

# INTRODUCCIÓN AL INTERNET DE LAS COSAS

## PROTOTIPADO RAPIDO EN IOT



**Javier Pastor**  
javier.pastor@carriots.com  
@javipstor



# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## ÍNDICE

1. PORQUÉ PLATAFORMA IOT
2. FUNCIONALIDADES DE LA PLATAFORMA
3. PROYECTOS IOT EN LA PLATAFORMA
4. INTEGRACIÓN
5. CUANDO HAY MUCHOS DATOS
6. CUANDO SE QUIERE REACCIONAR

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## ÍNDICE

1. PORQUÉ PLATAFORMA IOT
2. FUNCIONALIDADES DE LA PLATAFORMA
3. PROYECTOS IOT EN LA PLATAFORMA
4. INTEGRACIÓN
5. CUANDO HAY MUCHOS DATOS
6. CUANDO SE QUIERE REACCIONAR

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PORQUÉ PLATAFORMA IOT

- DESARROLLO RÁPIDO DE PROYECTOS
- POTENCIA
- COSTES
- ESCALABILIDAD / FIABILIDAD



SMART AGRICULTURE  
SMART BUILDINGS  
SMART RETAIL

SMART CITY  
SMART ENERGY  
SMART OIL & GAS

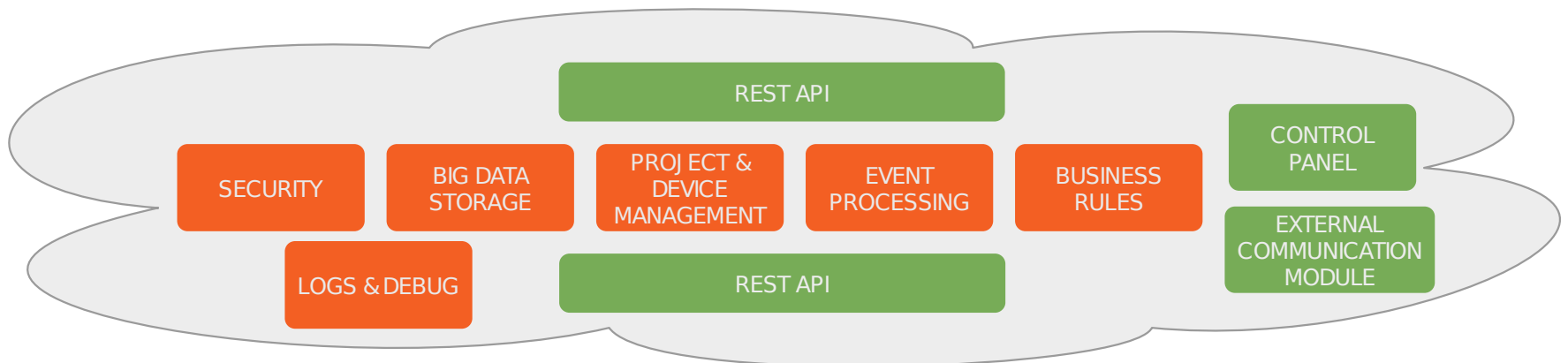
SMART BANKING  
SMART CONSUMER PRODUCTS  
SMART LOGISTICS

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PORQUÉ PLATAFORMA IOT

### DESARROLLO RÁPIDO

- INTEGRACIÓN
- GESTIÓN INTEGRADA
  - PROYECTOS
  - DISPOSITIVOS
- SEGURIDAD
- LÓGICA ACCESIBLE
- HERRAMIENTAS DE DESARROLLO

