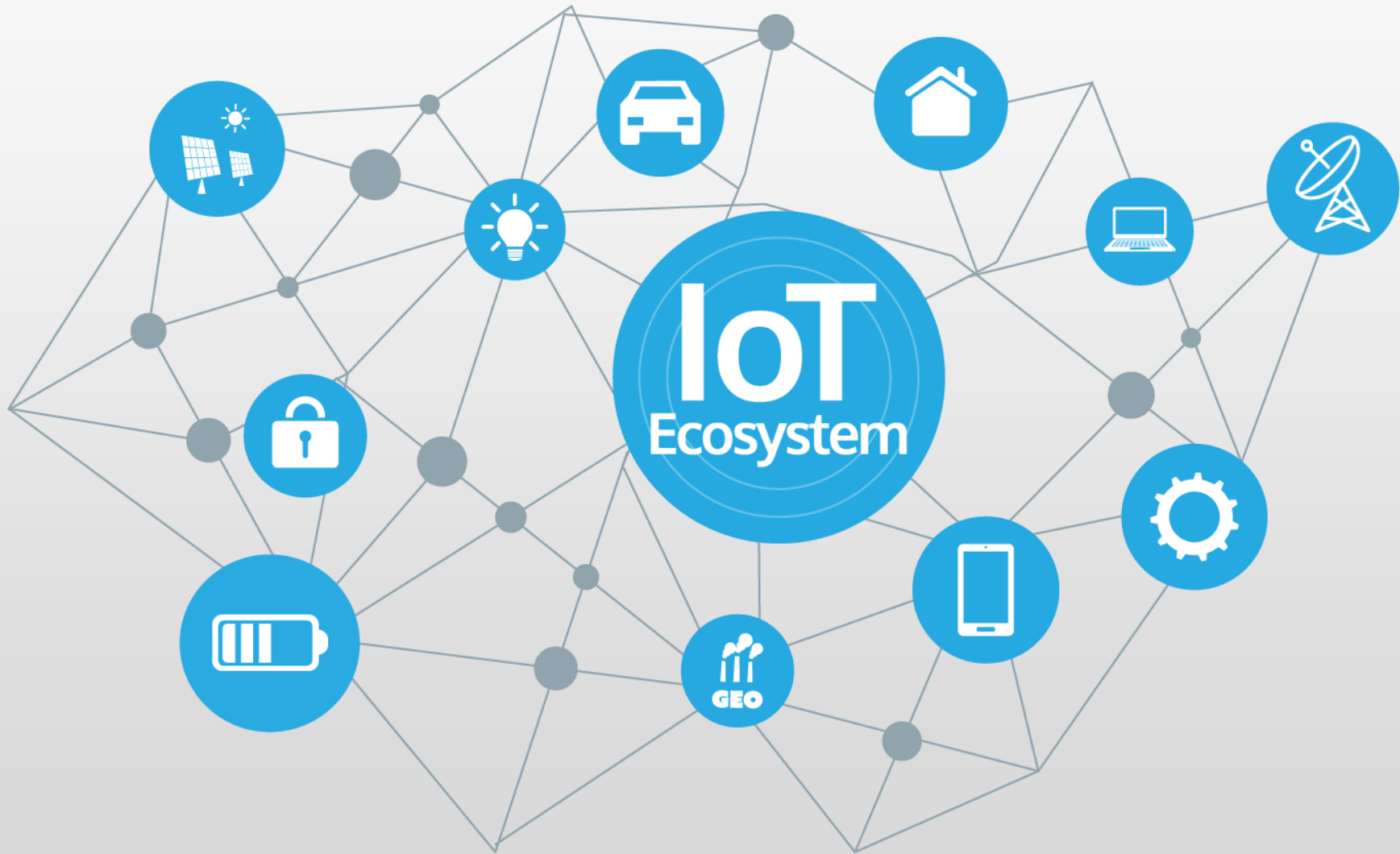


Social Access Controller

Daniel Salgado Población

Internet of Things (IoT)



Web Thing (WT)

