MINES PARIS
CRI

ARMINES

GeLaBa

A Framework to Define Classes of XML Documents and to Automatically Derive Specialized Infrastructures (demonstration)

Benoît PIN and Georges-André SILBER
Centre de Recherche en Informatique
Mines Paris, France
{pin,silber}@cri.ensmp.fr

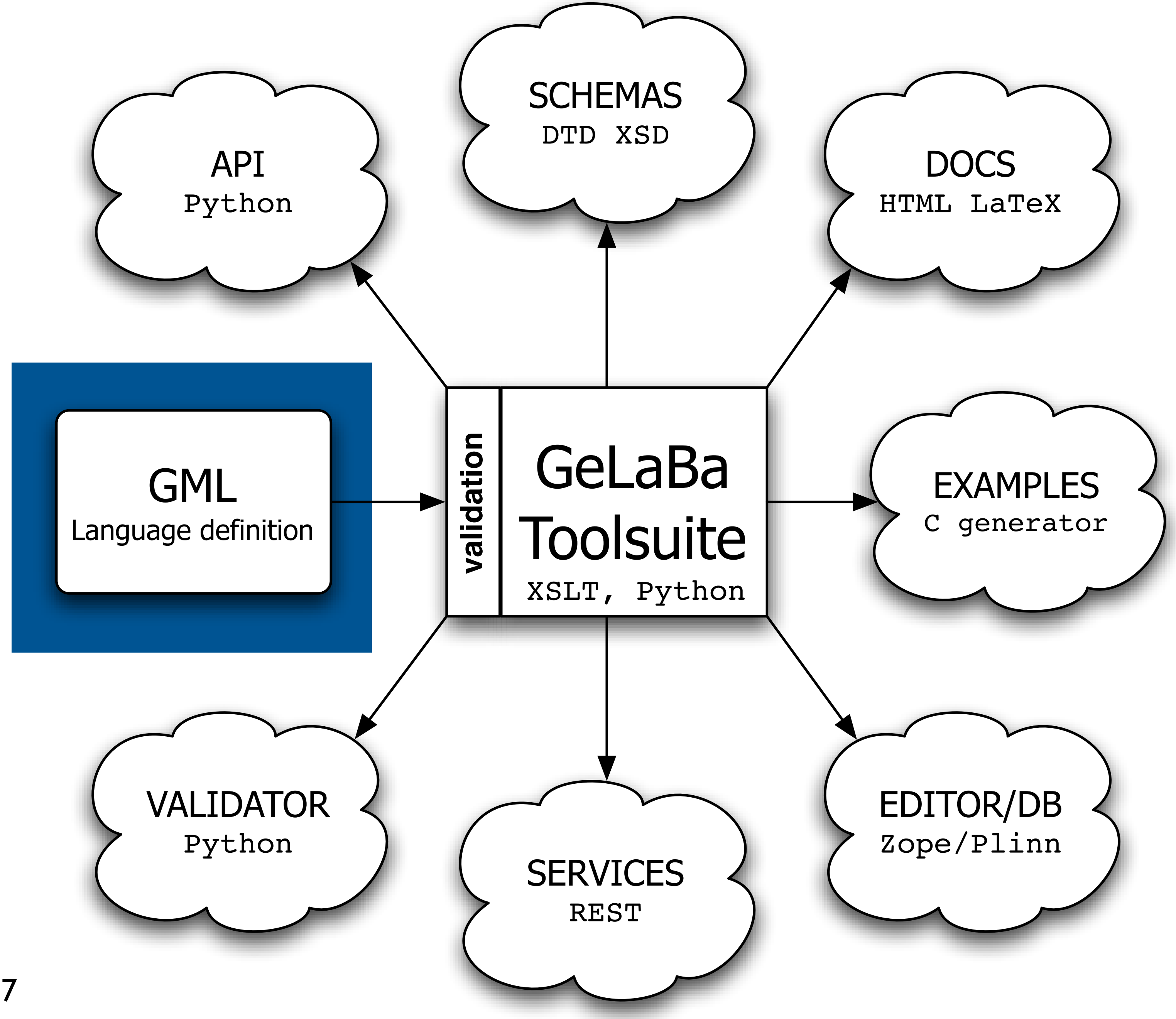
Genesis of GeLaBa

- Générateur de Langage à Balises
    - Markup Language Generator
- Project LHEO (2003), French Ministry of Work
    - First, just a DTD (127 elements)
    - Second, a documentation
    - Third, a W3C Schema
    - Then, a Web site, tools, services, etc...
- We needed a framework to maintain all of this

MINES PARIS
CRI

GeLaBa Toolsuite

GeLaBa



GML

GeLaBa Markup Language

- Another schema language
- Sequence: (a+,b?,c*)
- Choice: (a|b|c)+
- Data types: integer, float, date, text, ...
- Constraints on data types: size, regexp
- Dictionaries as data type: list of (key, value)
- Documentation for every item: french, english, ...
- Free properties

Why GML?

- Why not using W3C Schemas, DTD or RelaxNG?
- Simpler content model
  - Satisfiability is tractable and efficient with GML
  - Leads to robust and simple tools
- Original features
  - Multi-lingual documentations
  - Dictionaries
  - Properties

Running example

```
<definitions xmlns=”http://www.gelaba.org/gml/1.0”>
   <definition name=”adresse”>
    <fullname xml:lang="fr">Adresse postale</fullname>
    <fullname xml:lang="en">Postal Address</fullname>
    ...
    <element_type>
      <sequence>
        <element name="lignead" min="1" max="3" />
        <element name="codepostal" min=”1” max=”1” />
        <element name="ville" min=”1” max=”1” />
        <element name="departement" min=”1” max=”1” />
        <element name="region" min=”1” max=”1” />
        <element name="pays" min="0" max="1" />
      </sequence>
    </element_type>
  </definition>
  ...
</definitions>
```

adresse.xml

Running example

```
<definition name=”lignead”>
  <fullname xml:lang="fr">Ligne d’adresse</fullname>
  <fullname xml:lang="en">Address line</fullname>
  ...
  <element_type>
    <basic_text lgmin="1" lgmax="60"/>
  </element_type>
</definition>

<definition name="departement">
  <fullname xml:lang="fr">Département</fullname>
  ...
  <element_type>
    <basic_key ref="dict-departements-france"/>
  </element_type>
</definition>
```

