IBM Watson Cloud API

First contact and API promenade

Alexandre Quemy - aquemy@pl.ibm.com

April 5, 2016

IBM Analytics

PLAN

IBM Watson

IBM Watson Cloud Services

Available Services

Focus on some services

IBM WATSON

WHAT IS WATSON?

What is watson?

Difficult to answer at first sight.

WHAT IS WATSON?

What is watson?

It is an **ecosystem**:
When a researcher or engineer develop a cool feature, it becomes part of Watson ecosystem.

WHAT IS WATSON ?

Watson Equations

- AI methods + NLP = Watson Core
- Watson Core + (REST) API + Bluemix = Watson Cloud Services
- WCS + Engineer imagination = Cognitive era

IBM WATSON CLOUD SERVICES

IBM WATSON CLOUD SERVICES

IBM Watson

IBM Watson Cloud Services

- Generalities - Development - Deployement
- Programming models
- A note on Bluemix
- A note on AlchemyAPI

Available Services

Focus on some services

Generalities

- Each service provides a **REST API**.
- Some services provide additional interfaces (e.g. Websocket).
- **Bluemix Cloud Service** to deploy applications.
- Two priviledged languages: **Java** and **Node.js**.
- But also Python and Android SDK in beta.

Development Tools

- Using the Bluemix web interface.
- Using the Eclipse IDE (dedicated pluggin).
- Using the Cloud Foundry CLI (PaaS by VMWare).

IBM WATSON CLOUD SERVICES

IBM Watson

IBM Watson Cloud Services

- Generalities - Development - Deployement
- Programming models
- A note on Bluemix
- A note on AlchemyAPI

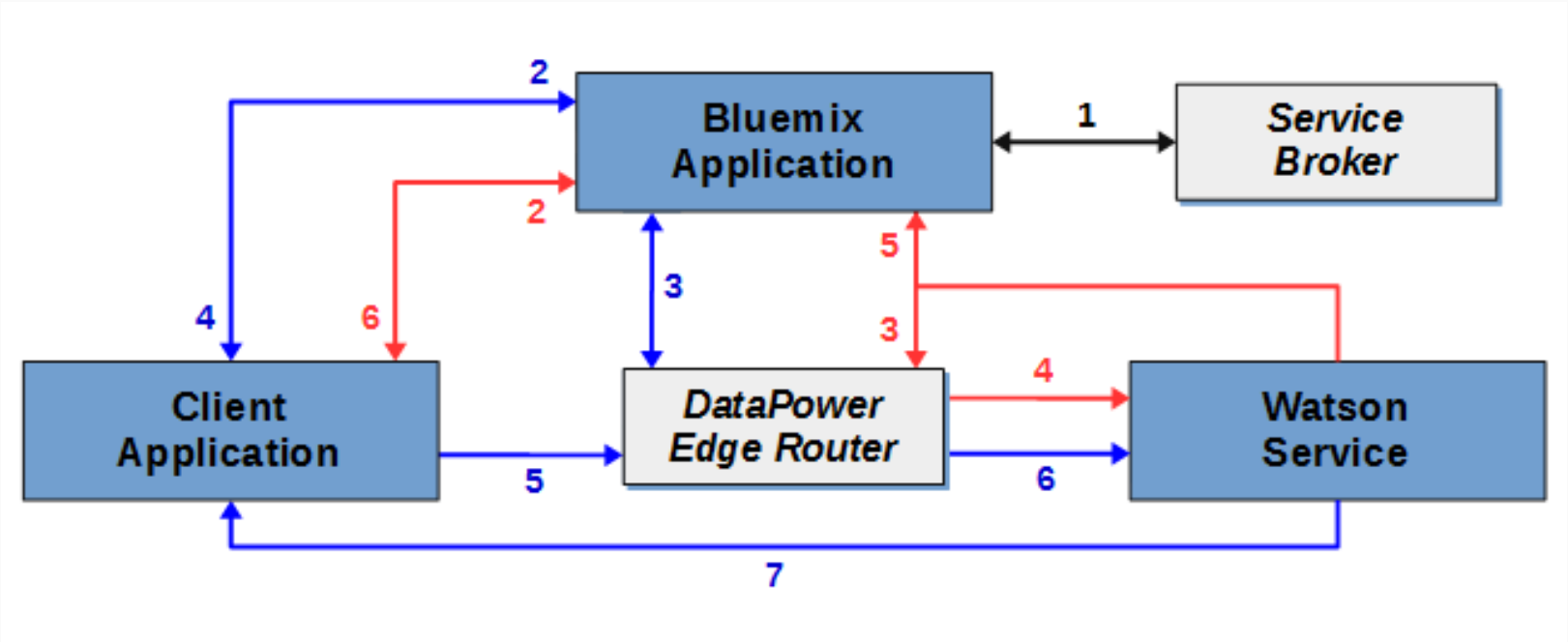
Available Services

Focus on some services

Programming models

- Server-side proxy.
  1. Relies only on credentials.
  2. Data stored by Bluemix.
  3. Safer by default, even on HTTP.
  4. Higher latency and lower performances.
- Direct interaction with a service.
  1. Relies on credentials + tokens.
  2. Data stored only at client-side.
  3. One hour lifetime token.
  4. XSS vulnerable. Be careful !

IBM WATSON CLOUD SERVICES



IBM WATSON CLOUD SERVICES

IBM Watson

IBM Watson Cloud Services

- Generalities - Development - Deployement
- Programming models
- A note on Bluemix
- A note on AlchemyAPI

Available Services

Focus on some services

Instant instances

- Support Java, Javascript, Go, Python, PHP, Ruby.
- Free for 30 days, up to 2Go.
- All the time free with restrictions:
  1. 1 concurrent instance.
  2. up to 512Mo.
  3. $<$ 100 clients.
  4. $<$ 300 API requests per second.

Containers

All the time free with restrictions:

1. 4 containers.
2. 20 Go external storage.
3. 2 public IP adresses.
4. 128Mo RAM per container.

IBM WATSON CLOUD SERVICES

IBM Watson

IBM Watson Cloud Services

- Generalities - Development - Deployement
- Programming models
- A note on Bluemix
- A note on AlchemyAPI

Available Services

Focus on some services

IBM WATSON CLOUD SERVICES

AlchemyAPI

- Part of IBM Watson Services since march 2015.
- Very rich API: would require a lecture juste for one service!

- http://alchemyapi.com/developers/getting-started-guide
- http://alchemyapi.com/products/demo

USEFUL RESOURCES

To click:

- Mixing several Watson services: The API Mashup Guide.
- More about Deep Learning: deeplearning.net.
- Processing Engine: Spark.
- Data visualization: Tableau.
- State of art in ML: TAO Inria Publications.
- LuaJIT over C/CUDA: Torch.
- Internet of Things the easy way: crossbar.io and autobahn.

