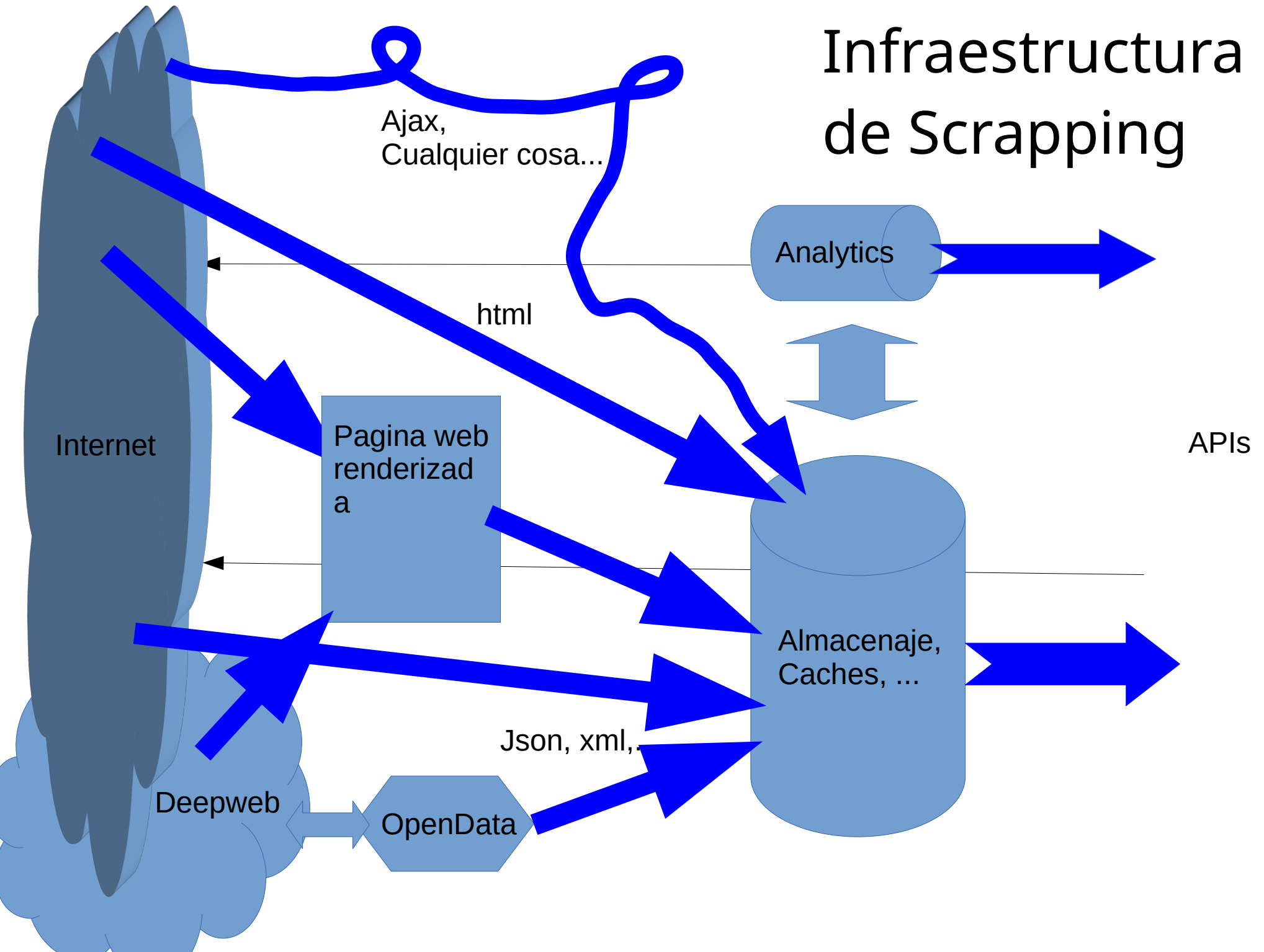
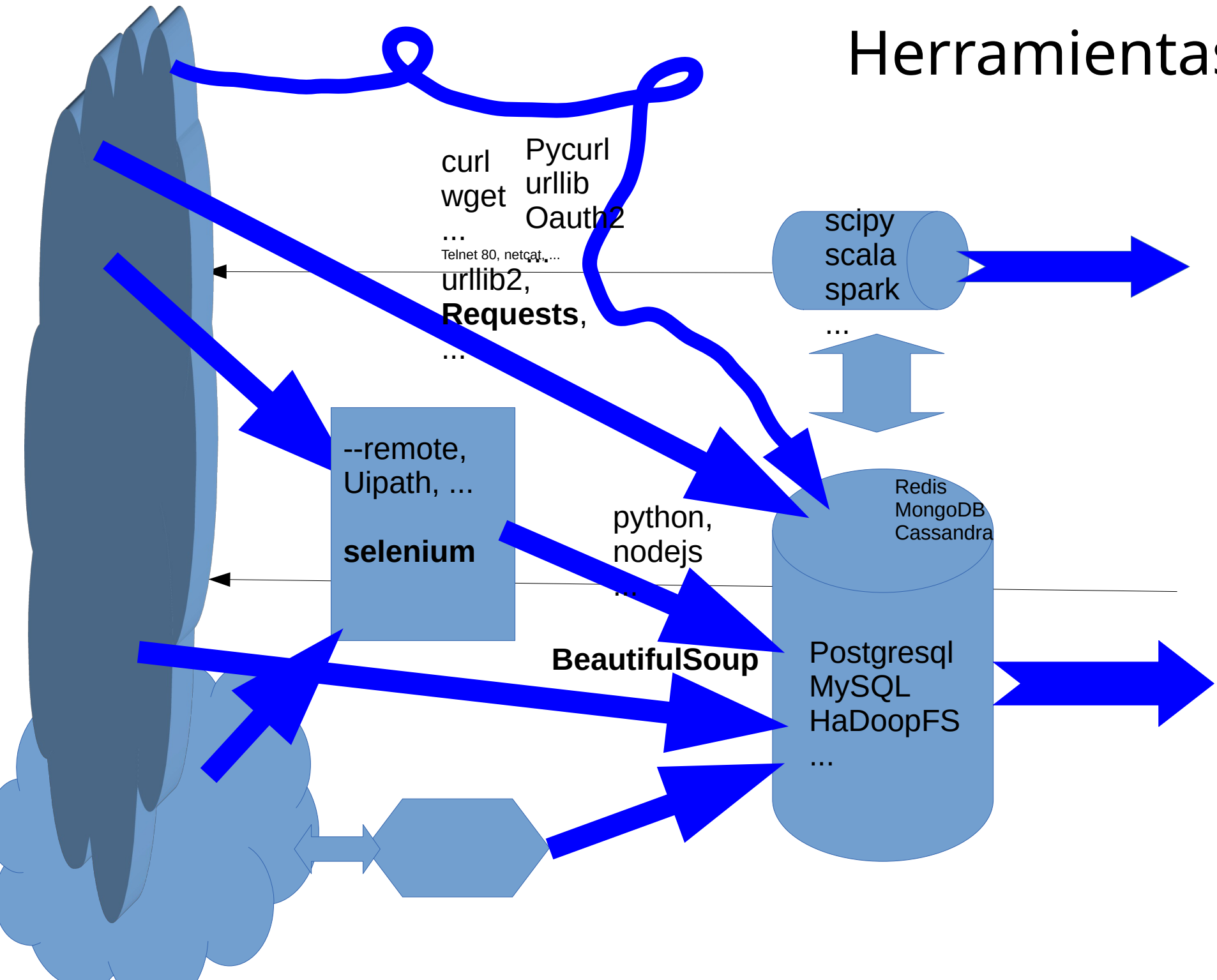


