

API REST
HYPERMEDIA
BACKBONES
MARIONETTE

VS

HEXAGONAL
ARCHITECTURE
ANGULARJS

CHAN
CHAN

@artolamola



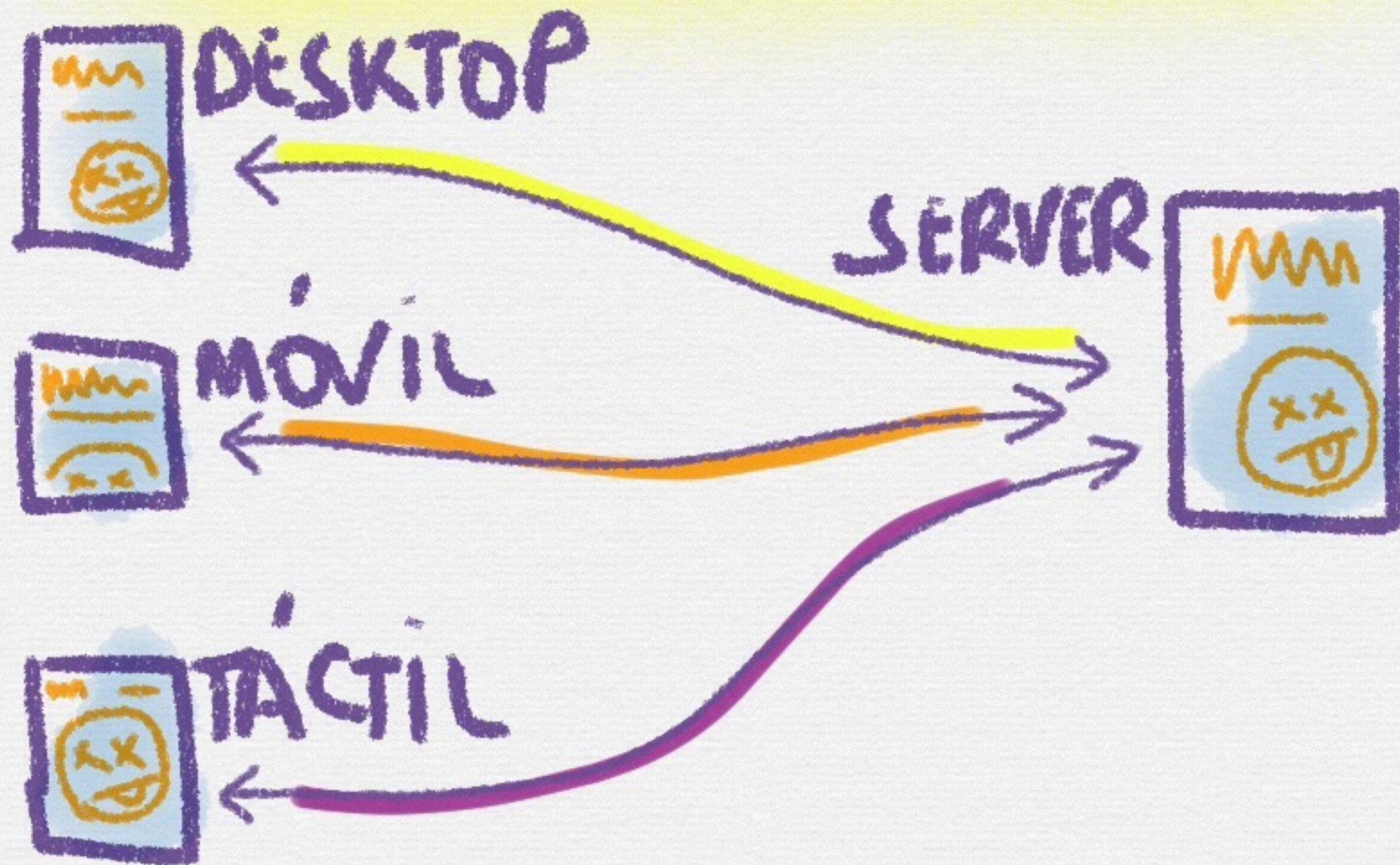
@ggalmator



BilboStack;

2014

¿UNA APP?



RESPONSIVE 1

1 HTML
1 LÓGICA DE NEGOCIO

DESDE EL MÓVIL
HACIA "ARRIBA" → LO CONTRARIO
A "GRACEFUL DEGRADATION"

2 THIN CLIENT / FAT SERVER

1 HTML x PLATAFORMA
1 LÓGICA DE NEGOCIO

¿PARA QUÉ AISLARSE? → HYPERMEDIA SERVER

FAT CLIENT / THIN SERVER 3

REST API
↓
DATA CENTRIC

1 HTML + LÓGICA x PLATAFORMA
1 PROYECTO PARAMETRIZADO
PARA CADA PLATAFORMA

GRANT, BOWEN

OPCIONES

Angular JS

BackboneJS

Data-centric

Hypermedia

Hexagonal

Acoplado

DATA CENTRIC

ACOPUADO A TABLAS

↓
"STANDARD"

↓
SENCILLO

BAJA UX

↳ TIENES QUE CONOCER
LAS URLS...