FURTHER TRAINING

Upcoming 2016 Sessions

- Social Media Monitoring Enhanced with Watson APIs - April 20
- Social Media Monitoring APIs in a BI Dashboard - May 4
- Enhancing the Results of Your BI Dashboard - May 18
- How To Partner with the Watson Ecosystem - June 1
- Social Media Monitoring in Practice - June 15

More info: http://www.pages03.net/ibmwatson/building-with-watson-web-series

AVAILABLE SERVICES

AVAILABLE SERVICES

IBM Watson

IBM Watson Cloud Services

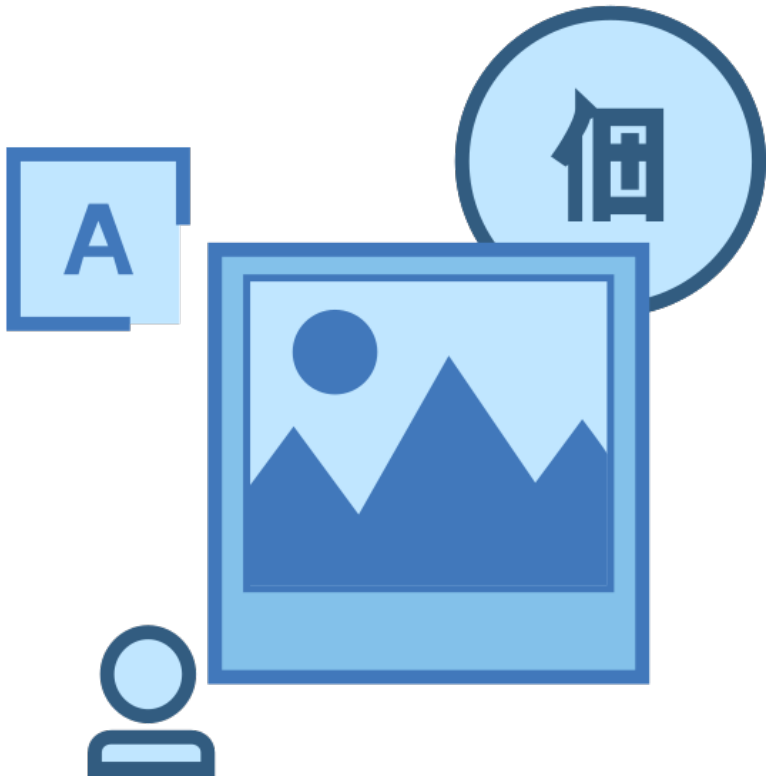
Available Services

- Services presentation: Language
- Services presentation: Speech
- Services presentation: Vision
- Services presentation: Data Insights

Focus on some services

ALCHEMYLANGUAGE

ALCHEMY LANGUAGE?



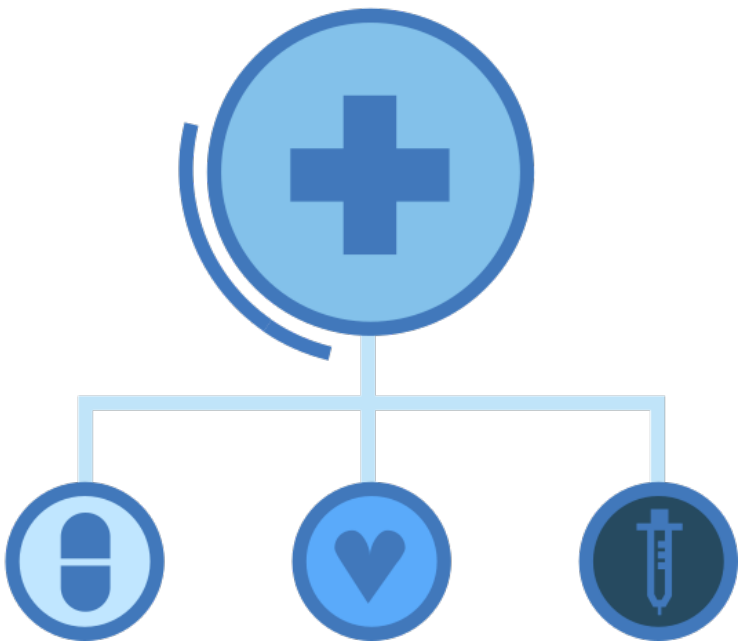
AlchemyLanguage

Collection of API for Natural Language Processing:

- Entity Extraction
- Sentiment Analysis
- Keyword Extraction
- Concept Tagging
- Relation Extraction
- Taxonomy Classification
- Author Extraction
- Language Detection
- Text Extraction
- ...

CONCEPT EXPANSION

CONCEPT EXPANSION



CONCEPT EXPANSION

Concept Expansion

Analyze text to create dictionnary of concepts and related words.

Usage example

Creating cross-domain dictionnary to understand different words used for the same thing.

CONCEPT INSIGHT

CONCEPT INSIGHTS

Concept Insights

- Analyze a text input to extract concepts and propose related content from different sources.
- Generate concept graphes or used default one (Wikipedia).

Usage example

Automatically provide further readings for articles, targeted advertisement, automatic content tagging.

DIALOG


I'm Watson! I can help you order a pizza. What size?
Great! Can I get a medium?
Perfect. What toppings are you in the mood for today?

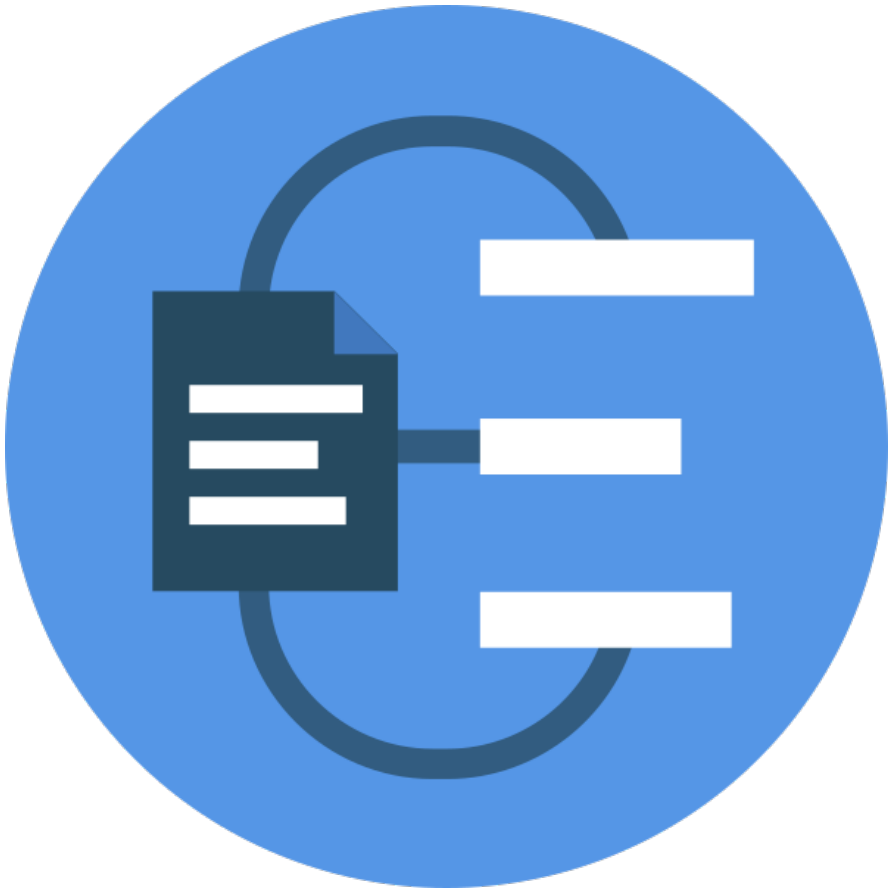
Dialog

- Creation of natural language dialogues.
- Include profile tracking for learning.

Usage example

Ordering online, first line online support, intelligent home automation.

DOCUMENT CONVERSION



Document Conversion

- Convert document in another format.
- Mainly use to prepare document for other services.

LANGUAGE TRANSLATION

Language Translation

Based on Statistical Machine Translation developped by IBM:

- Bayesian approach, with maximum likelihood estimation.
- Different from Ruled-Based Machine Translation.
- Different from Example-based Machine Translation.

