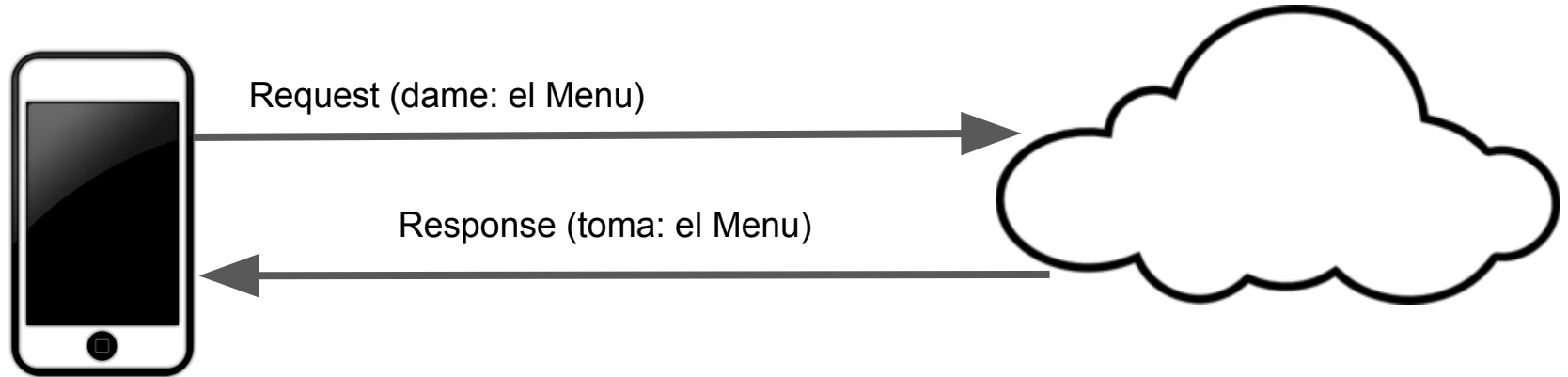
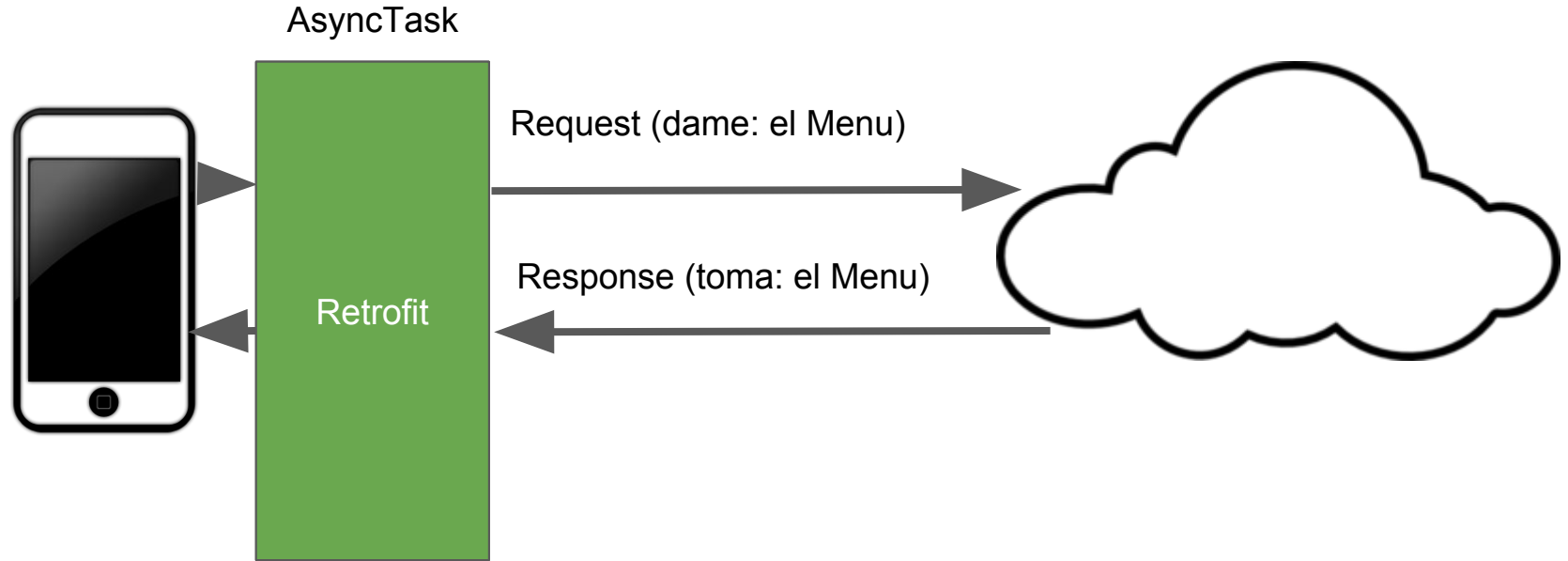