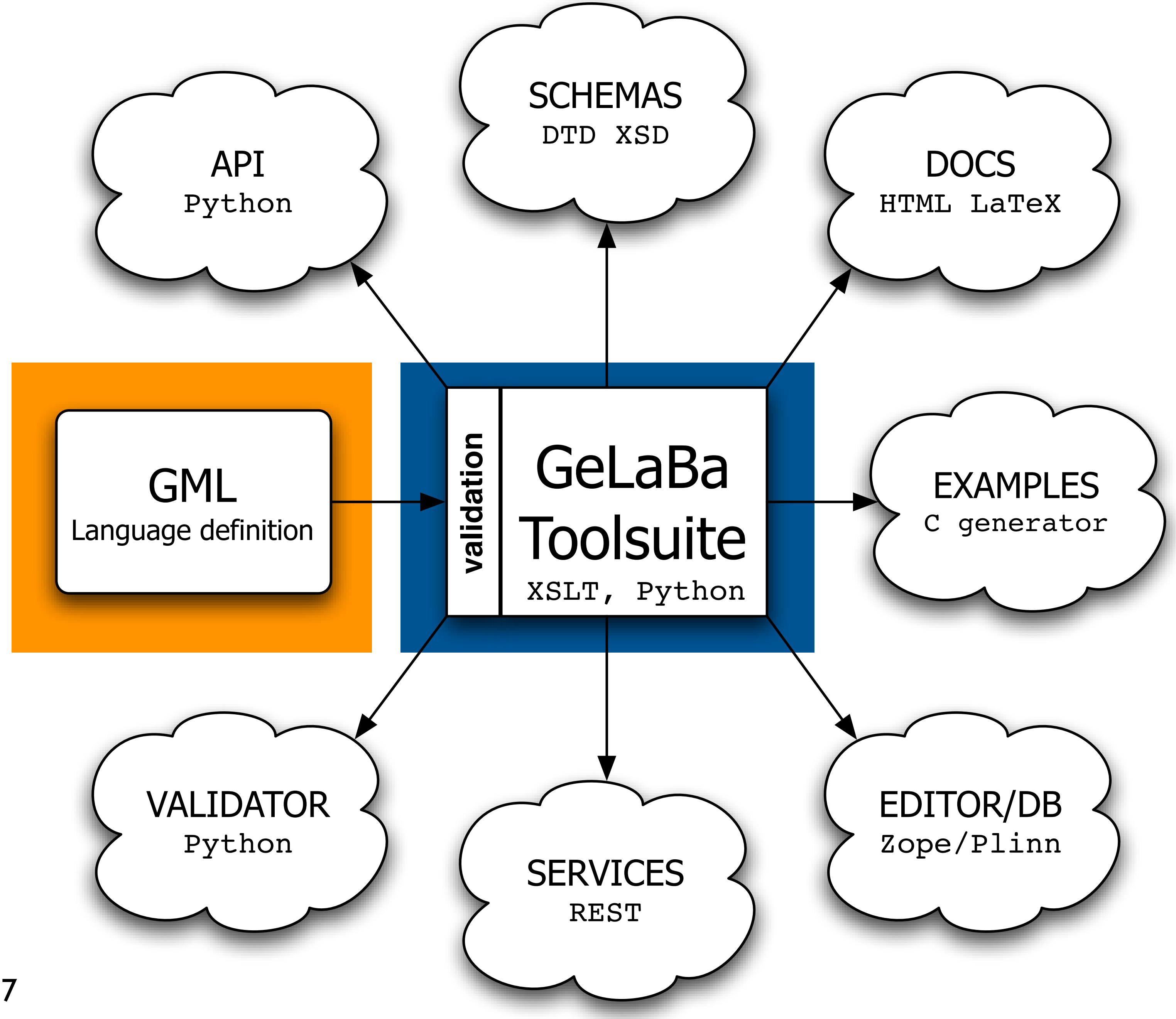
Running example

```
<dict name="dict-departements-france">
  <fullname xml:lang="fr">Départements français</fullname>
  <entry>
    <key val="14"/>
    <value xml:lang=”fr” val="Calvados"/>
  </entry>
  <entry>
    <key val="6"/>
    <value xml:lang=”fr” val="Alpes Maritimes"/>
  </entry>
  <entry>
    <key val="68"/>
    <value xml:lang=”fr” val="Haut-Rhin"/>
  </entry>
  ...
</dict>
```

XML example

```
<?xml version="1.0" encoding="utf-8"?>
<adresse>
  <lignead>17 rue Pastorelli</lignead>
  <codepostal>06000</codepostal>
  <ville>NICE</ville>
  <departement>06</departement>
  <region>93</region>
  <pays>FR</pays>
</adresse>
```