NATURAL LANGUAGE CLASSIFIER

NATURAL LANGUAGE CLASSIFIER

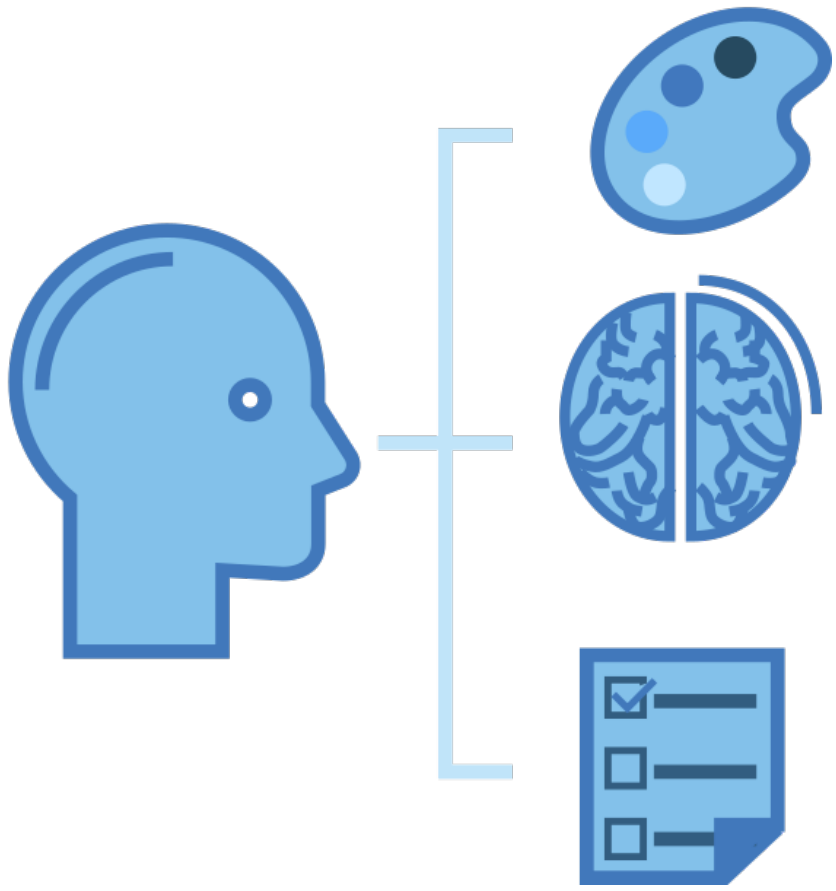
Natural Language Classifier

- Deep learning classification for Natural Language queries.
- Provide classification with confidence interval.

Usage example

Automatic tagging of content, classification of emails, intelligent dialogues.

PERSONALITY INSIGHTS



PERSONALITY INSIGHTS

Personality Insights

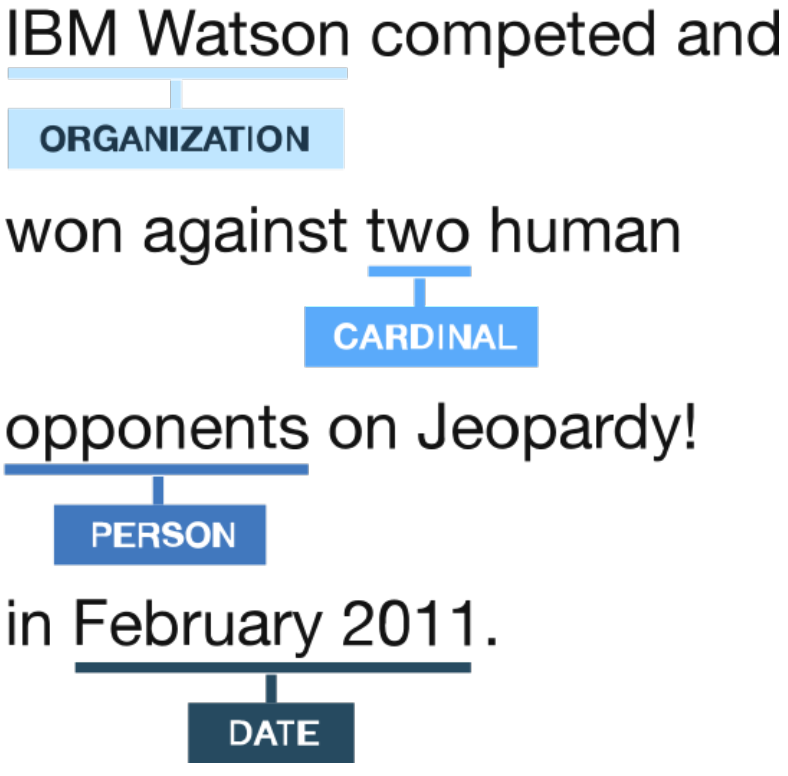
- Personality (Five Factors Model), needs and values extraction.
- Needs at least 3.5k words.
- Any source: twitter, book,...

Usage example

Individual customer service or marketing, job applicant analysis, market analysis.

RELATIONSHIP EXTRACTION

RELATIONSHIP EXTRACTION



RELATIONSHIP EXTRACTION

Relationship Extraction

- Linguistic analysis combined with Machine Learning.
- Extraction of entities in non-structured documents.

Usage example

Trend change detection, semantic search engine.

RETRIEVE AND RANK

Retrieve and Rank

- Find the most relevant documents to answer a question.
- Combination of search engine and Machine Learning.

Usage example

Intelligent diagnosis system.

TONE ANALYZER

TONE ANALYZER

Tone Analyzer

- Analyze emotion and language tones in text.
- Provide Emotion, Language and Social analysis.

Usage example

Study on the impact of your anouncements.

AVAILABLE SERVICES

IBM Watson

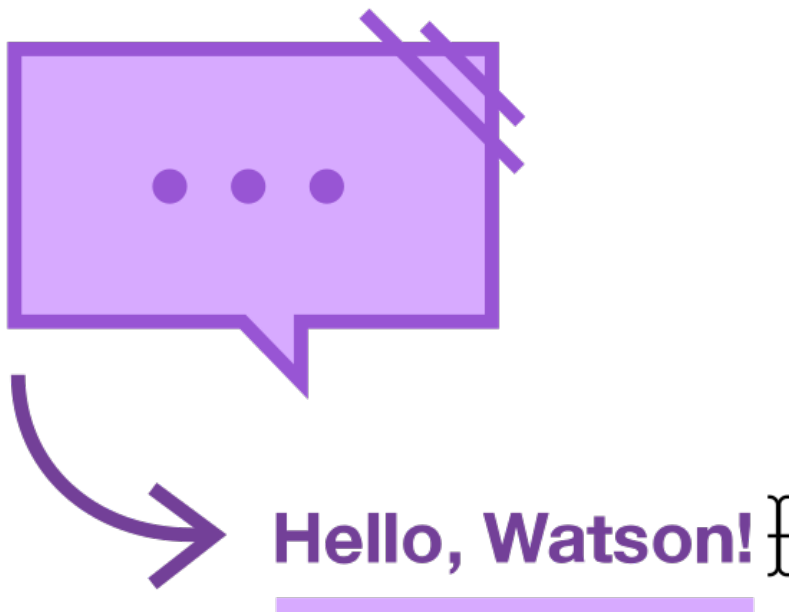
IBM Watson Cloud Services

Available Services

- Services presentation: Language
- Services presentation: Speech
- Services presentation: Vision
- Services presentation: Data Insights

Focus on some services

SPEECH TO TEXT



SPEECH TO TEXT

Speech to Text

- IBM speech recognition to text output.
- Stream or recorded input.
- Several languages supported.
- Keyword extraction.