**RETROFIT**

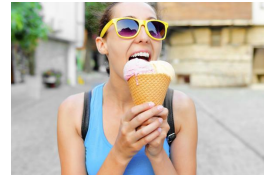
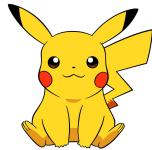
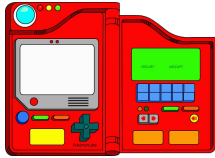
# COMO HACER UN PEDIDO A LA NUBE



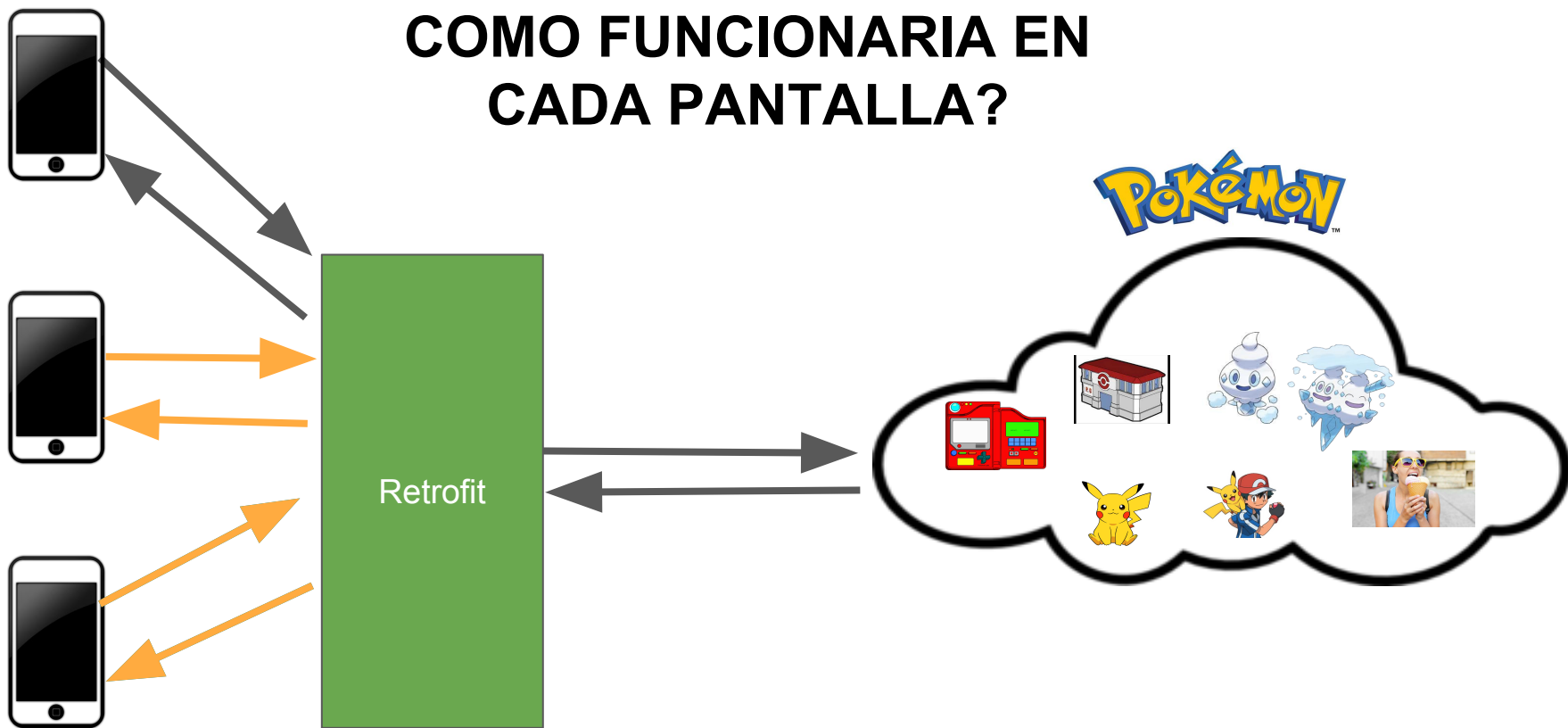
# COMO HACER UN PEDIDO A LA NUBE