GeLaBa Toolsuite



SCHEMAS
DTD XSD

API
Python

DOCS
HTML LaTeX

GML
Language definition

validation

GeLaBa
Toolsuite
XSLT, Python

EXAMPLES
C generator

VALIDATOR
Python

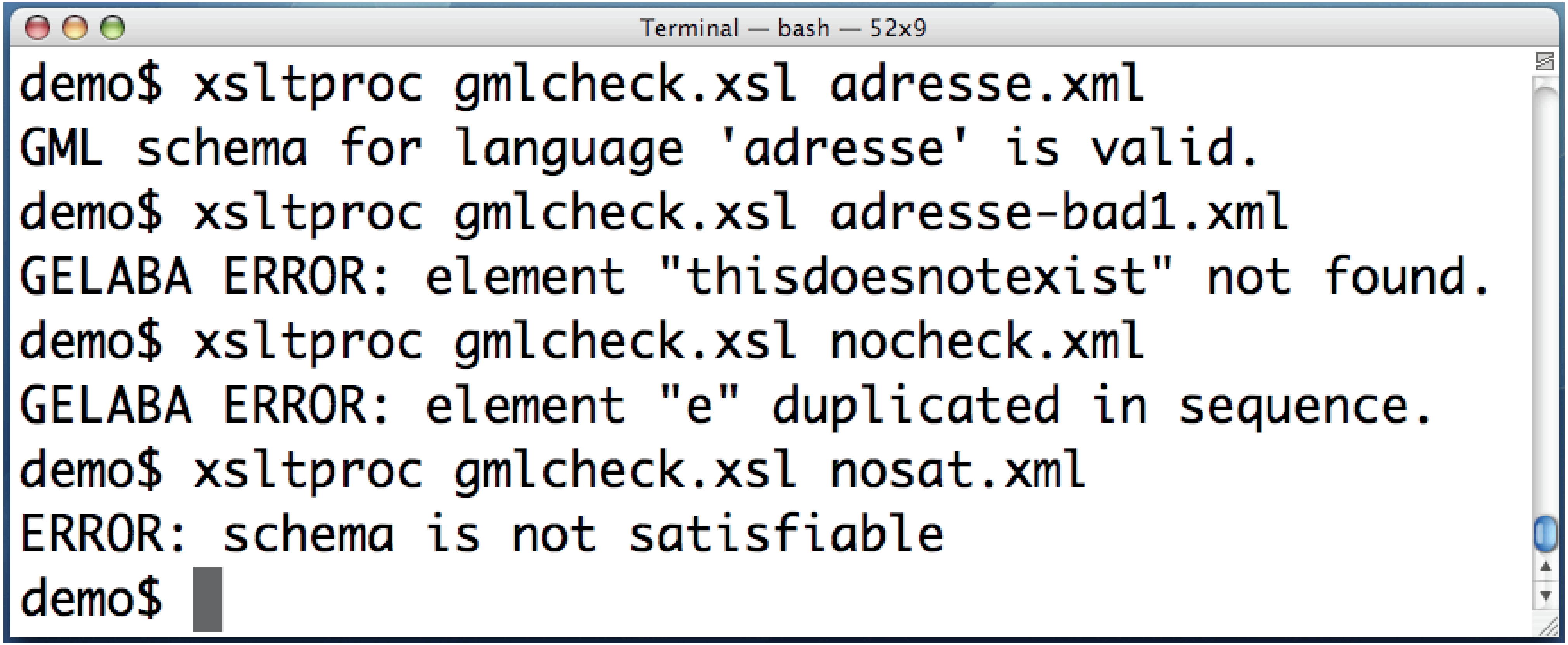
SERVICES
REST

EDITOR/DB
Zope/Plinn

Validation

- Verify GML constraints
  - Duplicated dictionary keys
  - Duplicated documentations
  - Duplicated sequence elements ...
- Satisfiability of the schema (polynomial comp.)
- All checks are done by a XSLT program!

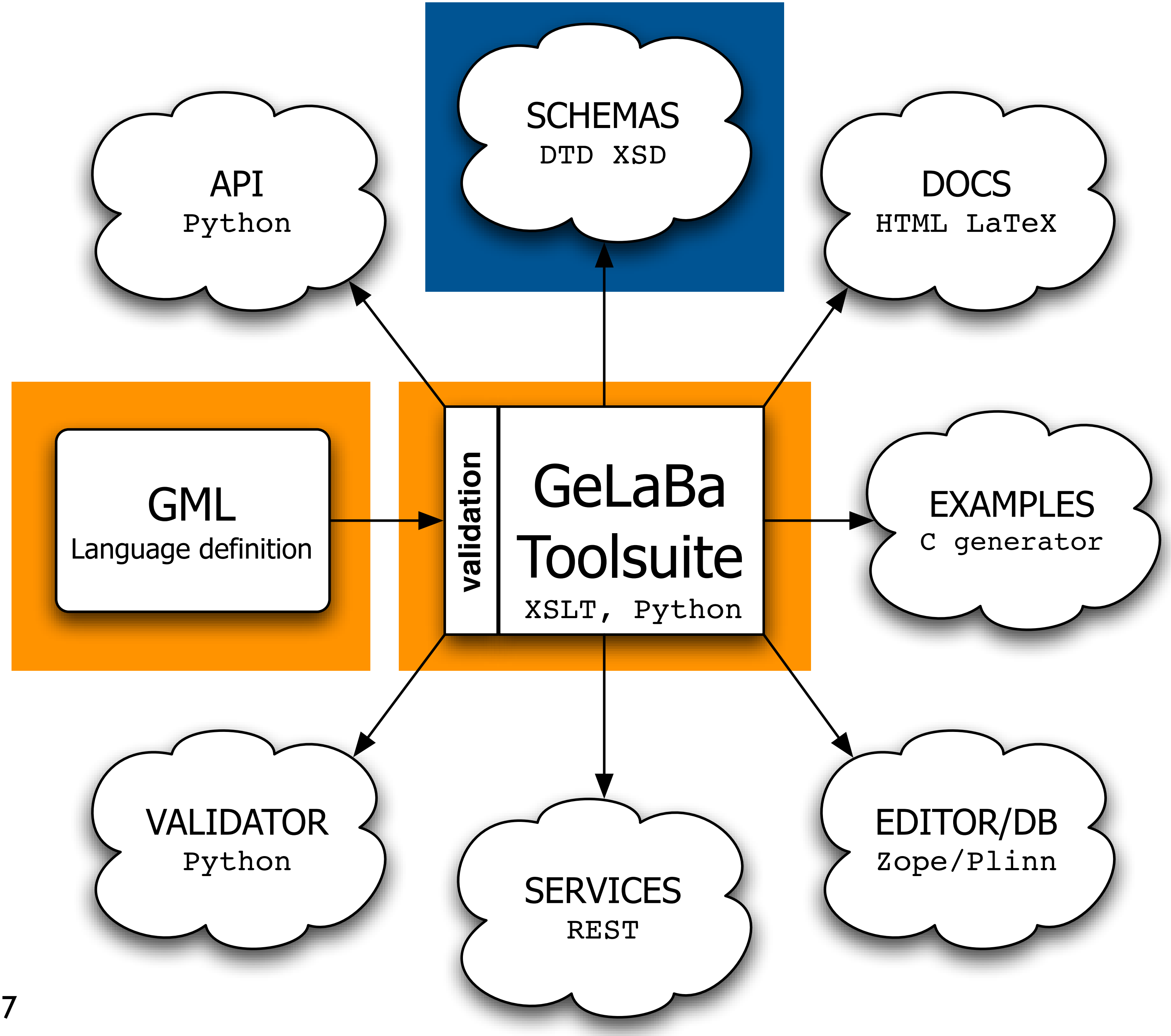
Validation demo



```
demo$ xsltproc gmlcheck.xsl adresse.xml
GML schema for language 'adresse' is valid.
demo$ xsltproc gmlcheck.xsl adresse-bad1.xml
GELABA ERROR: element "thisdoesnotexist" not found.
demo$ xsltproc gmlcheck.xsl nocheck.xml
GELABA ERROR: element "e" duplicated in sequence.
demo$ xsltproc gmlcheck.xsl nosat.xml
ERROR: schema is not satisfiable
demo$ 
```