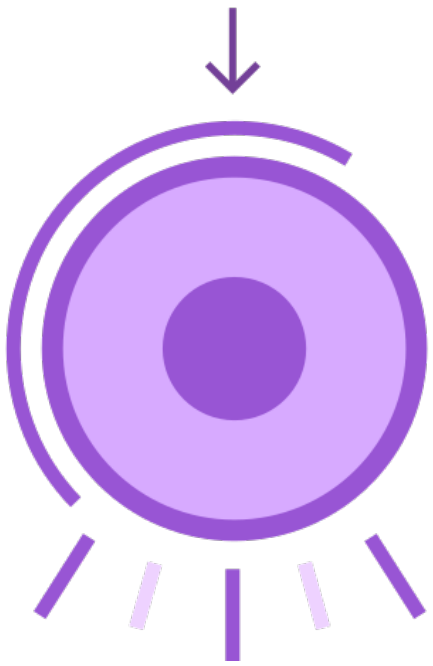
Usage example

Accessibility, transcription of conferences or meetings.

TEXT TO SPEECH

TEXT TO SPEECH

What can I do for you today



TEXT TO SPEECH

Text to Speech

- Low latency audio synthesis.
- Several languages supported.
- Specific words pronunciation tuning through the REST API.

Usage example

Accessibility.

AVAILABLE SERVICES

IBM Watson

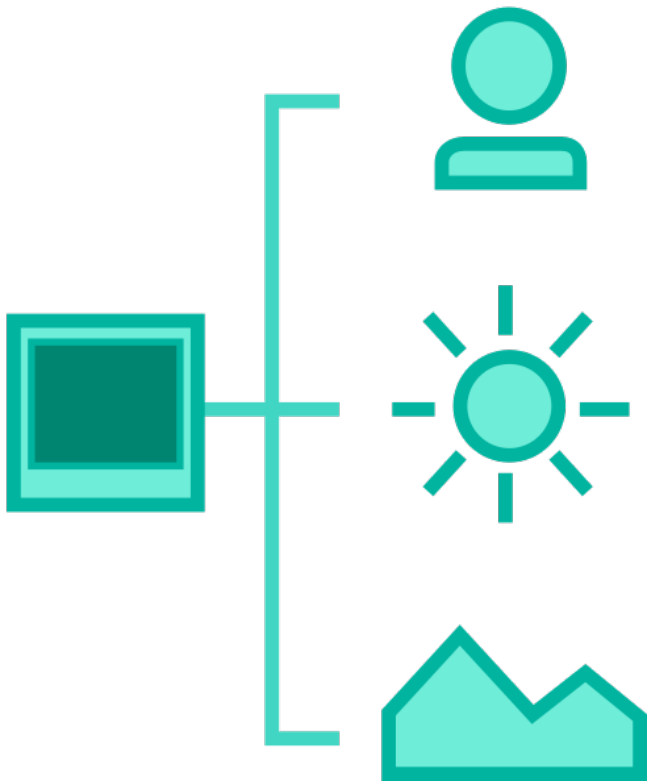
IBM Watson Cloud Services

Available Services

- Services presentation: Language
- Services presentation: Speech
- Services presentation: Vision
- Services presentation: Data Insights

Focus on some services

A_CHEMY VISION



ALCHEMYVISION

AlchemyVision

- Face, words, persons detection.
- Age estimation, tagging and classification.
- Similar content suggestions.

Usage example

Targeted add, help for criminal recognition, image search engine.

VISUAL RECOGNITION

SATURN

VISUAL RECOGNITION

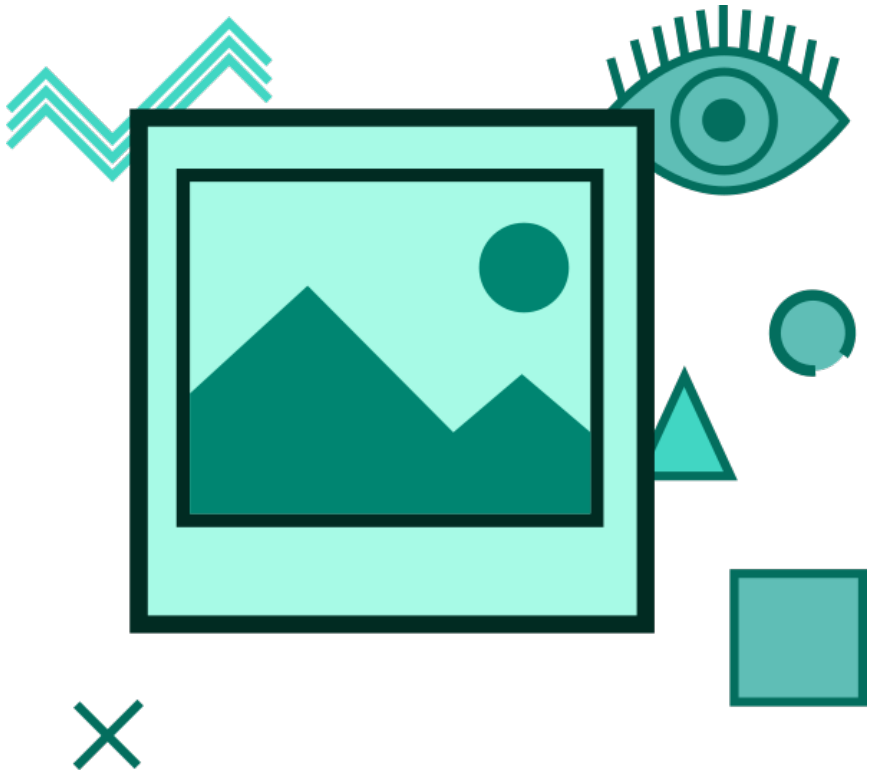
Visual Recognition

- Deep-learning methods to classify images.
- Output tags with confidence interval.

Usage example

Automatic organization of images, social media segmentation.

VISUAL INSIGHTS



VISUAL INSIGHTS

Visual Insights

- Deep-learning methods to describe **collections** of images.
- Returns tagging profile (weighted tags).
- Tags are clustered into categories.

Usage example

Automatic organization of images, social media segmentation.

AVAILABLE SERVICES

IBM Watson

IBM Watson Cloud Services

Available Services

- Services presentation: Language
- Services presentation: Speech
- Services presentation: Vision
- Services presentation: Data Insights

Focus on some services

ALCHEMY DATA

AlchemyData News

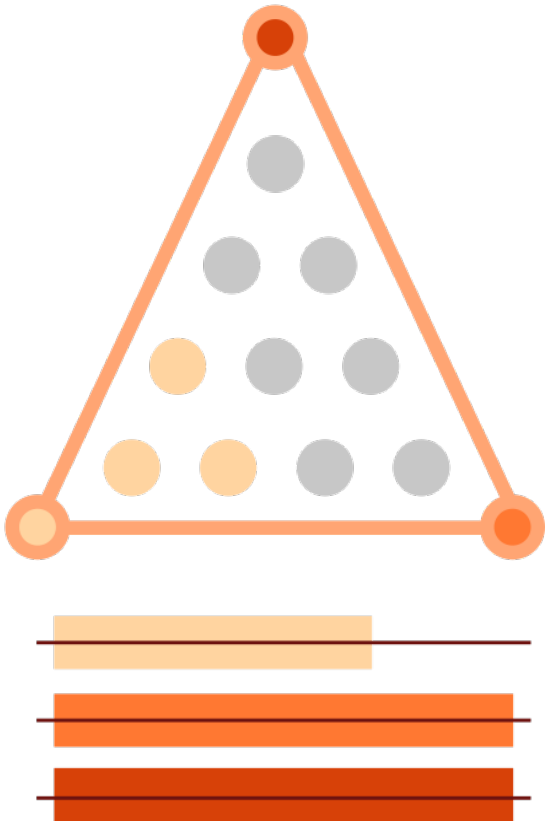
- NL query over news and blog articles.
- Over 250k articles indexed.