.Objeto Físico

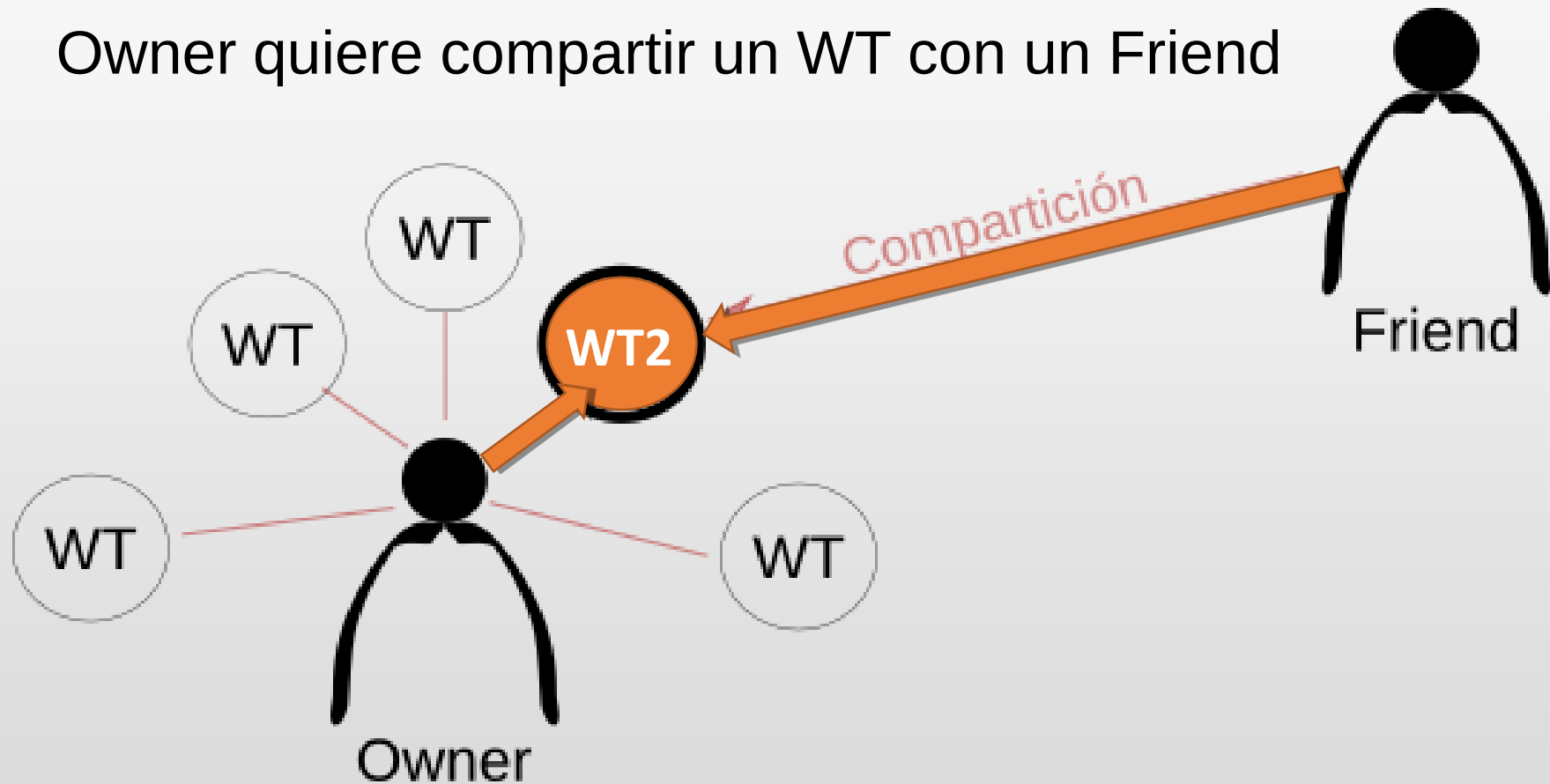


.Representación Virtual

```
{  
  "id": 1,  
  "name": "cafetera plus R10",  
  "brand": "Acme",  
  "links": {  
    "actions": {  
      "link": "/actions",  
      "resources": {  
        "calentar_agua": {  
          "values": "0"  
        }  
      }  
    }  
  }  
}
```