nosat.xml

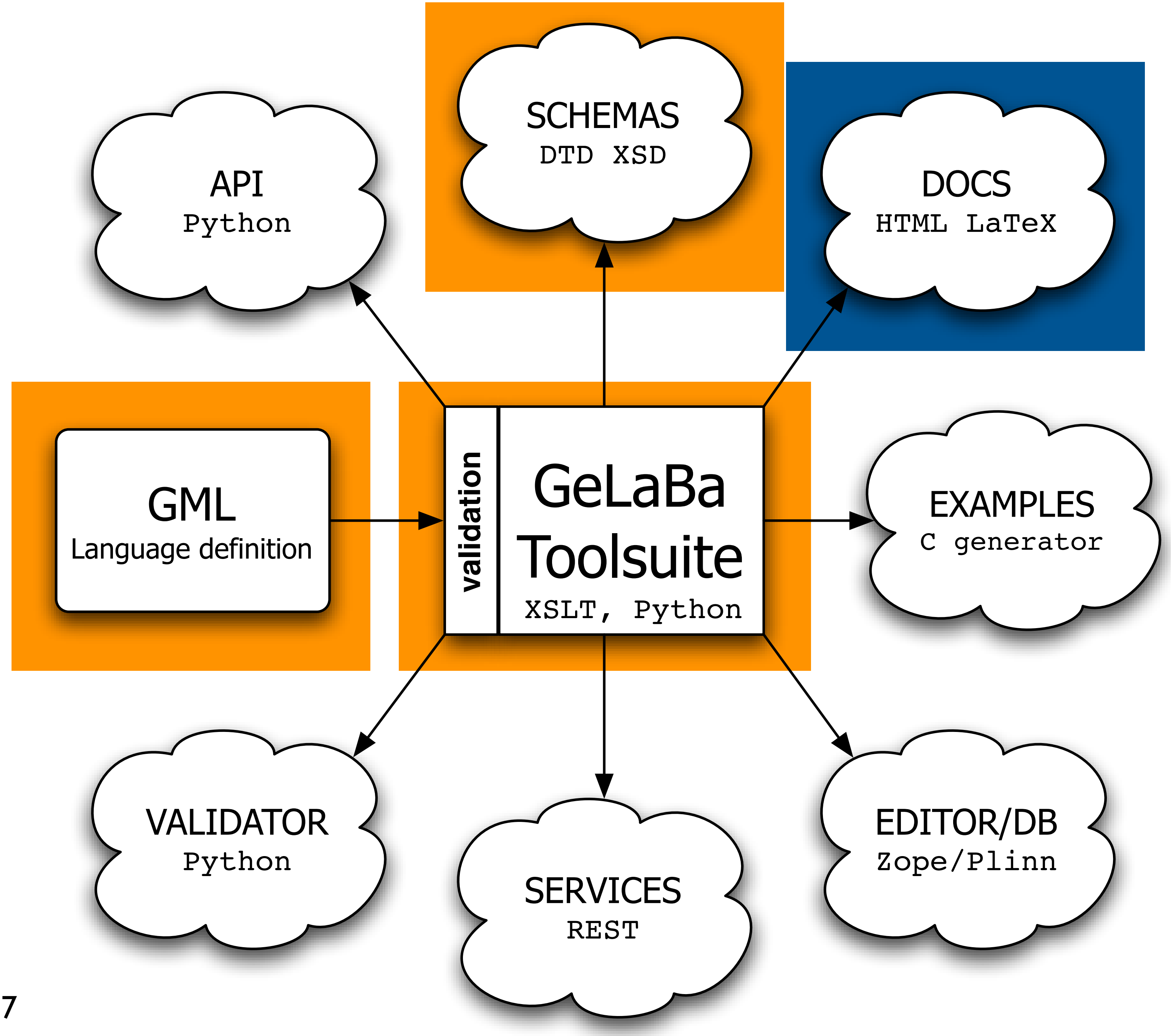
GeLaBa Toolsuite



Schemas

- XSLT programs to generate usual schemas
- DTD (Exemple 1, Exemple 2)
  - structural constraints
- W3C Schema (Exemple)
  - structural constraints + datatypes + sizes