Usage example

Automatic content suggestions, trend analysis.

TRADEOFF ANALYTICS

TRADEOFF ANALYTICS



Tradeoff Analytics

- Provide the set of best compromises in multiobjective optimization decision problem.
- Simple Pareto Set extraction with nice interface.

Usage example

Selection of medical treatment, sorting candidates, investiment analyzis.

FOCUS ON SOME SERVICES

FOCUS ON SOME SERVICES

IBM Watson

IBM Watson Cloud Services

Available Services

Focus on some services

- A more complex use case
- Speech to text
- Natural Language Classifier
- Dialog
- A prototype

Let’s motivate
our study...

A MORE COMPLEX USE CASE

Home Assistant aka Jarvis (or almost)

1. Speech to text to get input from persons in a room.
2. Natural Language Classifier to determine the user.
3. Dialog to take order and trigger actions (turn on or off the light, put some music on, ecosystem).
4. Text to speech to communicate to the user.
5. Tone analyzer to make suggestions of music.

FOCUS ON SOME SERVICES

IBM Watson

IBM Watson Cloud Services

Available Services

Focus on some services

- A more complex use case
- Speech to text
- Natural Language Classifier
- Dialog
- A prototype

Overview

1. Transcription continuously returned.
2. Retroactively updated with new text.
3. Trained by usage.
4. REST or Websocket interface.
5. Two beta mobile SDK for Android and iOS.

Overview

Websocket prefered over http protocol.

SPEECH TO TEXT: OVERVIEW

Input configuration

1. Language: language to transcript.
2. Model: broadband (16kHz) or narrowband (8kHz).
3. Format: FLAC, PCM, WAV, Ogg (opus codec).
4. Transmission: one-shot (4MB size max), streaming (max 100MB)

Supported languages: Brazilian Portuguese, Japanese, Mandarin, Arabic, Spanish, UK and US English.

SPEECH TO TEXT: OVERVIEW

Output configuration

1. Keyword spotting with user-defined confidence level.
2. Max alternative and interims.
3. Profanity filtering.
4. Word alternatives, confidence and timestamps.

General advices

1. Use nice hardware for sound capture.
2. Boardband model for real time transcription.
3. Narrowband for sources that filter the signal (phones).
4. Tweaking is the solution to overcome sensitivity.

SPEECH TO TEXT: A FAST EXAMPLE

```
curl -u aquemy:yoloPassword -X POST
    --header ”Content-Type: audio/flac”
    --header ”Transfer-Encoding: chunked”
    --data-binary @my_record_to_transcript.flac
    ”https://stream.watsonplatform.net/speech-to-
        text/api/v1/recognize?continuous=true”
```

SPEECH TO TEXT: A FAST EXAMPLE


```
{
    "results": [
    {
        "alternatives": [
        {
            "confidence": 0.8691191673278809,
            "transcript": "this is a test"
        }
        ],
        "final": true
    }
    ],
    "result_index": 0
}
```

SPEECH TO TEXT: WEBSOCKET INTERFACE

Why Websocket?

Single-socket and full-duplex communication:

1. Simpler and more powerful than REST over HTTP.
2. Lower latency, faster (full-duplex).
3. Lighter and lower bandwidth consumption (single socket).
4. Streaming directly from web browser (part of HTML5).

SPEECH TO TEXT: WEBSOCKET INTERFACE

Single (secure) end-point:

`wss://stream.watsonplatform.net/speech-to-text/api/v1/recognize`

```
1  var token = <authentication-token>;
2  var wsURI = "<endpoint>?watson-token=" + token
3    + "&model=es-ES_BroadbandModel";
4  var websocket = new WebSocket(wsURI);
5  websocket.onopen = function(evt) { onOpen(evt) };
6  websocket.onclose = function(evt) { onClose(evt) };
7  websocket.onmessage = function(evt) { onMessage(evt) };
8  websocket.onerror = function(evt) { onError(evt) };
```

SPEECH TO TEXT: WEBSOCKET INTERFACE

```
1 var message = {'action': 'start',
2     'content-type': 'audio/l16;rate=22050'};
3 websocket.send(JSON.stringify(message));
```

```
1 {"state":"listening"}
```

Minimal initialization

- action must be start
- content-type must be in:
  1. audio/flac
  2. audio/l16
  3. audio/wav
  4. audio/ogg;codecs=opus

SPEECH TO TEXT: WEBSOCKET INTERFACE

```
var message = {'action': 'start',
    'content-type': 'audio/l16;rate=22050',
    'continuous': True, # Do not stop after 0.5s
    'inactivity_timeout': -1 # Unlimited
    'keywords': ['word'],
    'keywords_threshold': 0.95,
    'max_alternatives': 3,
    'interim_results': True,
    'word_alternatives_threshold': 0.80,
    'word_confidence': True,
    'timestamps': True,
    'profanity_filter': False
};
```

SPEECH TO TEXT: WEBSOCKET INTERFACE

```
1         socket.send(blob);
2         function onMessage(evt) {
3           console.log('recognition result in json format
                : ' + evt.data);
4         }
```

Remarks:

- Binary data only.
- No need to wait for the answer of 'start'.

SPEECH TO TEXT: KEYWORD SPOTTING RESULT

```
{
   "results": [
      ...
      "keywords_result": {
         "coffee": [
            {
               "normalized_text":"coffee",
               "start_time":0.76,
               "confidence":0.984,
               "end_time":1.18,
            },
            ...
         ],
        ...
      }
   ]
}
```

SPEECH TO TEXT: MAXIMUM ALTERNATIVES RESULT


```
{
   "results": [
      {
         "alternatives": [
            {
               "confidence": 0.8691263198852539,
               "transcript": "the first transcript"
            },
            {
               "transcript": "the second transcript"
            },
         ],
      }
   ],
}
```

SPEECH TO TEXT: WORD ALTERNATIVES RESULT


```
{
   "results": [
      "word_alternatives": [
         {
            "start_time":0.76,
            "alternatives": [
               {
                  "confidence":0.9837,
                  "word":"Chuck"
               },
               ...
            ],
            "end_time":1.18,
         },
      ],
   ]
}
```

SPEECH TO TEXT: WORD TIMESTAMPS RESULT

```
{
   "results": [
      "alternatives": [
            {
               "timestamps": [
                  ["several", 1.0, 1.51],
                  ["tornadoes", 1.51, 2.15],
                  ["touch", 2.15, 2.5],
                  . . .
               ]
            },
         ]
    ]
}
```

SPEECH TO TEXT: WEBSOCKET INTERFACE

Stop the service:

```
1        socket.send(null);
```

Or preferably:

```
1        var message = {"action": "stop"}
2        websocket.send(JSON.stringify(message));
```

Close the connection:

```
1        websocket.close();
```

FOCUS ON SOME SERVICES

IBM Watson

IBM Watson Cloud Services

Available Services

Focus on some services

- A more complex use case
- Speech to text
- Natural Language Classifier
- Dialog
- A prototype

NATURAL LANGUAGE CLASSIFIER: FAST EXAMPLE

```
curl -u aquemy:yoloPassword -X POST
    -F training_data=@data_train.csv
    -F training_metadata="{\"language\":\"en\",
        \"name\":\"TutorialClassifier\"}"
    "https://gateway.watsonplatform.net/
        natural-language-classifier/api/
            v1/classifiers"
```

NATURAL LANGUAGE CLASSIFIER: FAST EXAMPLE

Answer:

```
{
    "name": "TutorialClassifier",
    "language": "en",
    "status": "Training",
    "url": "https://gateway.watsonplatform.net/natural-
        language-classifier/api/v1/classifiers/10D41B-nlc
        -1",
    "classifier_id": "10D41B-nlc-1",
    "created": "2015-05-28T18:01:57.393Z",
    "status_description": "The classifier instance is in
         its training phase, not yet ready to accept
        classify requests"
}
```

