

# Salesforce APIs – Composite resources

Iñaki Hernando Urcullu



dreamOlé<sup>®</sup>

Madrid 1st June 2017

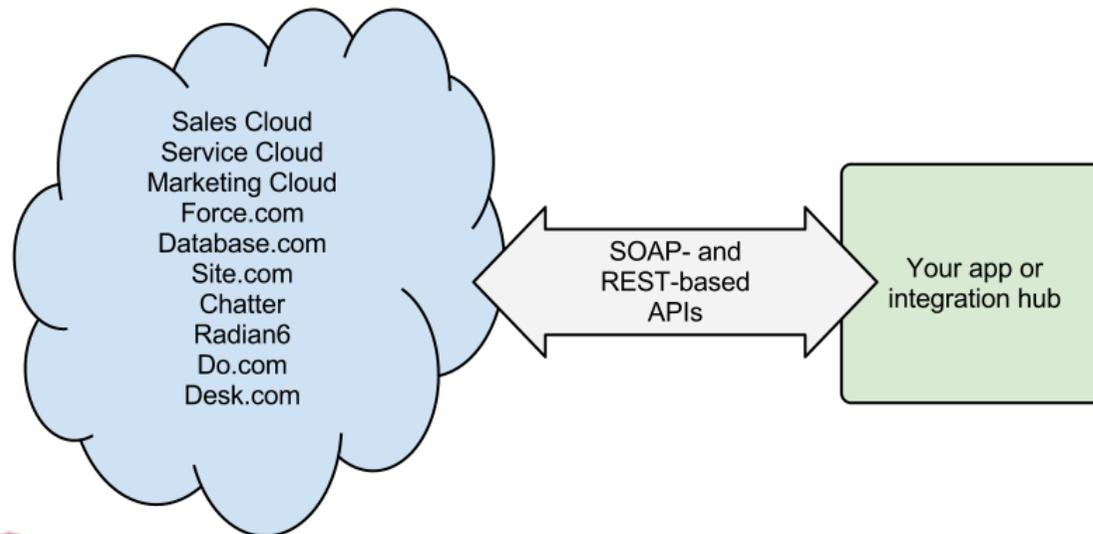
# Cloud Coaches

- 2015** Año de fundación de Cloud Coaches
- Salesforce** Nuestro objetivo como organización es formar a Partners y Clientes en tecnologías Salesforce
- Adopción** Ayudamos a clientes en procesos de transformación digital, implementación de proyectos especiales, con un foco especial en la adopción
- Creación** Colaboramos con clientes y partners en la creación de sus equipos Salesforce, desde la identificación, transformación y captación de profesionales
- 40** Trabajamos como más de 40 clientes a nivel nacional e internacional
- 400** Hemos participado en 400 certificaciones de las conseguidas en el último año en Salesforce

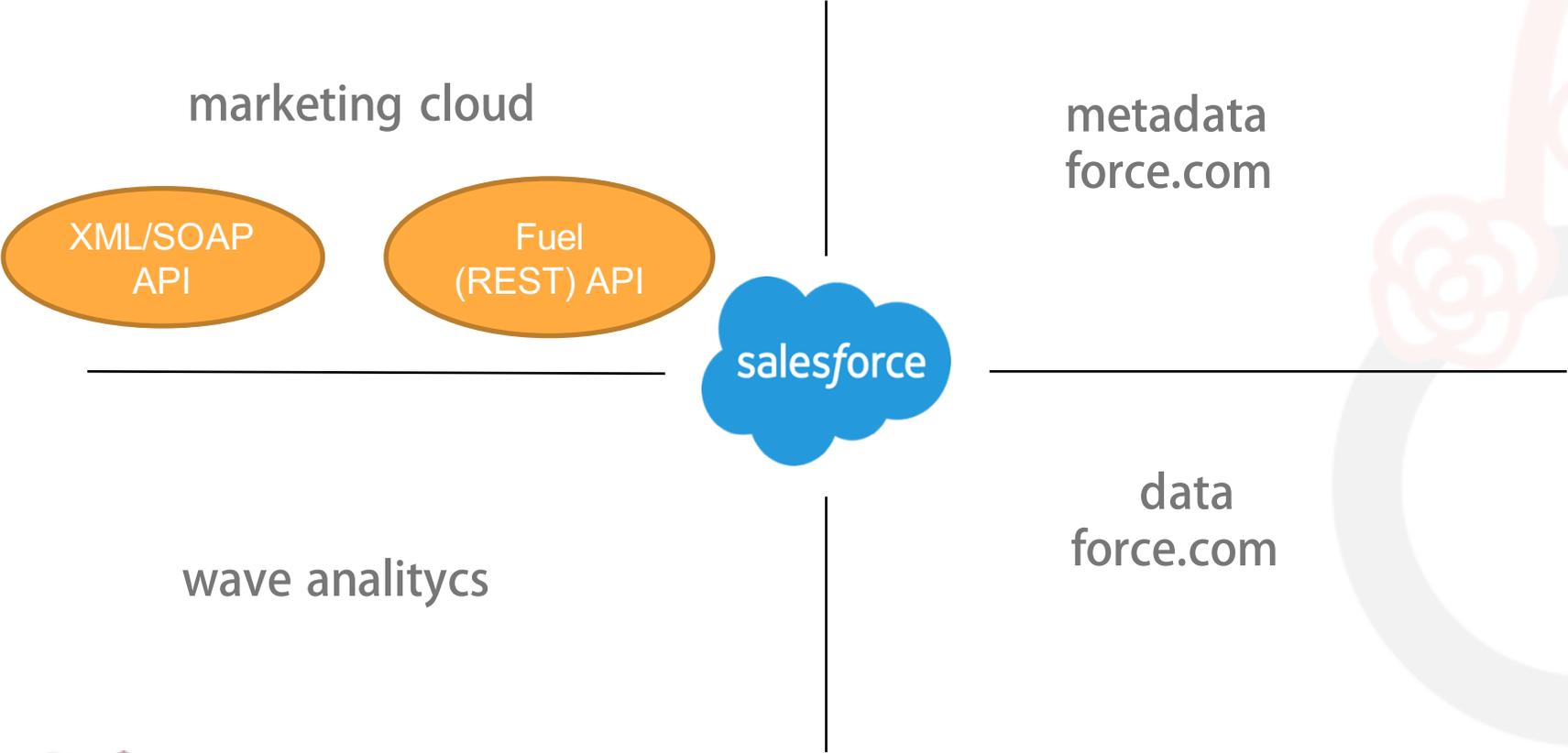
# APIs de Salesforce

Salesforce sabe que sus clientes y socios siempre están pensando en nuevas maneras de extender la funcionalidad de Salesforce. Por lo que ha proporcionado unas herramientas para realizar esto.

Esto lo permiten las distintas APIs que provee salesforce y número de estas APIs es amplio y aumenta con las nuevas funcionalidades y las nuevas tecnologías que va apareciendo.