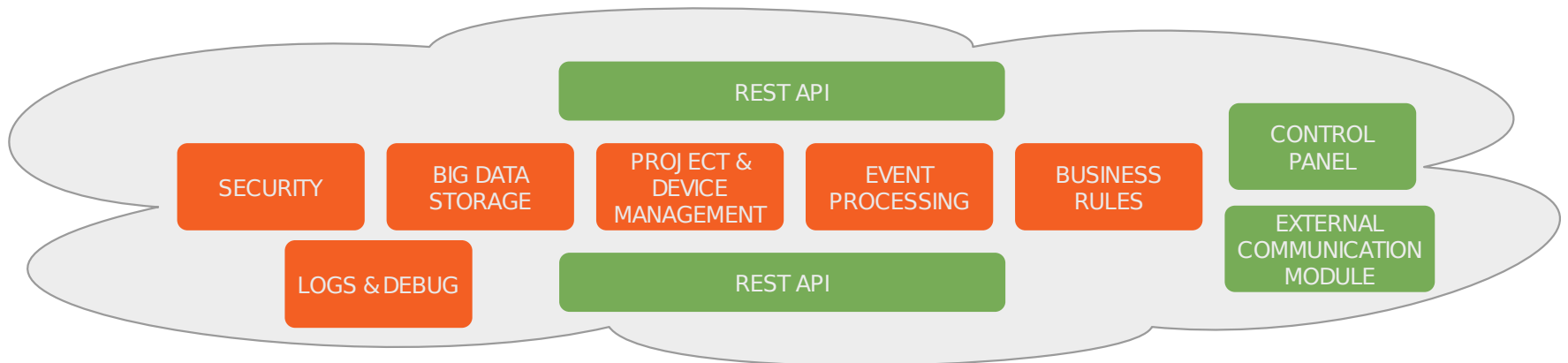


# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PORQUÉ PLATAFORMA IOT

### **COSTES**

- INFRAESTRUCTURA EN LA NUBE
- FACILIDAD DE INTEGRACIÓN
- TIME TO MARKET
- MVP / PROTIPO RÁPIDO

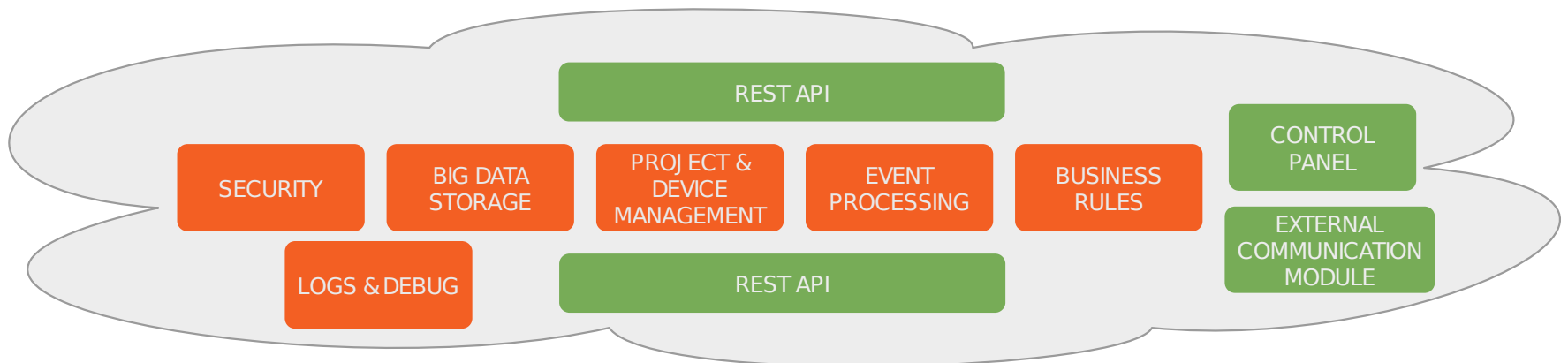


# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PORQUÉ PLATAFORMA IOT

### ESCALABILIDAD / FIABILIDAD

- ARQUITECTURA SW
- ARQUITECTURA HW
- KNOW - HOW
- COMUNIDAD

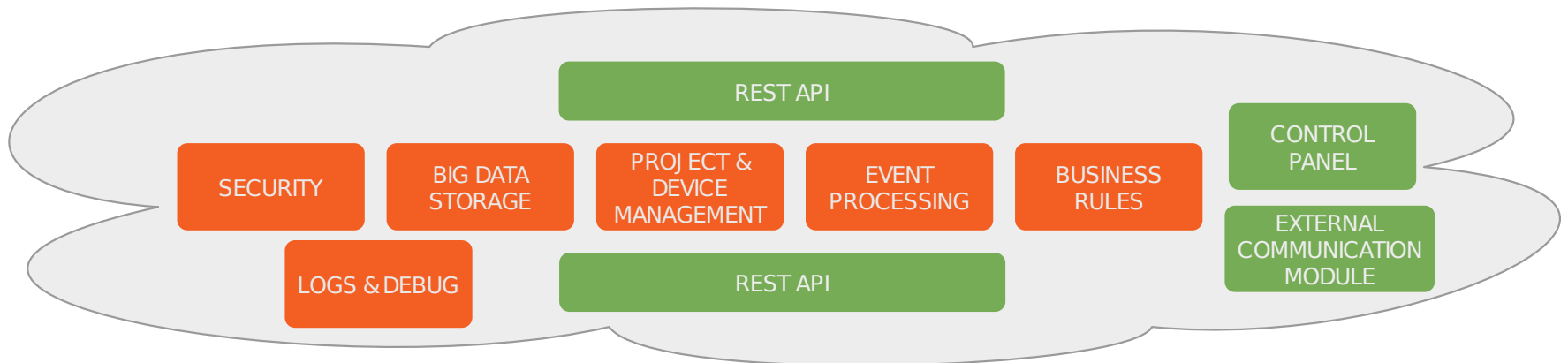


# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PORQUÉ PLATAFORMA IOT

### **POTENCIA**

- ORIENTADO A EVENTOS (O SU AUSENCIA)
- SDK - SCRIPTING & SANDBOXING
- PUSH/PULL APPROACHES
- BIG DATA





# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## ÍNDICE

1. PORQUÉ PLATAFORMA IOT
2. FUNCIONALIDADES DE LA PLATAFORMA
3. PROYECTOS IOT EN LA PLATAFORMA
4. INTEGRACIÓN
5. CUANDO HAY MUCHOS DATOS
6. CUANDO SE QUIERE REACCIONAR

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## FUNCIONALIDADES DE LA PLATAFORMA

### API REST

Siguiendo un estándar ampliamente adoptado en Internet, Carriots implementa una API REST sobre HTTPS para:

- **Recibir** cantidades masivas de datos de los dispositivos.
- **Interactuar con todas las entidades de Carriots** para construir paneles de control personalizados, dashboards e informes

Ej: Sensores de parking envían su estado (libre u ocupado) a Carriots con una simple petición HTTP. Un panel de control personalizado usa la API REST para gestionar dispositivos.

### ALMACENAMIENTO BIG DATA

Grandes cantidades de datos se almacenan en una arquitectura de big data con estructura schemaless. Proporciona a los proyectos de IoT la flexibilidad de gestionar datos heterogéneos de diferentes dispositivos.

Ej: Temperatura, posición geográfica, estado de una plaza de parking, etc.

### GESTIÓN DE DISPOSITIVOS Y PROYECTOS

Para cubrir cualquier requisito de un proyecto, Carriots propone una jerarquía lista para usar y de uso flexible para la complejidad de cada caso.

