


API REST  
HYPERMEDIA  
BACKBONEJS  
MARIONETTE

**VS.**

HEXAGONAL  
ARCHITECTURE  
ANGULARJS

CHAN  
CHAN

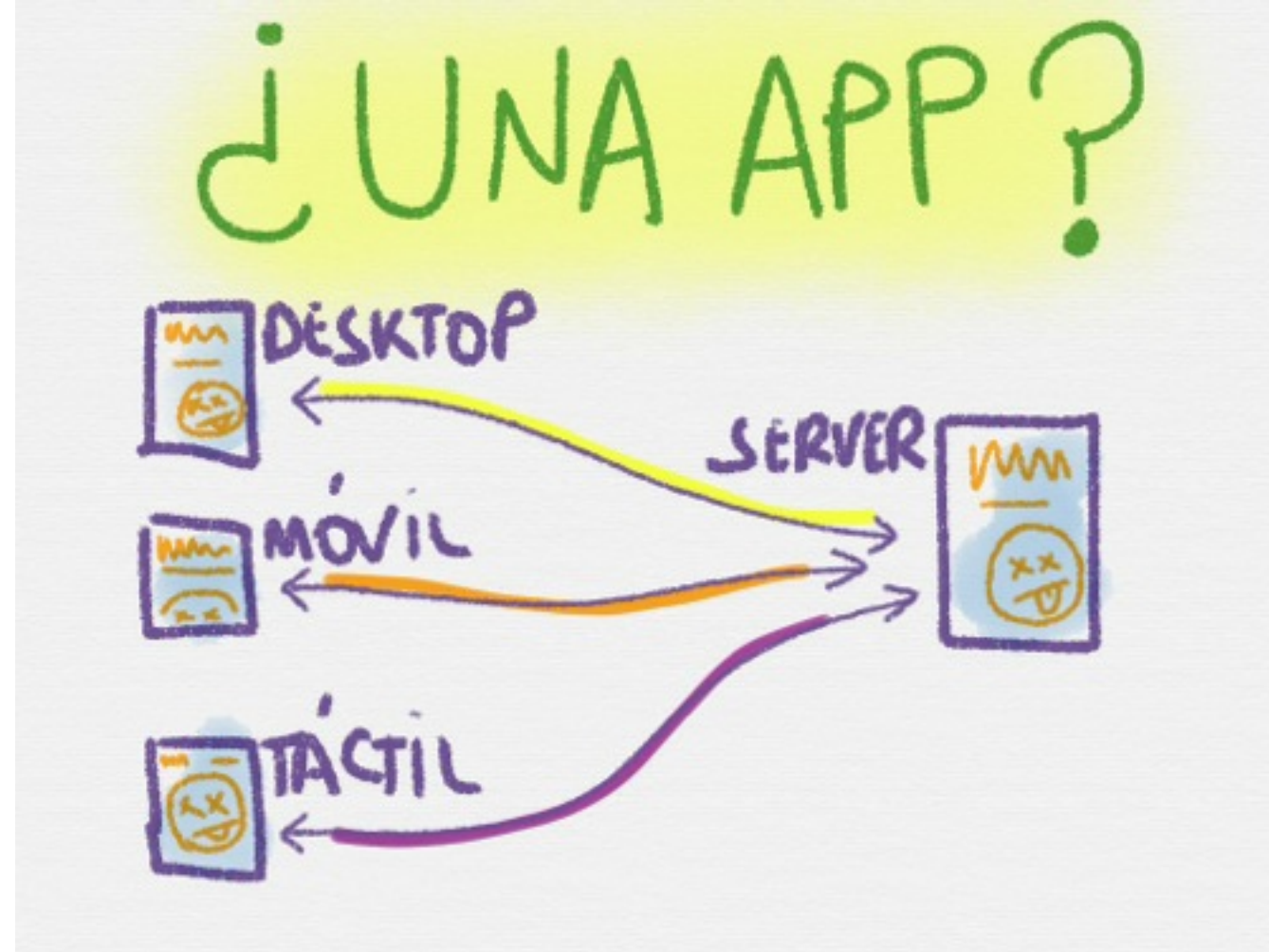
@artolamola



@ggalmazor

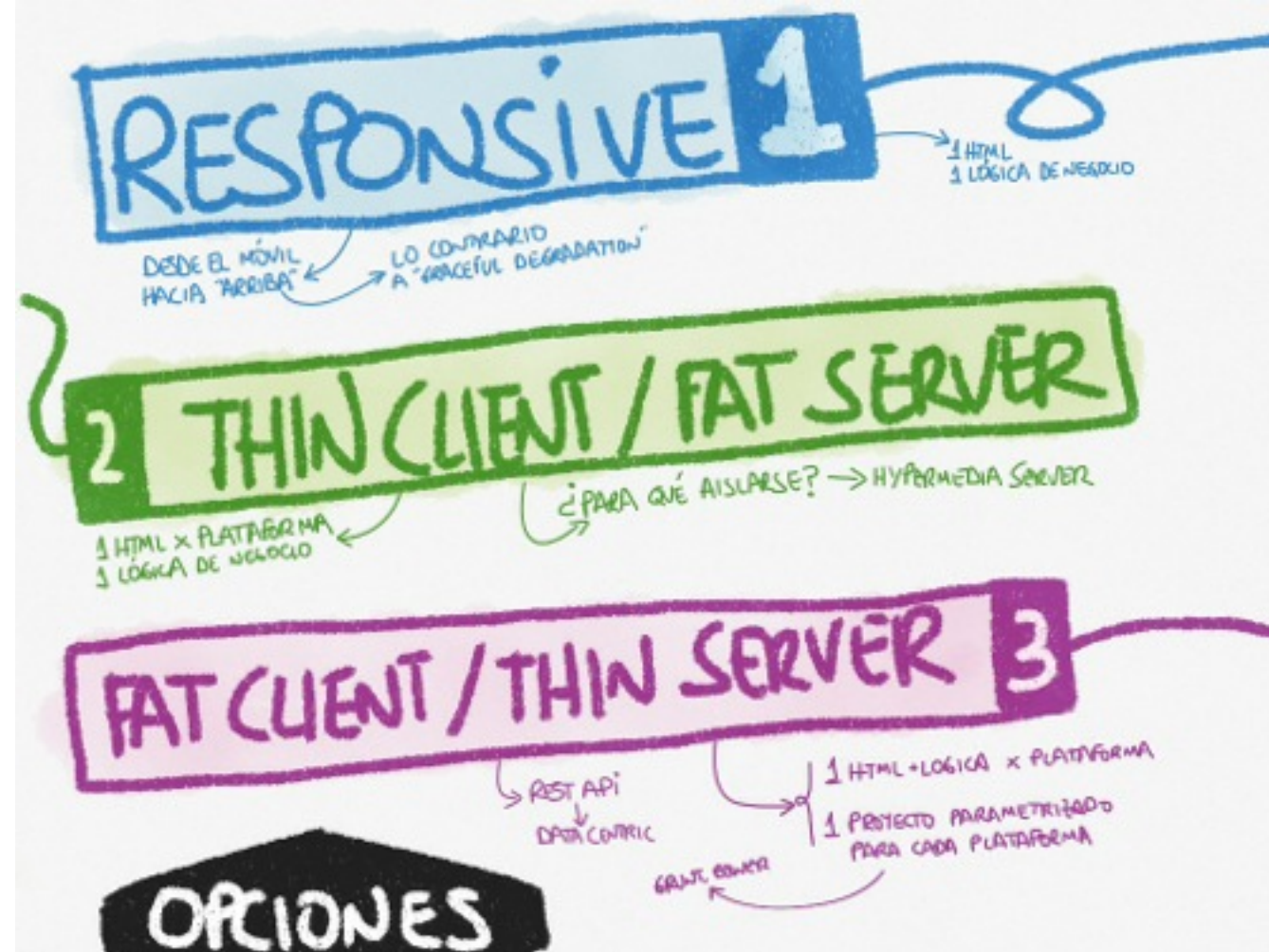


BilboStack;  
2014



Este es el **problema** que nos empuja a estudiar Hexagonal u otros paradigmas.

Puede parecer una sola app, pero la realidad es que podemos organizar **varias arquitecturas** para conseguir este resultado



Preguntas que debo hacerme: ¿Cómo son y dónde están los datos que presento? ¿Es un punto clave la UX de mi app para mis usuarios? ¿Existe una lógica de negocio pesada o crítica? etc.

**No hay una bala de plata**, solo opciones y nuestra capacidad de juicio

<i>Angular JS</i>	<b>BackboneJS</b>
Data-centric	<i>Hypermedia</i>
<b>Hexagonal</b>	Acoplado

Estas son las **3 claves** a la hora de definir una app: Frameworks - Tipo de servidor - Arquitecturas



## DATA CENTRIC

ACOPADO A TABLAS

↓  
"STANDARD"      SENCILLO

BAJA UX

↳ TIENES QUE CONOCER  
LAS URLS...