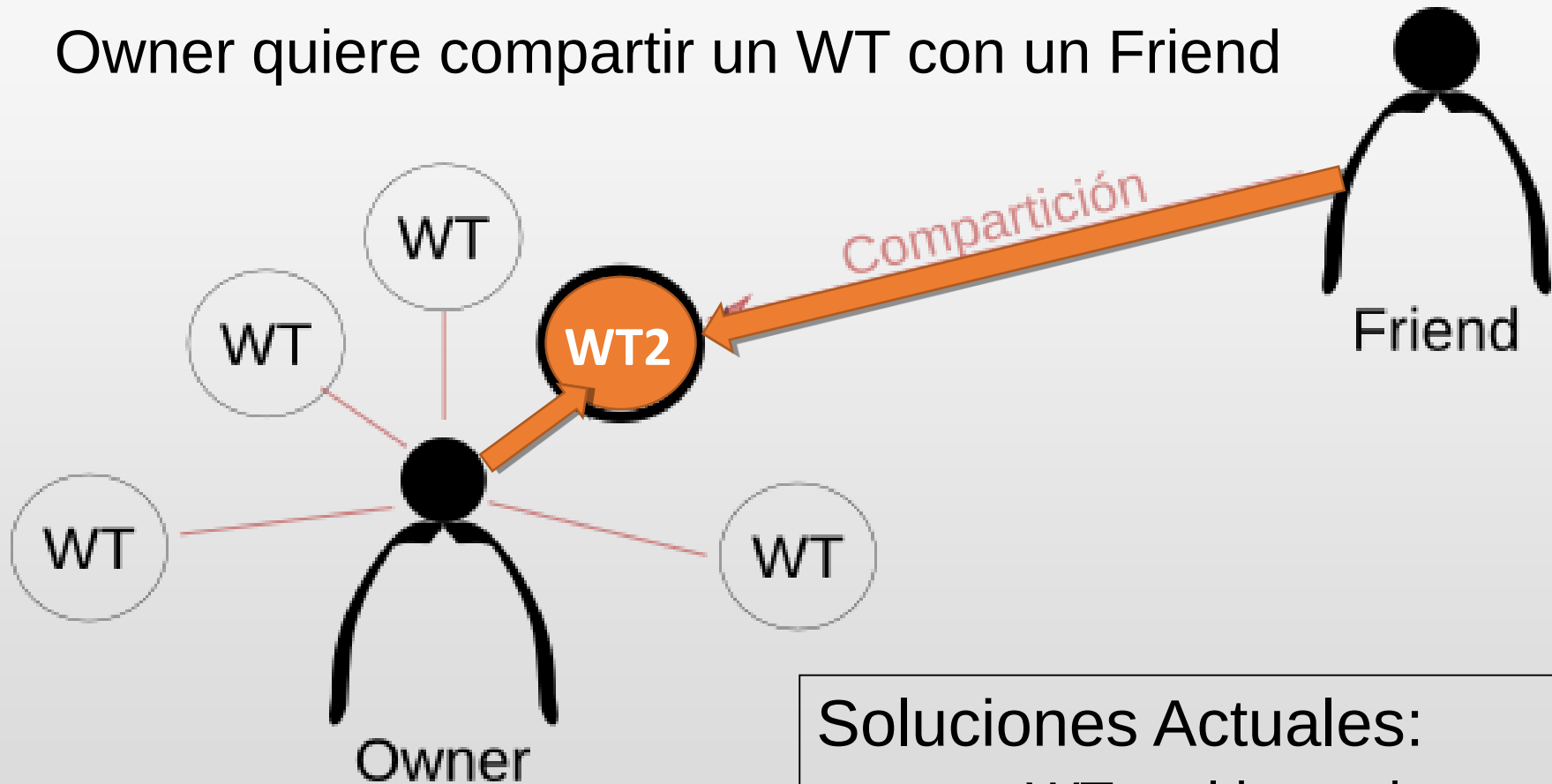
Problema a resolver

Owner quiere compartir un WT con un Friend



Motivación

Owner quiere compartir un WT con un Friend

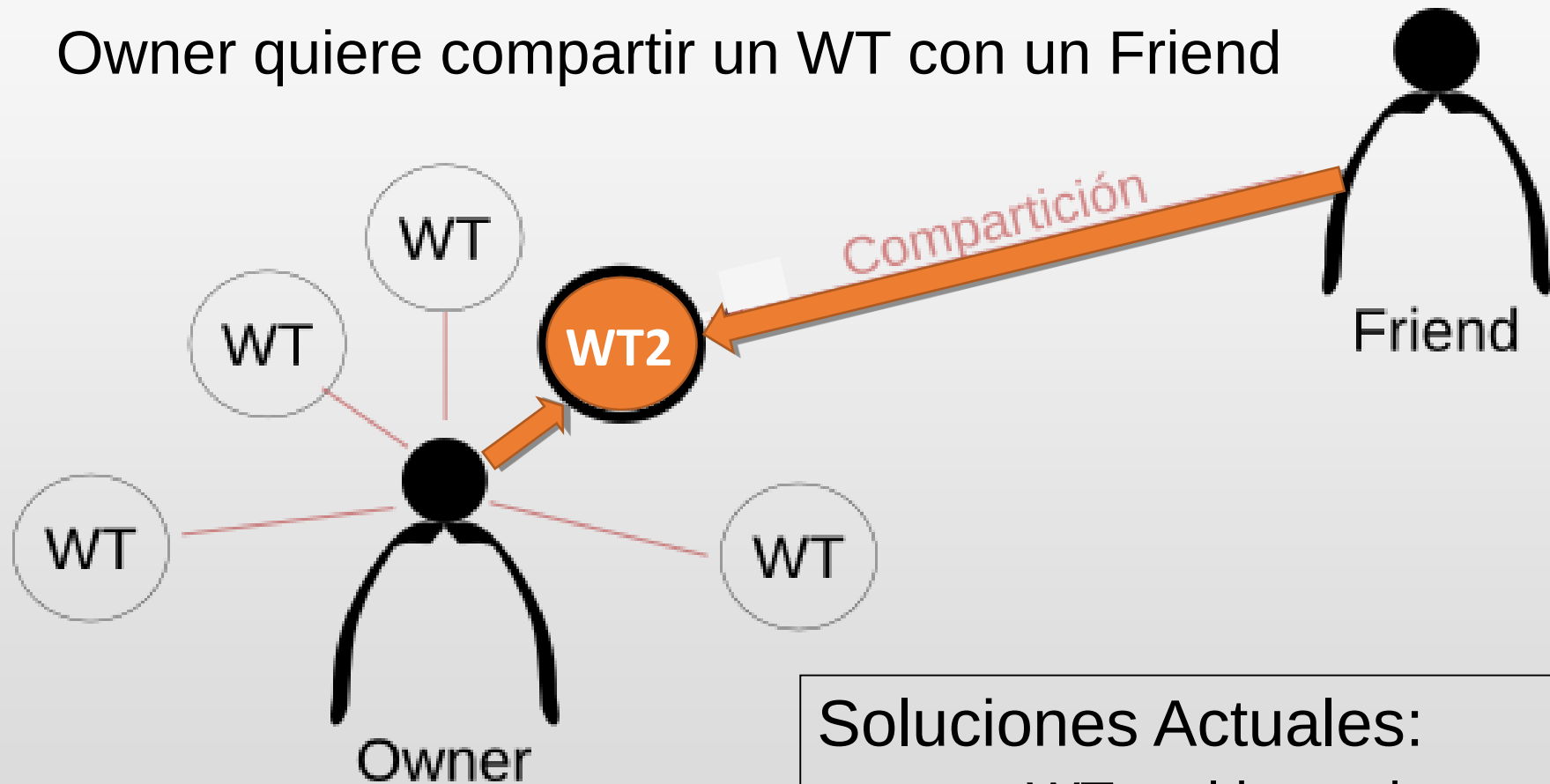


Soluciones Actuales:

- WT multiusuario
- Compartir Credenciales

Motivación

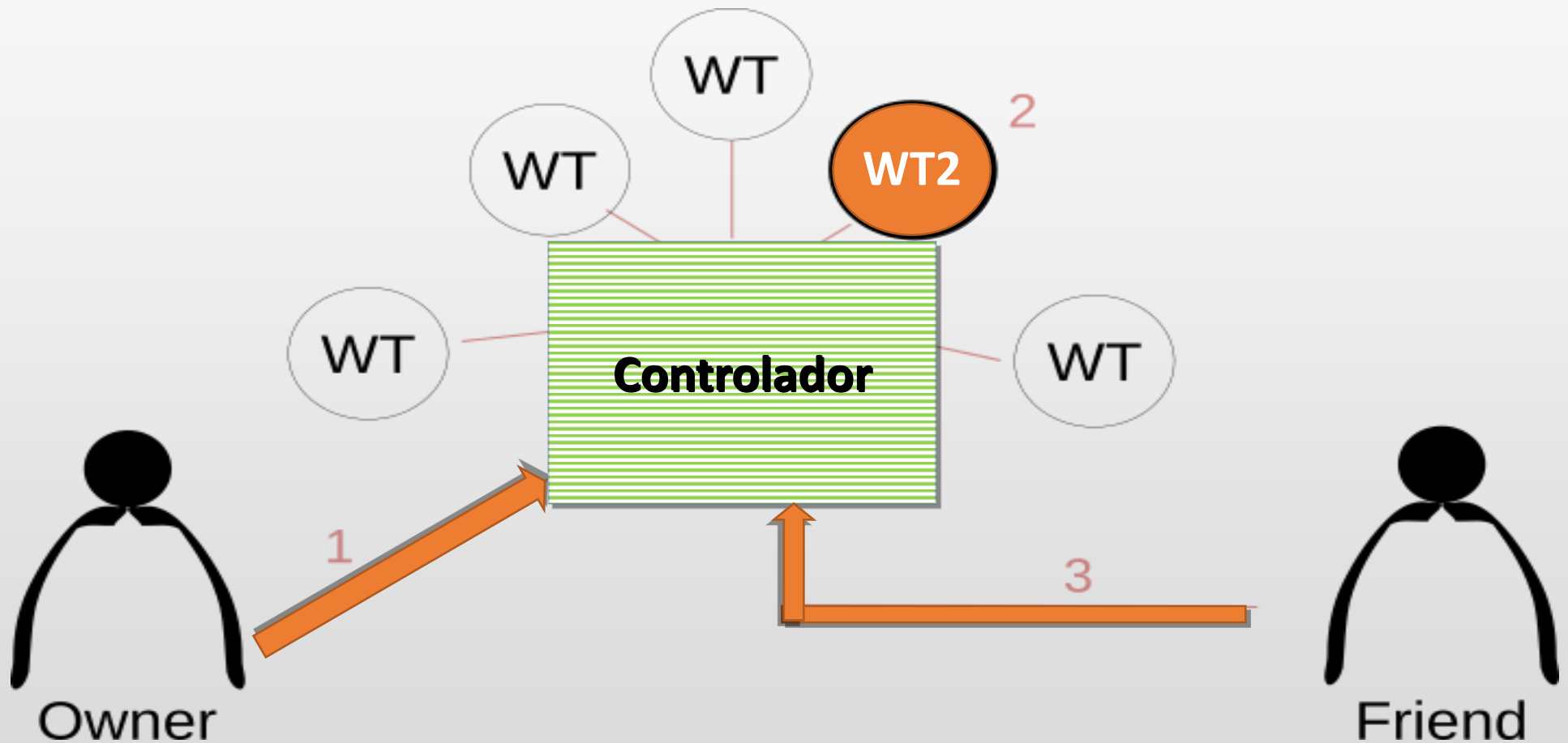
Owner quiere compartir un WT con un Friend



Soluciones Actuales:

- WT multiusuario
- Compartir Credenciales
- **Comparticion Atómica**

Esquema Solución General



Modelos

***WT**

***Controlador**

Modelo WT

[W3Consortium](#)

W3Consortium

Como deben ser los WT



Web Thing Model

W3C Member Submission 24 August 2015

A large blue circle with a black drop shadow, containing the letters "WT" in a bold, black, sans-serif font.