Actualización de firmware y configuración remota de dispositivos también disponibles.

Ej: Ajustar la periodicidad del muestreo de los sensores o especificar un mensaje por defecto en un panel de señalización.

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## FUNCIONALIDADES DE LA PLATAFORMA

### LÓGICA DE NEGOCIO Y EVENTOS

La lógica del proyecto de IoT se aloja y se ejecuta en la plataforma. Scripts de Groovy se ejecutan (aislados y securizados) en el motor SDK y se lanzan en base a eventos tratados con una aproximación tipo **if-then-else**. Desde scripts sencillos a complejas reglas.

Ej: If parking\_sensor="ocupado" then parking\_area.plazas -1.  
If parking\_area.plazas=0 then display.mensaje="Parking lleno"

### SEGURIDAD

En todos los proyectos IoT la seguridad debe formar parte de su diseño. La aproximación de Carriots a este tema:

- **Apikeys** definen privilegios y visibilidad.
- **HTTPS** para cifrar conversaciones con la API REST.
- **HMAC hash** y contraseñas pre-compartidas para firmar el mensaje.
- **Cifrado** personalizado o medidas adicionales de seguridad en el código de los eventos.

### LOGS & DEBUG

Una consola para **debug**, necesaria para facilitar el desarrollo y una serie de alarmas que se levantan ante cualquier fallo.

### MÓDULO DE COMUNICACIÓN

Enviar emails o SMS, interactuar con otros sistemas o APIs. Todo esto se realiza con un módulo específico para garantizar la seguridad y el rendimiento.

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## FUNCIONALIDADES DE LA PLATAFORMA

- APLICA
  - Prototipado rápido
  - Despliegues masivos
  - Integraciones de diferentes fuentes de datos
  - Proyectos que no puedan disponer de backoffice: sensores + frontend
  - Proyectos de data logging
  - Etc.
- NO APLICA
  - Proyectos de tiempo real en la nube
  - Silos en los que no hay salida a Internet
  - Escenarios que requieren de baja latencia en las comunicaciones
  - Proyectos en los que se deba entregar el código fuente

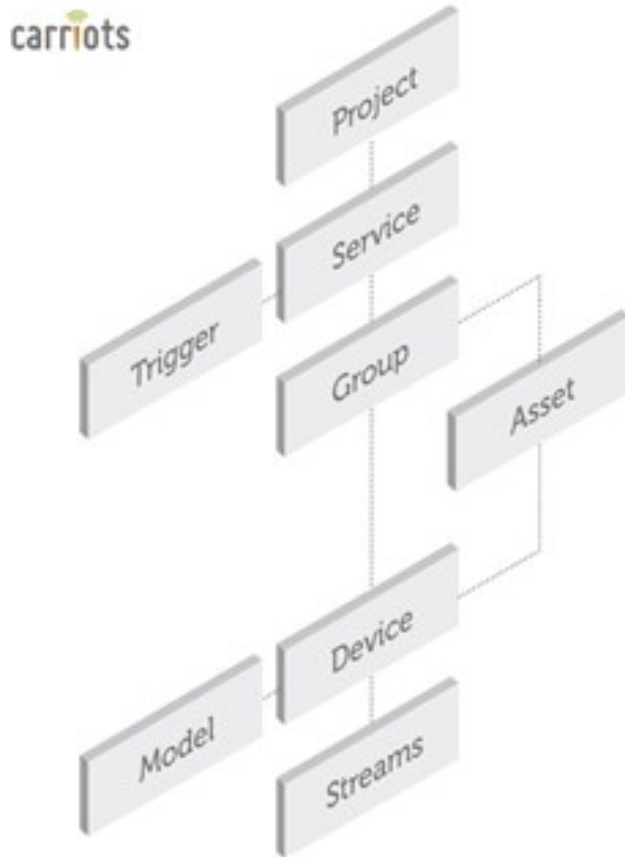
# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## ÍNDICE

1. PORQUÉ PLATAFORMA IOT
2. FUNCIONALIDADES DE LA PLATAFORMA
3. PROYECTOS IOT EN LA PLATAFORMA
4. INTEGRACIÓN
5. CUANDO HAY MUCHOS DATOS
6. CUANDO SE QUIERE REACCIONAR