## HYPERMEDIA

ALTA UX { FORMS YA  
RELLENOS  
• JSONES VACIOS  
• URI TEMPLATES  
...

↓  
COMPLEJO

↓ NO STANDARD → EL SERVER DECIDE  
ESTRUCTURAS DE  
DATOS  
↳ HAL, SIREN



## Principios de diseño de APIs REST

<http://leanpub/u/ear>

# ALTERNATIVA

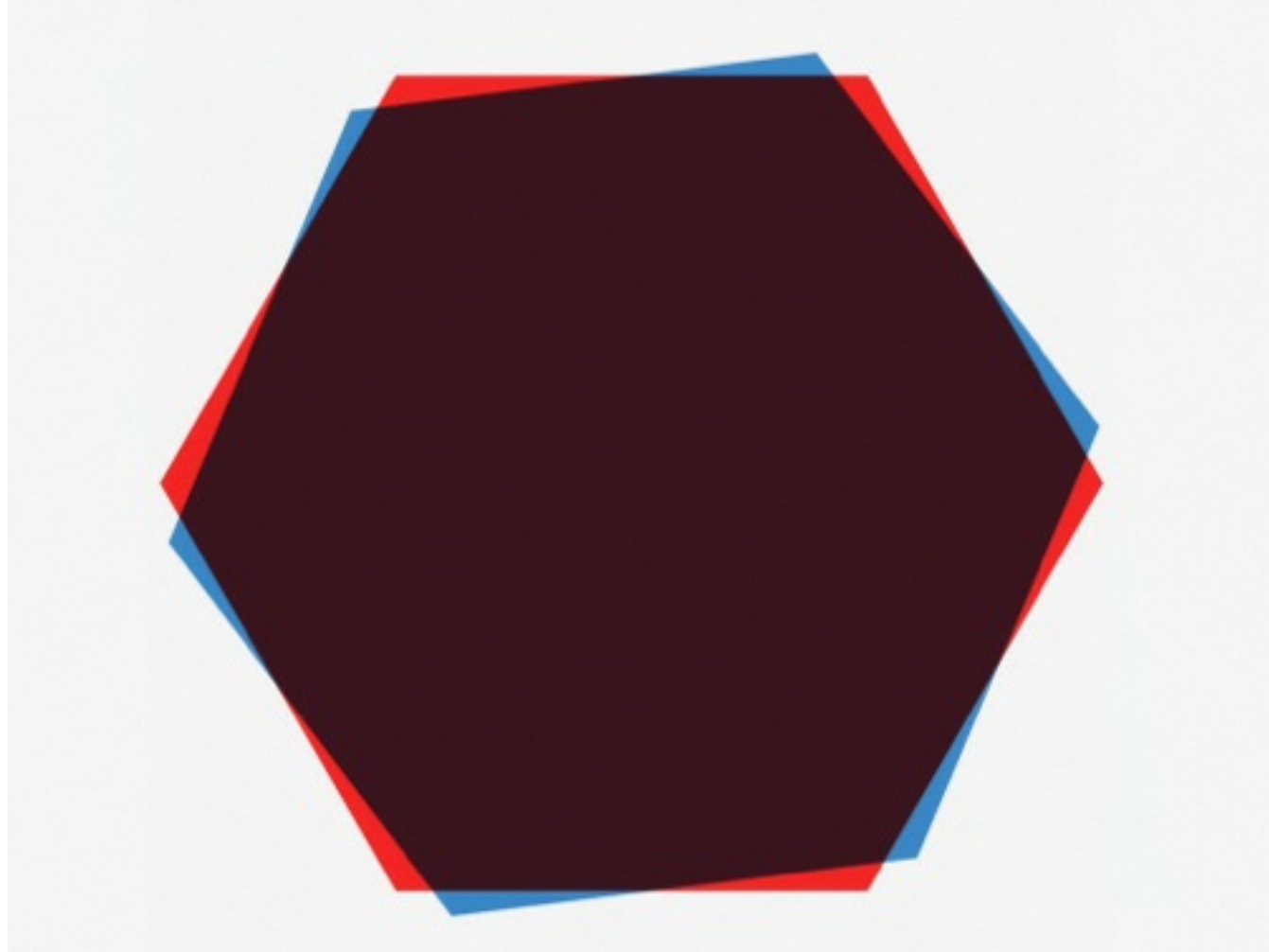
PUT /rest/compra

NO ES UNA  
TABLA

<u>ANGULAR</u>	<u>BACKBONE</u>
LIBERTAD	MODELS COLLECTIONS ACTIVE RECORD
BINDEO BIDIRECCIONAL	BINDEO DECLARATIVO
ECOSISTEMA (KARMA, TOOLS, ...)	DOLOR