WT

Ajuste con modelo W3Consortium

Nivel DEBE

Definición de requisito	Nivel de cumplimiento
A WT MUST at least be an HTTP/1.1 server	No. Usamos un proxy para acceder a los WTs
A WT MUST have a root resource accesible via an HTTP URL	Sí
A WT MUST support GET, POST, PUT and DELETE HTTP verbs	Sí
A WT MUST implement HTTP status codes 200, 400, 500	Sí
A WT MUST support JSON as default representation	Sí
A WT MUST support GET on its root URL	Sí

Ajustamos a un 60% a Nivel DEBERÍA

Ajustamos al 0% en Nivel PODRÍA

Modelo Controlador

Dominique Guinard y otros

Como debe ser el Controlador

Guinard

Sharing Using Social Networks
in a Composable Web of Things

Dominique Guinard, Mathias Fischer, Vlad Trifa
Institute for Pervasive Computing, ETH Zurich
and SAP Research CEC Zurich
8092 Zurich, Switzerland
Contact Email: dguinard@ethz.ch

Controlador

Ajuste modelo Dominique Guinard y otros

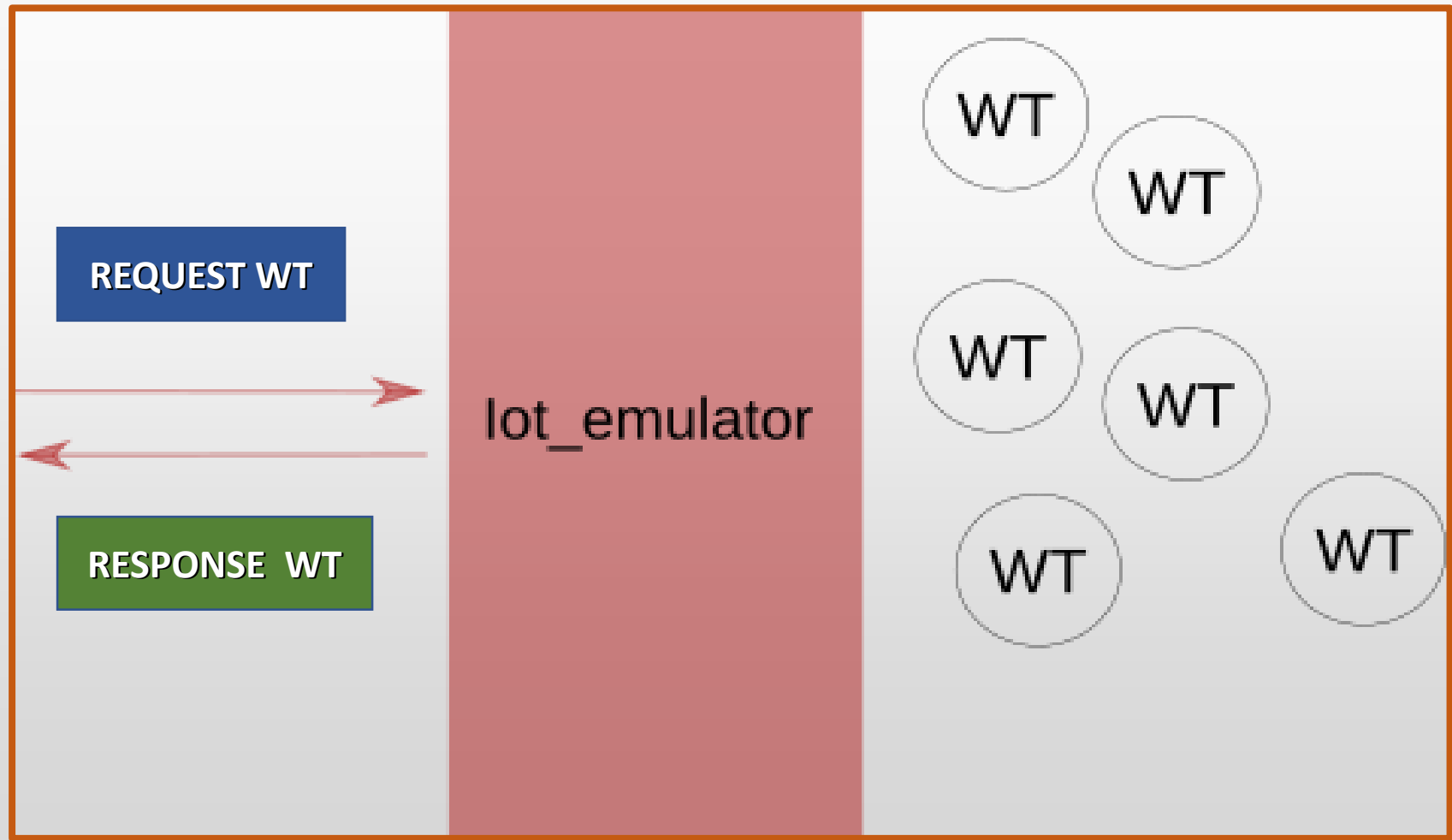
- Requisitos que **NO** hemos implementado:
 - Independencia de Red social usada
 - Descubrimiento de Wts automática
 - Discernir verbo HTTP compartido



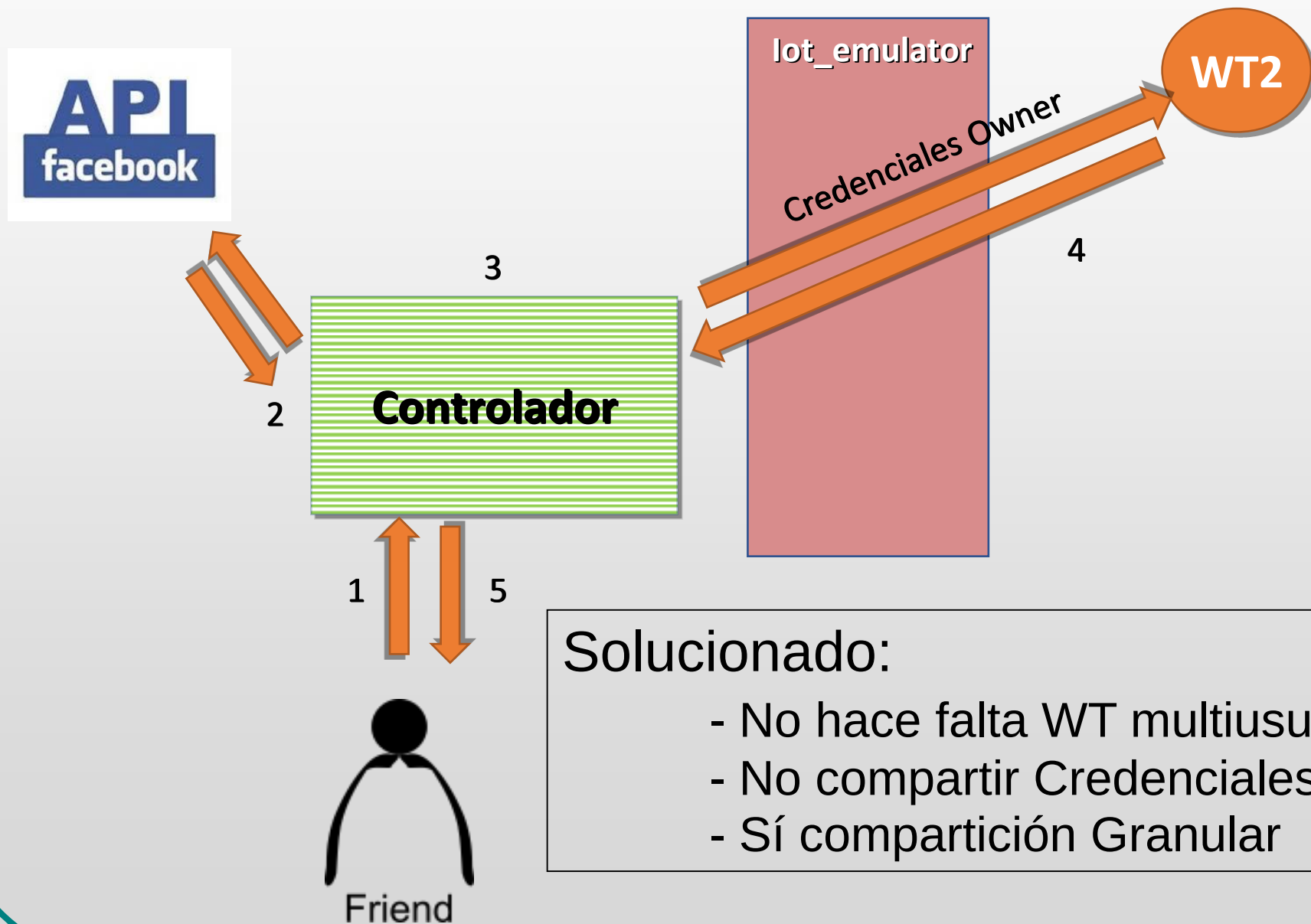
¿Cómo acceder a los WT?