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

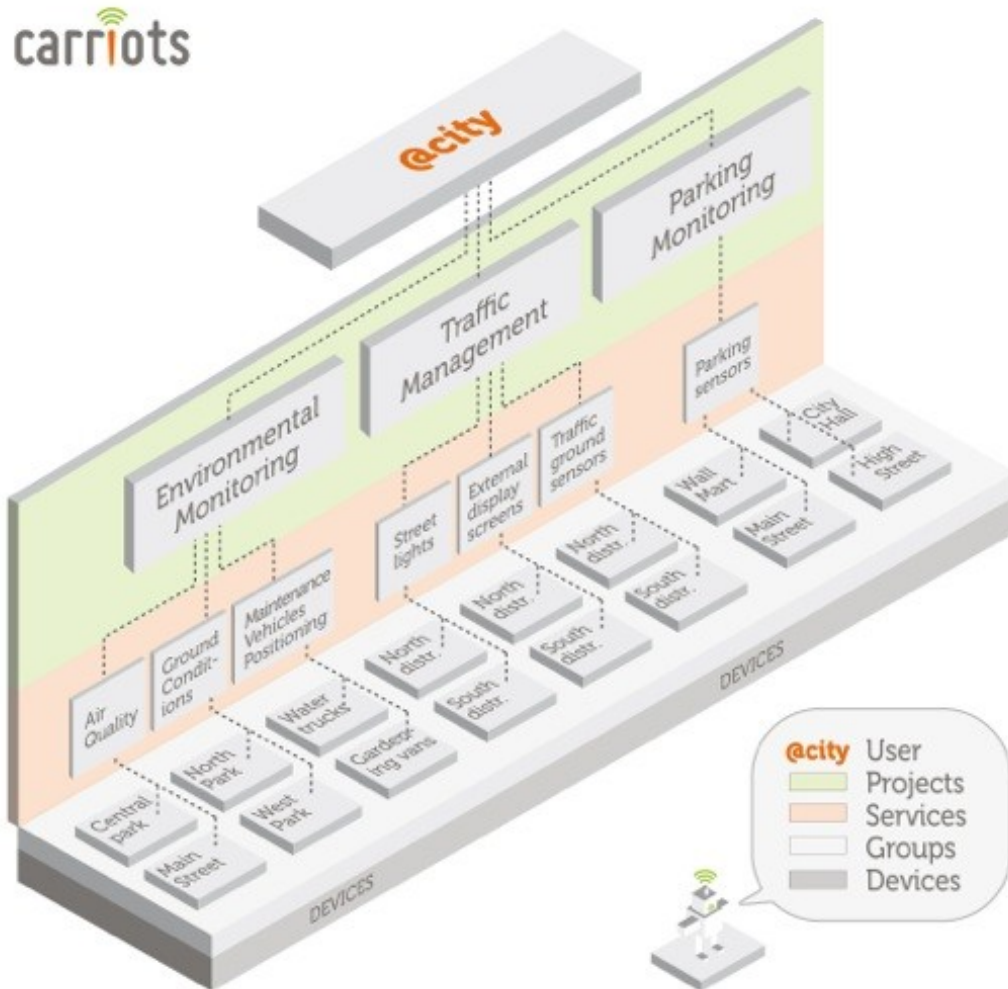
## PROYECTOS IOT EN LA PLATAFORMA



- Carriots modela los proyectos IoT en entidades
- Carriots ofrece una jerarquía por defecto
- **Proyectos, Servicios, Grupos y Asset** son entidades en una jerarquía para agrupar y estructurar sus dispositivos
- Una buena definición de la jerarquía mejorará la gestión del proyecto
- Centrado en dispositivos
- Las tramas se asocian a dispositivos

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

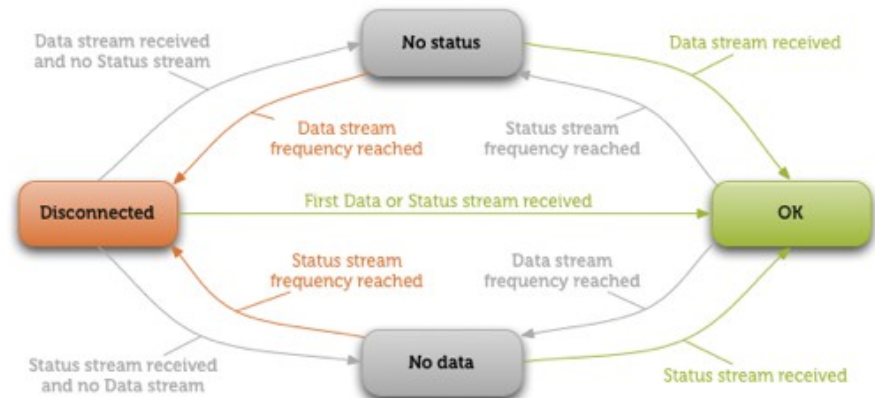
## PROYECTOS IOT EN LA PLATAFORMA



# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PROYECTOS IOT EN LA PLATAFORMA

- Dos tipos de tramas
  - Stream. <http://api.carriots.com/streams>
  - Status. <http://api.carriots.com/status>
- Dos propiedades de dispositivo para comprobar la recepción en el tiempo esperado
  - Data stream frequency
  - Status stream frequency
- Un dispositivo puede estar en 4 estados diferentes
  - Disconnected
  - No\_data
  - No\_status
  - Ok





# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PROYECTOS IOT EN LA PLATAFORMA

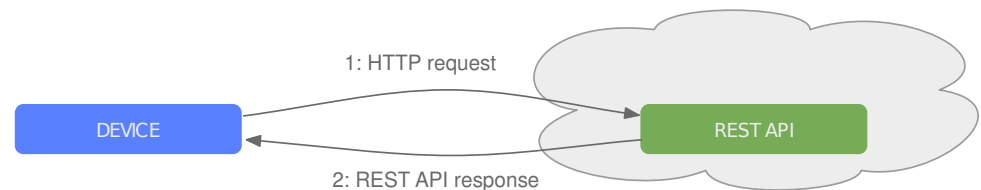
- PAYLOAD de las tramas (stream y status)
  - Protocol: Protocolo de comunicación
  - Device: Dispositivo que manda la trama
  - At: Unix timestamp o bien “now”
  - Data: datos a enviar
- Ejemplo envío de trama (plaza de parking)

### HTTP request

```
POST /streams HTTP/1.1
Host: api.carriots.com
Accept: application/json
User-Agent: place_524@smartparking
Content-Type: application/json
carriots.apikey:98346673a637...5a0d83045425407ab4
Content-Length: 182
Connection: close
```

### Data (PAYLOAD)

```
{
  "protocol": "v2",
  "at": "now",
  "device": "place_524@smartparking",
  "data": {"parking": "on"},
  "checksum": "2c0766329b4d4b3beb08...97ae7b7de2160be"
}
```

