ChemAxon
discovery informatics

JChem Web Services

Gabor Guta, PhD

Agenda

- Products Overview
- JChem Web Services „Classic”
  SOAP examples in Python
- „Brand New“ JChem Web Services
  REST examples in JavaScript
  and Python
- Future Plans

ChemAxon

Goal of the Product

- Provide a **simple and ergonomic** access to JChem functionality from Web Sites (JavaScript, PHP)
- Integrate JChem products from **non-Java SE/EE environments**: Python, Perl, R, etc.
- Provide robust way to add JChem products to your existing (Java) **Enterprise Service Bus**

ChemAxon

Product Overview

- **JChem „Classic" Web Services**

  This is our existing SOAP Web Service platform – Heavy, Robust

- **JChem Web Services**

  Our brand new REST Web Services product – Simple, Lightweight

ChemAxon

SOAP/REST Product Features

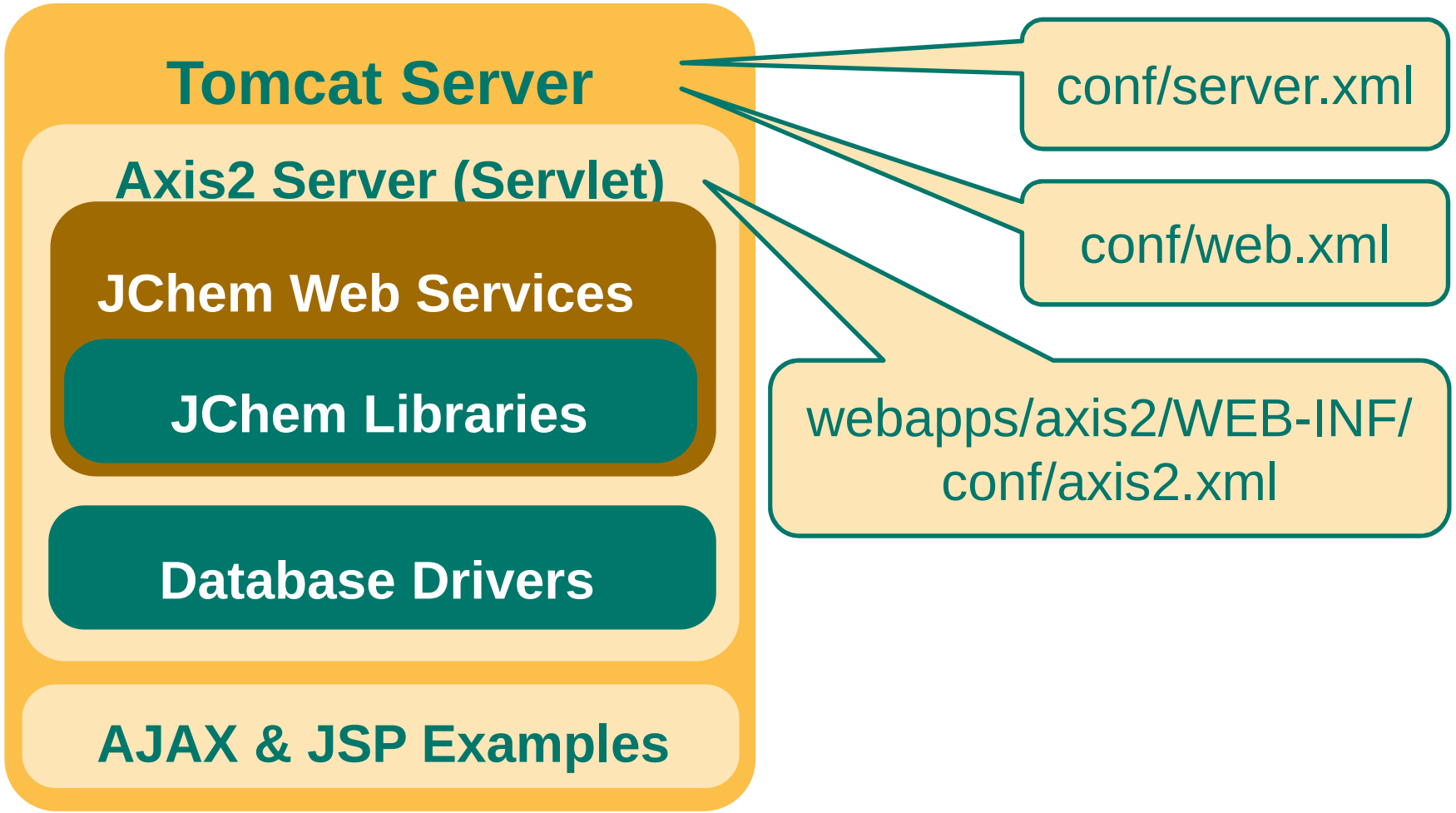
	SOAP	REST
 Database Related Services	Y	6.0
 Molecular Conversion Service	Y	6.0
 Standardization Service	Y	6.1
 Chemical Terms Service	Y	6.0
 Reactor Service	Y	6.1
• User Management	N	6.1