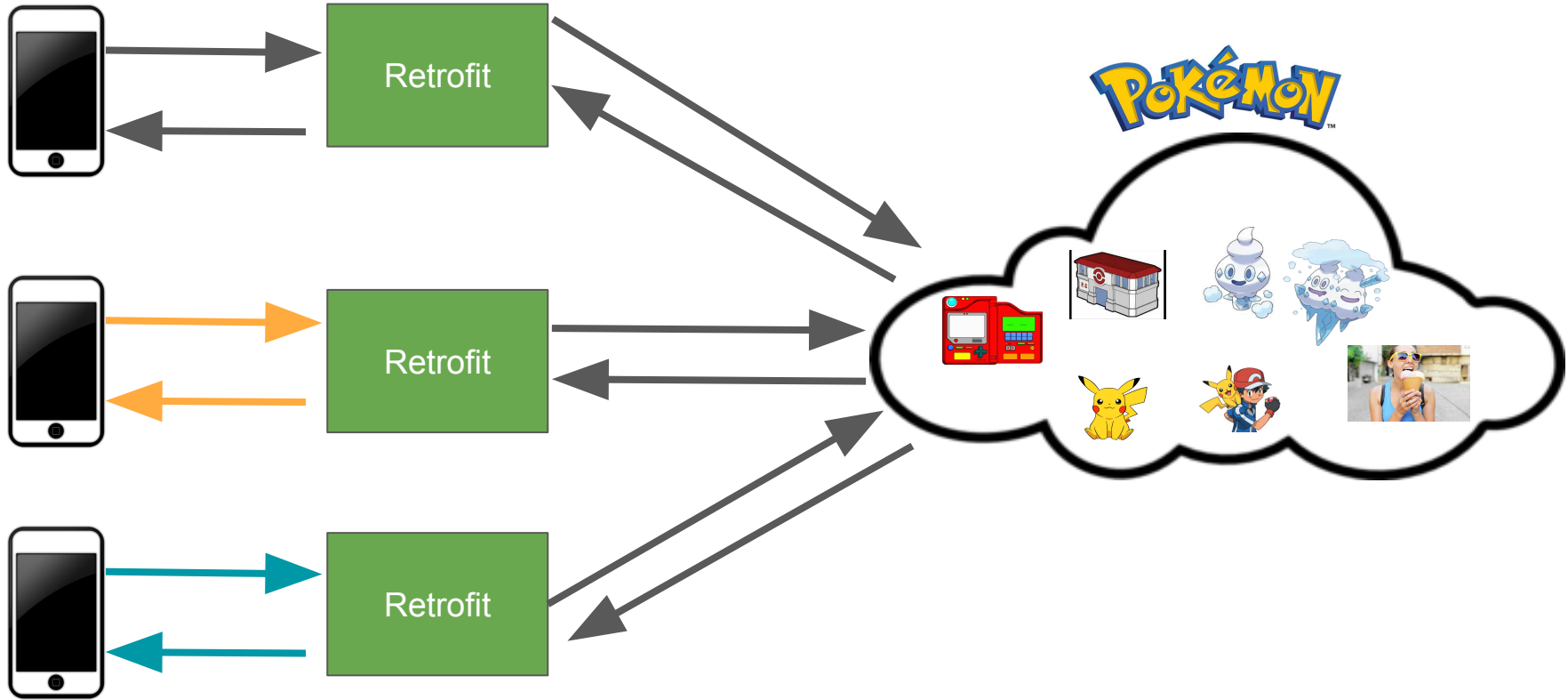
# POKÉMON™



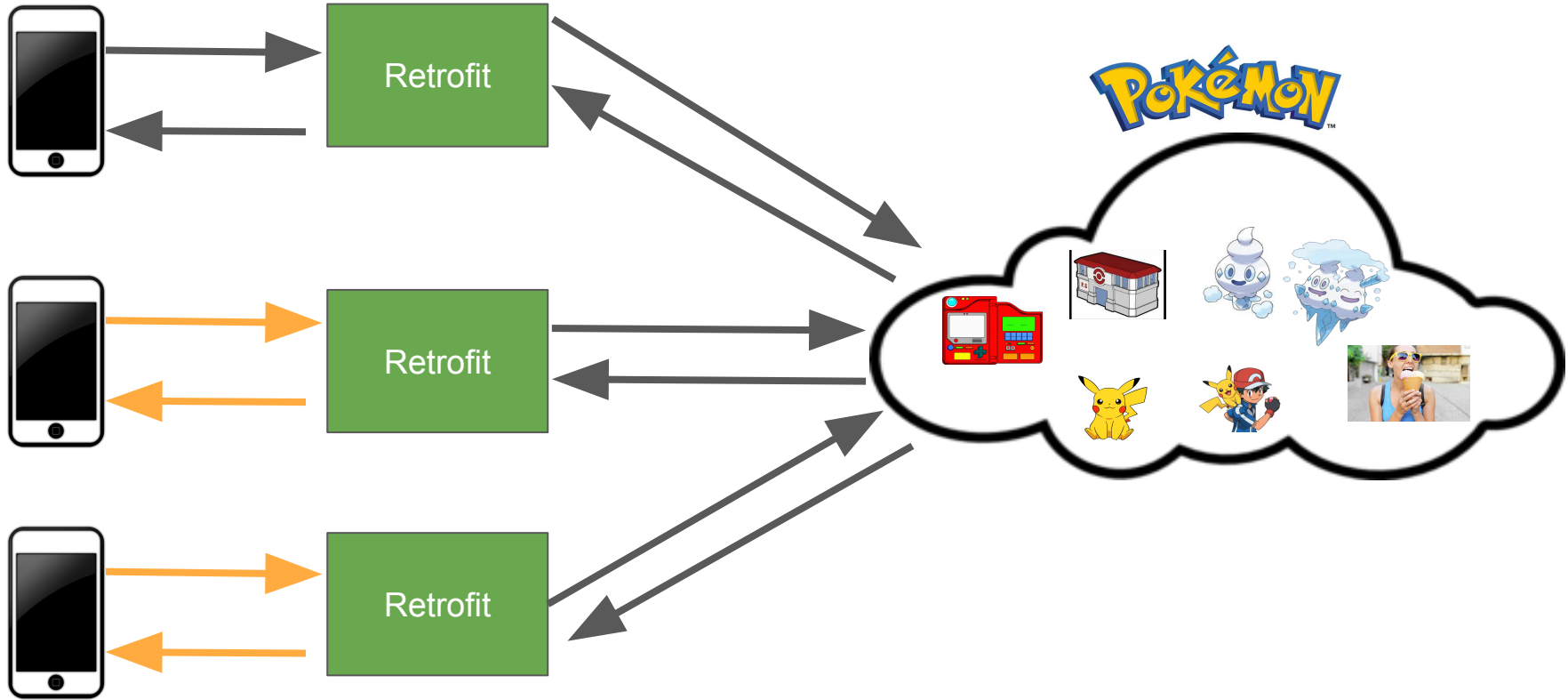
# COMO FUNCIONARIA EN CADA PANTALLA?