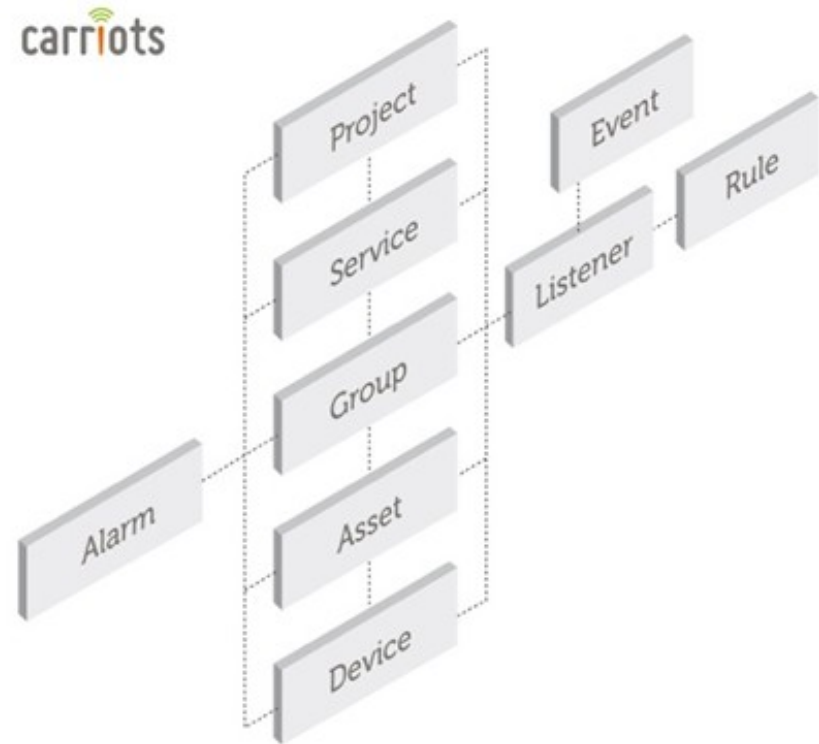


# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PROYECTOS IOT EN LA PLATAFORMA

### LÓGICA DE NEGOCIO

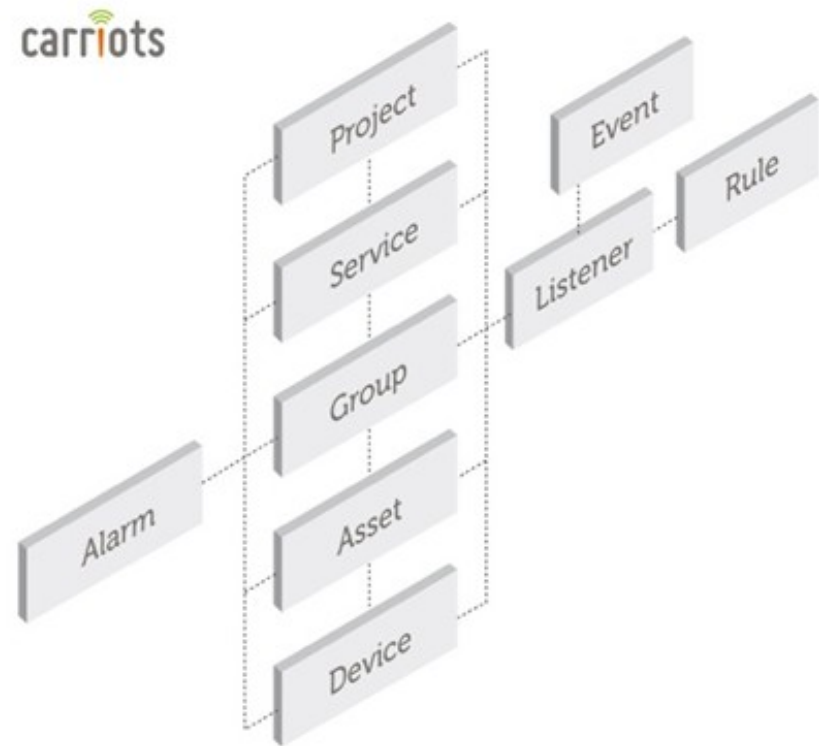
- La **lógica** de un proyecto IoT se aloja y se ejecuta en la plataforma
- Scripts en **GROOVY** son ejecutados en el motor SDK de forma aislada
- Enfoque **if-then-else** que simplifica la compresión y reduce el código
- Construye desde simples Scripts a **complejas reglas** de negocio



# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PROYECTOS IOT EN LA PLATAFORMA

- **SDK**: Motor de la inteligencia del núcleo en el ecosistema de Carriots. Basado en Groovy y con librerías para ayudarle a trabajar más rápido
- **Listener**: “Escuchador” asociable a cualquier elemento de la jerarquía esperando a que ocurra un evento para ejecutar una porción de código
- **Rule**: Porción de código definida por el usuario para ser utilizada en varios listeners
- **Trigger**: “Lanzador” asociado a un Service que permite realizar una petición HTTP de forma inmediata cuando se recibe una trama
- **Evento**: Suceso que ocurre en Carriots y que desencadena una acción



# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PROYECTOS IOT EN LA PLATAFORMA

