

MONTREAL JUG · PRÉSENTATIONS ÉCLAIRS · 5 NOVEMBRE 2026

From Backend to Hardware

D'une idée à un objet fabriqué en série, vu par un développeur backend.

Alexis Couvreur · CTO, Reeveel

Alexis Couvreur



- Développeur backend · Montréal
- Platform engineer ·  Banque Nationale
- Cofondateur et CTO ·  Reeveel
- Mainteneur ·  Sablier

CHAPITRE 1 / 5

Le produit

Ce qu'on fabrique, et ce qu'on exige de lui.

LE PRODUIT FINAL



Écran

e-paper couleur, 13,3"

Batterie

10 000 mAh

Capteur

accéléromètre

Stockage

carte microSD

Radio

Bluetooth, Wi-Fi

NFC

avec le téléphone

Bouton

tactile, caché

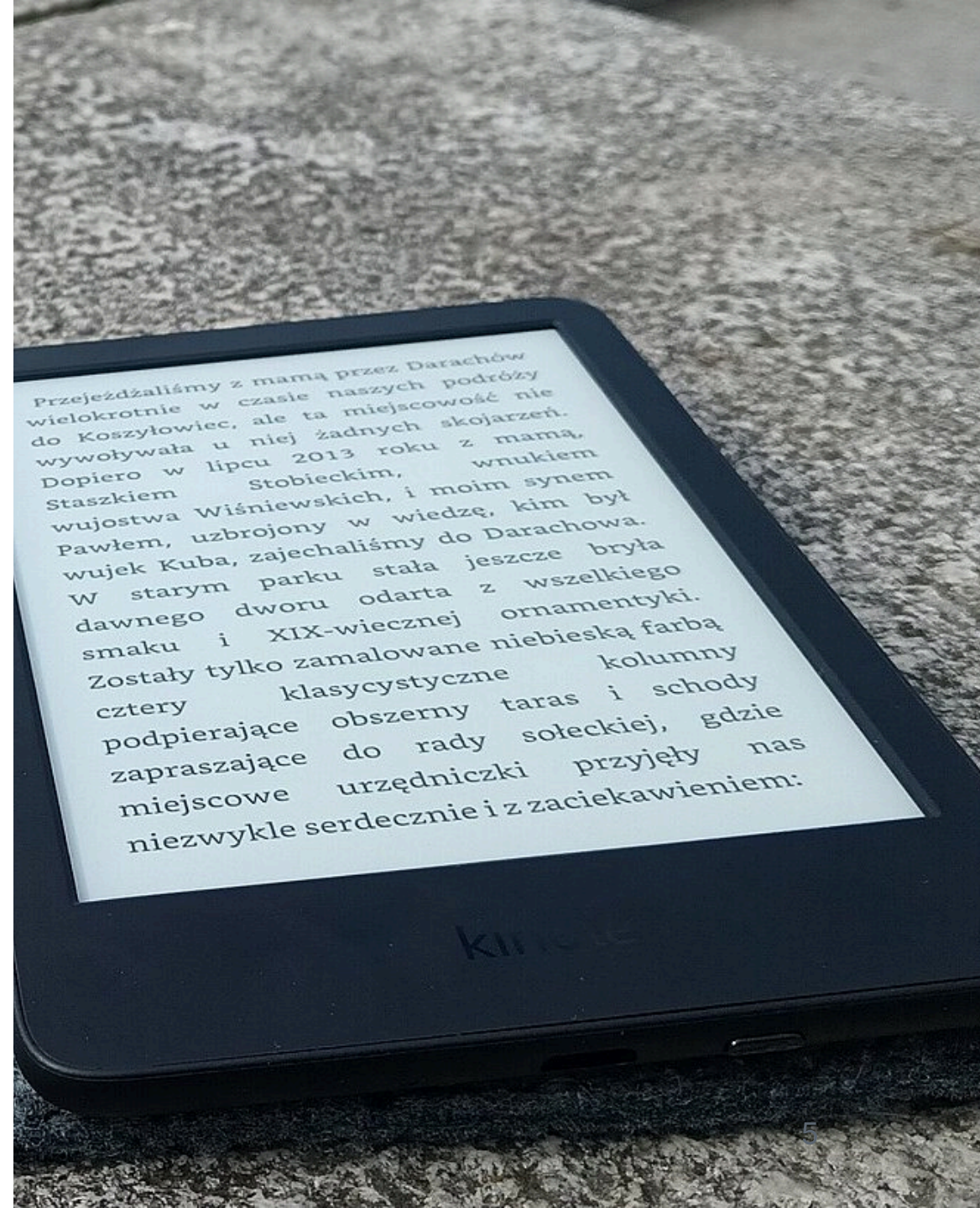
Passe-partout

aimanté



Le e-paper, comme une liseuse

- Comme un Kindle, en couleur
- Pas de rétroéclairage : il réfléchit la lumière
- Une image fixe ne consomme rien



Nos exigences

- Un an sur une charge
- Bluetooth d'abord, Wi-Fi en option
- Mise à jour à distance
- Des périphériques : NFC, accéléromètre, bouton caché, microSD