GeLaBa Toolsuite



GML
Language definition

validation

GeLaBa
Toolsuite
`XSLT, Python`

API
`Python`

SCHEMAS
`DTD XSD`

DOCS
`HTML LaTeX`

EXAMPLES
`C generator`

VALIDATOR
`Python`

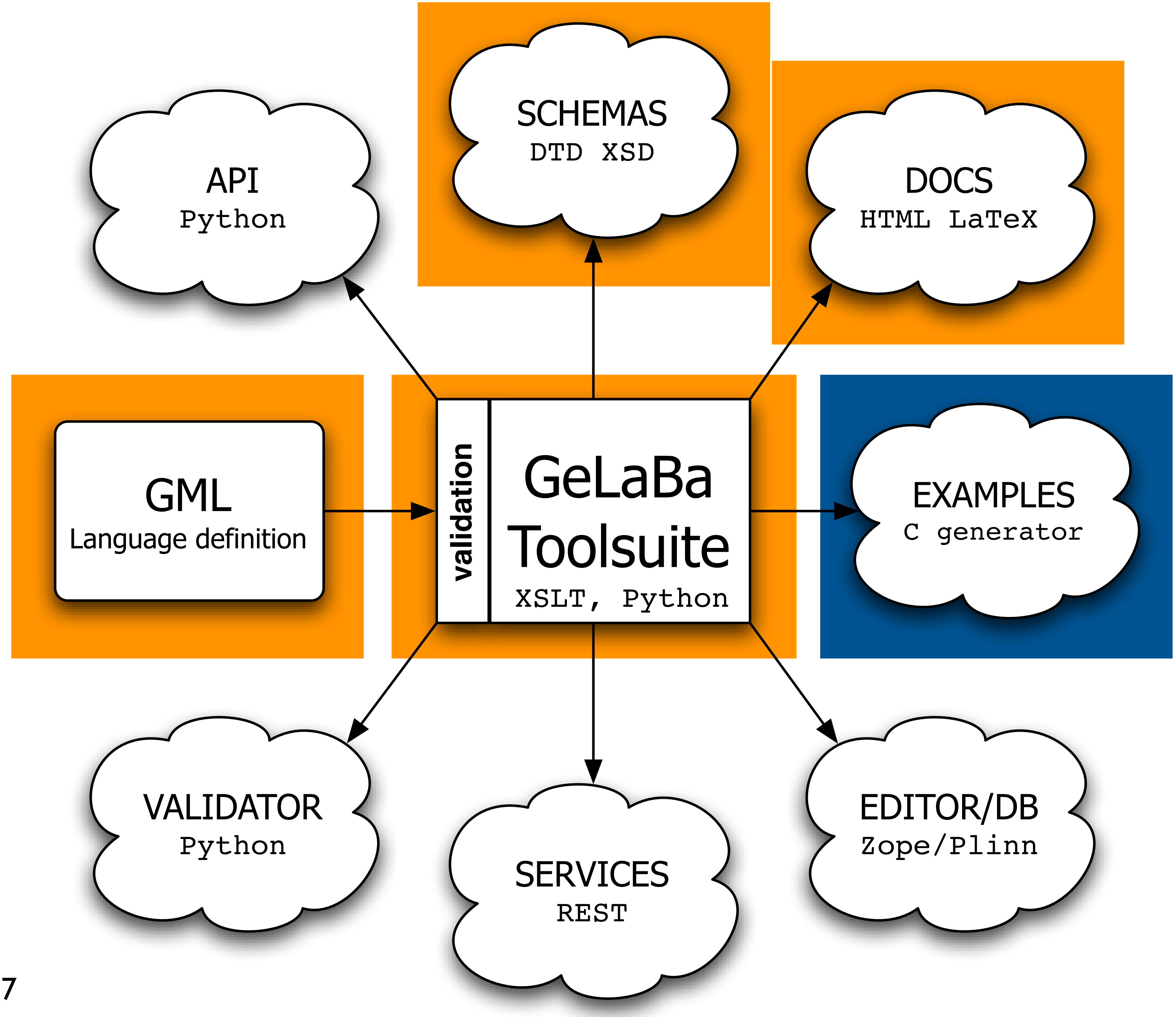
SERVICES
`REST`

EDITOR/DB
`Zope/Plinn`

Documentation

- Documentation in HTML
  - Simple "developer oriented" documentation
  - http://www.lheo.org/lheo.html
- Documentation in LaTeX
  - http://www.lheo.org/1.2/lheo.pdf