# ¿cuál es la API adecuada?



# Marketing Cloud API

## REST API:

- Contacts
- Journey Builder
- Mobile Connect
- MobilePush
- Campaigns
- Triggered Sends (también en SOAP)

## SOAP API

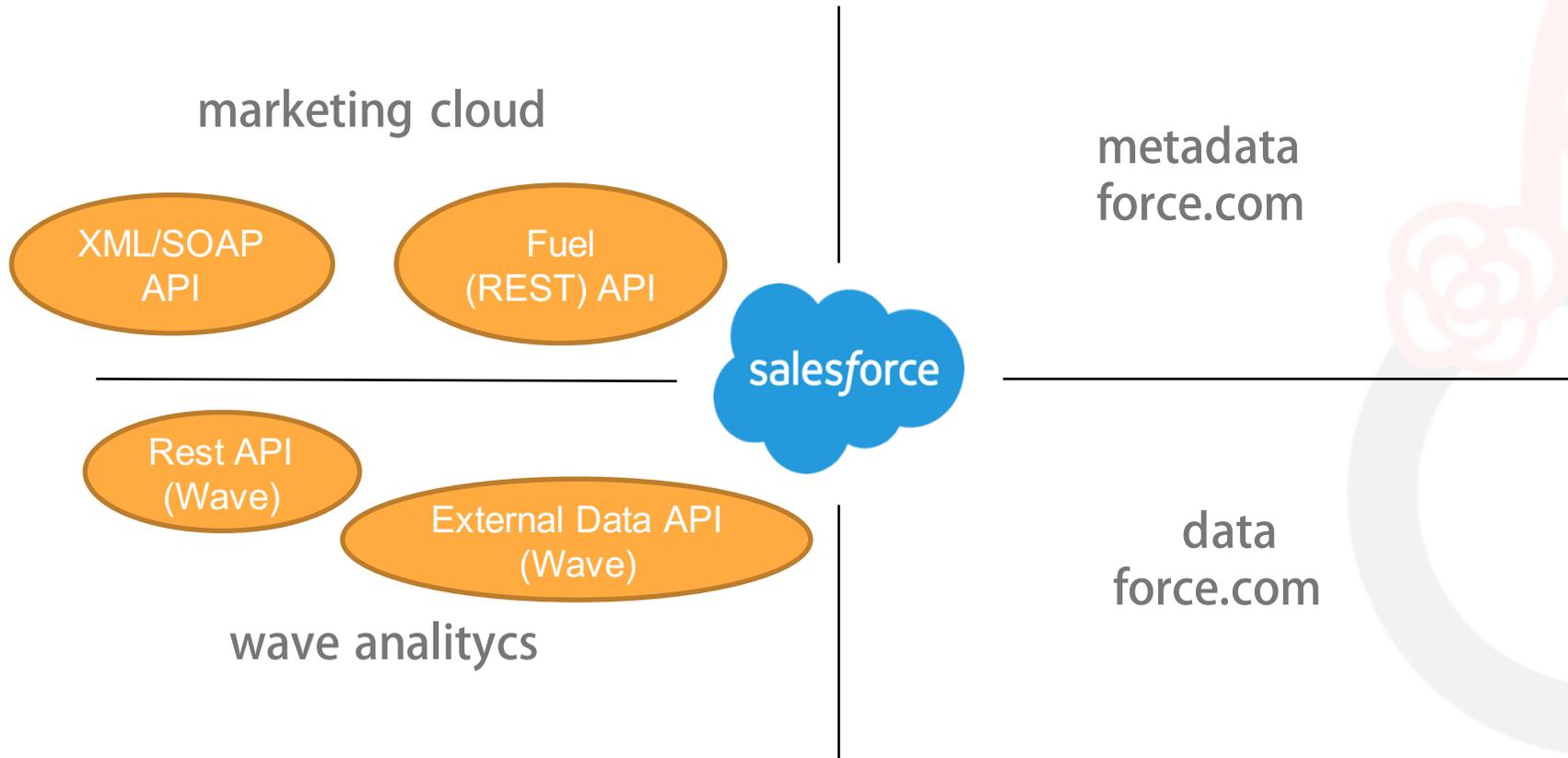
- Tracking
- Subscribers and lists
- Programs
- Triggered sends (también en REST)
- Content
- Most other email activities

## FUEL SDK:

Es una librería para distintos lenguajes que nos permite utilizar estas dos APIs



# Acceder y extraer datos de wave



# Wave Analytics

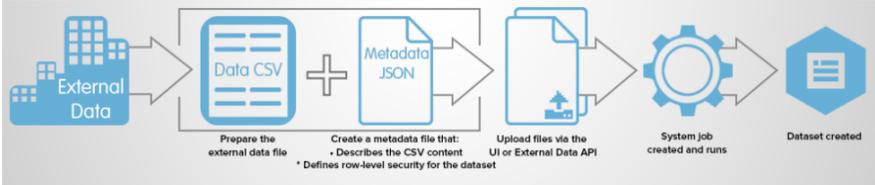
## REST API

Puede acceder a los activos de Wave Analytics.