ChemAxon

SOAP Examples in Python

ChemAxon

SOAP Web Services Bundeled with a Tomcat



ChemAxon

List Available Services

- The default address is of the installed tomcat is localhost:8180
- You can find the list of the services at: http://localhost:8180/axis2/services/listServices
- The WSDL description of the Chemical Terms service can be found at: http://localhost:8180/axis2/services/ChemicalTermsWS?wsdl
- Documentation: http://www.chemaxon.com/webservices/developersGuide.html

SOAP Request

```
<soapenv:Envelope
        xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:web="http://webservice.jchem.chemaxon">
  <soapenv:Header/>
  <soapenv:Body>
    <web:evaluate>
      <!--Optional:-->
      <web:target>c1ccccc1</web:target>
      <!--Optional:-->
      <web:expression>atomCount</web:expression>
    </web:evaluate>
  </soapenv:Body>
</soapenv:Envelope>
```

ChemAxon

SOAP Response

```
<soapenv:Envelope
        xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:evaluateResponse
        xmlns:ns="http://webservice.jchem.chemaxon">
      <ns:return>12</ns:return>
    </ns:evaluateResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

ChemAxon

Python Environment

- **The example is written for the following configuration:**
  - **Python versions 3.x**
  - **SUDS-JURKO (0.4.1.jurko.4)**

ChemAxon

MolConverter Demo

```
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import suds.client as cl
>>> molconverter=cl.Client(
        'http://localhost:8180/axis2/services/MolConvertWS?wsdl')
>>> molconverter.service.convert('OC=O', 'mol')
Mrv0541 09111208282D

  3  2  0  0  0  0            999 V2000
    1.2375   -0.7145    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    1.9520   -1.1270    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.6664   -0.7145    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  2  0  0  0  0
M  END
```

ChemAxon

MolConverter Demo (cont‘d)

```
>>> import binascii
>>> encImg=molconvert.service.convert('OC=O',
      'jpeg:w160,h100,Q95,#C0CDC0')
>>> imgFile=open('D:\\test.jpg', 'wb')
>>> imgFile.write(binascii.a2b_base64(bytes(encImg,
      'UTF-8')))
4456
>>> imgFile.close()
```

ChemAxon

Simple Database Usage Pattern

- ConnectionWS: Connect to JChem database
- JChemSearchWS: Search & Retrieve
  (multiple times)
- ConnectionWS: Close

ChemAxon

REST/Python Example

ChemAxon

REST Overview

- A single WAR file which can be deployed in a servlet container
- User Guide
- JS Example
- Python Example
  - Simple Query
  - Query with Calculation
  - Format Conversion

ChemAxon

Testing Details

- Usability Testing
  - We build our own Web Application on the top of this API
- Current Test Environments
  - DB: Oracle, MySQL, MSSQL, Derby
  - App Server: Jetty, Tomcat
- We are going to add additional systems on customer request

ChemAxon

Demo Environment

- https://www.chemaxon.com/download.php?d=/data/download/webservices2/0.8.0-developer-preview

ChemAxon

JavaScript Examples

- 1. Getting a Molecule from the Database
- 2. Display a Search Result
- 3. Displaying Calculation Results
- 4. Structure Table

ChemAxon

Python Example

```
import requests
import json
url="http://localhost:8080/webservices2/
rest-v0/data/sample/table
/ChEBI_lite_3star/search"
paging = {"offset": 0, "limit": 15}
result=requests.get(url, params=paging)
table=result.json()['data']
for row in table:
      print("("+str(row['cd_id'])+")„
            +row['c_chebi_id']+": „
            +row['c_chebi_name'])
```

ChemAxon

Python Example 2

```
import requests
import json
url = ...
searchOptions = ...
headers = {'content-type': 'application/json'}
resultRaw = requests.post(url, data
            =json.dumps(searchOptions),
            headers=headers)
result=resultRaw.json()
for row in result["data"]:
      print(row)
```

ChemAxon

Python Example

Similarity Settings

Mol. Mass < 500

Additional Chemical Terms Fields

```
{"s    chOptions": {
        "queryStructure": "c1ccccc1",
        "searchType": "SIMILARITY",
        "similarity": {"similarityThreshold": 0.55}},
 "filter": {
        "simpleConditions": "cd_molweight;lt;500",
        "orderBy": "cd_id"},
 "paging": {"offset": 0,   "limit": 10},
 "display": {
        "include": ["cd_id", "cd_molweight"],
        "additionalFields": {
                "logp": "chemicalTerms(logp)",
                "don": "chemicalTerms(donorCount)",
                "acc": "chemicalTerms(acceptorCount)"}}}
```

ChemAxon

Python Example 3

```
import requests
import json
url="http://localhost:8080/webservices2/
      rest-v0/util/calculate/stringMolExport"
conversionOptions = {"structure": "aspirin",
    "parameters": "mol"}
headers = {'content-type': 'application/json'}
resultRaw = requests.post(url, data
      =json.dumps(conversionOptions),
      headers=headers)
result=resultRaw.text
print(result)
```

ChemAxon

Plans

- Add Authentication
- Make Java side also extendable
- Improve ChemAxon Tool Coverage

- Update SOAP WS

ChemAxon