NATURAL LANGUAGE CLASSIFIER: FAST EXAMPLE

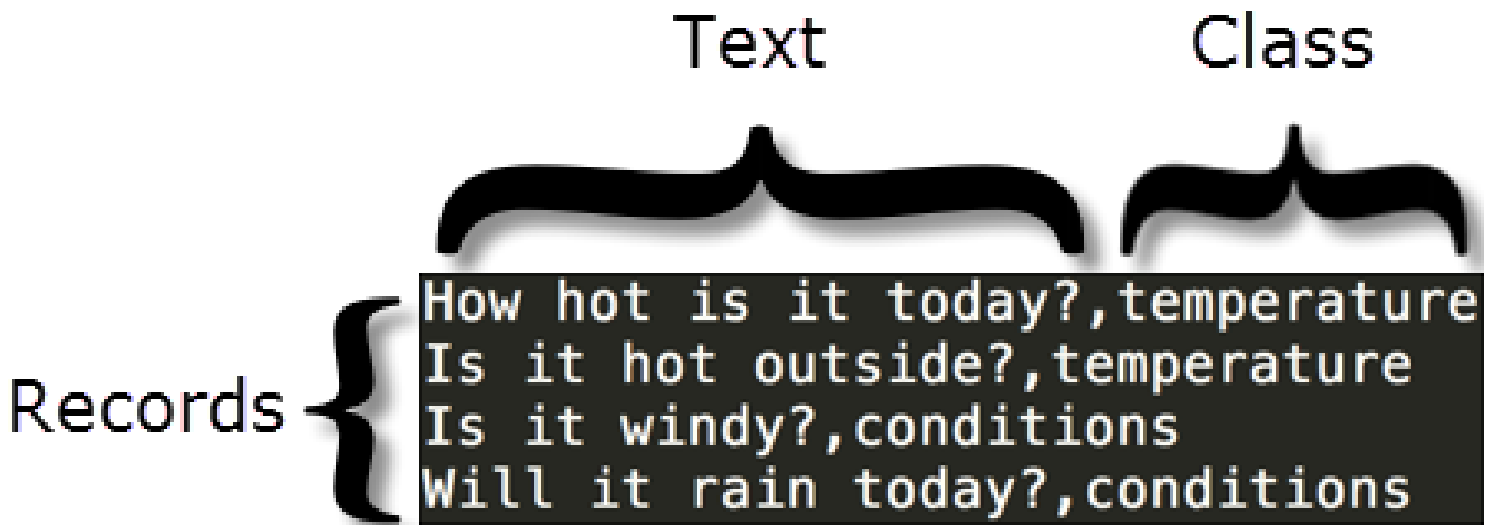
```
curl -u aquemy:yoloPassword -X POST
    "https://gateway.watsonplatform.net/
    natural-language-classifier/api/
        v1/classifiers/<classifier_id>/classify"
    --data-urlencode "text=How hot will
        it be today?"
```

NATURAL LANGUAGE CLASSIFIER: FAST EXAMPLE

Answer:

```
{
      "classifier_id": "10D41B-nlc-1",
      "url": "https://gateway.watsonplatform.net/natural
          -language-classifier/api/v1",
      "text": "How hot will it be today?",
      "top_class": "temperature",
      "classes": [
          {
              "class_name": "temperature",
              "confidence": 0.999820125854978 1
          },
          {
              "class_name": "conditions",
              "confidence": 0.00017987414502176904
          }
      ]
}
```

NATURAL LANGUAGE CLASSIFIER: USING YOUR OWN DATA



Remarks:

- CSV format, UTF-8, escaped, double quotation marks.
- Minimum of 5 records, maximum of 15000 records.
- Maximum of 1024 characters per records.
- Supported languages: en, ar, fr, it, pt, es.

NATURAL LANGUAGE CLASSIFIER: USING YOUR OWN DATA

Guidelines:

- Avoid record with more than 60 characters.
- Avoid to use more than 1̃0̃0 classes.
- Do not use less than 51̃0̃ records per class.
- Avoid to use more than 3 classes per record.

For you interest

You can use NLC Toolkit to import JSON and manage your datasets.

NATURAL LANGUAGE CLASSIFIER: REST API

GET /v1/classifiers

```
{
  "classifiers": [
    {
      "classifier_id": "10D41B-nlc-1",
      "url": "https://gateway.watsonplatform.net/natural
          -language-classifier/api/v1/classifiers/10D41B-
          nlc-1",
      "name": "My Classifier",
      "language": "en",
      "created": "2015-05-28T18:01:57.393Z"
    }
  ]
}
```

Alternatively: /v1/classifiers/<classifier_id>

NATURAL LANGUAGE CLASSIFIER: REST API

POST /v1/classifiers - REQUEST

```
{
  language: 'en',
  name: 'My Classifier',
  training_data: '<csv>', # See Node or Java example
  training_metadata: '<csv>' # Idem
}
```

NATURAL LANGUAGE CLASSIFIER: REST API

POST /v1/classifiers - OK ANSWER

```
{
  "classifier_id": "10D41B-nlc-1",
  "name": "My Classifier",
  "language": "en",
  "created": "2015-05-28T18:01:57.393Z",
  "url": "https://gateway.watsonplatform.net/natural-
     language-classifier/api/v1/classifiers/10D41B-nlc-1
     ",
  "status": "Training",
  "status_description": "The classifier instance is in
     its training phase, not yet ready to accept
     classify requests"
}
```

NATURAL LANGUAGE CLASSIFIER: REST API

GET /v1/classifiers/<classifier_id>/classify - OK ANSWER

```
{
  "classifier_id": "10D41B-nlc-1",
  "url": "https://gateway.watsonplatform.net/natural-
     language-classifier/api/v1/classifiers/10D41B-nlc
     -1/classify?text=How%20hot%20wil/10D41B-nlc-1",
  "text": "How hot will it be today?",
  "top_class": "temperature",
  "classes": [
    {
      "class_name": "temperature",
      "confidence": 0.999820125854978 1
    },
    ...
  ]
}
```

NATURAL LANGUAGE CLASSIFIER: REST API

POST /v1/classifiers/<classifier_id>/classify - REQUEST

```
{
  "classifier_id": "10D41B-nlc-1",
  "text": "How hot will it be today?",
}
```

For the answer, see GET on the same endpoint.

NATURAL LANGUAGE CLASSIFIER: REST API

DELETE /v1/classifiers/<classifier_id> - REQUEST

```
{
  "classifier_id": "10D41B-nlc-1",
}
```

FOCUS ON SOME SERVICES

IBM Watson

IBM Watson Cloud Services

Available Services

Focus on some services

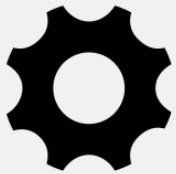
- A more complex use case
- Speech to text
- Natural Language Classifier
- Dialog
- A prototype

DIALOG: THE BIG PICTURE

Using the Dialog Service

Prepare data

- Gather representative questions
- Understand use case
- Understand business processes

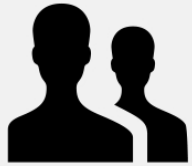


Design dialog flows

- Translate business processes to Dialog models
- Design personality
- Review potential for Watson services (NLC, STT, TTS, etc.)