CHAPITRE 2 / 5

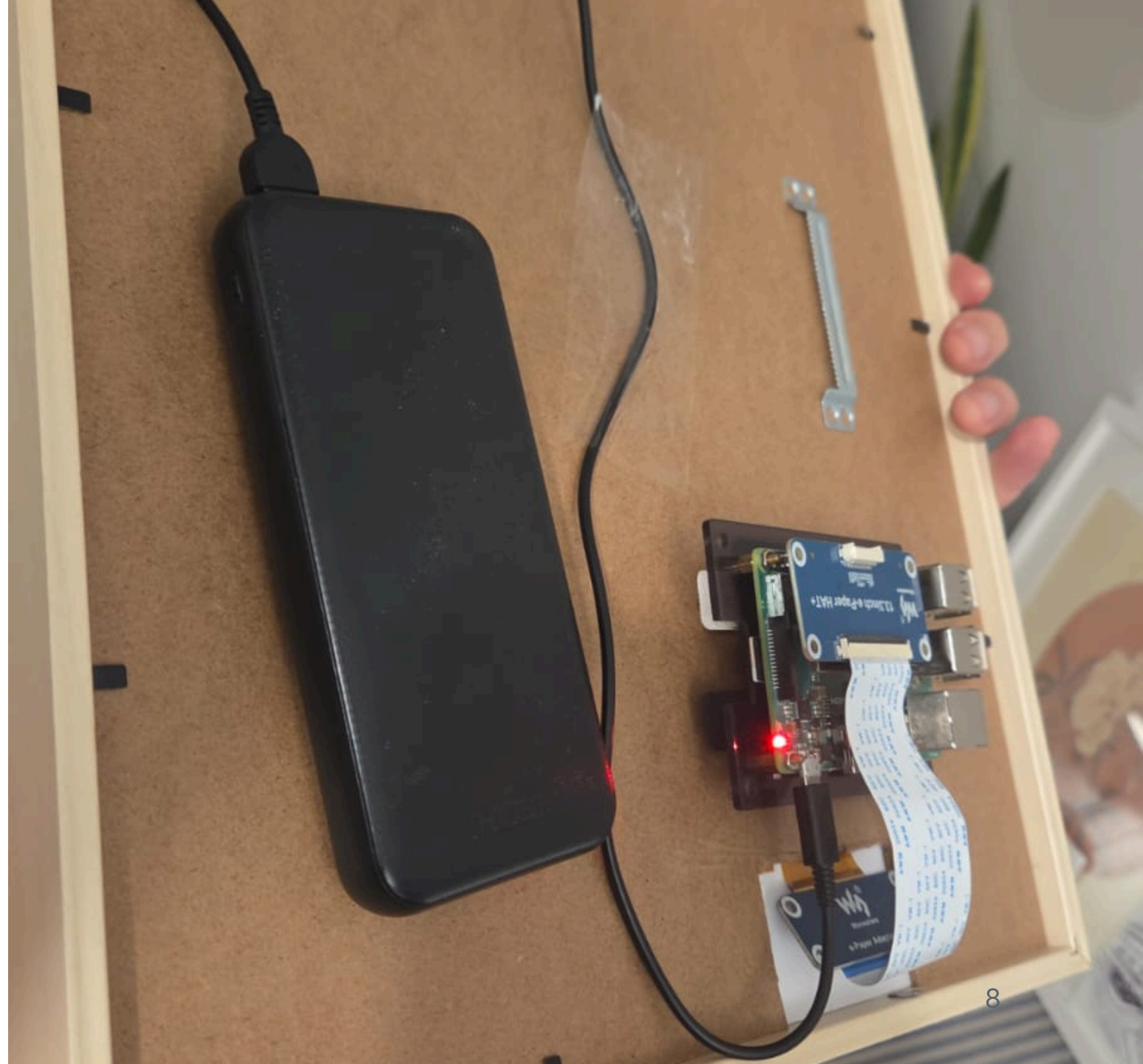
Les prototypes

Du Raspberry Pi à notre propre carte.

L'IDÉE

Raspberry Pi