GeLaBa Toolsuite

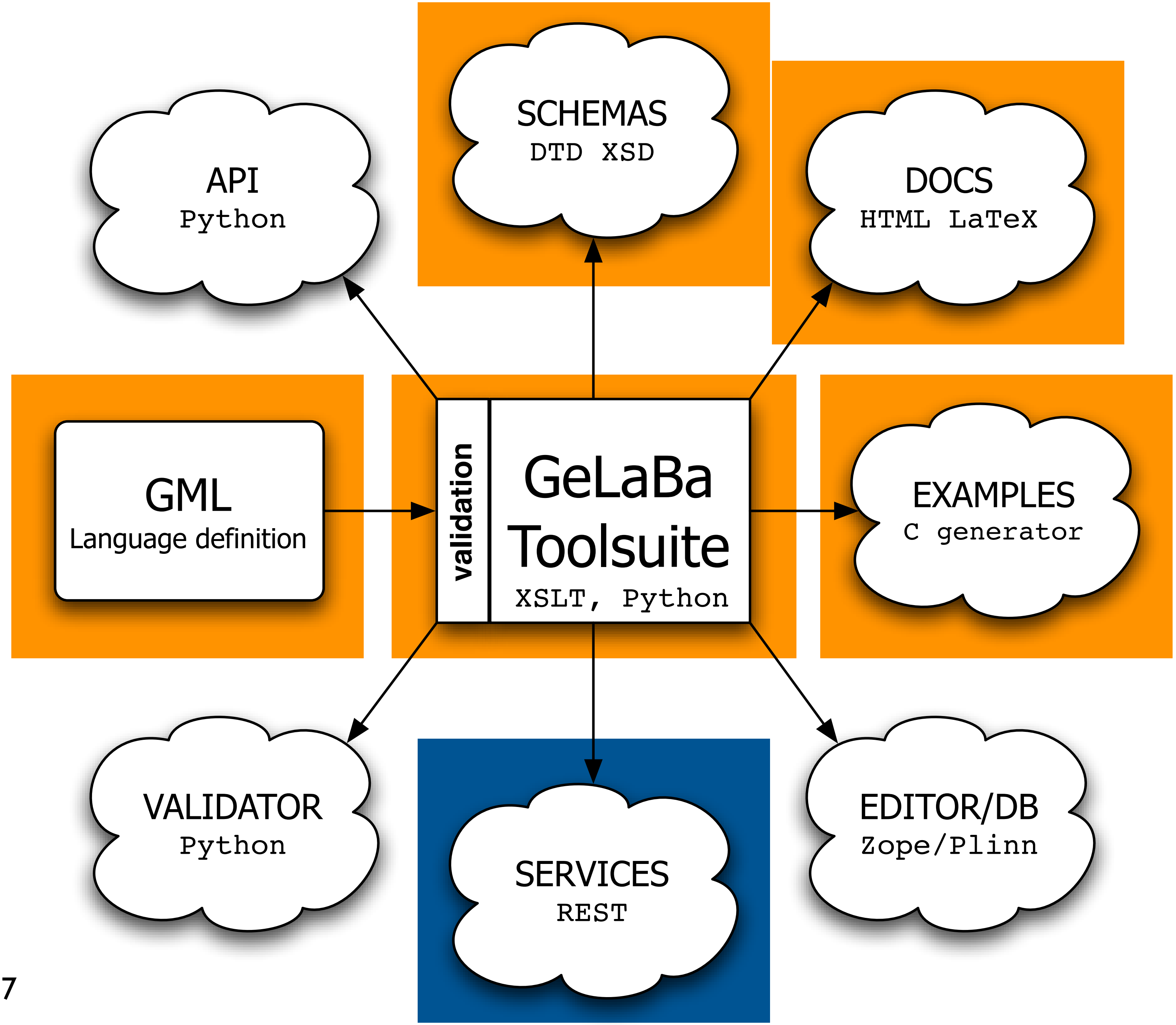


Examples

- Companion examples file
- Single Example Generator in XSLT
  - example1, example2
- Generated Random Example Generator in C
  - example

