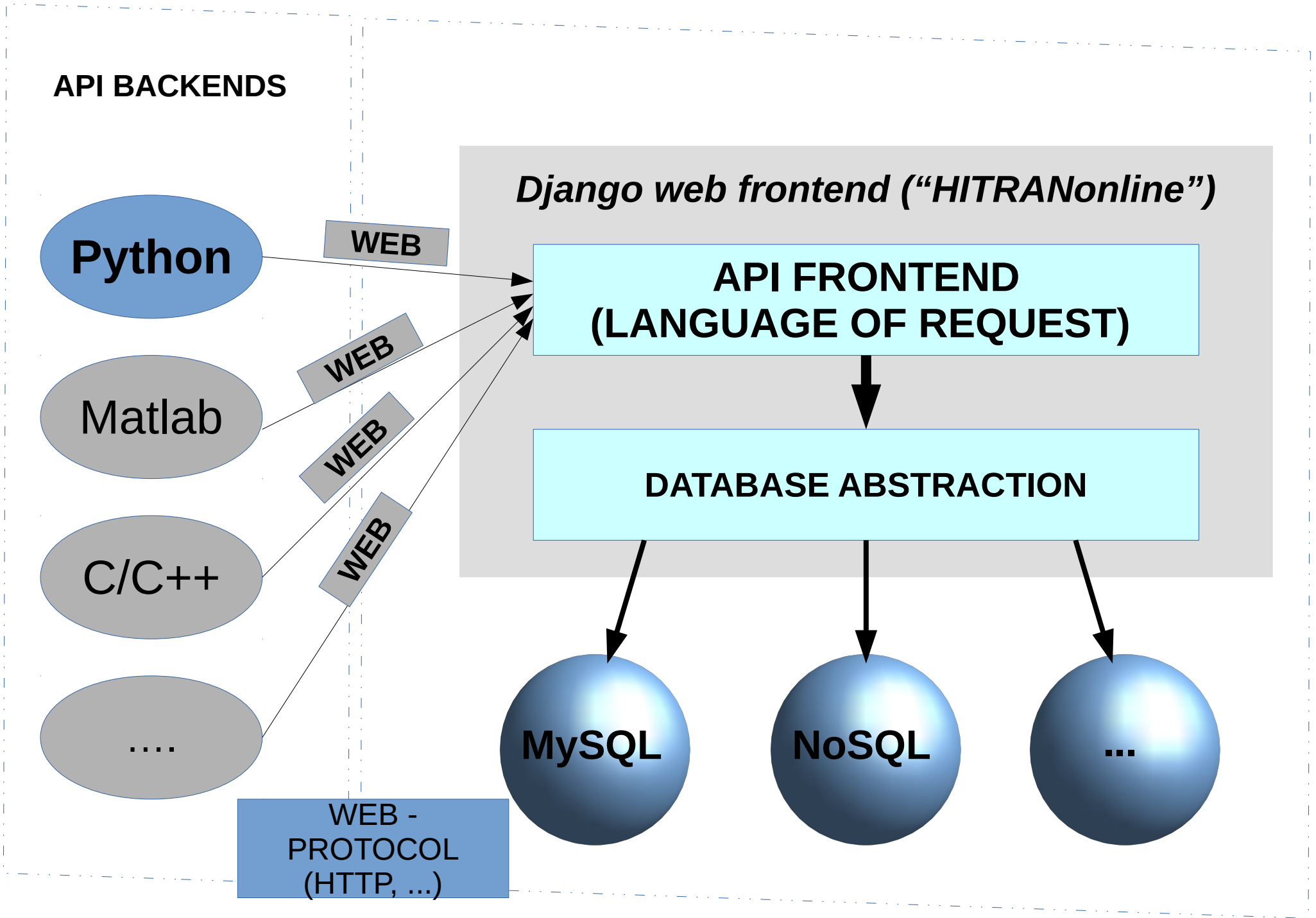
Features of HITRANonline API

- Basic capabilities: download and filter line lists (cross sections), partition functions, line profiles, cross section calculation
- Working with data locally (via DBMS)
- Formats: .par (default), CSV, XSAMS, HDF-5, JSON, “schema-free”
- Several layers of functionality (high, medium, low)
- Possibility to extend functionality by user's own routines
- Embedded documentation (getHelp)
- Fully compatible with HITRANonline database schema