HYPERMEDIA

ALTA UX

- FORMS YA RELLENOS
- JSONES VACIOS
- URI TEMPLATES
- ...

↓
COMPLEJO

↓
NO
STANDARD

↳ HAL, SIREN

→ EL SERVER DECIDE
ESTRUCTURAS DE
DATOS




Principios de diseño de APIs REST

<http://leanpub/u/ear>

ALTERNATIVA

PVT /rest /compra

NO ES UNA
TABLA



ANGULAR

LIBERTAD

BINDEO BIDIRECCIONAL

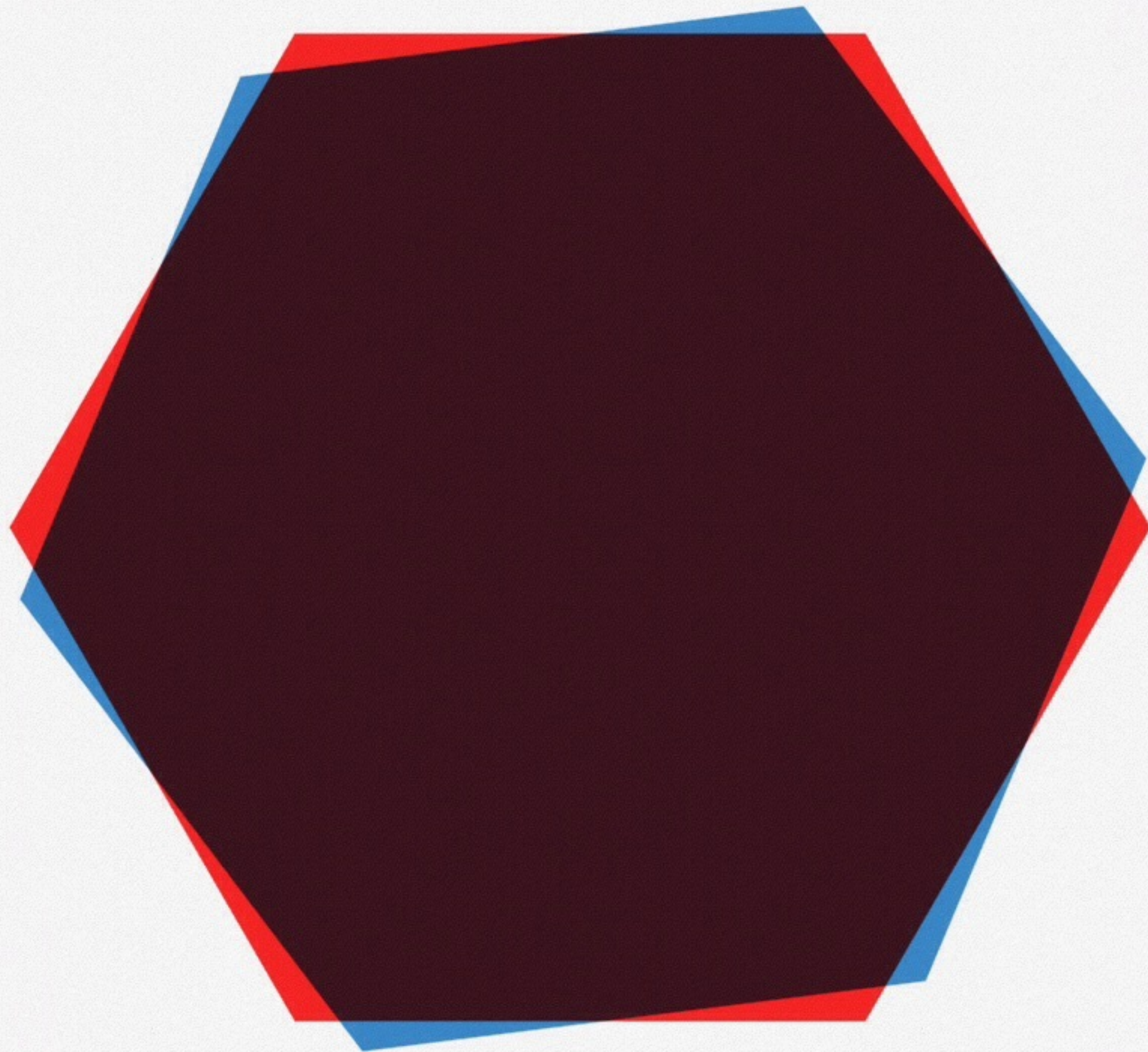
ECOSISTEMA
(KARMA.TOOLS,...)

