

Arquitectura

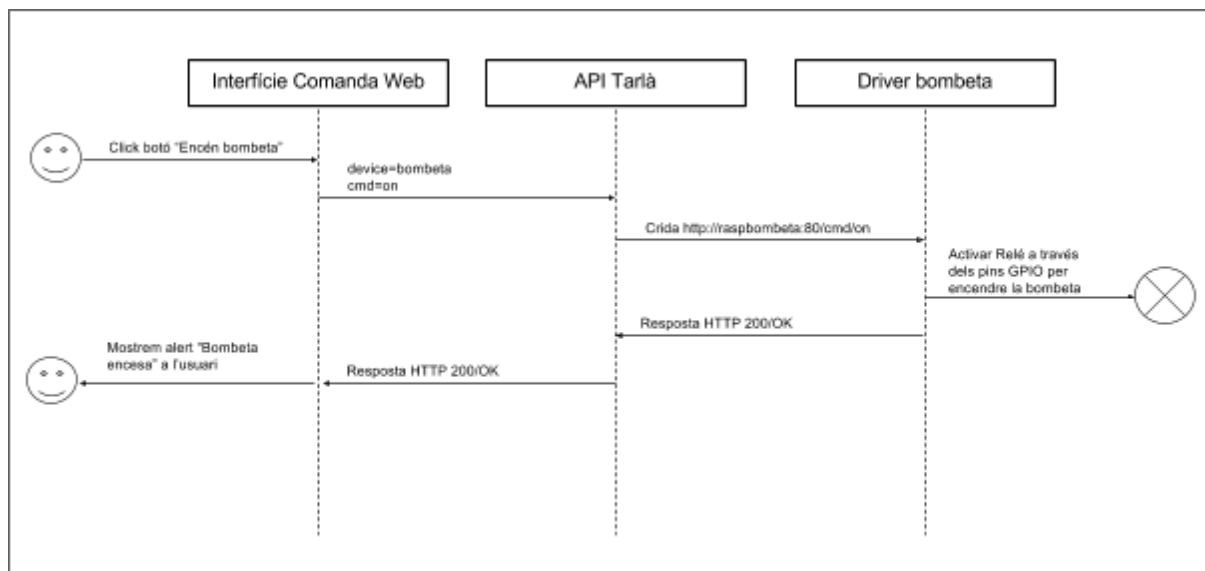
El projecte col·laboratiu del HackathonGi 2017 té 3 mòduls diferenciats. Des de la organització hem desenvolupat el mòdul principal (*API*), per a que els diferents equips puguin treballar de forma independent en els dos altres mòduls (*comandes* i *drivers*).

Els mòduls del projecte seran els següents:

- 1) **Interfícies de comandes.** Projectes software capaços de transformar una ordre d'un usuari a una crida HTTP cap a la *API Tarlà*
- 2) **API Tarlà.** Projecte software que rep comandes via HTTP i les encamina als diferents *drivers* configurats. Es tracta d'un nivell d'indirecció entre les interfícies i el hardware/software que interactua amb els habitants de la casa, una mena de proxie.
- 3) **Drivers.** Projecte software/hardware que rep comandes via HTTP o Socket TCP de la API central, i realitza la acció sol·licitada.

Els equips treballaran en mòduls o bé d'*interfície de comandes* o bé de *drivers*, l'*API Tarlà* està tancada i sota control de la organització :P

Exemple de flux



PETICIÓ DE INTERFÍCIE DE COMANDES A L'API TARLÀ

La API TARLÀ està basada en crides HTTP. La URL per atacar-la és del tipus:

<http://tarla.hackathongi.cat/devices/{bombeta}/cmds/{on}?param1=1¶m2=2>

on *bombeta* i *cmd* són l'ID del dispositiu i la comanda respectivament.

Codi per simular una petició a TARLÀ:

```
# Desde la xarxa interna (wifi hackathongi):  
$ curl http://192.168.4.250/devices/bombeta_http/cmds/on  
  
# Desde internet (4G):  
$ curl http://tarla.hackatongi.cat/devices/bombeta_http/cmds/on
```

PETICIÓ DE LA API TARLÀ ALS DRIVERS

Els *drivers* han de ser capaços de rebre un missatge de *TARLÀ*, fer la actuació corresponent, i retornar una resposta.

S'han de donar d'alta a la configuració de *TARLÀ*, indicant si fan servir el protocol HTTP o plain TCP socket.

Protocol HTTP

El dispositiu ha de córrer un servidor web, on *TARLÀ* enviarà la crida http amb els mateixos paràmetres.

El body de la resposta es retornarà a la petició en curs de *TARLÀ*

Codi per simular una petició HTTP de *TARLÀ* al *driver*:

```
$ curl http://<ip_interna_dispositiu>:80/bombeta_http/cmds/on
```

Protocol plain TCP socket

El dispositiu ha de escoltar connexions TCP. Un cop *TARLÀ* hagi establert la comunicació, enviarà un string del tipus "{cmd}?{query}" on query seria "param1=val1¶m2=val2&..."

El driver ha de parsejar la comanda, respondre o bé "OK-{string de resposta opcional}" o bé "KO-{string de resposta opcional}", i tancar el socket

Codi per simular una petició SOCKET de *TARLÀ* al *driver*:

```
$ (echo -n "on"; sleep 2) | nc <ip_interna_dispositiu> <port>
```

Exemples de mòduls per desenvolupar al hackathon

Exemples interfícies de comandes

- Chatbots
 - Via Telegram
 - Via facebook messenger
 - Via Slack
- Reconeixement de veu (p.ex. amb Google Speech API)
- SMS o trucades telefòniques (p.ex. Fent servir la API de twilio.com)
- Web interface
- App nativa iOS/Android amb botons per llançar comandes

Exemples drivers

- Hardware
 - **Encendre/apagar una bombeta** => Arduino/rasp + interruptor
 - **Torradora** => Arduino/rasp + interruptor
 - **Obtenir temperatura de la sala** => Arduino + Sensor temperatura
 - **Alarma** => Arduino + sensor de presència (o webcam) i que encengui la sirena i t'avisi
 - Sensor de llum=> poder preguntar-li la intensitat de llum
 - **Pantalla/media center** per veure-hi coses d'altres dispositius: (càmera de vigilància, internet controlat pel cervell central(mostrar cerques, visionar-hi coses...))
 - Control persianes => Sensor de llum + pujar i baixar persiana
- Software
 - Posar música a spotify
 - Explicar acudits
 - Previsió meteorològica
 - Google calendar del dia
 - Cerques a google/youtube i visionar-les a alguna pantalla