Infraestructura de Scrapping



Herramientas



Caso Ideal: WGET \$URL > /tmp/mypage.html

Nunca es el caso ideal

```
self.conn = pycurl.Curl()
# Restart connection if less than 1 byte/s is received during "timeout" seconds
if isinstance(self.timeout, int):
    self.conn.setopt(pycurl.LOW_SPEED_LIMIT, 1)
    self.conn.setopt(pycurl.LOW_SPEED_TIME, self.timeout)
self.conn.setopt(pycurl.URL, API_ENDPOINT_URL)
self.conn.setopt(pycurl.USERAGENT, USER_AGENT)
# Using gzip is optional but saves us bandwidth.
self.conn.setopt(pycurl.ENCODING, 'deflate, gzip')
self.conn.setopt(pycurl.POST, 1)
self.conn.setopt(pycurl.POSTFIELDS, urllib.urlencode(POST_PARAMS))
self.conn.setopt(pycurl.HTTPHEADER, ['Host: stream.twitter.com',
                                     'Authorization: %s' % self.get_oauth_header()])
# self.handle_tweet is the method that are called when new tweets arrive
self.conn.setopt(pycurl.WRITEFUNCTION, self.handle_tweet)
```

- Necesitamos PUTs, no Solo GETS
- A veces queremos scrapppear un Stream, con reconexiones
- Hay que enviar cabeceras, cookies de sesion...
- ¡En la DeepWeb hace falta user y password!

Requests: “HTTP for Humans”

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
u'{"type":"User"... '
>>> r.json()
{'private_gists': 419, 'total_private_repos': 77, ...}
```

<http://docs.python-requests.org/en/latest/>

¡ajax, sesiones, navegación!

- Si con curl o requests no basta, hay que emular un navegador.
- Webdriver, en Selenium
<http://www.seleniumhq.org/projects/webdriver/>

```
WebDriver driver = new FirefoxDriver();  
// And now use this to visit Google  
driver.get("http://www.google.com");  
// Alternatively the same thing can be done like this  
// driver.navigate().to("http://www.google.com");  
// Find the text input element by its name  
WebElement element = driver.findElement(By.name("q"));  
// Enter something to search for  
element.sendKeys("Cheese!");  
// Now submit the form. WebDriver will find the form for us from the element  
element.submit();
```

Nettiquette

(Para que no digas que nunca te lo han dicho).

- Mira el robots.txt de los sitios que vayas a scrappear.
- Controla la velocidad. Si el sitio va lento, baja la presión.
- Indica en el UserAgent una forma de contactarte. Email, web...

Parsing

- Html/xml: Sax, Xpath, ... xsltproc :-(
- Json: .loads(), etc ...
- JS en el server: nodejs
- BeautifulSoup

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="text"/>
  <xsl:template match="/">
    <xsl:for-each select="response/results/entry">
      <xsl:value-of select="field[@id='content']" />
      <xsl:text>&#xA;</xsl:text>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

<http://www.crummy.com/software/BeautifulSoup/>

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc)
print(soup.prettify())
```

```
for link in soup.find_all('a'):
    print(link.get('href'))
```

Almacenando

- PostgreSQL: tiene extensiones json
- Mysql: ...
- Hdfs/hive/etc: si tienes mas de una máquina.
 - (o una con muchos cores)
 - (o podrias tenerlas y quieres usar spark, mapreduce, etc)

Y sirviendo (la API)