Codes included (fortran)

- Total internal partition sum (**TIPS**): R. Gamache et al. Icarus 215 (2011) 391–400
  - Included information about 51 molecules
  - 70-3000K temperature range
- **PcqSDHC** line profile: Ngo et al. JQSRT 129 (2013) 89–100.
  - a.k.a. Hartmann-Tran profile, HTP
  - Can be reduced to VP, RP, qSDVP, qSDRP

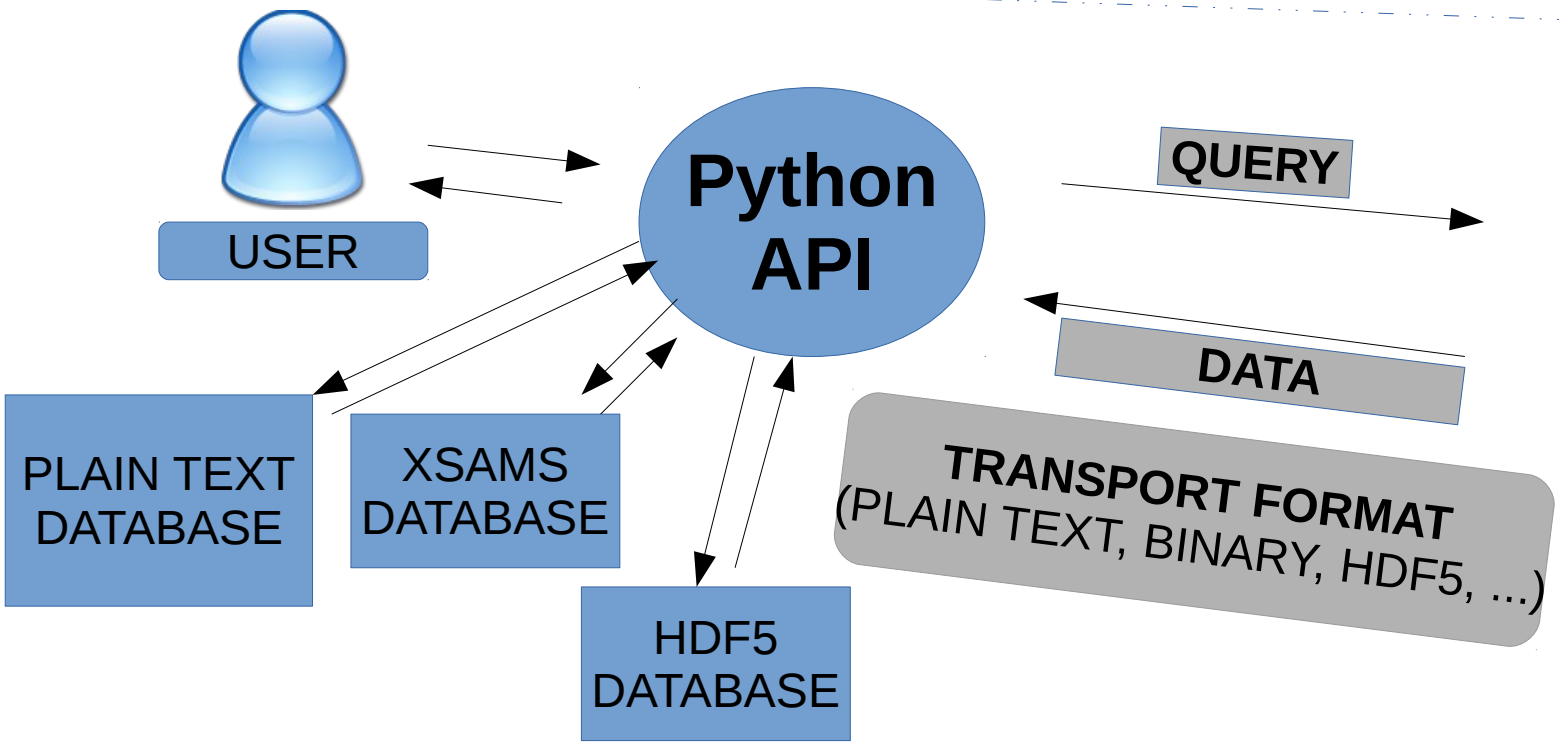
HITRANonline API Architecture



API architecture summary

- Frontend: web service based on some formal “language”
  - E.g. VAMDC query:
  - http://lts.iao.ru/node/cdsd-4000-xsams1/tap/sync?LANG=VSS2&REQUEST=doQuery&FORMAT=XSAMS&QUERY=select * where InchiKey = CURLTUGMZLYLDI-UHFFFAOYSA-N
- Backend: local library of Python functions

API structure (local)



LOCAL DATABASE operations:

→ partitionSum(…)
→ absorptionCoef(…)
→ transmittanceFunc(…)
→ calcProfile(…)
→ getColumn (…)
→ describe (…)
→ etc ...

LOCAL

REMOTE

HITRANonline

API functionality: level 0

- **Just one type of function: fetch(...)**

- This function queries HITRANonline system to get a desired piece of data and put it to the local storage (folder)
- "**RequestOptions**":
  - Molecule number?
  - Isotopologue number?
  - Wavenumber range
  - Intensity cutoff (!)
  - Desired line parameters etc.

API functionality: level 1

- API functions are expressed in physical terms:
  - partitionSum(...M,I, temp...)
  - calcProfile(ProfileType,ProfileParameters)
    - ProfileType: HT, Voigt, Lorentz, Gaussian, Galatry ...
  - absorptionCoefficient(... spectral parameters …)
    - Include: environment dependence (temperature, pressure, path length), line shifts, mixtures...