### TIPOS DE EVENTO

#### **Cron**

Cada intervalo de tiempo definido por el usuario

#### **Data received**

Cuando se recibe una trama

#### **Data persisted**

Cuando se almacena una trama

#### **Device change state**

Cuando un dispositivo cambia de estado

#### **Stream data received**

Cuando se recibe una trama de datos

#### **Stream data persisted**

Cuando se almacena una trama de datos

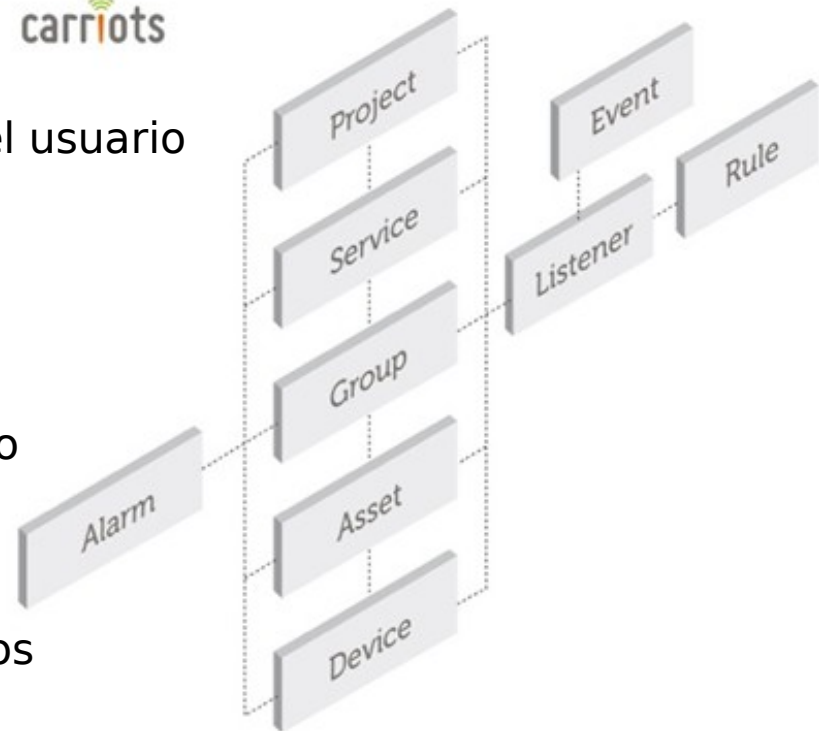
#### **Status data received**

Cuando se recibe una trama de estado

#### **Status data persisted**

Cuando se almacena una trama de estado

carriots



# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PROYECTOS IOT EN LA PLATAFORMA

### Creación de listener

SMARTCITY ADMINISTRATION MY SETTINGS DEBUG & LOG HELP ALARMS 30 LOGOUT

CARRIOTS CONTROL PANEL

### Listener creation

Name

Description

Entity type

Id

Event

If expression

Then expression

Then rule

Else expression

Else rule

Enabled ☒

Create List

ENTIDAD VIGILADA

EVENTO QUE  
DESENCADENA LA  
LÓGICA

APROXIMACIÓN  
IF-THEN-ELSE

SCRIPTS DE GROOVY

REGLAS PREDEFINIDAS

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## PROYECTOS IOT EN LA PLATAFORMA

// Libraries

```
import com.carriots.sdk.Device;  
import com.carriots.sdk.utils.BasicHttp;
```

← SDK LIBRARIES IMPORT

// Free places counter update

```
def device = Device.find('ParkingControl@smartparking');  
def places = new Integer(device.device_properties.FreePlaces);  
device.device_properties.FreePlaces=places-1;  
device.update();
```

← DATA BASE ACCESS

// Place update

```
def place = Device.find(context.device);  
place.device_properties['free']='no';  
place.update();
```

← CUSTOM PROPERTIES MANAGEMENT

// Display location

```
def led = Device.find('LedDisplay-12@smartparking');  
def address = led.device_properties.address;  
def token = led.device_properties.token;
```

← CARRIOTS -> DEVICE COMMUNICATION  
IP address stored as device property

// Display update

```
def basicHttp = new BasicHttp();  
basicHttp.url = "http://" + address + "/message";  
basicHttp.params=["text":"Free places: " + device.device_properties.FreePlaces, "token":token];  
basicHttp.send();
```



# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## ÍNDICE

1. PORQUÉ PLATAFORMA IOT
2. FUNCIONALIDADES DE LA PLATAFORMA
3. PROYECTOS IOT EN LA PLATAFORMA
4. INTEGRACIÓN
5. CUANDO HAY MUCHOS DATOS
6. CUANDO SE QUIERE REACCIONAR

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## INTEGRACIÓN

### PUSH APPROACH

- Tipo de comunicación donde el **servidor** es el que inicia la comunicación
- Gracias al uso de Trigger o Listener, Carriots permite el envío de peticiones PUSH a otros Servicios
- Ejemplo Ducksboard

[https://public.ducksboard.com/mvG\\_u7xHr3QNYGEuogiQ/](https://public.ducksboard.com/mvG_u7xHr3QNYGEuogiQ/)