## External Data API

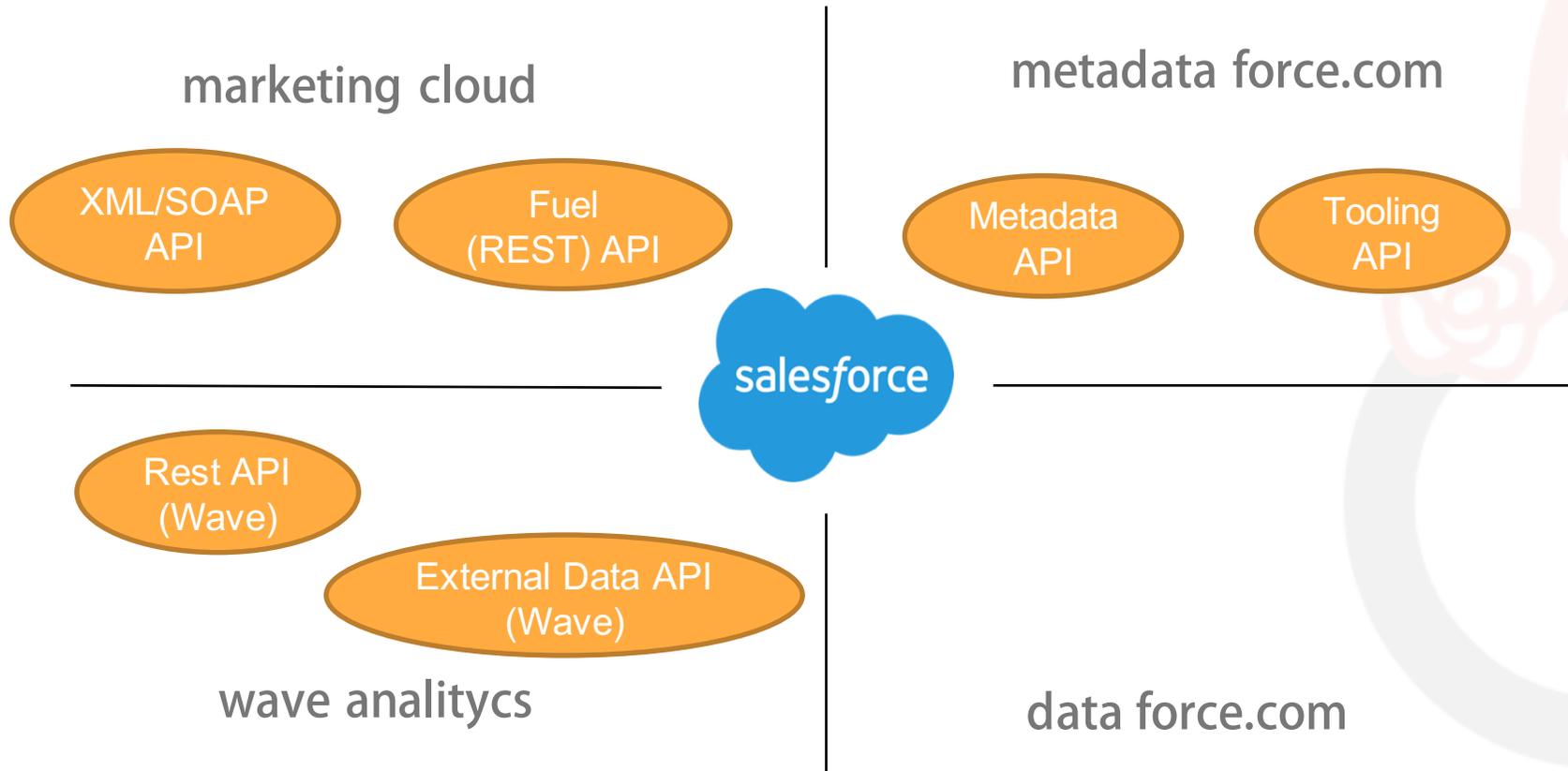
Permite cargar archivos de datos externos a Wave Analytics.



wave analytics



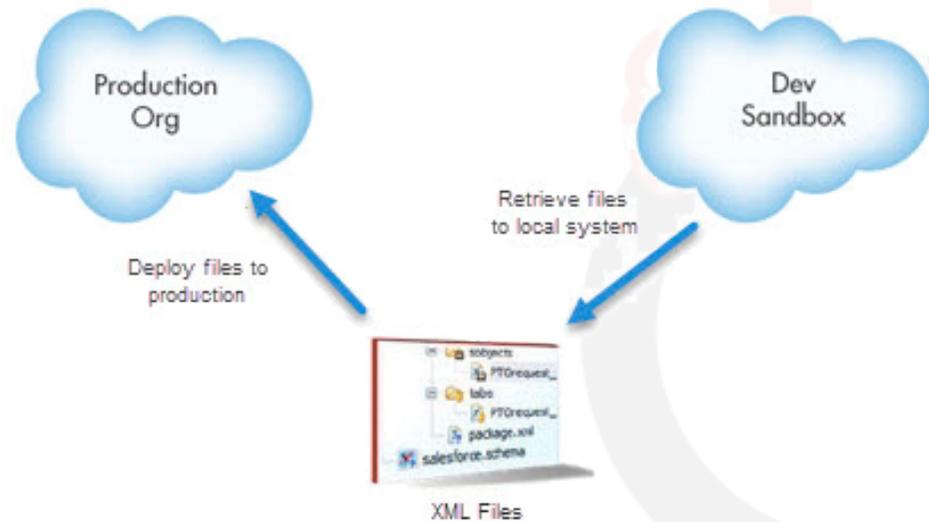
# Modificar configuración de la organización



# Force.com Metadata API

La API de metadatos está diseñada para administrar personalizaciones y para crear herramientas que pueden administrar el modelo de metadatos, no los propios datos.