    - different types of spectra (absorption coefficient, absorption spectra, transmittance, radiance)
  - getEnergyLevels(LineList)
  - sortByEnergy(Options)
  - sortByBand(Options) etc...

API functionality: level 2

- Lower-level: API functions are expressed in terms of database tables
- Approach similar to SQL: line list is a table of columns (nu, S, A, etc...)
- **+** queries are much more expressive
- **–** queries are more complex
- Basic functions: select(...), getColumns(...), dropTable(...), sort(...), group(...)
- Full documentation and tutorials on API will be available on hitranazure.cloudapp.net or by using the function getHelp( )

Typical function/subroutine

select(TableName,DestinationTableName,ParameterNames,Conditions)

TableName = source table

DestinationTableName = table or stdout

ParameterNames = list of required parameters:

e.g. ('M','I','nu','S')

Conditions = structure containing all restrictions on parameters:

('BETWEEN' , 'nu' , 99.5 , 100) => 99.5 <= nu <=100

('AND',('IN','I',('SET',[1,2,3])),('<','nu',50)) => I in {1,2,3} and nu<50

Conditions = **any logical expression** containing the following operations:

{ 'AND', 'OR', 'NOT', 'RANGE', 'IN', '<', '>', '<=', '>=', '==', '!=', 'LIKE', 'STR', '+', '-' ,'*' ,'/' ,'MATCH' ,'SEARCH' ,'FINDALL' }

API live demonstration (IPython notebook)

Plans

- High-speed computation of cross-sections
- More data formats (XML, JSON, netCDF …)
- More profiles (soft collisions, line mixing)
- Verify with FASCODE, Kelly Chance's code (HIT-CROSS), HITRAN-on-the-WEB
- Add unit conversion
- Release: end of 2014

Thank you for your attention!