Python, et un écran e-paper : la preuve que l'image s'affiche.



PROTOTYPE 1

ESP32, fait main

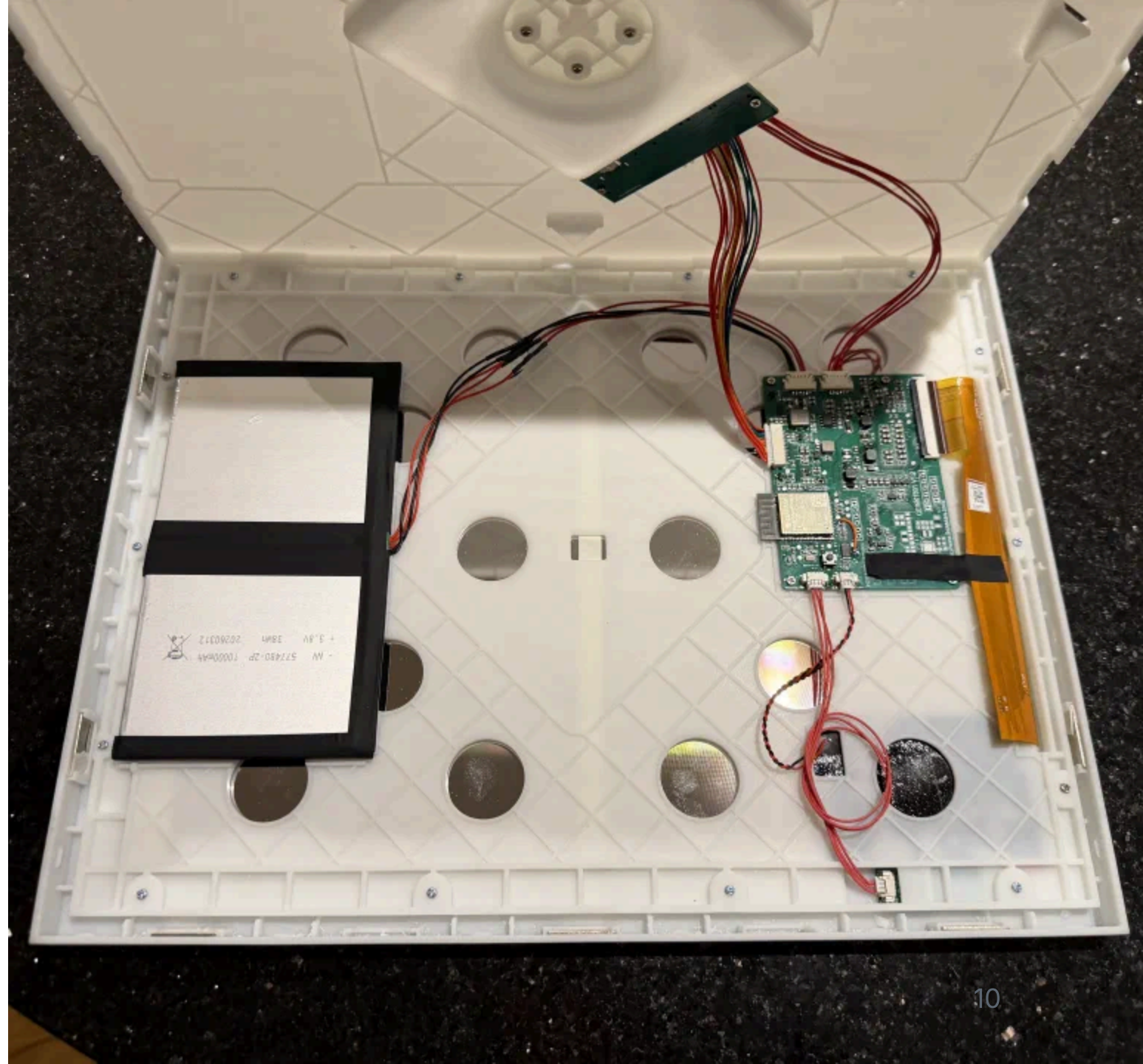
Une puce qui sait dormir profondément.