Otras features de AngularJS

- Plantillas
- MVC
- DIP
- Directivas



a.k.a Ports & Adapters

Metáfora hexágono

Interior - Exterior

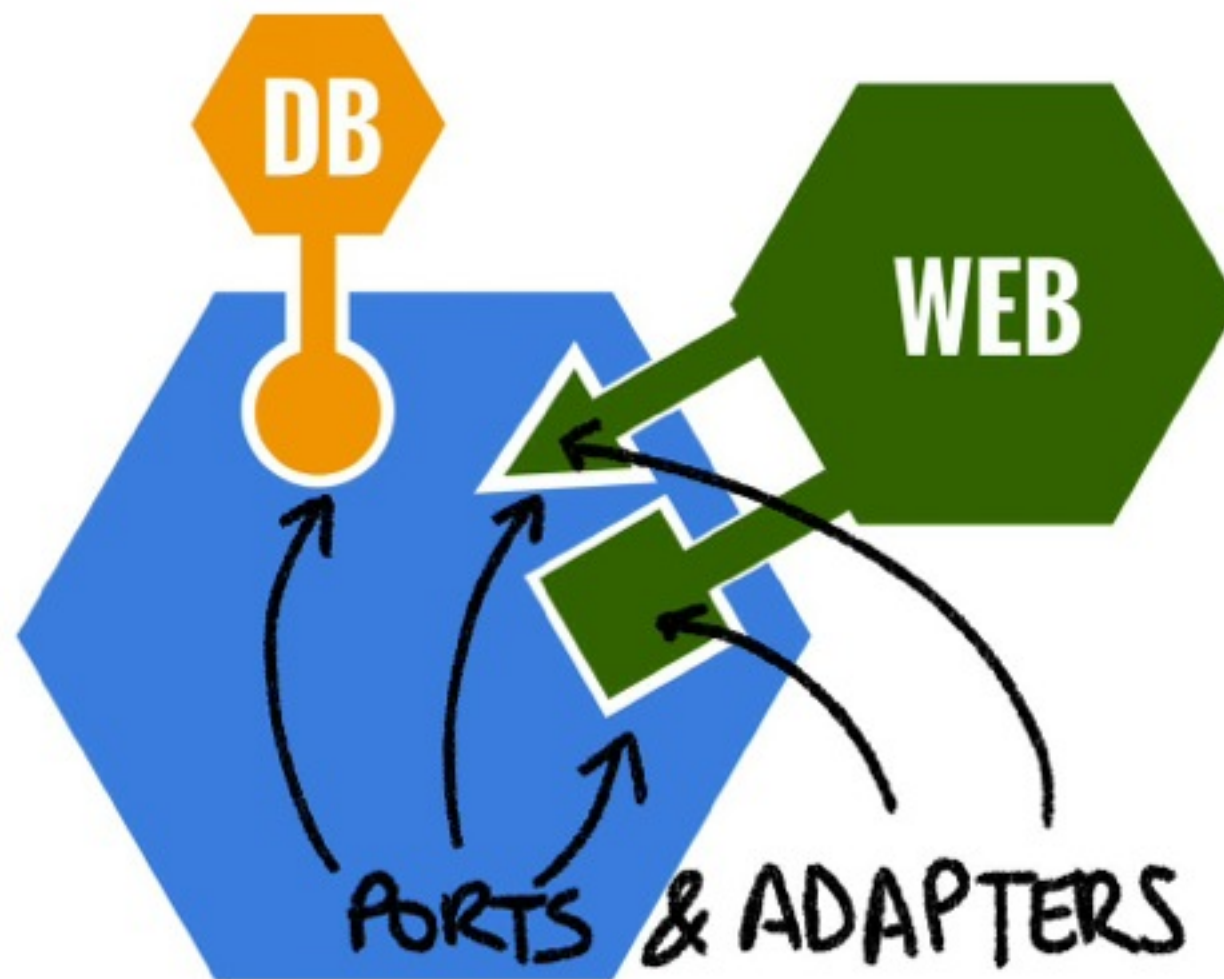
**Varias caras** (integraciones con tecnologías y sistemas terceros)

**Rompe** con el diagrama de **capas** clásico unidimensional





Son cualidades que pueden parecer poco valiosas para el que no está acostumbrado a desarrollar con prácticas XP