Capa de Accesibilidad

lot_emulator



Esquema Solución General



Solucionado:

- No hace falta WT multiusuario
- No compartir Credenciales
- Sí compartición Granular

Simplificaciones

Un único Owner

Identificador de WT es el PK de la tabla

Action 

- Nombre del Action ES el nombre del property
- Ya que Action name es la union entre

Sac lot_Emulator

Teconlogías

Tecnologías Software Backend



Tecnologías Software Backend



Symfony 4



Tecnologías Software Frontend



Tecnologías Software Frontend



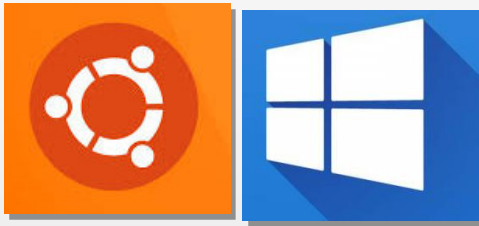
Dominio y Subdominio

Proyecto	Dominio
SAC	https://socialaccesscontroller.tk
lot_emulator	http://iot.socialaccesscontroller.tk

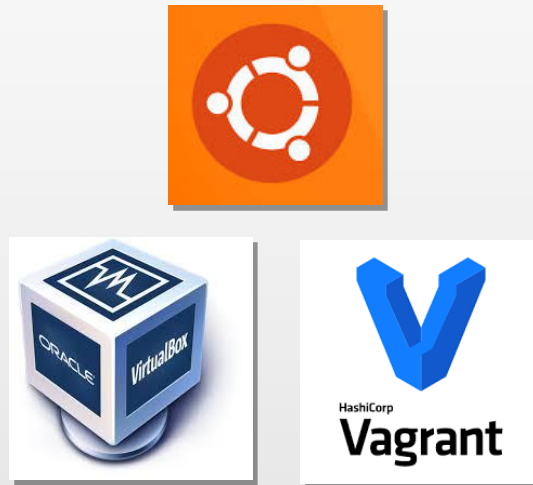


Entornos

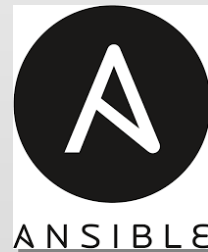
Local



Desarrollo



Producción



Gestores dependencias



Herramientas desarrollo



Arquitectura Común

SAC

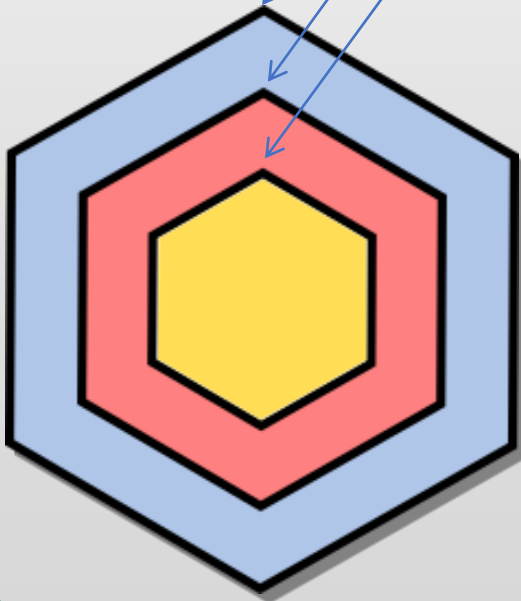
src/Domain
src/Aplication
src/Infrastructure