PROTOTYPE 3

Avec notre fabricant

Notre propre carte,
dans le boîtier
imprimé en 3D, puis
usiné en CNC.



CHAPITRE 3 / 5

Le firmware

Organiser un logiciel stateful, en C.

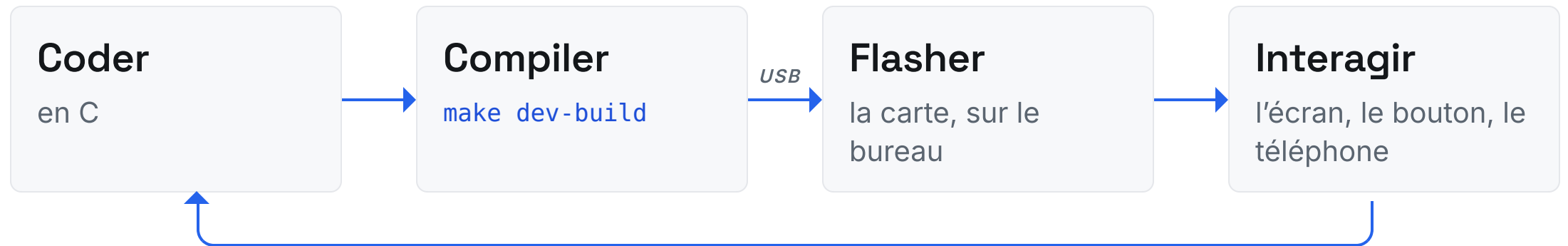
LA STACK TECHNIQUE



Le framework d'Espressif : bas niveau, en C, au plus près de la puce.

Pas de Linux, pas de processus, pas de garbage collector.

Une journée de dev



En backend, on lit un JSON. Ici, on regarde l'objet réagir.

Architecturer un firmware

```
components/features/  
├─ display/  
│   ├── ctrl_ble_display      Bluetooth  
│   ├── ctrl_http_display     HTTP  
│   ├── ctrl_mqtt_display     MQTT  
│   ├── ctrl_cli_display      console  
│   ├── svc_display           le cœur, et sa state machine  
│   ├── drv_epd_el133uf3      l'écran UF3  
│   └── drv_epd_el133uf1      l'écran UF1, une autre variante  
└─ playlist/  
    ├── ctrl_ble_playlist     Bluetooth  
    ├── ctrl_http_playlist    HTTP  
    ├── svc_playlist          le cœur  
    └── repo_playlist         les playlists, sur la carte SD
```

Une feature, un dossier : **controllers**, **service**, **repository**, **drivers**.

Le socle : la plateforme

display

nfc

playlist

battery

ota

...