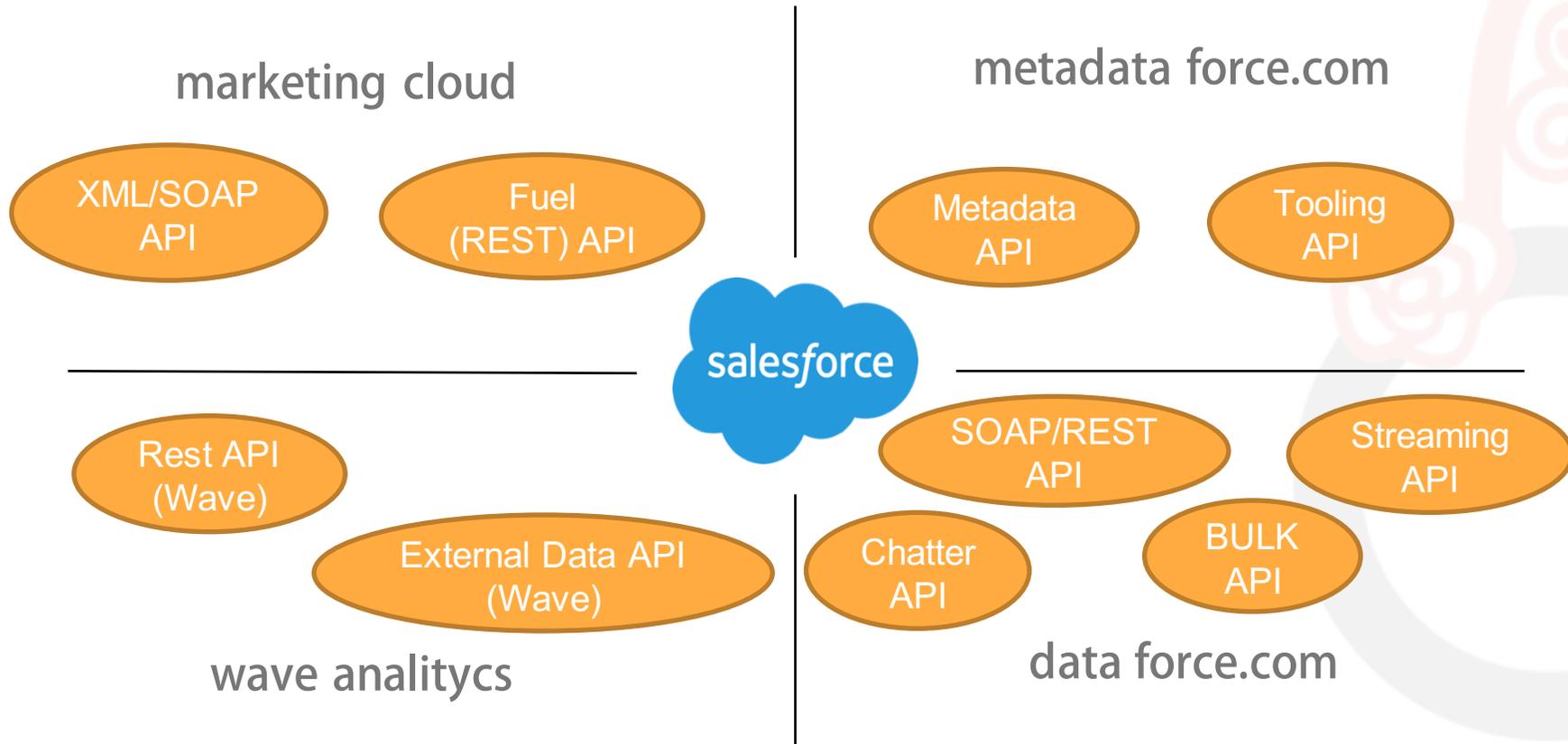
force.com™  
ide



## Force.com Tooling API

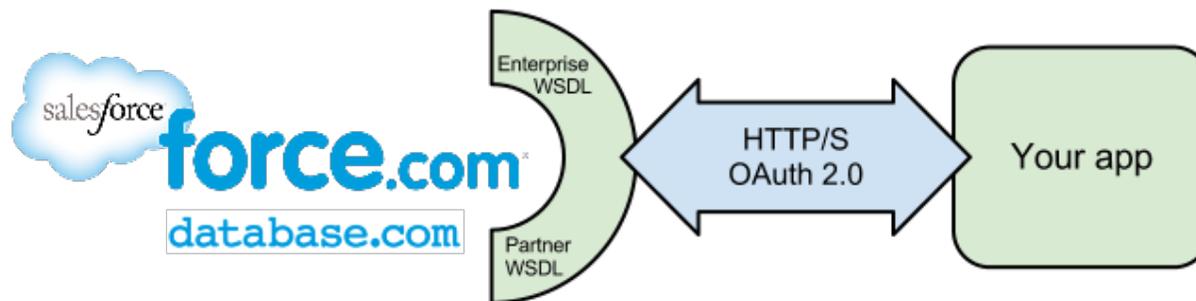
- Apex Logs
- Code Coverage
- Trace flag for debug logs

# Acciones y manejo de datos en force.com



# Force.com SOAP API

Usa SOAP API para crear, extraer, actualizar o borrar registros como cuenta, candidatos o objetos personalizados. SOAP API tiene mas de 20 llamadas diferentes, permitiendo consultar gestionar la contraseña, describir los objetos, buscar y muchas otras mas.



Manage data using the Force.com SOAP API

SOAP API puede ser usado desde cualquier lenguaje que soporte Webservices, para ello es posible descargarse la definición de **Setup | Develop | API**

# Enterprise vs Partner SOAP API

Existen dos WebService Force.com o definiciones de WSDL que se pueden obtener desde la organización.

```
<complexType name="Account">
  <complexContent>
    <extension base="ens:sObject">
      <sequence>
        <element name="Name"
          nillable="true"
          minOccurs="0"
          type="xsd:string"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<complexType name="sObject">
  <sequence>
    <element name="Id" type="tns:ID" nillable="true"/>
    <element name="type" type="xsd:string"/>
  </sequence>
</complexType>
```

## Enterprise Web Services WSDL

- Tipos para cada Objeto
- Especifico para cada organización
- Simplifica el uso de la misma
- La definición cambia si se realizan cambios en el metadata de la organización

## Partner Web Services WSDL

- Definición genérica de los Objetos
- Común para las distintas organizaciones.
- Mucho mas dinámico.
- La definición cambio cuando hay una nueva versión

# API Limits

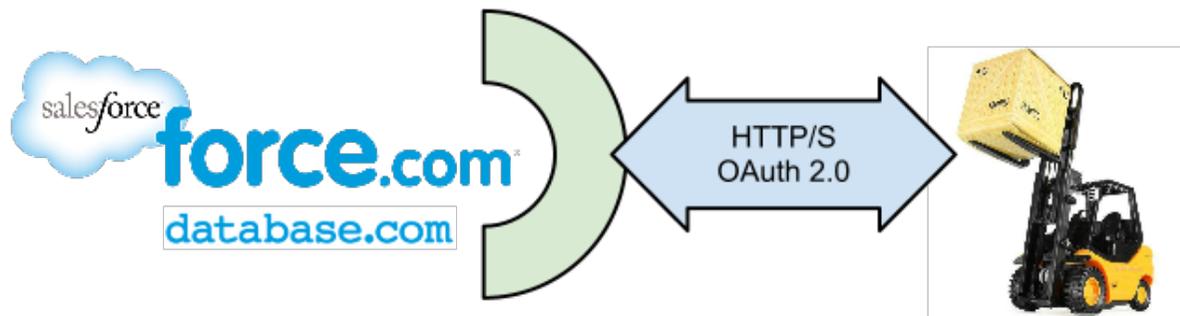
Salesforce Edition	API Calls por tipo de licencia		Mínimo	Máximo
<ul style="list-style-type: none"> <li>Developer</li> </ul>			15.000	15.000
<ul style="list-style-type: none"> <li>Enterprise</li> <li>Professional</li> </ul>	Salesforce: 1,000 Force.com Light App: 200 Force.com Enterprise App: 200 Salesforce Platform: 1,000	Partner Community: 200 Force.com – One App: 200 * Gold Partner: 200 *	15.000	1.000.000
<ul style="list-style-type: none"> <li>Unlimited</li> <li>Performance</li> </ul>	Salesforce: 5,000 Force.com Light App: 200 Force.com Enterprise App: 200 Force.com App Bundle: 1,000	Salesforce Platform: 5,000 Partner Community: 200 Force.com - One App: 200 * Gold Partner: 200 *	15.000	Unlimited
<ul style="list-style-type: none"> <li>SandBox</li> </ul>			N/A	5.000.000