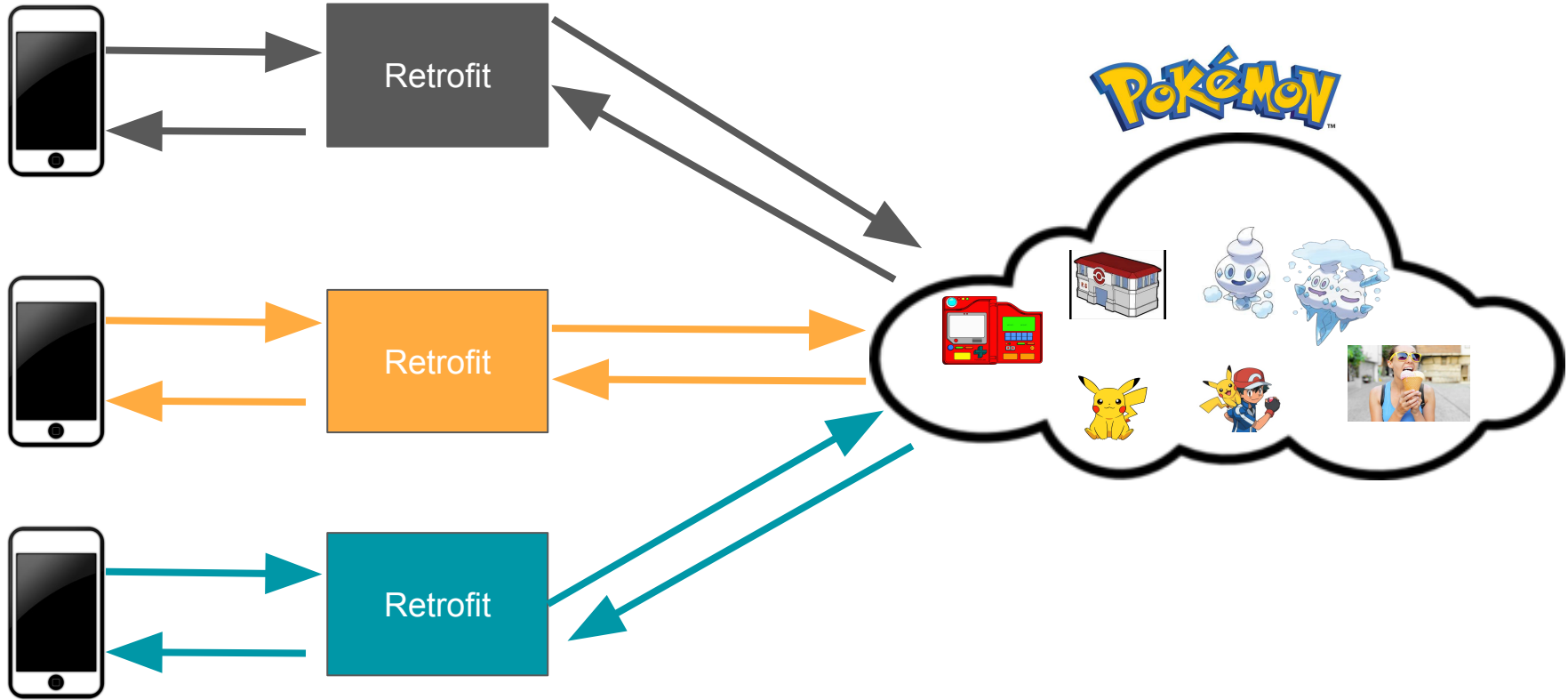
# 1\* INTENTO DE ARREGLAR EL PROBLEMA



# 1\* INTENTO DE ARREGLAR EL PROBLEMA

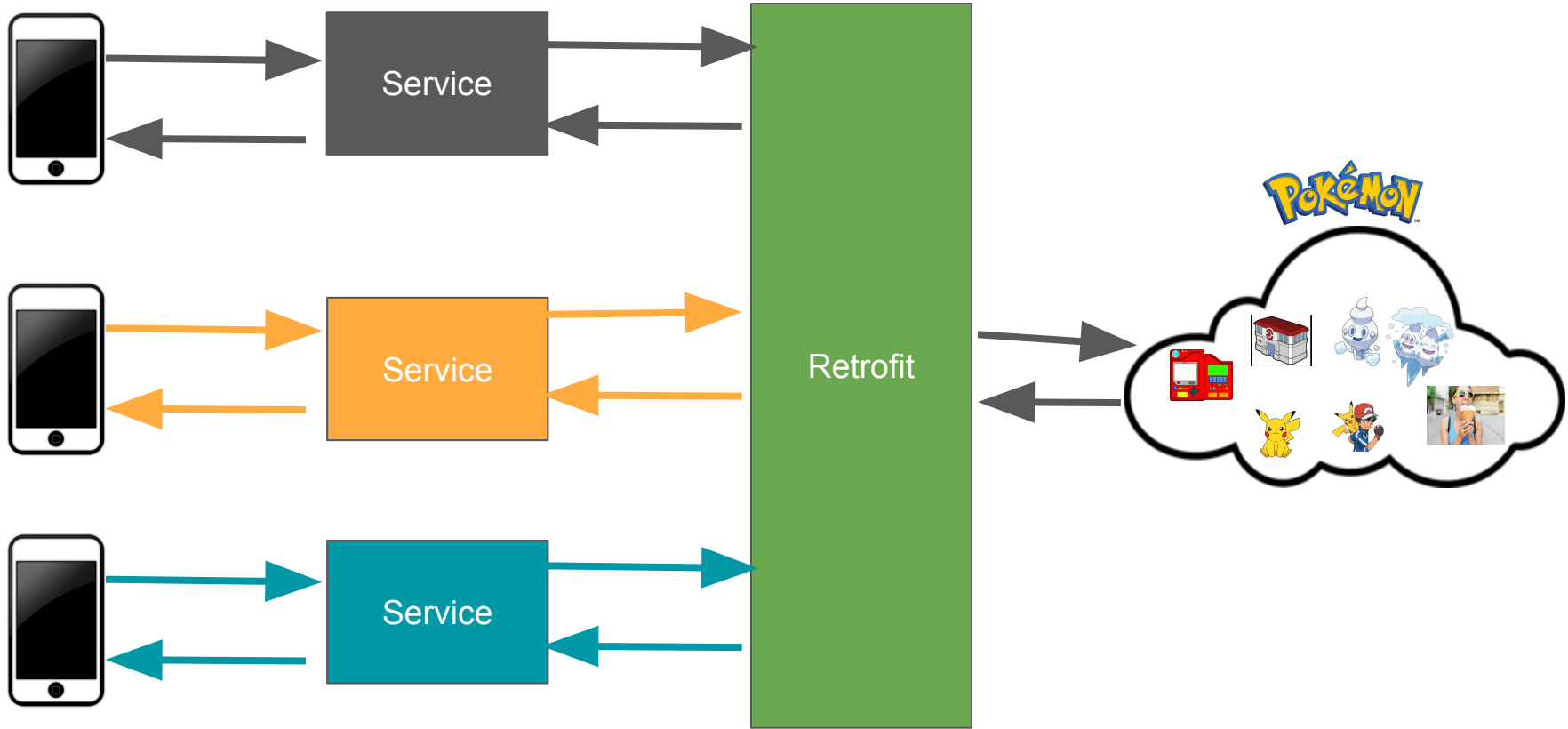


## 2\* INTENTO ACEPTABLE





### 3\* INTENTO MEJORADO ACEPTABLE

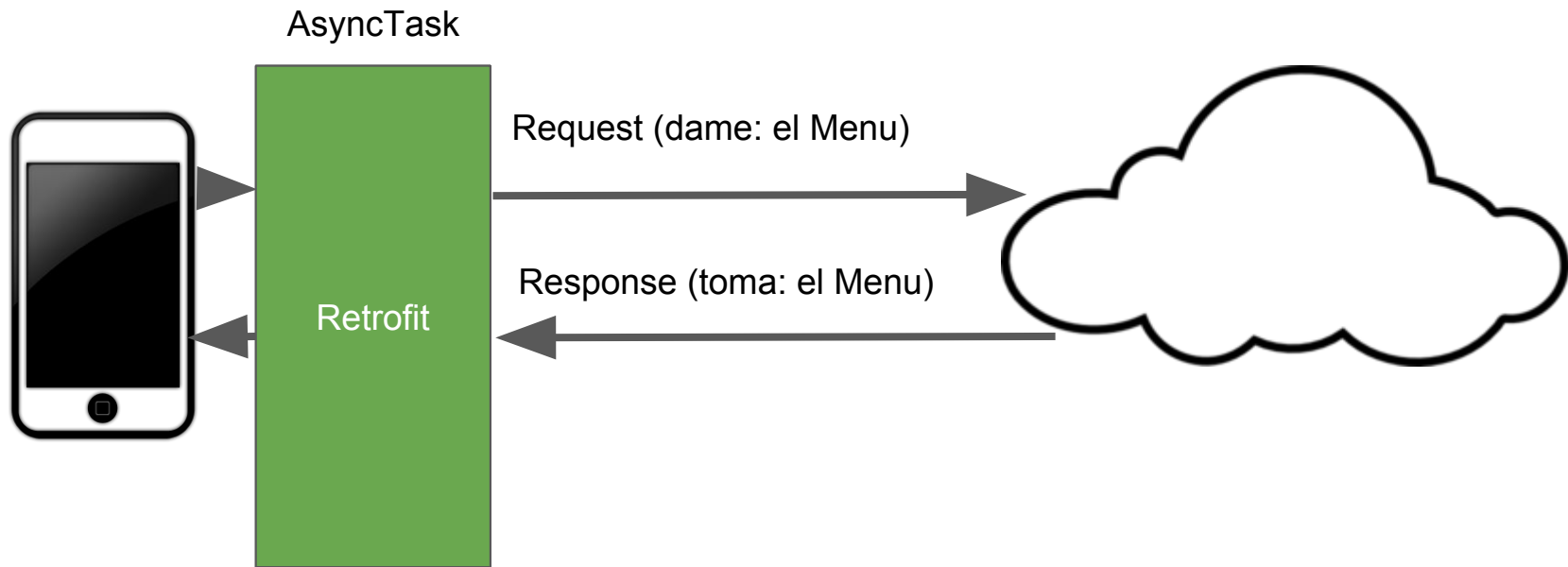


## Service

```
public interface PokedexService {  
  
    @GET("pokemon")  
    Call<List<Pokemon>> pokedex();  
}
```

## Retrofit

```
public class PokemonDAO {  
  
    private String baseURL;  
    private Retrofit retrofit;  
    private PokedexService service;  
  
    public PokemonDAO() {  
        baseURL = "http://pokeapi.co/api/v2/";  
        retrofit = new Retrofit.Builder()  
            .baseUrl(baseURL)  
            .addConverterFactory(GsonConverterFactory.create())  
            .build();  
  
        service = retrofit.create(PokedexService.class);  
    }  
}
```