Push live to users

- Connect to User Interface and publish publicly
- Monitor conversations and review activity



Iterate, optimize and add content

- Learn from real interactions
- Adjust existing content to suit user needs
- Add new content based on user activity

DIALOG: CREATING THE XML

General information

- Description in XML.
- Really rich API but also complex.

```
1  <dialog xmlns:xsi="http://www.w3.org/2001/XMLSchema-
       instance" xsi:noNamespaceSchemaLocation="
       DialogDocFormat_v0.8_manual.xsd">
2    <flow>
3      ...
4    </flow>
5    <entities></entities>
6    <constants></constants>
7    <variables></variables>
8  </dialog>
```

Let's start by the end...

DIALOG: CREATING THE XML

Variables

- Store user's inputs to be passed to the backend.
- Separated by categories in 'var_folder'.

```
<variables>
    <var_folder name="Home">
        <var_folder name="Phones" type="VAR">
            <var name="Color" type="TEXT" description="
                Phone color preference"/>
            <var name="Storage" type="TEXT" description=
                "Storage size preference"/>
        </var_folder>
    </var_folder>
</variables>
```

DIALOG: CREATING THE XML

```
1 <constants>
2     <var_folder>
3             <var name="VA" type="TAG">Virtual Agent</var
                  >
4     </var_folder>
5 </constants>
```

DIALOG: CREATING THE XML

```
1 <settings>
2     <setting name="DISPLAYNAME" type="USER">test</
          setting>
3     <setting name="LANGUAGE" type="USER">EN</setting>
4     <setting name="RESPONSETIME" type="USER">-2</setting
          >
5     <setting name="CONCEPTMATCHING" type="USER">0</
          setting>
6 </settings>
```

Exhaustive list in API documentation!

DIALOG: CREATING THE XML

Entities

Used to define enumerations.

```
<entity name="COLOR" entityType=GENERIC>
    <value name="Black" value="Black" />
    <value name="White" value="White" />
    <value name="Silver" value="Silver" />
</entity>
```

General information

- LOCATION, DATE_TIME_RANGE
- CONDITION
- AMOUNT, CURRENCY_AMOUNT, PERCENTAGE_AMOUNT
- PERSONA_NAME, PHONE_NUMBER, CREDIT_CARD_NUMBER

The folders

Group similar content. First to be defined:

- Main: entry point.
- Library: dialog structure.
- Global: global functions or nodes.
- Concepts: local concept of a dialog.

DIALOG: CREATING THE XML

```
1  <dialog xmlns:xsi="http://www.w3.org/2001/XMLSchema-
       instance" xsi:noNamespaceSchemaLocation="
       WatsonDialogDocument_1.0.xsd">
2    <flow>
3      <folder label="Main"></folder>
4      <folder label="Library"></folder>
5      <folder label="Global"></folder>
6      <folder label="Concepts"></folder>
7    </flow>
8    ...
9  </dialog>
```

DIALOG: CREATING THE XML

Ouput

Contains the response to user's input.

```
<folder label="Main">
    <output>
        <prompt selectionType="RANDOM">
            <item>Hello, welcome to Mike's Pizza</item>
        </prompt>
        <goto ref="getUserInput_2449614"/>
    </output>
</folder>
```

DIALOG: CREATING THE XML

```
<folder label="Main">
    <output>...</output>
    <getUserInput id="getUserInput_2449614">
        <search ref="folder_2449611"/>
        <default>
            <output>
                <prompt selectionType="RANDOM">
                    <item>
                        I am sorry, I did not understand
                        your question. Please try asking
                        another one.
                    </item>
                </prompt>
            </output>
        </default>
    </getUserInput>
</folder>
```

DIALOG: CREATING THE XML

```
<folder label="Library">
  <folder label="Live Content" id="folder_2449611">
    <input>
      <grammar>
        <item>What type of toppings do you have?</item>
        <item>$ What type of toppings do you have?</item
            >
        <item>$ list of toppings</item>
        <item>What * toppings * have</item>
      </grammar>
      <output>
        <prompt selectionType="RANDOM">
          <item>We have Pepperoni, Mushrooms, and
              Sausage</item>
        </prompt>
      </output>
    </input>
  </folder>
</folder>
```

DIALOG: CREATING THE XML

Search

Contains the response to user's input.

```
<output>
  <prompt>
    <item>Hi, how can I help you?</item>
  </prompt>
  <getUserInput>
    <search ref="about"></search>
  </getUserInput>
</output>
</folder>
  <folder label="Library">
    <folder label="About insurance" id="about"></folder>
</folder>
```

DIALOG: CREATING THE XML

Concept

Store some variations around a single word.

```
<concept id="123456">
    <grammar>
        <item>Hello</item>
        <item>Hi</item>
        <item>Hola</item>
        <item>Hey</item>
    </grammar>
</concept>
```

Tip

Use Concept Expansion to build concepts.

DIALOG: CREATING THE XML

Conditional Structures:

```
<if>
    <cond varName="Topic" operator="Equals">
        X-Phone 7
    </cond>
</if>
```

Some operators:

- EQUALS, EQUAL_TO_YES, EQUAL_TO_NO
- CONTAINS
- MATCHES_PATTERN
- LESS_THEN, GREATER_THEN

DIALOG: CREATING THE XML

Action

Used to set variable for profile variables

```
1 <action varName="PostalCode" operator="SET_TO">
2     33801
3 </action>
```

Some operators:

- DO_NOTHING_STR
- SET_TO_USER_INPUT(_CORRECTED)
- INCREMENT_BY, DECREMENT_BY
- SET_TO_YES, SET_TO_NOT

DIALOG: CREATING THE XML

```
1 <function>
2     <script name="Calculate">Name=Calculate Line= "{
          Phone_Number}".replace(/-/g, "").replace("(", "")
          .replace(")", "").replace(/ /g, "")</script>
3     <output>
4         <prompt/>
5             <action varName="Phone_Number" operator="
                  SET_TO">{MCT:CUSTOM:Calculate:value}</
                  action>
6     </output>
7 </function>
```

Good to know

Support type NLC_MODEL to call NLC !

DIALOG: THE REST API

Create and get the dialogs

- POST /v1/dialogs:
  send the XML file and return the id a new dialog.
- GET /v1/dialogs:
  get the list of dialogs.

DIALOG: THE REST API

Create and get a specific dialog

- DELETE /v1/dialogs/dialog_id:
  remove a specific dialog.
- GET /v1/dialogs/dialog_id:
  get a specific dialog.
- PUT /v1/dialogs/dialog_id:
  set another XML file.