MINES PARIS
CRI

GeLaBa Toolsuite



SCHEMAS
DTD XSD

API
Python

DOCS
HTML LaTeX

GML
Language definition

validation

GeLaBa
Toolsuite
XSLT, Python

EXAMPLES
C generator

VALIDATOR
Python

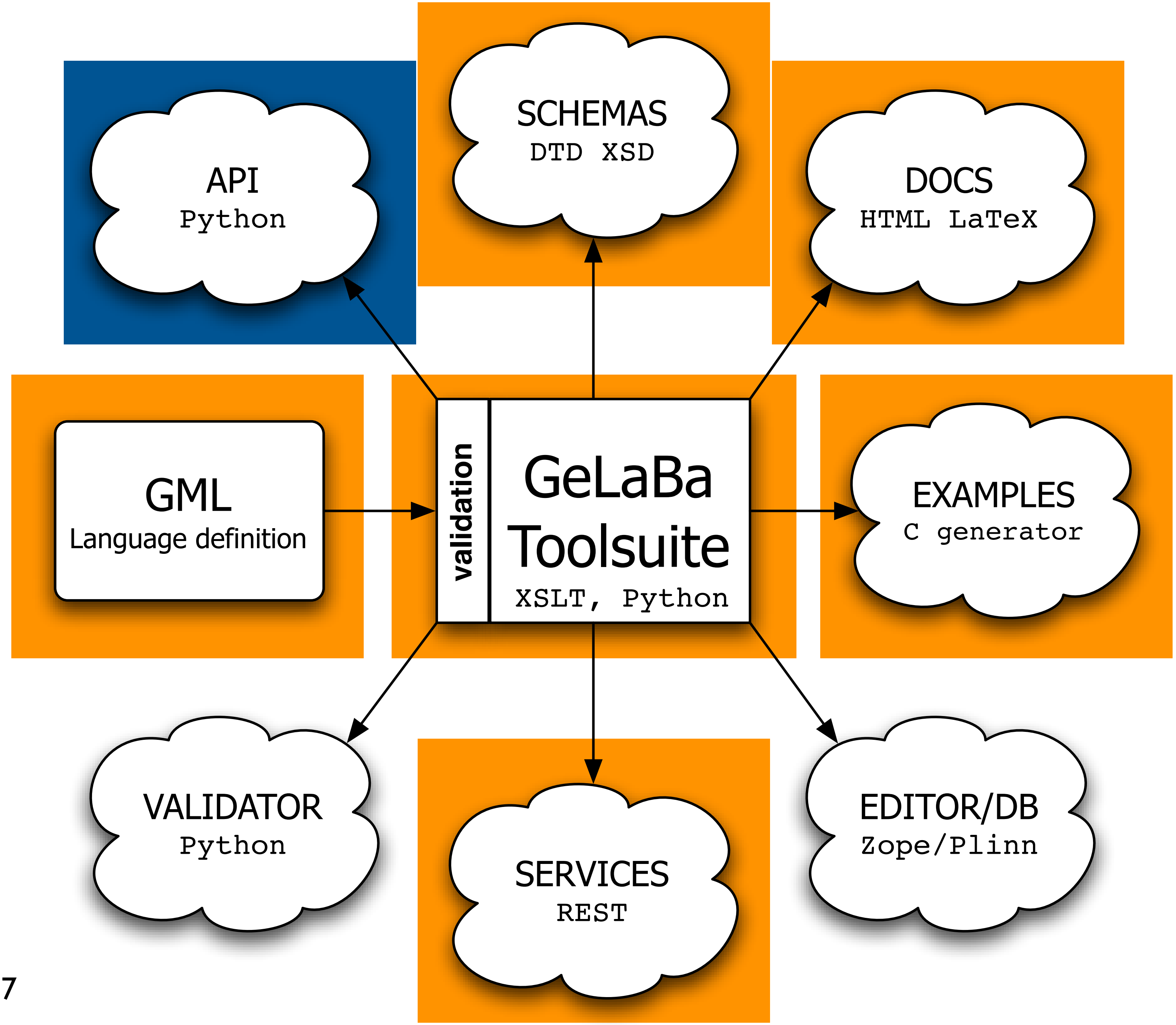
SERVICES
REST

EDITOR/DB
Zope/Plinn

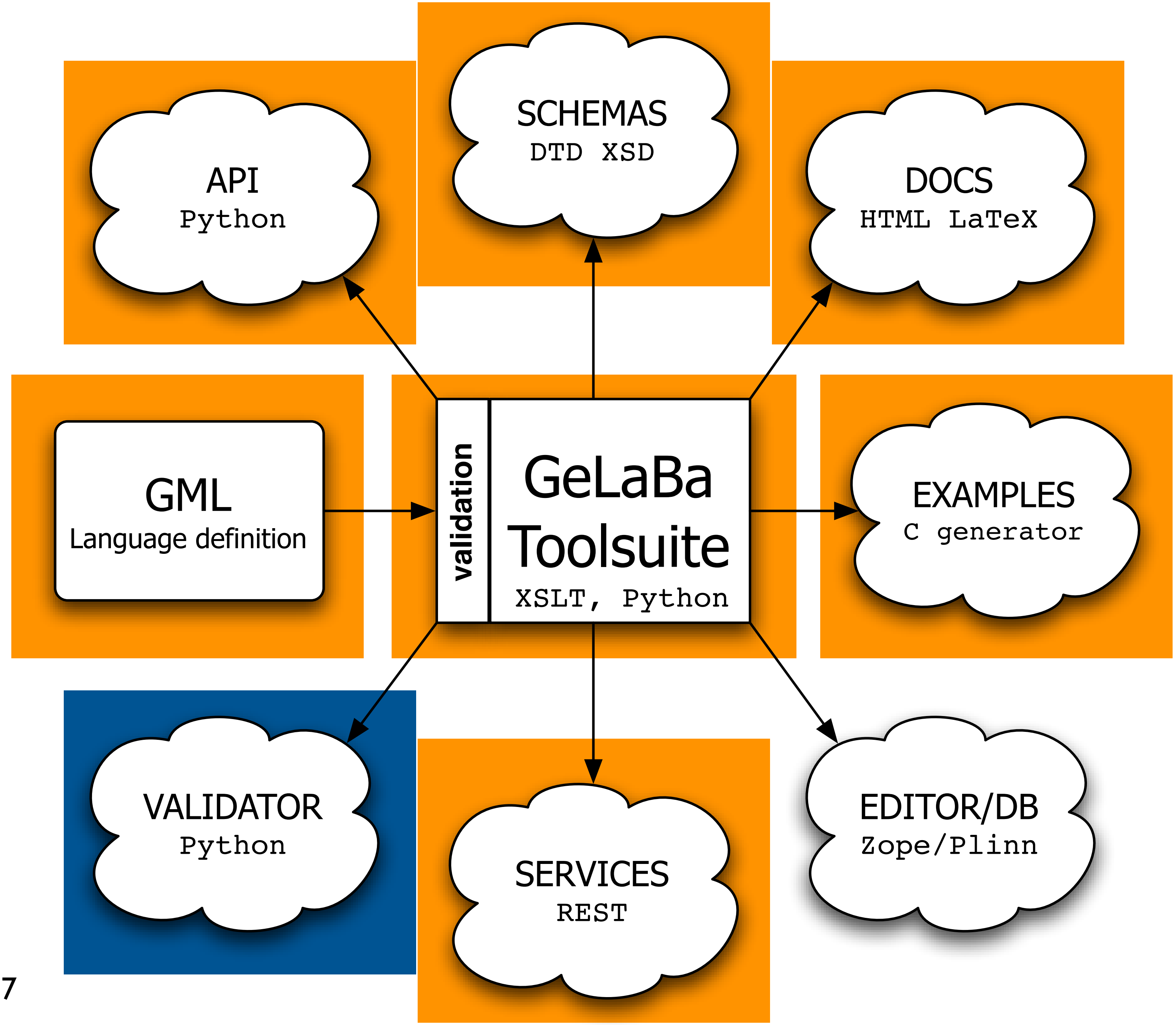
Services

- Services to retrieve XML parts of dictionaries
  - Example for the countries dictionary