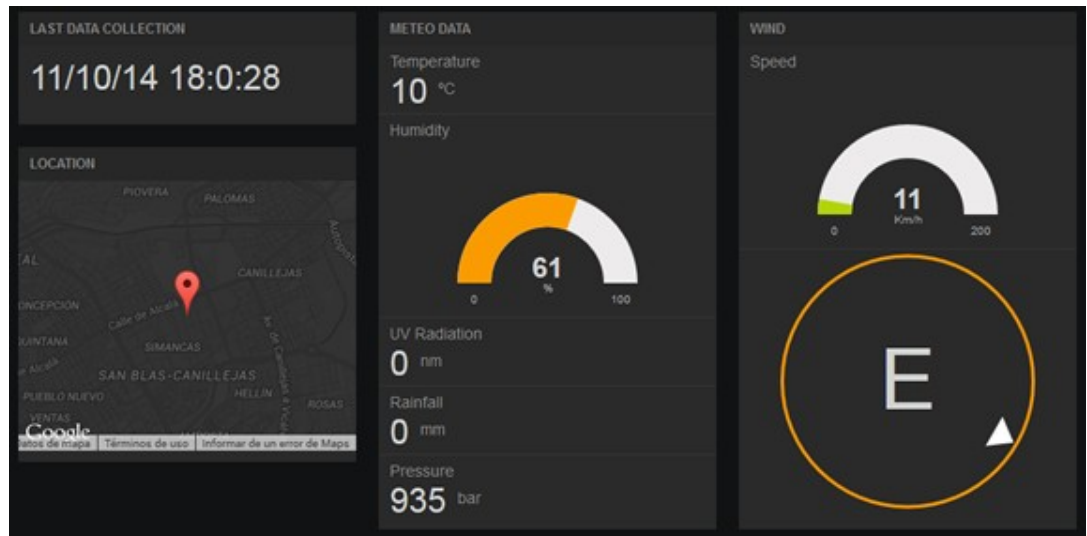
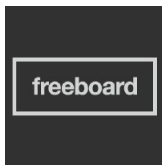
# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## INTEGRACIÓN

### PULL APPROACH

- Tipo de comunicación donde el **cliente** es el que inicia la comunicación
- Podemos hacer PULL a la REST API para construir nuestro dashboard
- Ejemplo Freeboard

<https://freeboard.io/board/54044523f43cebd83b00001c>

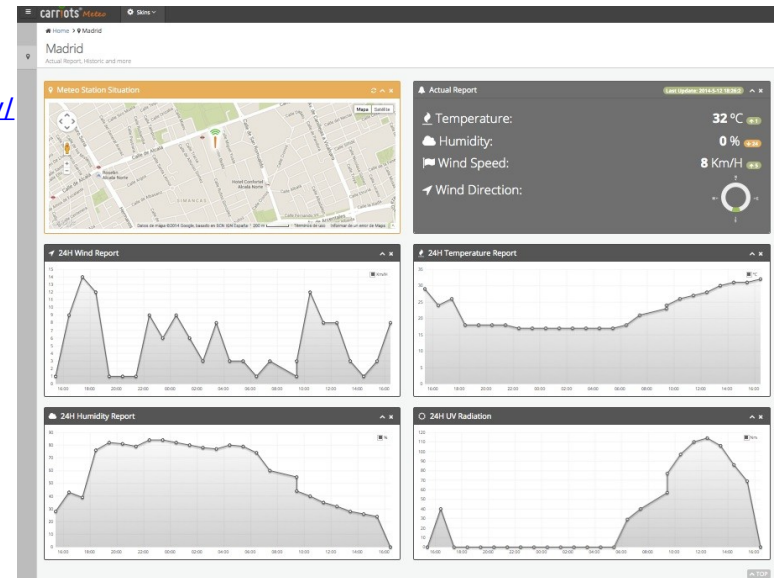


# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## INTEGRACIÓN

- Podemos usar **Javascript** para crear nuestro propio dashboard personalizado
- Carriots tiene habilitado **CORS** para que puedas construir tu **client-only frontend**
- Ejemplo Javascript

<https://blog.carriots.com/carriots-public-weather-station-come-and-play/>  
<http://demo.carriots.com/meteocarriots/index.html>



# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## ÍNDICE

1. PORQUÉ PLATAFORMA IOT
2. FUNCIONALIDADES DE LA PLATAFORMA
3. PROYECTOS IOT EN LA PLATAFORMA
4. INTEGRACIÓN
5. CUANDO HAY MUCHOS DATOS
6. CUANDO SE QUIERE REACCIONAR

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## CUANDO HAY MUCHOS DATOS

- La mejor opción es contar con una herramienta especializada
- La plataforma se queda con
  - La gestión de los datos en bruto
  - La gestión de las conexiones con los dispositivos
  - La seguridad e integridad de la plataforma “hacia abajo”
- La plataforma envía los datos (PUSH) o pone a disposición de (PULL) la herramienta especializada en Big Data, Business Intelligence, Business Analytics, Machine learning o Deep learning

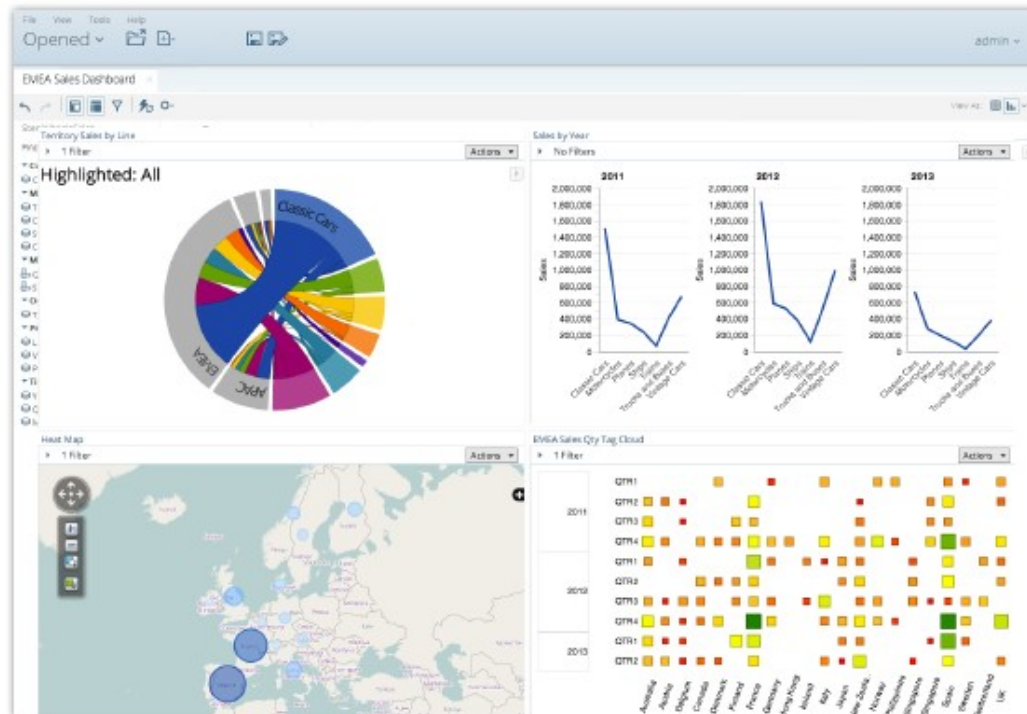
# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