\* Licencia no disponible para nuevos clientes

# Force.com Bulk API

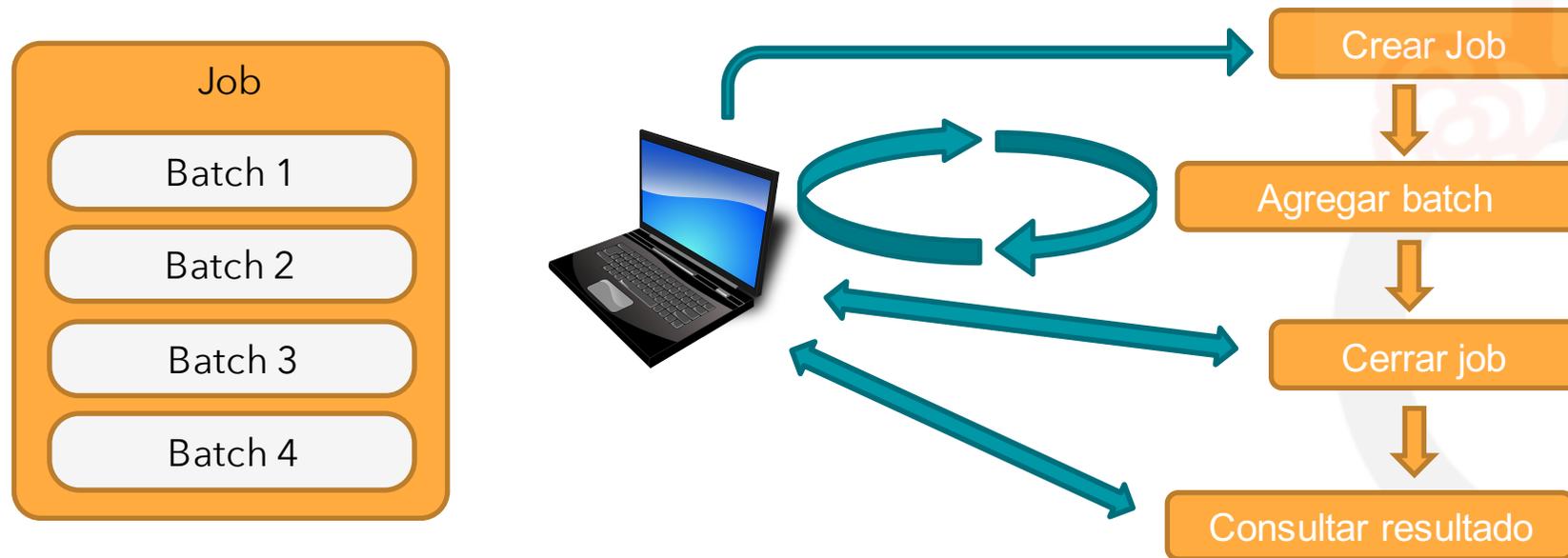
Bulk API es un Webservice basado en los principios REST que esta optimizada para la carga y borrado de grandes cantidades de datos. Puedes utilizar query, queryAll, insert, update, upsert o delete de muchos registros de forma asíncrona. Esto se hace mediante jobs y batches en background.



Load tons of data using the Force.com Bulk API

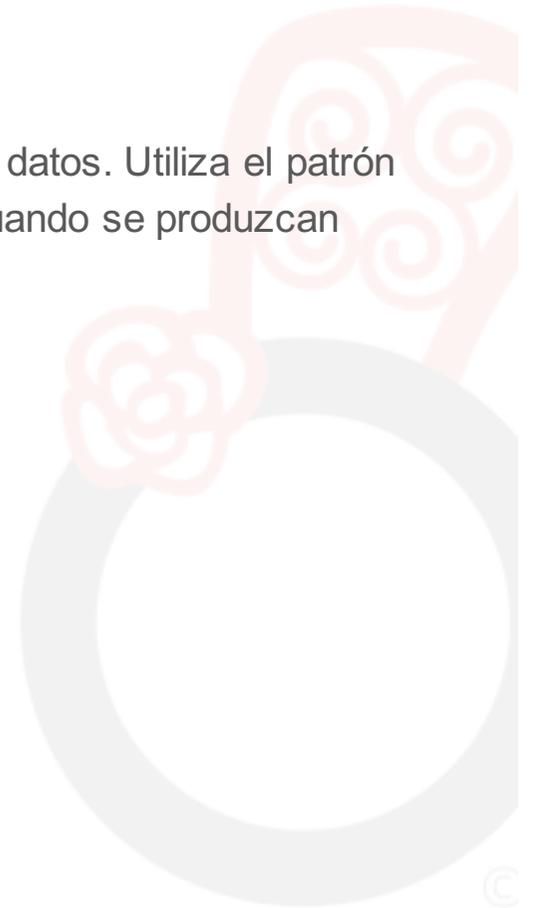
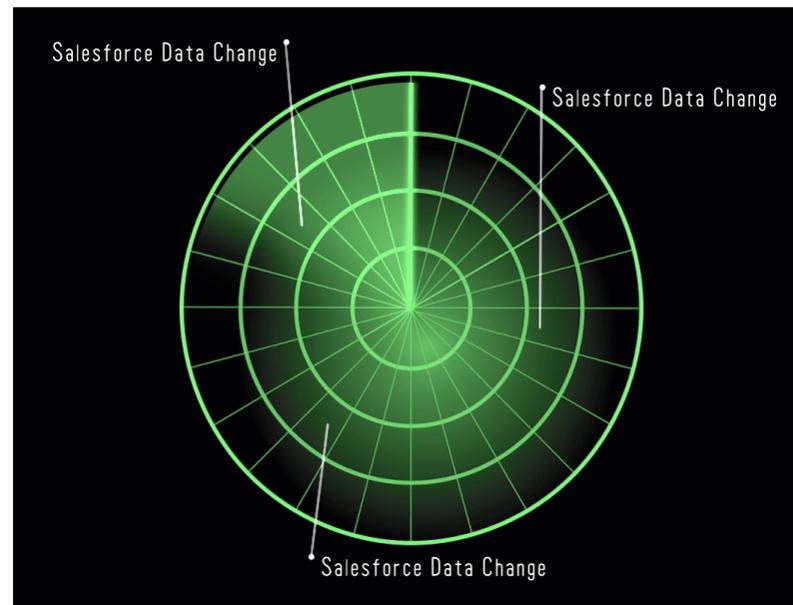
# Force.com Bulk API

Para realizar el uso de Bulk api es necesario un trabajo (job) e ir introduciendo batch con los bloques.



# Force.com Streaming API

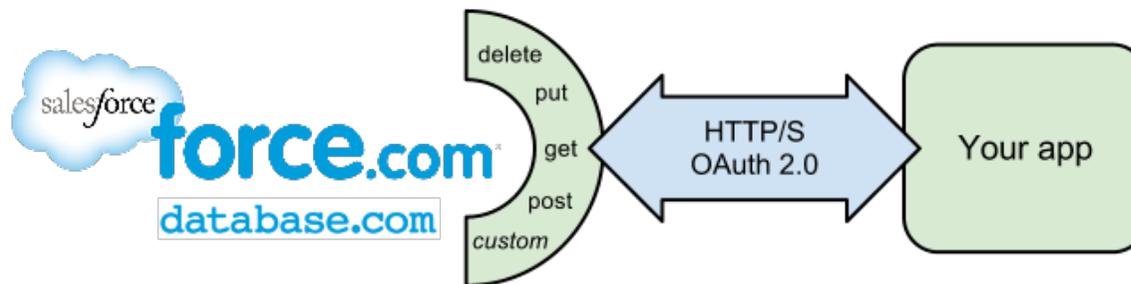
Streaming API permite generar notificaciones cuando se realiza un cambio en los datos. Utiliza el patrón de publish-subscribe, la app se suscribirá a un canal que transmitirá los datos cuando se produzcan cambios en los datos que se han auditado.