CHACQUE FEATURE S'ENREGISTRE SUR ↓

ble_host

Bluetooth

http_host

serveur HTTP

console_host

console

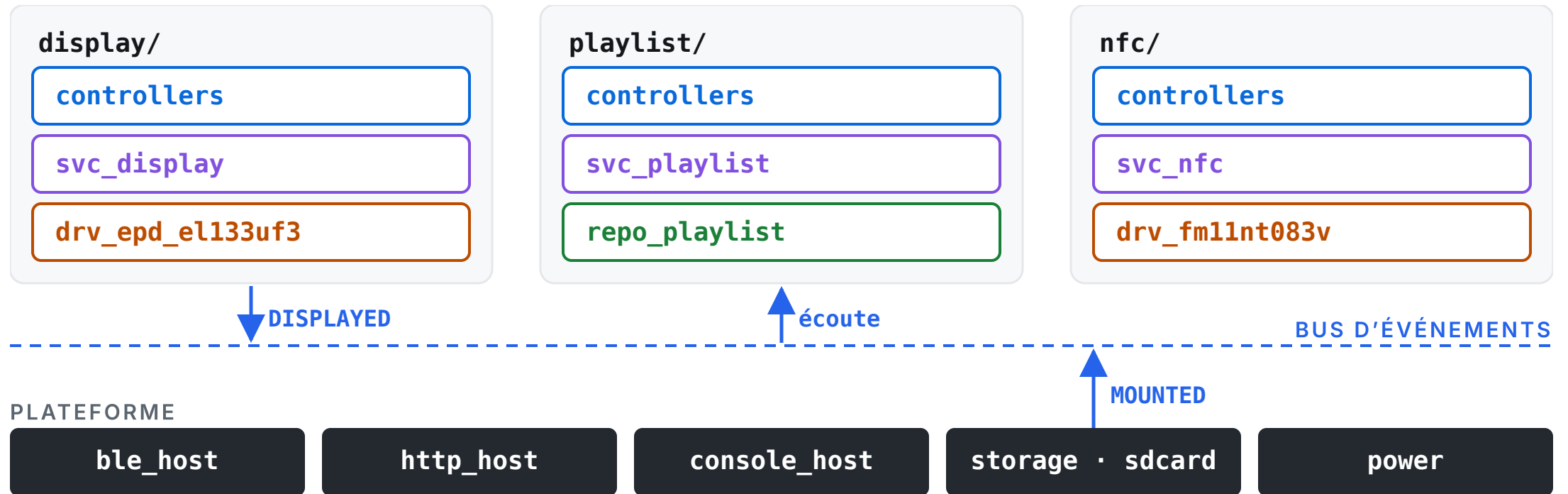
storage · sdcard

carte SD

power

énergie

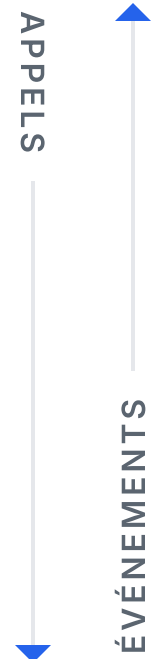
Features et socle : comment ils se parlent



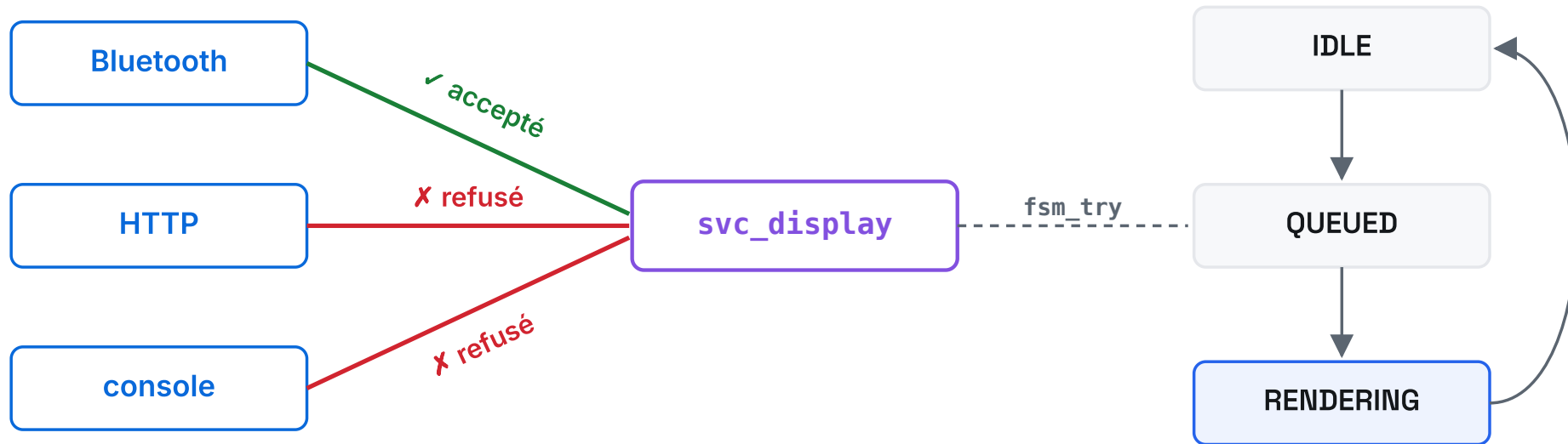
Dans une feature, les appels descendent. Entre features, des événements.