CUANDO HAY MUCHOS DATOS

EJEMPLOS DE HERRAMIENTAS / PLATAFORMAS



<http://www.pentaho.com>



# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

CUANDO HAY MUCHOS DATOS

EJEMPLOS DE HERRAMIENTAS / PLATAFORMAS



<http://www.oracle.com/es/solutions/business-intelligence/>



<https://powerbi.microsoft.com>



<http://aws.amazon.com/es/>



Google Cloud Platform

<https://cloud.google.com>

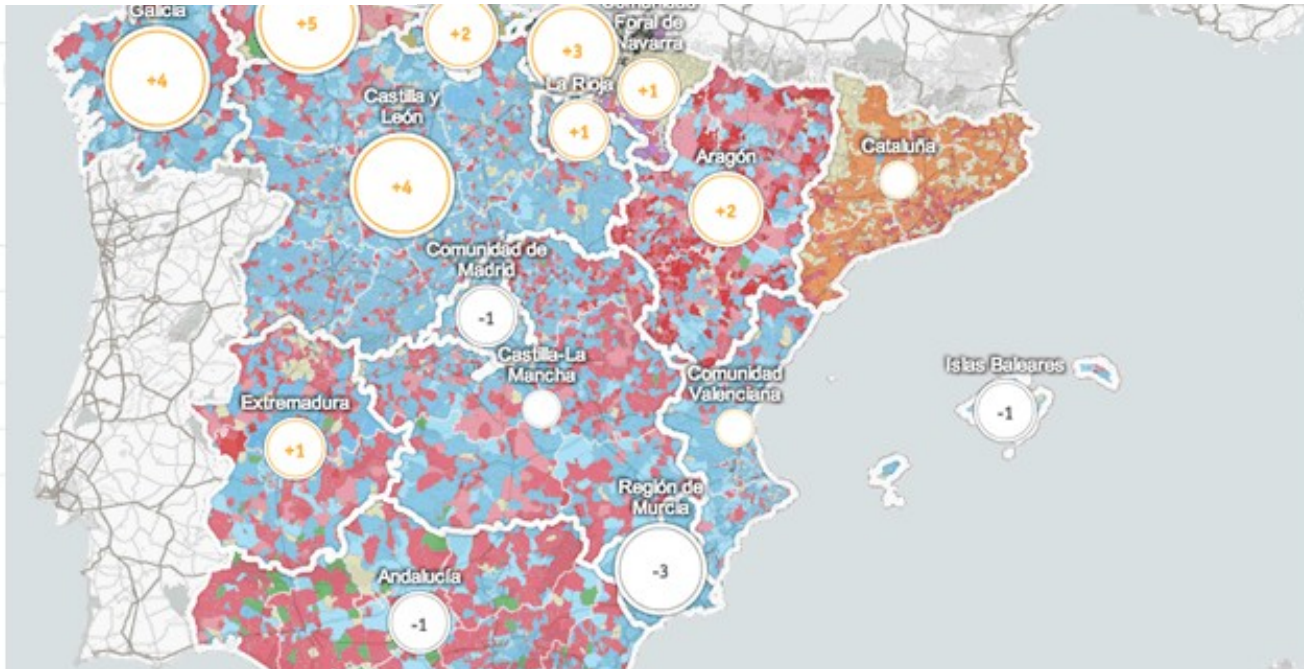
# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

CUANDO HAY MUCHOS DATOS

## EJEMPLOS DE HERRAMIENTAS / PLATAFORMAS



<http://cartodb.com>



# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## ÍNDICE

1. PORQUÉ PLATAFORMA IOT
2. FUNCIONALIDADES DE LA PLATAFORMA
3. PROYECTOS IOT EN LA PLATAFORMA
4. INTEGRACIÓN
5. CUANDO HAY MUCHOS DATOS
6. CUANDO SE QUIERE REACCIONAR



# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

## CUANDO SE QUIERE REACCIONAR

- Desde la plataforma todo Internet está disponible
- La ubicuidad de las APIS REST facilitan la integración
- Algunas ofrecen valor añadido:
  - Control de uso de la API
  - Monetización de la API
  - Autenticación federada

# TECNOLOGÍAS DE PROTOTIPADO RÁPIDO EN IOT

CUANDO SE QUIERE REACCIONAR

EJEMPLOS DE HERRAMIENTAS / PLATAFORMAS



Google Drive



foursquare®



# INTRODUCCIÓN AL INTERNET DE LAS COSAS

## PROTOTIPADO RAPIDO EN IOT



**Javier Pastor**  
javier.pastor@carriots.com  
@javipstor