# Force.com REST API

Force.com REST API permite interactuar con salesforce a través del Webservice con tecnología Rest.

Cada recursos en REST API tiene un nombre URI que se consume mediante HTTP method: HEAD, GET, POST, PATCH, or DELETE. Este WebService soporta autenticación OAuth 2.0 y soporta tanto JSON como XML.



Manage data using the Force.com REST API

# Force.com REST API

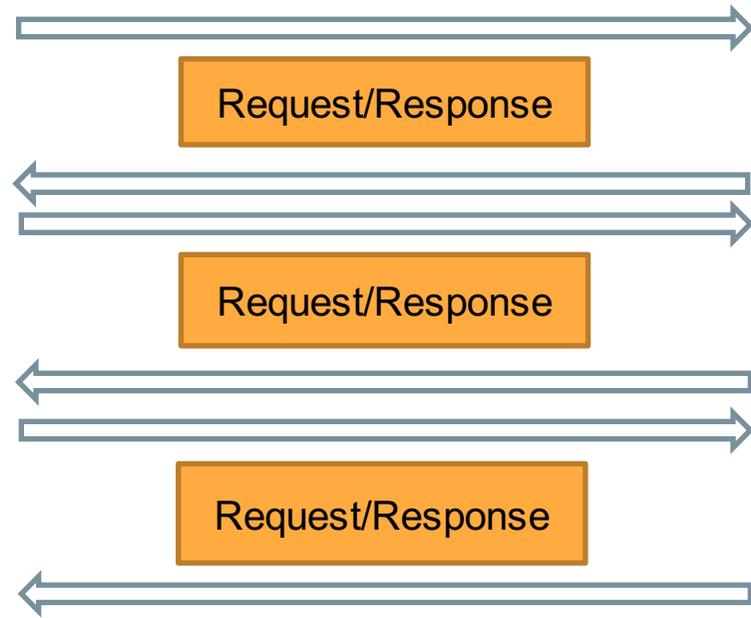
Nos permite multitud de acciones pudiendo verlas accediendo a

<https://instance.salesforce.com/services/data/vNN.N/Action>

Action	URL	Http Method
Consultar 1 registro	/subjects/Account/Id	GET
Crear registro	/subjects/Account	POST
Borrar registro	/subjects/Account/Id	DELETE
Consultar limites	/limits	GET
Query	/query?q=SOQL	GET
Describe Object	/subjects/Account	GET

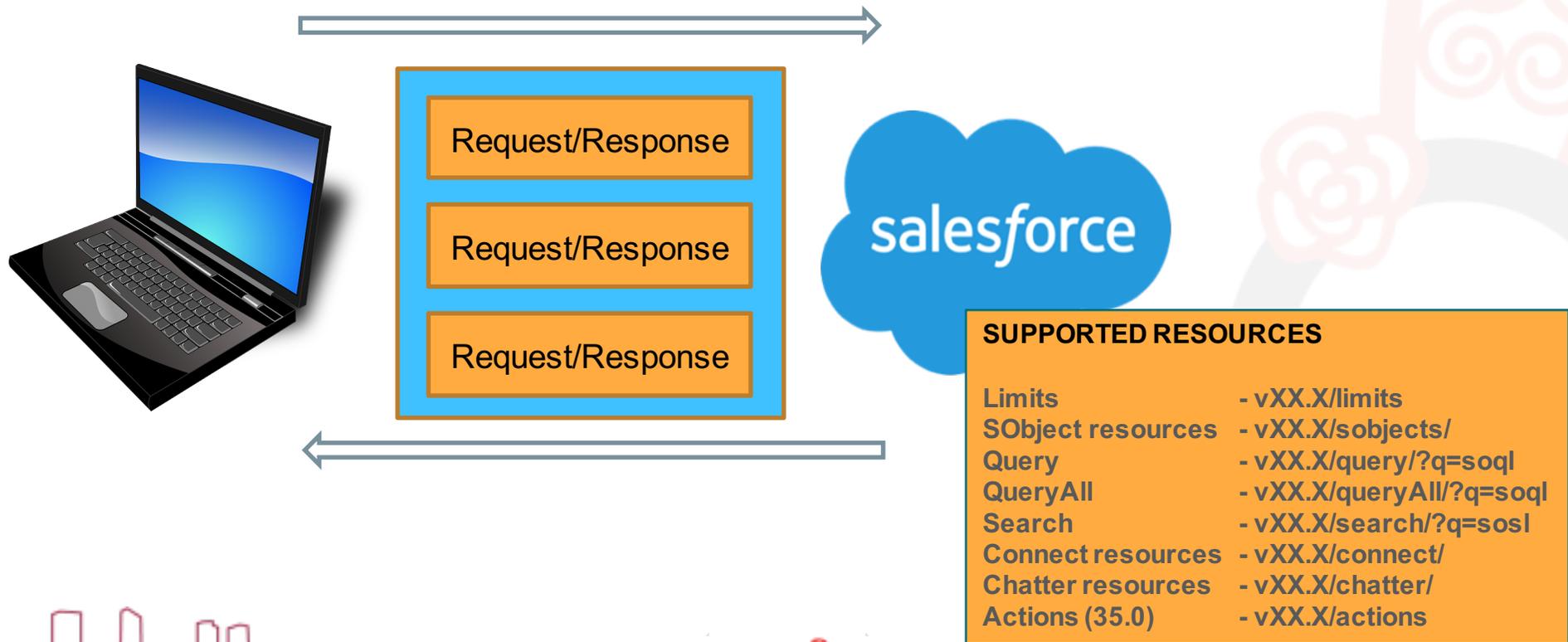
# Flujo de requests

Usa REST API composite resources para mejorar el performance de tu aplicación minimizando el número de envíos y respuestas que se realizan entre el cliente y salesforce.



# \composite\batch

Se utiliza para realizar un bloque de *request* independientes sin relación entre las mismas.



# \composite\batch Request

```
{
  "batchRequests" : [
    {
      "method" : "GET",
      "url" : "v38.0/subjects/Account/describe/"
    },
    {
      "method" : "POST",
      "url" : "v38.0/subjects/Account/",
      "richInput" : { "Name" : "Dream Olé 2017" }
    },
    {
      "method" : "GET",
      "url" : "v38.0/limits/"
    }
  ]
}
```

URI:

/services/data/v39.0/composite/batch

Formats: JSON, XML

HTTP method: POST

Authentication (Header)

Authorization: Bearer *token*

Body:

```
{ "batchRequests" : [
  subrequests
], "haltOnError" :
  boolean
```

**SUB-REQUEST:**  
method: GET, POST, ...  
url: url del servicio  
richInput: body

# \composite\batch Response

## hasErrors:

Indica si en alguna de las sub-request se ha producido un error al realizarla (false si ha ido todo correcto).