Comme dans Spring, plus un Driver

Controller	Bluetooth, HTTP, MQTT, console	AUCUNE LOGIQUE
Service	les règles métier, la state machine	
Repository	le stockage : flash, carte SD	
Driver	la puce, l'écran, les capteurs	NOUVEAU



Des state machines partout



```
W SVC_DISPLAY: show(src=http) rejected: still RENDERING
```

```
if (!fsm_try(&s_fsm, QUEUED, "show", src)) return ESP_ERR_NOT_FINISHED;
```

20 machines, 67 états. Combinés : plus de 5 milliards.

Un code, plusieurs cartes

```
sdkconfig.defaults          commun à toutes les cartes  
sdkconfig.esp32s3-proto1.EL133UF3  puce · révision · écran  
sdkconfig.esp32s3-proto2.EL133UF3  + LED avant, buzzer  
  
make dev-build BOARD=esp32s3-proto2.EL133UF3
```

Une carte, un profil. Un écran, un driver. Le reste ne bouge pas.

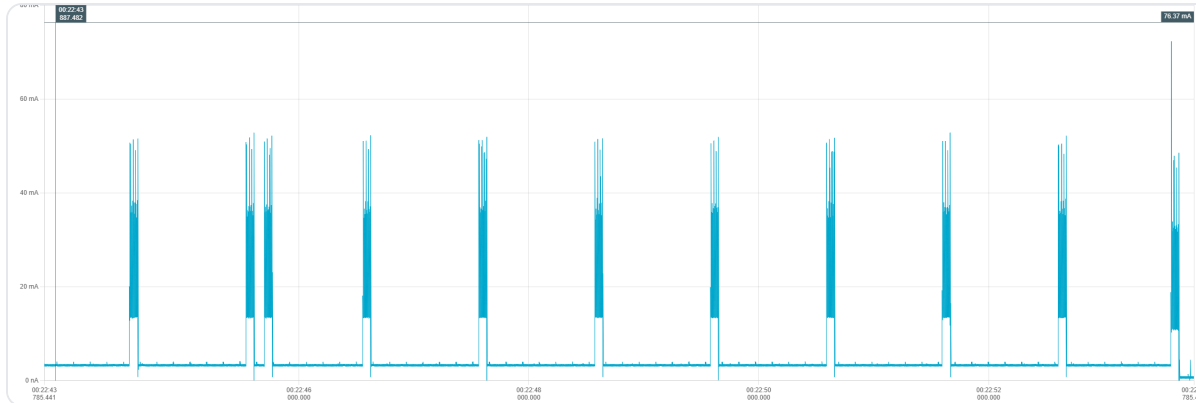
CHAPITRE 4 / 5

L'énergie

Tenir un an sur une charge.

EXIGENCE · UN AN SUR UNE CHARGE

D'abord, mesurer



4,23 mA **72 mA** **10 s**
en moyenne au pic de mesure

Un Nordic PPK2, en série avec la batterie, USB débranché.



Les leviers, un par un

Sans rien

CPU à 160 MHz, qui attend



27,6 mA

+ CPU adaptatif

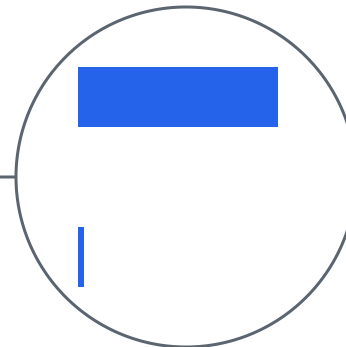
descend à 40 MHz



13,2 mA ÷ 2

+ Light sleep

dort entre deux tâches



× 20

240 µA ÷ 115

+ Deep sleep

jusqu'au prochain réveil



7 µA ÷ 3 900

Et pour y rester : des interruptions plutôt que du polling, et un nom par raison de rester éveillé.