# PEDIDO ASINCRONICO

```
public void damePokemons(String numeroPokemon, final ResultListener<Pokemon>
listenerDelController){

    service.pokemon(numeroPokemon).enqueue( new Callback<Pokemon>() {

        @Override
        public void onResponse(Call<Pokemon> call, Response<Pokemon> response) {

            Pokemon pokemonx = response.body();
            listenerDelController.finish(pokemonx);
        }

        @Override
        public void onFailure(Call<Pokemon> call, Throwable t) {

            Log.e("retrofit", "fallo");
        }
    });
}
```

# USOS AVANZADOS

- URLs Compuestas
- PUT
- Converter
- Interceptor

# URL COMPUESTAS

```
public interface PokedexService {  
  
    @GET("pokemon")  
    Call<List<Pokemon>> pokedex();  
  
    @GET("pokemon/{numero}")  
    Call<Pokemon> pokemon(@Path("numero") String user);  
  
}
```

# PUT

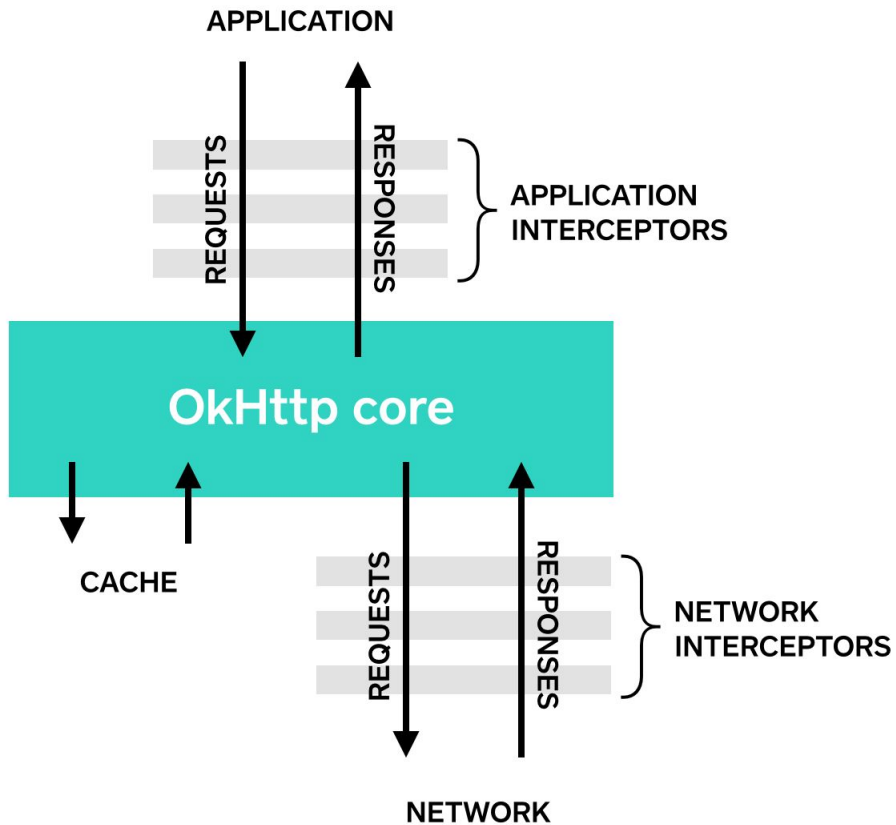
```
@PUT("pokemon/{id}/") Call<Pokemon> updatePokemon(@Path("id")  
int pokemonId, @Body Pokemon pokemon);
```

# CONVERTER

```
Retrofit retrofit = Retrofit.Builder()  
    .baseUrl("https://your.api.url/v2/");  
    .addConverterFactory(ProtoConverterFactory.create())  
    .addConverterFactory(GsonConverterFactory.create())  
    .build();
```



# INTERCEPTOR



# Bibliografia

- <https://futurestud.io/tutorials/retrofit-love-working-with-apis-on-android>