**results:** responses de las distintas sub-request realizadas.

**result:**  
statusCode: httpcode  
result: body response

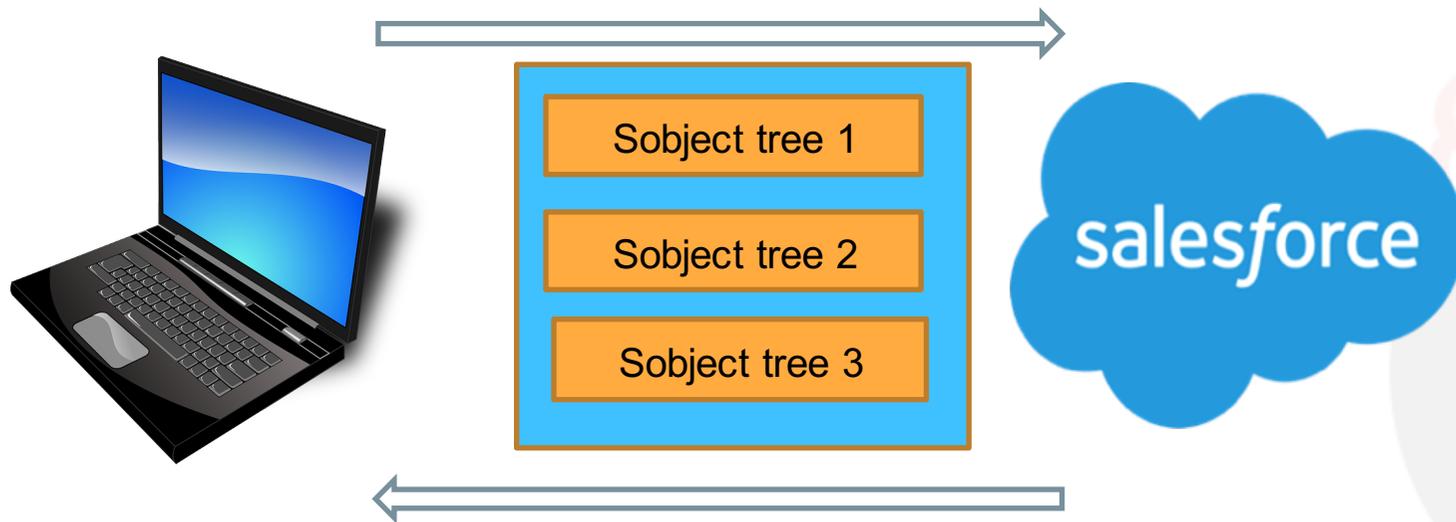
## Bad Request (Http Status: 400)

```
[ {  
  "errorCode" : "POST_BODY_PARSE_ERROR",  
  "message" : "Invalid value for request method: UPDATE"  
} ]
```

```
{  
  "hasErrors" : true,  
  "results" : [ {  
    "result" : [ {  
      "errorCode" : "METHOD_NOT_ALLOWED",  
      "message" : "HTTP Method '&#39;PUT&#39; not allowed. Allowed are HEAD,GET"  
    } ],  
    "statusCode" : 405  
  }, {  
    "statusCode" : 201,  
    "result" : {  
      "id" : "0010Y00000RmYQRQA3",  
      "success" : true,  
      "errors" : [ ]  
    }  
  }  
] ]  
}
```

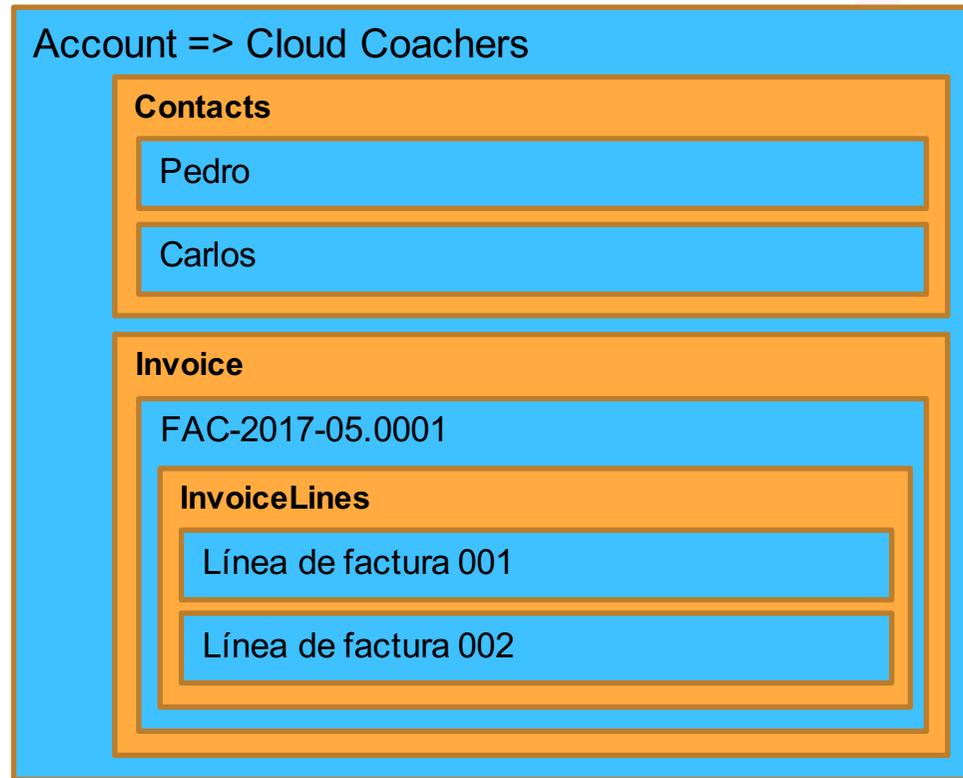
# \composite\tree

Se utiliza para crear múltiples registros del mismo objeto y registros hijos referentes al principal.



# SObject tree

- Lista de registros de un mismo tipo.
- El numero máximo de registros que se pueden crear es de 200.
- 5 tipos de SObject en total
- Máximo 5 niveles de profundidad



# \composite\tree Request

```
{
  "records" : [{
    "attributes" : {
      "type" : "Account", "referenceId" : "ref1"},
    "name" : "Cloud Coachers",
    "Contacts" : {
      "records" : [{
        "attributes" : {
          "type" : "Contact", "referenceId" : "ref2"},
          "lastname" : "Contact 1",},
        {
          "attributes" : {
            "type" : "Contact", "referenceId" : "ref3"}
          },
          "lastname" : "Contact 2",}]
      }
    }, {
      "attributes" : {
        "type" : "Account", "referenceId" : "ref4"}
      },
      "name" : "YXelp Software",
    }
  ]
}
```

URI:

/services/data/v39.0/composite/tree/*Subject*

Formats: JSON, XML

HTTP method: POST

Authentication (Header)

Authorization: Bearer *token*

Body:

```
{ "records" : [
  subjectTrees
]
}
```

**SUBREQUEST:**

```
attributes: type
            referenceld
```

```
field_1 : value_1
```

```
field_2 : value_2
```

```
childrecords : [
```

```
  attributes: type
```

```
              referenceld
```

```
  field_1 : value_1
```

```
  field_2 : value_2
```

```
]
```

# \composite\tree Response

```
{
  "hasErrors" : false,
  "results" : [ {
    "referenceId" : "ref1",
    "id" : "0010Y00000RmYaDQAV"
  }, {
    "referenceId" : "ref4",
    "id" : "0010Y00000RmYaEQAV"
  }, {
    "referenceId" : "ref2",
    "id" : "0030Y00000Ng25BQAR"
  }, {
    "referenceId" : "ref3",
    "id" : "0030Y00000Ng25CQAR"
  } ]
}
```

## hasErrors:

Indica si en alguna de las sub-request se ha producido un error al realizarla (false si ha ido todo correcto).