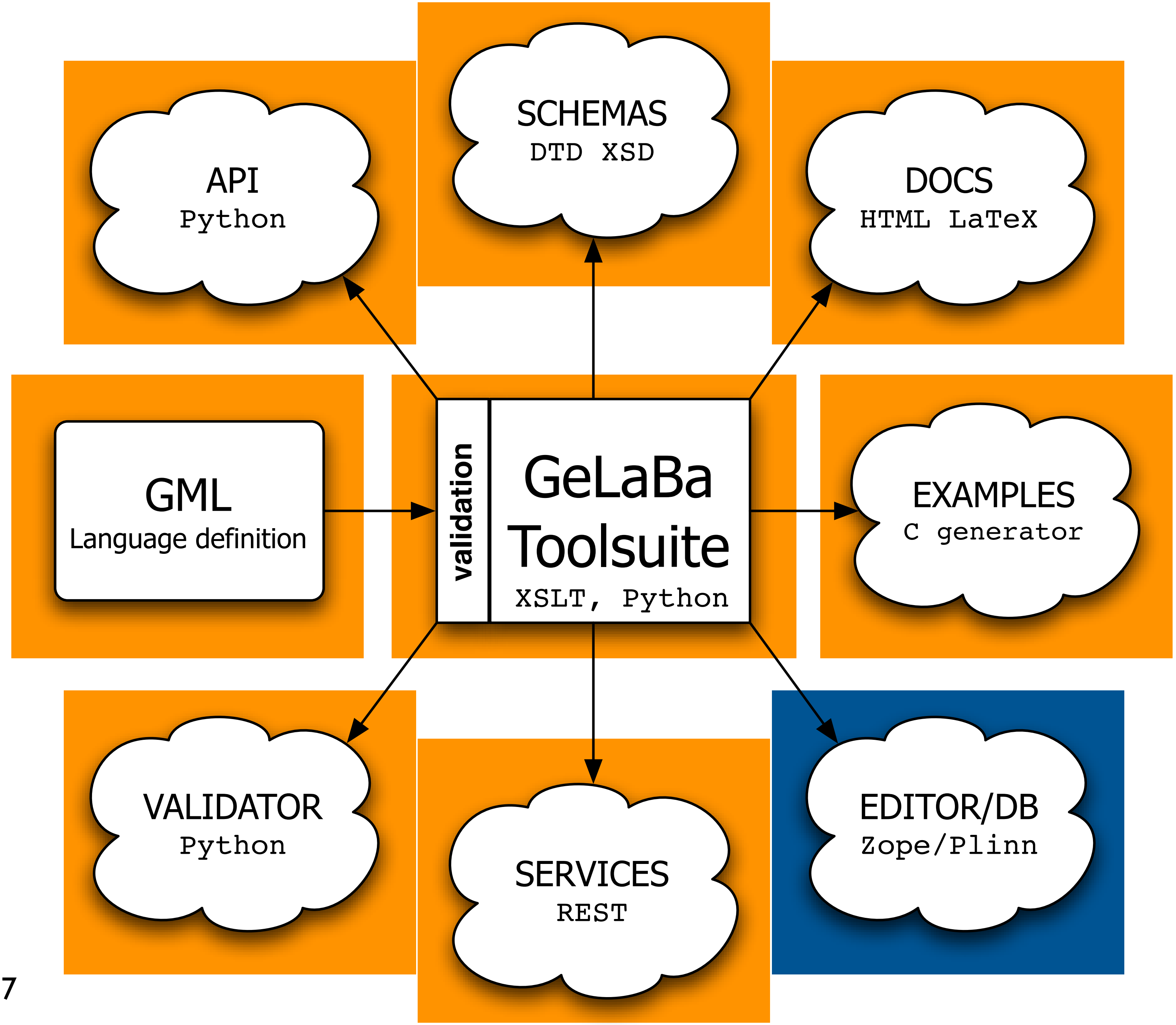
GeLaBa Toolsuite



GeLaBa Toolsuite



GeLaBa Toolsuite



API
Python

SCHEMAS
DTD XSD

DOCS
HTML LaTeX

GML
Language definition

validation

GeLaBa
Toolsuite
XSLT, Python

EXAMPLES
C generator

VALIDATOR
Python

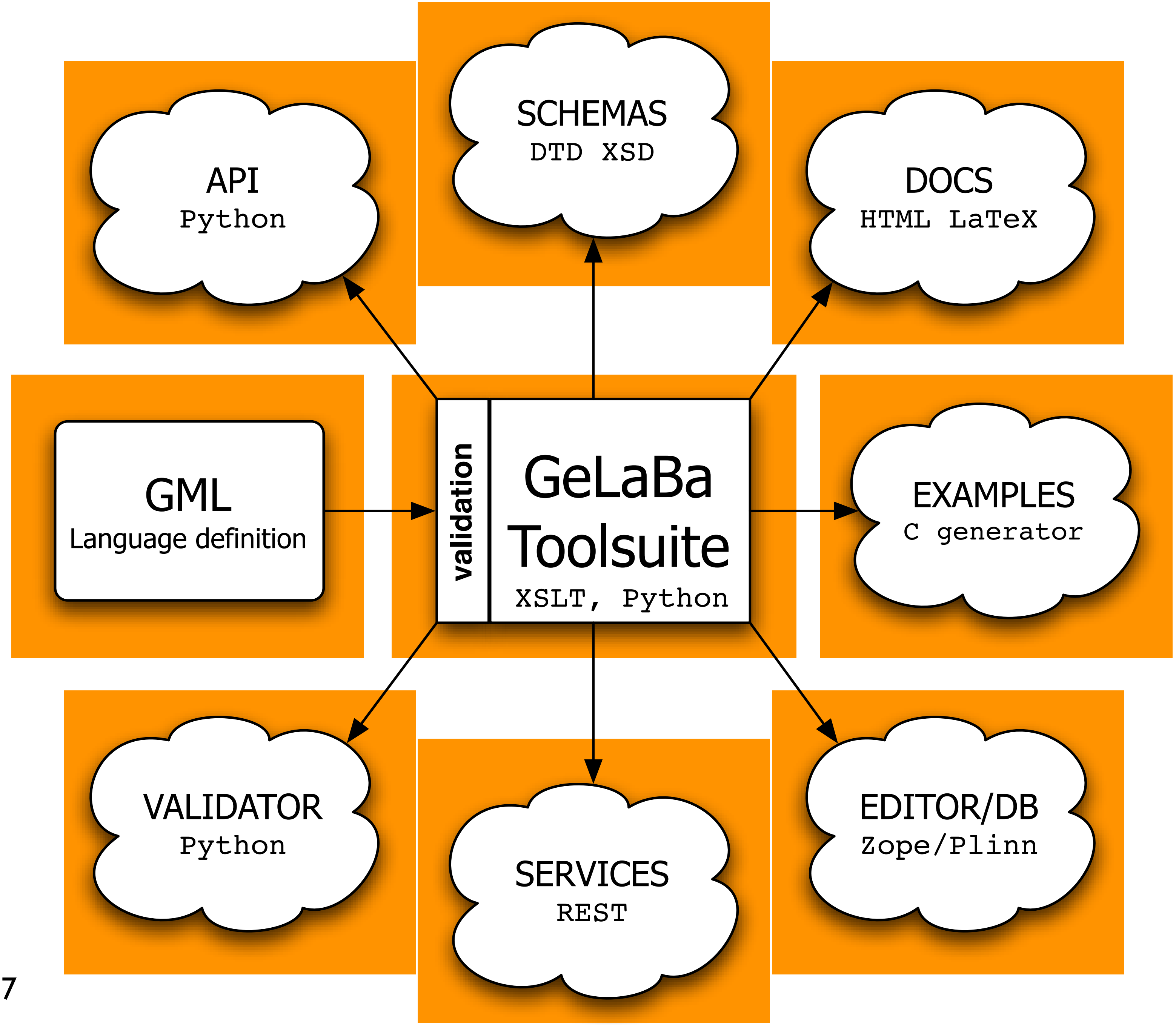
SERVICES
REST

EDITOR/DB
Zope/Plinn

Editor / Database

- A GML language definition can be placed into a Plinn/Zope application server
- Dynamic classes following the definition are created on the server, and can be instanciated
- Features
  - Guided content edition
  - Content view
  - XML import/export

GeLaBa Toolsuite



Final remarks

- Written almost completely in XSLT (+ Python)
- GML is reflexive
  - GeLaBa Toolsuite can be used to create new langages in GML (gml.xml)
- Web site
  - http://www.gelaba.org
- Public Subversion repository
  - http://svn.cri.ensmp.fr/svn/gelaba