DIALOG: THE REST API

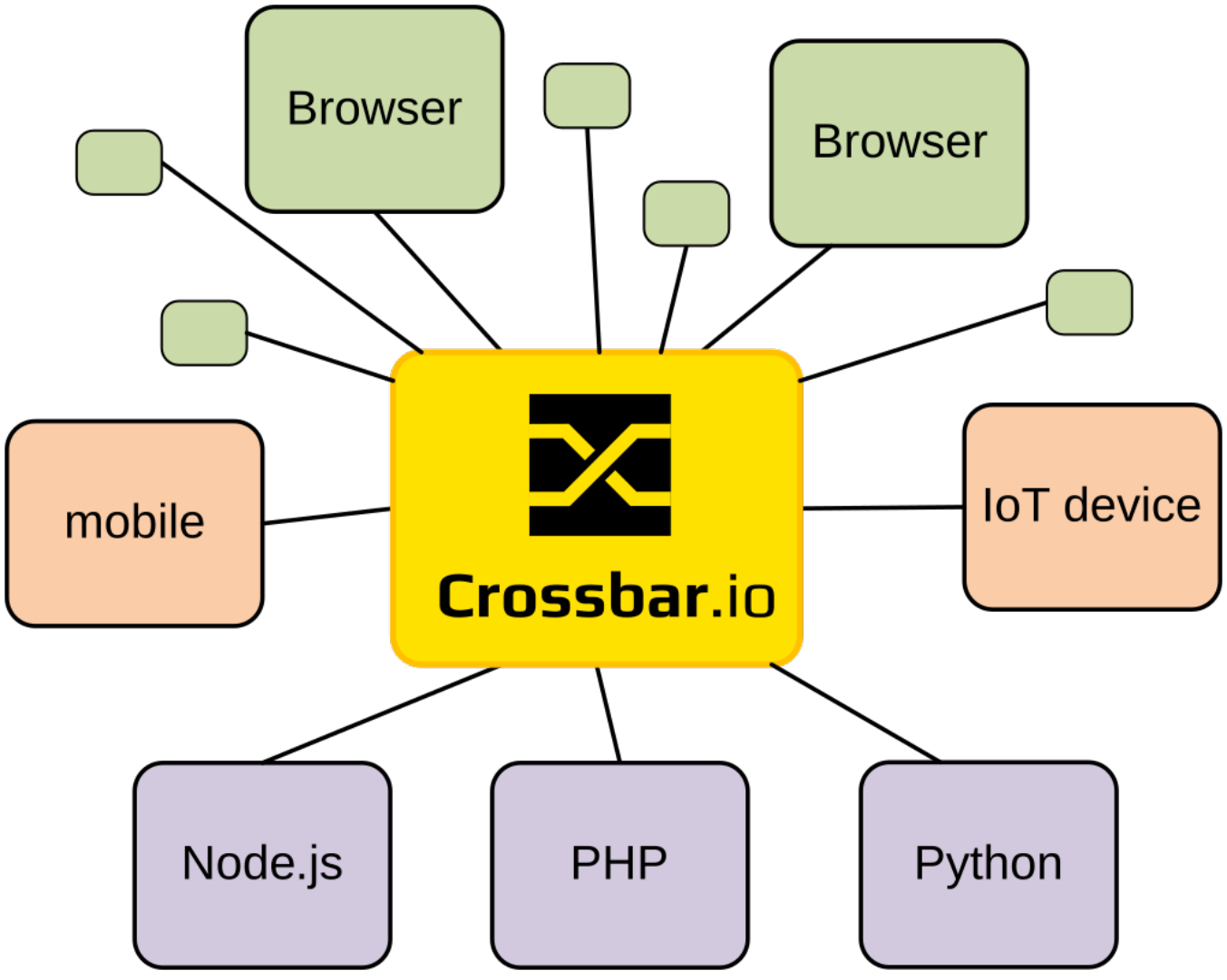
Create and get a specific dialog

- POST /v1/dialogs/dialog_id/conversation:
  start a new conversation or get response.
- GET /v1/dialogs/dialog_id/conversation:
  get history for a given range.

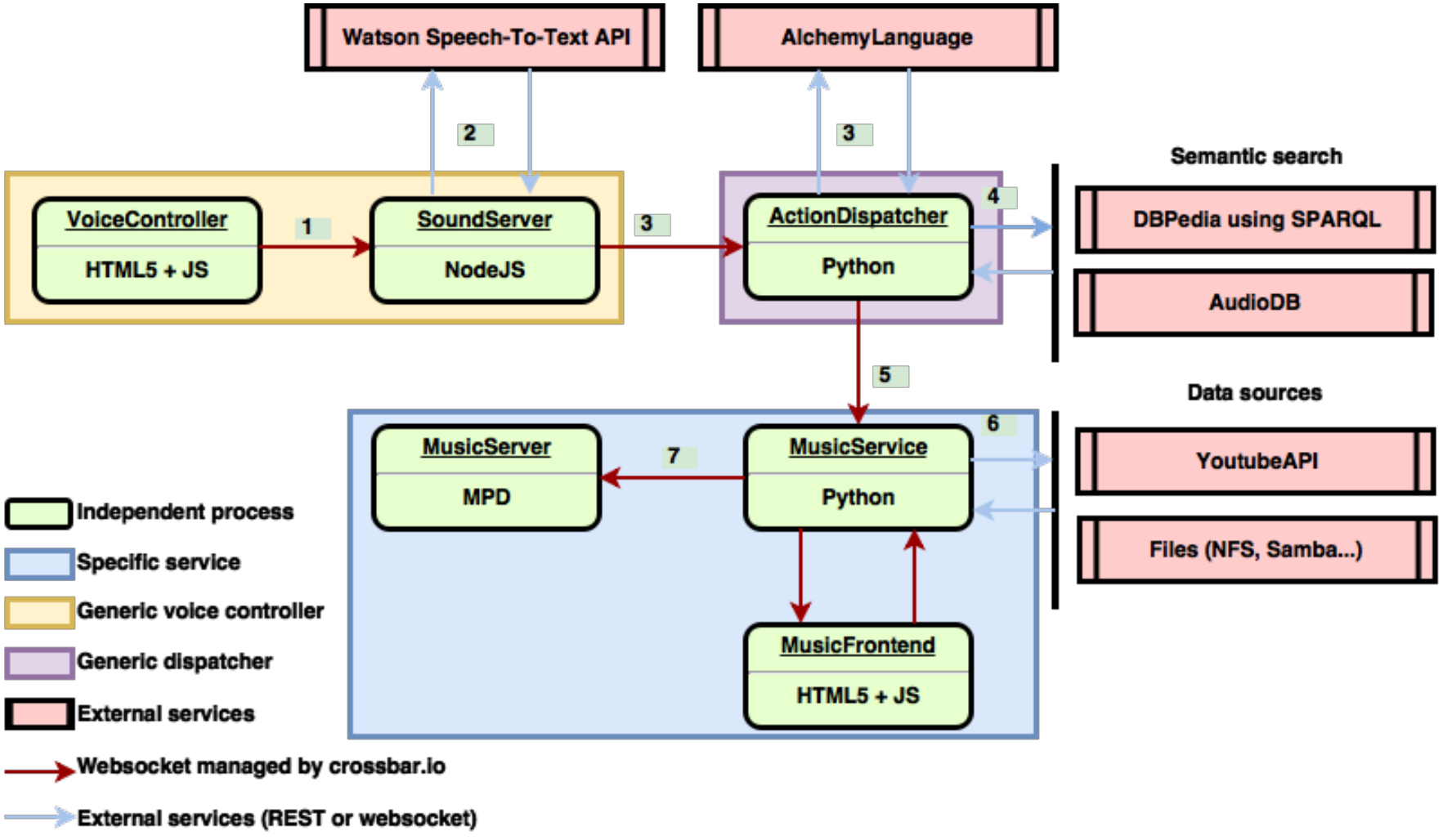
FOCUS ON SOME SERVICES

- IBM Watson
- IBM Watson Cloud Services
- Available Services
- Focus on some services
  - A more complex use case
  - Speech to text
  - Natural Language Classifier
  - Dialog
  - A prototype

A PROTOTYPE


Browser
Browser
mobile
Crossbar.io
IoT device
Node.js
PHP
Python

A PROTOTYPE


Watson Speech-To-Text API
AlchemyLanguage
2
3
Semantic search
VoiceController
HTML5 + JS
1
SoundServer
NodeJS
3
ActionDispatcher
Python
4
DBPedia using SPARQL
AudioDB
5
Data sources
MusicServer
MPD
7
MusicService
Python
6
YoutubeAPI
Files (NFS, Samba...)
MusicFrontend
HTML5 + JS
Independent process
Specific service
Generic voice controller
Generic dispatcher
External services
Websocket managed by crossbar.io
External services (REST or websocket)

A PROTOTYPE

https://github.com/aquemy/watson-homeautomation

THE END

THE END

Thank you for your attention!
Questions?