**results:** responses de las distintas sub-request realizadas.

## result:

referenceId: definido en la request  
id: id del nuevo registro

**errors:** error que ha producido que no se realice ninguna operación

```
{ "hasErrors" : true,
  "results" : [ {
    "referenceId" : "ref3",
    "errors" : [ {
      "statusCode" : "REQUIRED_FIELD_MISSING",
      "message" : "Faltan campos obligatorios: [LastName]",
      "fields" : [ "LastName" ]
    } ]
  } ]
}
```

# \composite

Se utiliza para realizar un bloque de subrequest las cuales pueden recibir como parámetro la response de la anterior.

Por ejemplo se podría insertar una cuenta y una oportunidad con varios contactos y las funciones de los contactos dentro de la oportunidad.

Este funcionamiento:

- Solo un request API para los limites.
- Puedo realizar el proceso en bloque.

## SUPPORTED RESOURCES

All sObject resources - vXX.X/subjects/

The Query resource - vXX.X/query/?q=soql

The QueryAll resource - vXX.X/queryAll/?q=soql

25 sub-request (10 query & queryMore)

# \composite Request

URI:

/services/data/v39.0/composite/

Formats: JSON, XML

HTTP method: POST

Authentication (Header)

Authorization: Bearer *token*

Body:

```
{ "allOrNone" : boolean
  "compositeRequest" : [
    composite SubRequest
  ]
}
```

**Composite subrequest:**  
**method:** GET, POST, ...  
**url:** url del servicio  
**referenceId:**  
  identificador referencia  
**body:** body

```
{
  "allOrNone" : true,
  "compositeRequest" : [{
    "method" : "POST",
    "url" : "/services/data/v38.0/subjects/Account",
    "referenceId" : "refAccount",
    "body" : { "Name" : "Cloud Coachers" }
  }, {
    "method" : "POST",
    "url" : "/services/data/v38.0/subjects/Contact",
    "referenceId" : "refContact",
    "body" : {
      "LastName" : "Contact 001",
      "AccountId" : "@{refAccount.id}"
    }
  }, {
    "method" : "POST",
    "url" : "/services/data/v38.0/subjects/Opportunity",
    "referenceId" : "refOpportunity",
    "body" : {
      "Name" : "Opportunity 001",
      "AccountId" : "@{refAccount.id}",
      "CloseDate" : "2017-07-01"
    }
  }, {
    "method" : "POST",
    "url" : "/services/data/v38.0/subjects/OpportunityContactRole",
    "referenceId" : "refOppContactRole",
    "body" : {
      "OpportunityContactRole" : "Technical",
      "ContactId" : "@{refContact.id}",
      "OpportunityId" : "@{refOpportunity.id}"
    }
  }
]}
}
```

# \composite Response – HALTED fail

```
{ "compositeResponse" : [ {  
  "body" : [ {  
    "errorCode" : "PROCESSING_HALTED",  
    "message" : "The transaction was rolled back since another  
    operation in the same transaction failed."  
  } ],  
  "httpHeaders" : { },  
  "httpStatusCode" : 400,  
  "referenceId" : "refAccount"  
}, {  
  "body" : [ {  
    "message" : "Faltan campos obligatorios: [StageName]",  
    "errorCode" : "REQUIRED_FIELD_MISSING",  
    "fields" : [ "StageName" ]  
  } ],  
  "httpHeaders" : { },  
  "httpStatusCode" : 400,  
  "referenceId" : "refOpportunity"  
} ] }  
}}
```

Al definir trabajar como una sola transacción, las operaciones que no se realicen debido al fallo de una anterior mostrarán el error code “PROCESSING\_HALTED”

**compositeResponse []**  
**body:**  
    **errorCode**  
    **message**  
**httpHeaders:** responses.  
**httpStatusCode:.**  
**referenceId:.**

## \composite Response - OK

```
{
  "compositeResponse" : [ {
    "body" : {
      "id" : "0010Y00000RmYbQAV",
      "success" : true,
      "errors" : [ ]
    },
    "httpHeaders" : {
      "Location" : "/services/data/v38.0/subjects
                  /Account/0010Y00000RmYbQAV"
    },
    "statusCode" : 201,
    "referenceId" : "refAccount"
  }
]
```

**CompositeResource[]**: responses de las sub-requests

- **body**: body de respuesta
- **httpHeaders**: url del recurso y otros datos
- **statusCode**: Código de estado
- **referenceId**: identificador indicado en la request

# Tipos de Composite Resources



Batch	Tree	Composite
Request de forma secuencias	Creación de un registro con sus dependencias de forma transaccional	Request dependientes y con lógica compleja

# Apex APIs personalizadas

Apex REST y SOAP API aumenta esta funcionalidad y hace posible que los desarrolladores creen sus propios servicios web utilizando Apex. Permitiendo utilizar la autenticación ya integrada de la plataforma.

## REST API

Global class con la siguientes anotaciones:

```
@RestResource(urlMapping='/yourUrl')
```

```
global class
```

```
@HttpDelete
```

```
global static
```

```
@HttpGet
```

```
@HttpPatch
```

```
@HttpPost
```

```
@HttpPut
```

## SOAP API

Global class con funciones marcadas como webservice

```
global class MyWebService {  
  
    webservice static Id  
    makeContact(String pLastName,  
                String pFirstName) {
```

Sharin Pix



copado

atmiradigital



Q & A

dreamOlé<sup>®</sup>

Madrid 1st June 2017

-  @inakihu
-  ihernandourcullu
-  ihernando@cloudcoachers.com
-  www.cloudcoachers.com

The image shows a nighttime cityscape of Madrid with several illuminated skyscrapers. The text "Q & A" is overlaid in a large white font on the left side. On the right side, the "dreamOlé" logo is displayed in white, with a stylized owl head above the "O". Below the logo, the date "Madrid 1st June 2017" is written. At the bottom right, there are four social media icons (Twitter, LinkedIn, Email, Home) with their corresponding contact information.