BACKBONE

MODELS
COLLECTIONS
ACTIVE RECORD

BINDEO DECLARATIVO

DOLOR

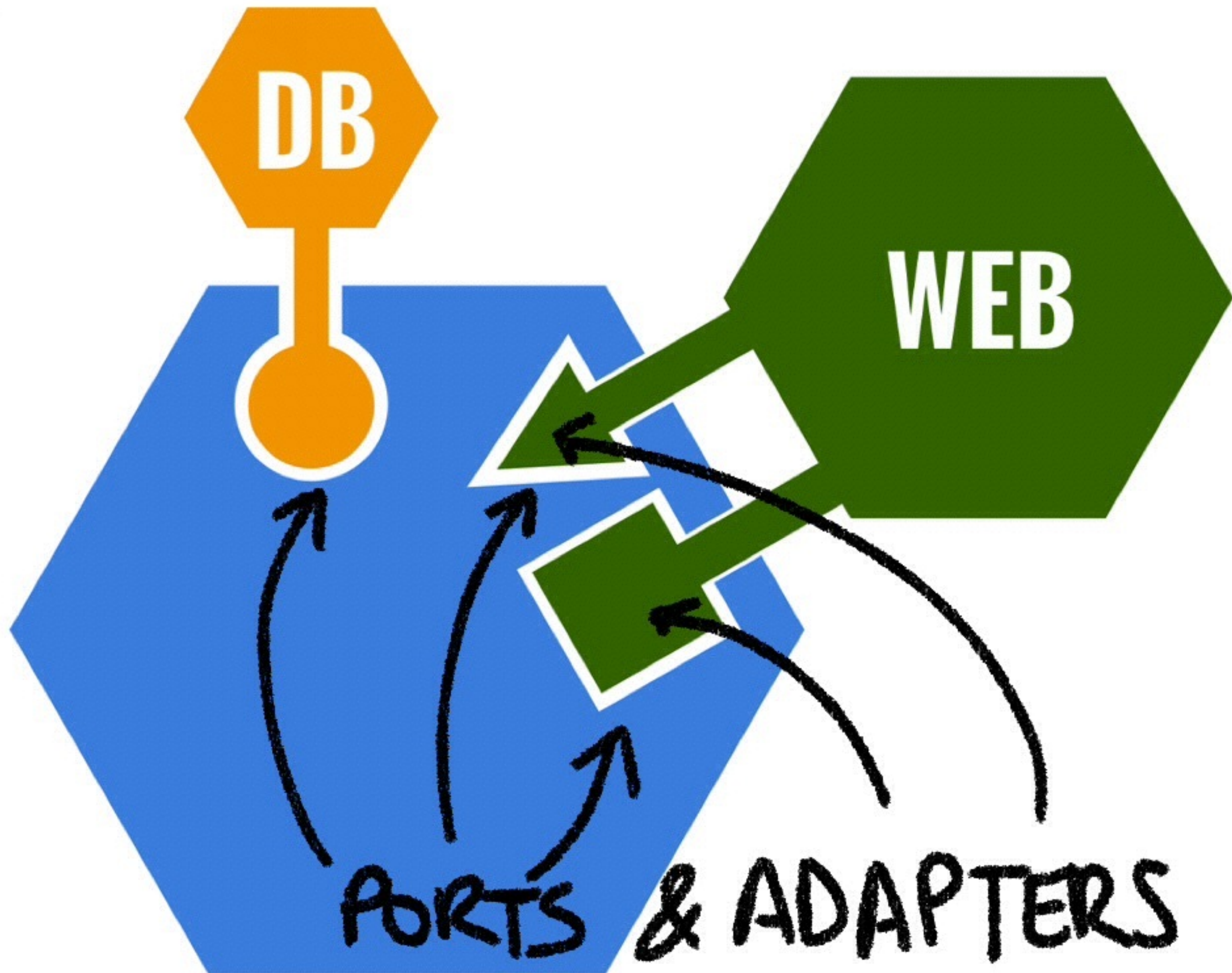


DESARROLLO SIN DEPENDENCIAS
DEL RUN-TIME

TESTEO DE LA APP AL MAYOR
NIVEL DE ABSTRACCIÓN

DIRIGIR LA APP DESDE
CUALQUIER CONTEXTO

IGNORAR TECNOLOGÍAS
DE TERCEROS



DENTRO DE UN PUERTO

APIs
PROPÓSITOS
CLAROS
INTERFACES
CASOS DE
USO