tests/

fixtures/

CI/

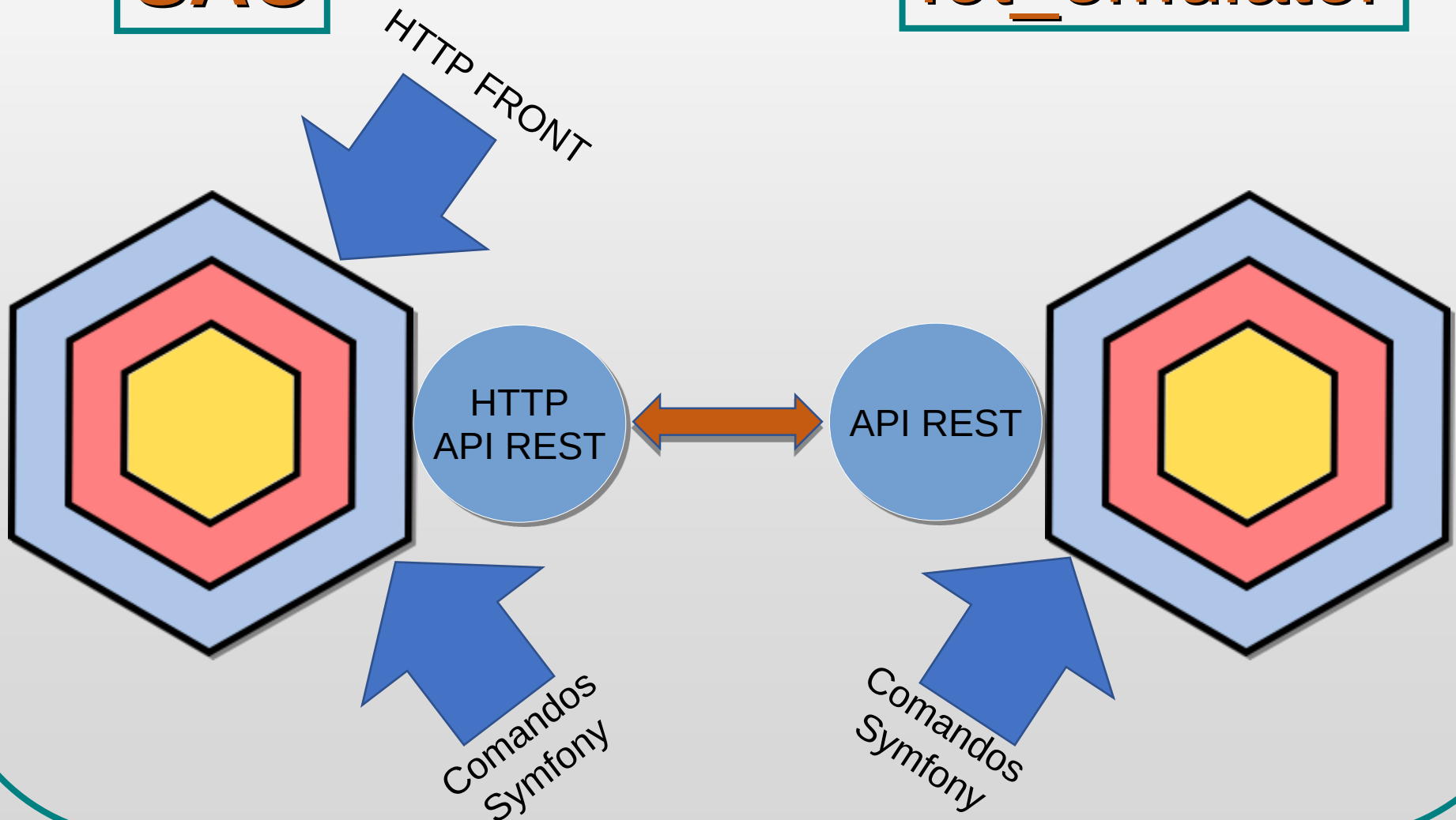
lot_emulator



Arquitectura Hexagonal y API

SAC

lot_emulator



SAC

Dominio SAC

Entidades

```
src/Domain/Entity/Action.php  
src/Domain/Entity/Friend.php  
src/Domain/Entity/Owner.php  
src/Domain/Entity/Thing.php
```

Repositorios

```
src/Domain/Repository/OwnerRepository.php  
src/Domain/Repository/ThingConnectedRepository.php  
src/Domain/Repository/ThingRepository.php  
src/Domain/Repository/ActionRepository.php  
src/Domain/Repository/FriendRepository.php
```

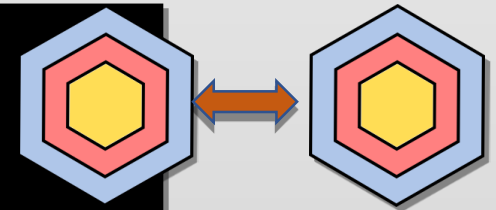
Dominio SAC

Entidades

```
src/Domain/Entity/Action.php  
src/Domain/Entity/Friend.php  
src/Domain/Entity/Owner.php  
src/Domain/Entity/Thing.php
```

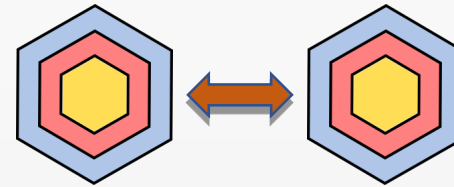
Repositorios

```
src/Domain/Repository/OwnerRepository.php  
src/Domain/Repository/ThingConnectedRepository.php  
src/Domain/Repository/ThingRepository.php  
src/Domain/Repository/ActionRepository.php  
src/Domain/Repository/FriendRepository.php
```



Aplicación SAC

Ejemplo del CommandHandler relacionado con



```
src/Application/CommandHandler/Thing/ThingConnected/GetThingConnectedCompleteHandler.php
```

Ejecución de comando Symfony

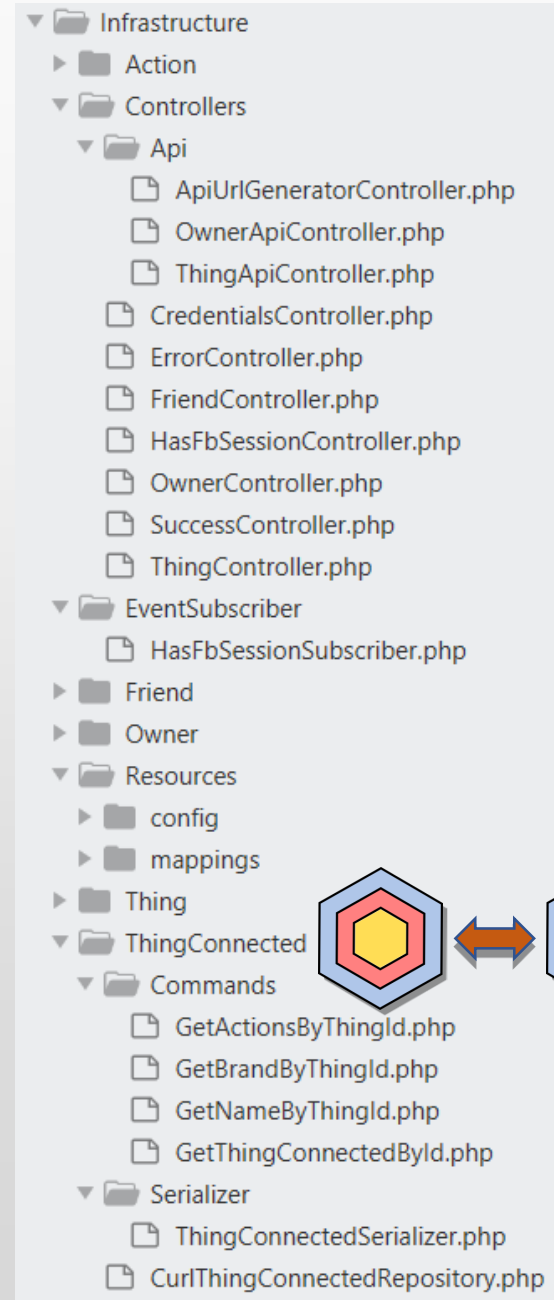
```
ubuntu@ip-172-31-45-14:/var/www/sac$ php bin/console app:ThingConnected:GetThingConnectedCompleteById 1
array:3 [
    "message" => ""
    "status" => true
    "data" => {#357
        +"id": 1
        +"name": "thing_name1"
        +"brand": "thing_brand1"
        +"links": {#351
            +"actions": {#353
                +"link": "/actions"
                +"resources": {#329
                    +"action_name1": {#365
                        +"values": "property_value1"
                    }
                }
            }
        }
    }
]
```