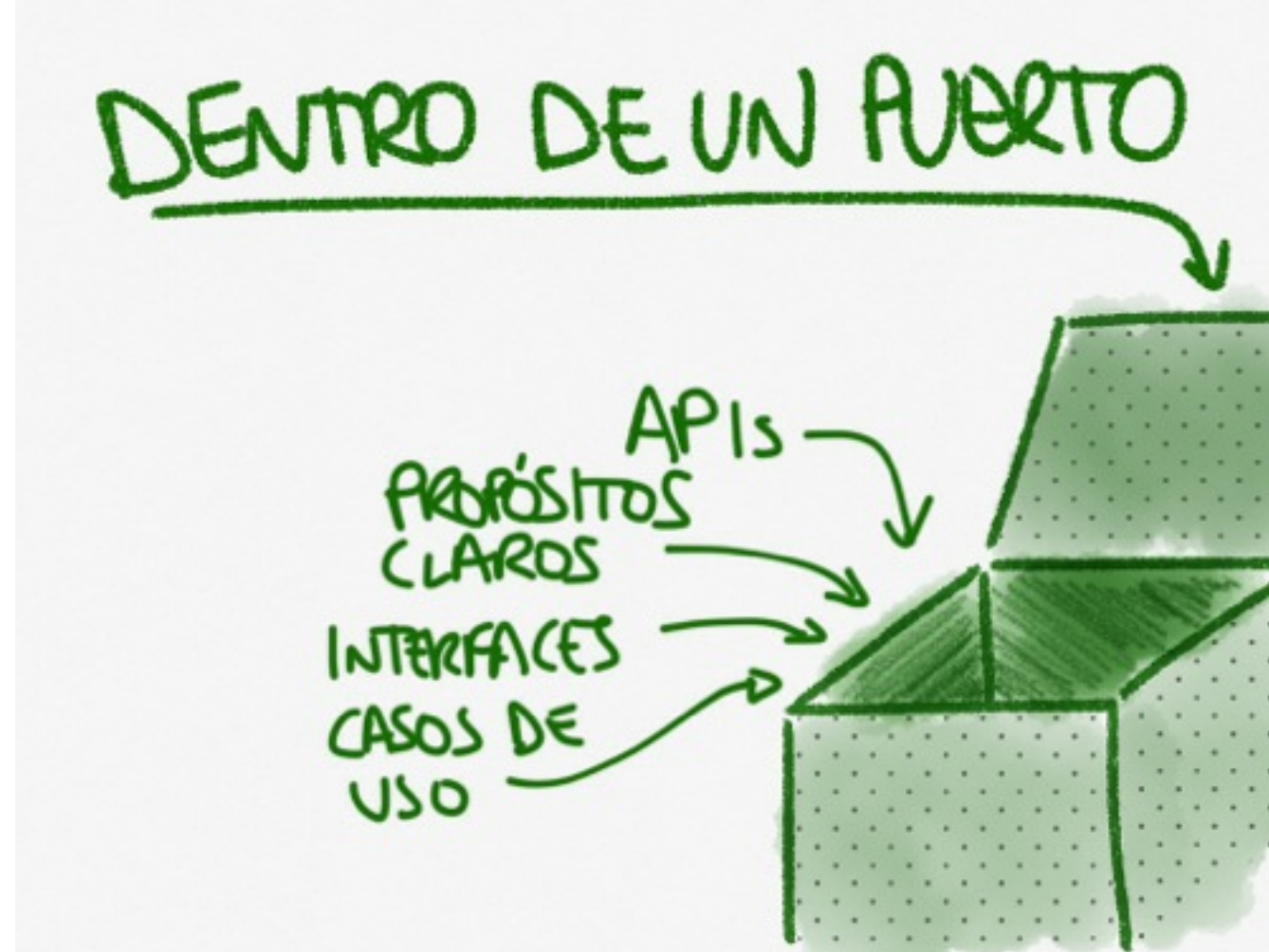


Estímulos del exterior, consultas de nuestra app o mensajes al exterior viajan a través de Puertos

Un Adaptador de una tecnología específica traduce lo que llega o se dirige a un puerto

El interior del hexágono no se filtra al exterior, ni viceversa

A.K.A: Un hexágono ignora los detalles de las tecnologías que le rodean y viceversa



Un Puerto representa un propósito u objetivo claro y acotado: Hacer consultas al repositorio de datos, enviar o recibir emails, ejecutar casos de uso, etc.

Los Puertos definen los APIs con los que se puede o se debe interactuar con una aplicación

Los Puertos se dividen en primarios (cuando permiten a terceros poner en marcha nuestra app) y secundarios (los que nuestra app usa para realizar consultas al exterior o enviar mensajes a terceros)