CHAPITRE 5 / 5

Les mises à jour

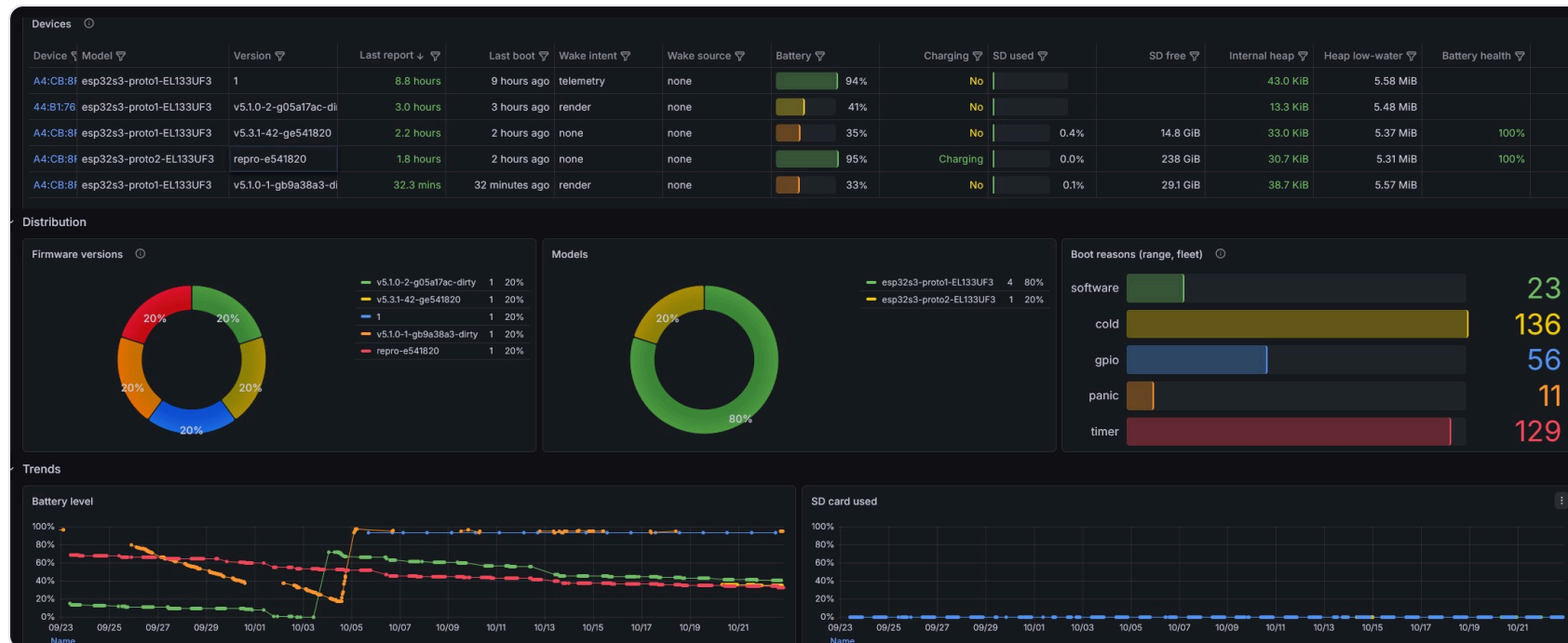
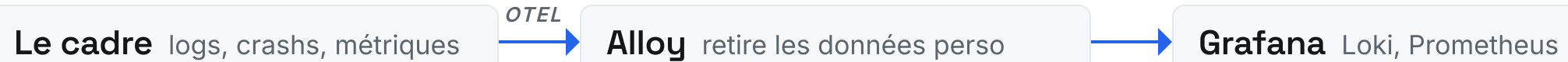
Corriger un cadre déjà installé chez quelqu'un.

Mettre à jour : du blue-green dans la puce



Par Bluetooth : 1,9 Mo en 30 s. Ou tout seul, en HTTP.

Télémétrie : débbugger à l'échelle



Seulement en développement, ou si un client l'active pour qu'on l'aide. Désactivée en production.

CONCLUSION

Et maintenant ?

D'une idée à un Kickstarter, en 15 mois.

CONCLUSION

Où on en est



QUESTIONS ?

Merci !

Inkolor est sur Kickstarter en ce moment.

 gettreeveel.com ·  [Kickstarter](https://www.kickstarter.com) ·
alexiscouvreur.fr