Infraestructura SAC

- Controladores
- Event Subscriber
- Resources

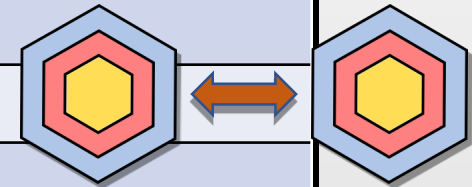


- Comandos Symfony
- Serializadores
- Repositorios

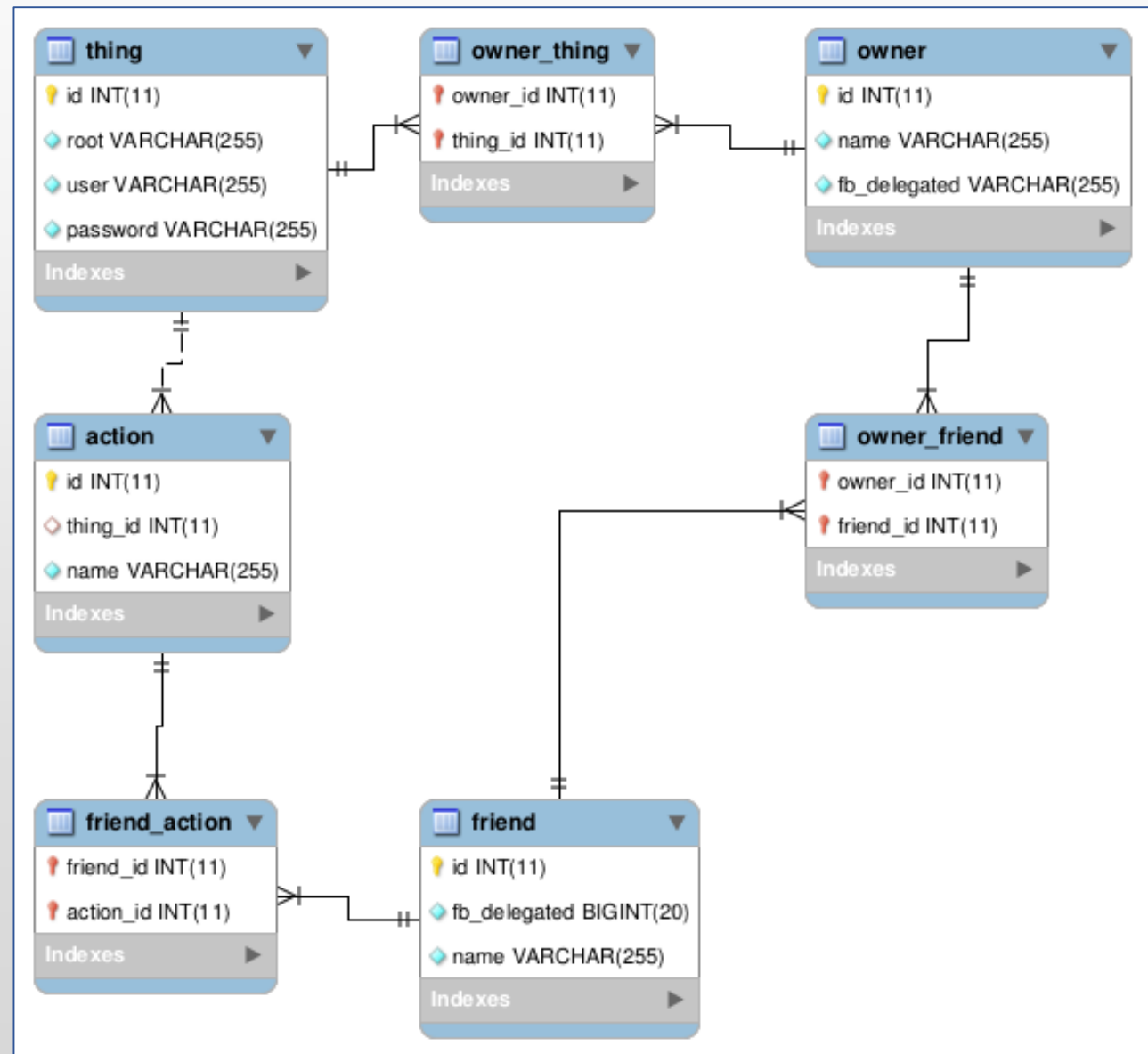


REST API SAC

Verbo HTTP	Endpoint
GET	/api/owner
GET	/api/thing/{thingId}
GET	/api/url/provider/thing
GET	/api/url/provider/api/thing
GET	/api/url/provider/api/share/action



Esquema Base de Datos SAC

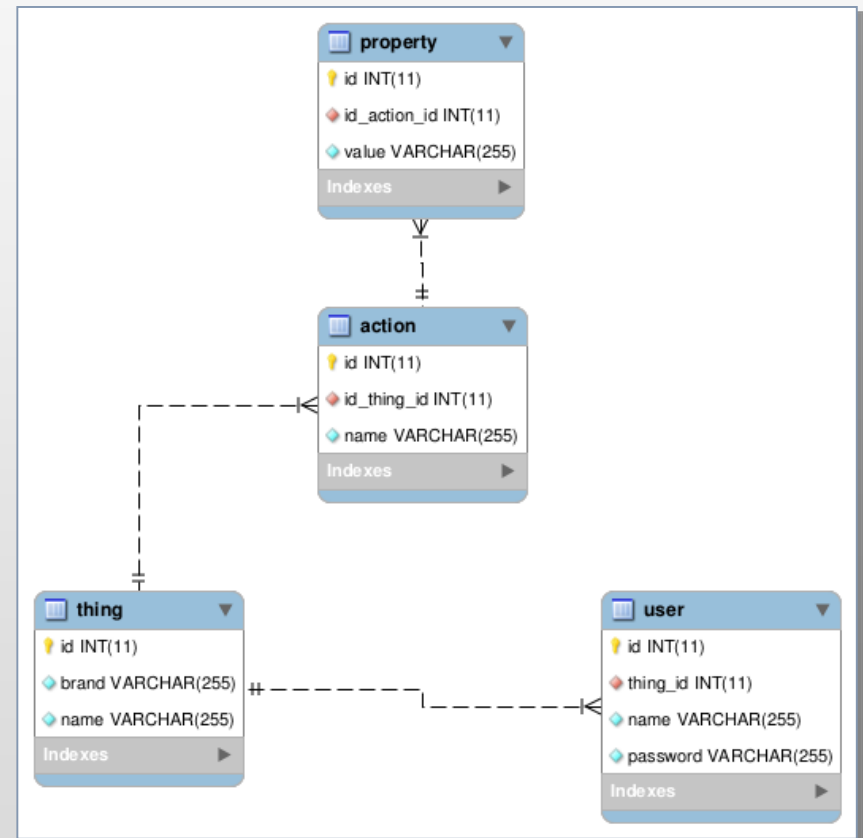


lot_emulator

Dominio lot_emulator y esquema de Base de Datos

Existe un Repositorio por cada Entidad

```
src/Domain/Entity/Action.php  
src/Domain/Entity/Property.php  
src/Domain/Entity/Thing.php  
src/Domain/Entity/User.php
```



Comparación de Dominios

lot_emulator

```
src/Domain/Entity/Action.php  
src/Domain/Entity/Property.php  
src/Domain/Entity/Thing.php  
src/Domain/Entity/User.php
```

SAC

```
src/Domain/Entity/Action.php  
src/Domain/Entity/Friend.php  
src/Domain/Entity/Owner.php  
src/Domain/Entity/Thing.php
```

Comparación de Dominios

lot_emulator

```
src/Domain/Entity/Action.php  
src/Domain/Entity/Property.php  
src/Domain/Entity/Thing.php  
src/Domain/Entity/User.php
```

SAC

```
src/Domain/Entity/Action.php  
src/Domain/Entity/Friend.php  
src/Domain/Entity/Owner.php  
src/Domain/Entity/Thing.php
```

Aplicación lot_emulator

Existe un Comando por cada CommandHandler

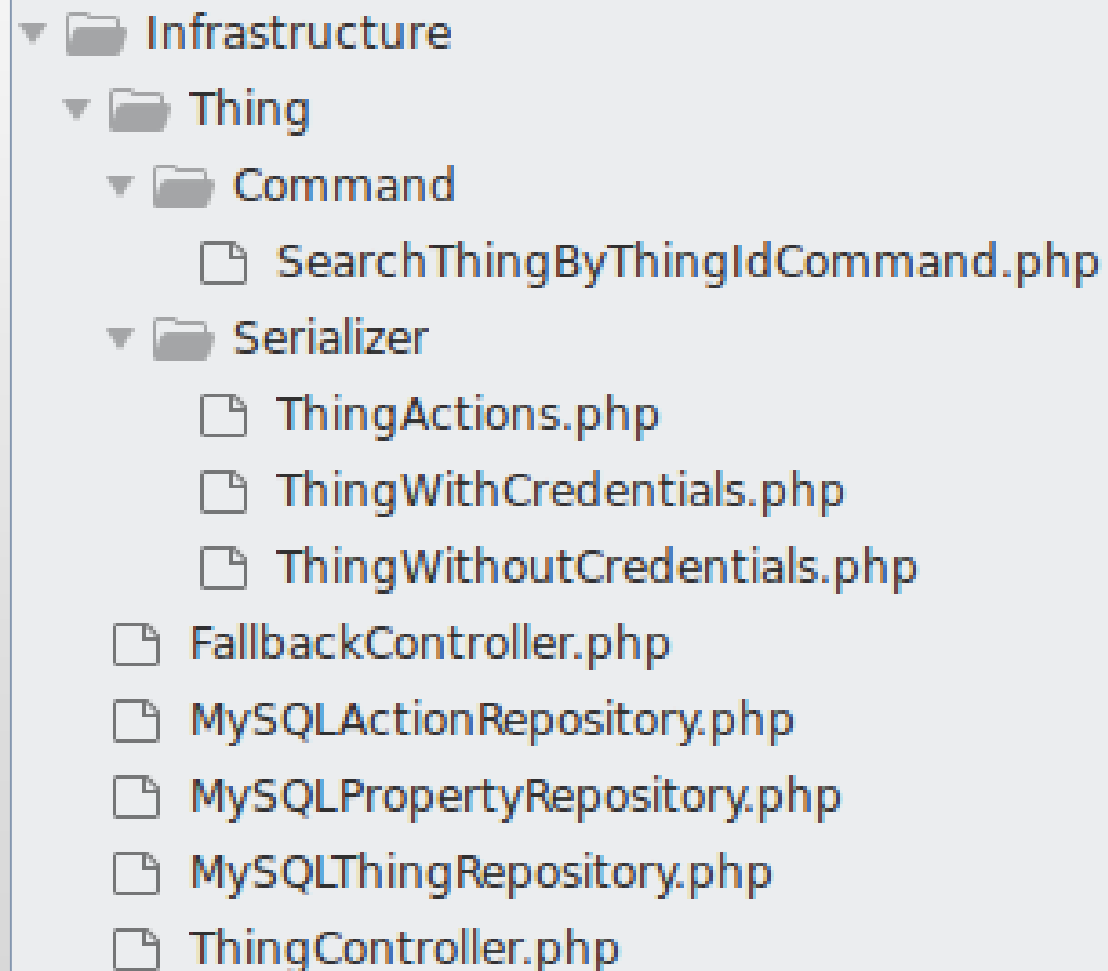
```
src/Application/CommandHandler/Thing/CreateThingHandler.php  
src/Application/CommandHandler/Thing/ExecuteActionHandler.php  
src/Application/CommandHandler/Thing/SearchThingByIdHandler.php  
src/Application/CommandHandler/Thing/SearchThingByIdWithoutCredentialsHandler.php
```

DTO para ayudar con las credenciales



```
src/Application/Dto/UserCredentialsDto.php
```


Infraestructura lot_emulator



```
▼ Infrastructure
  ▼ Thing
    ▼ Command
      SearchThingByThingIdCommand.php
    ▼ Serializer
      ThingActions.php
      ThingWithCredentials.php
      ThingWithoutCredentials.php
  FallbackController.php
  MySQLActionRepository.php
  MySQLPropertyRepository.php
  MySQLThingRepository.php
  ThingController.php
```

The image shows a file explorer view of a project structure. The root directory is 'Infraestructura'. Inside it, there is a 'Thing' directory. The 'Thing' directory contains two sub-directories: 'Command' and 'Serializer'. The 'Command' directory contains a single file 'SearchThingByThingIdCommand.php'. The 'Serializer' directory contains three files: 'ThingActions.php', 'ThingWithCredentials.php', and 'ThingWithoutCredentials.php'. Outside the 'Thing' directory, there are five more files: 'FallbackController.php', 'MySQLActionRepository.php', 'MySQLPropertyRepository.php', 'MySQLThingRepository.php', and 'ThingController.php'.

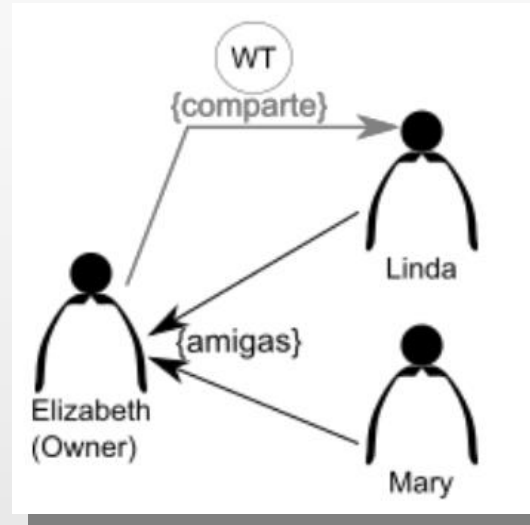
Rest lot_emulator

Verbo HTTP	Endpoint
GET	/
GET	/ {id} (Sin credenciales)
GET	/ {id} (Con credenciales)
POST	/create
POST	/ {id}/actions/ {action_name} 
GET	/ {id}/properties/ {property_name}
GET	/ {id}/actions

**Datos de Pruebas
(mocks)**

Explicación datos de prueba

Facebook



lot_emulator



```
{"name":"thing_name1","brand":"thing_brand1","links":{"actions":["action_name1"],"properties":[{"action_name1":"property_value1"}]}}  
{"name":"thing_name2","brand":"thing_brand2","links":{"actions":["action_name1","action_name2"],"properties":[{"action_name1":"property_value1"}, {"action_name2":"property_value2"}]}}  
{"name":"thing_name3","brand":"thing_brand3","links":{"actions":["action_name1","action_name2","action_name3"],"properties":[{"action_name1":"property_value1"}, {"action_name2":"property_value2"}, {"action_name3":"property_value3"}]}}
```

SAC



php bin/console doctrine:fixture:load

CONCLUSIONES

El SAC aporta estas ventajas:

- Manera segura de compartir WT
- Usar estructura social de Redes Sociales
- Compartir granularmente

MEJORAS SAC

- Multi-owner
- Cacheado
- Actualización de Friends
- Descubrimiento de WTs

MEJORAS lot_emulator

- Actualización propuesta 2019
- Action no acoplada a Property
- NoSQL para estructuras más flexibles

