Progressive Web Applications

Aperçu et mise en place

UE Software Architecture & Mobile Programming

Frédéric Dadeau

Semaine du 23 novembre

Principes

~2015

- Exécuter une application dans une fenêtre de navigateur mobile spécifique
  - Application Shell : visualisation en HTML/CSS, logique applicative en Javascript
  - Nombreuses possibilités : événements tactiles, accès caméra/micro, synthèse vocale, gyroscope, accéléromètre, GPS, lecture audio/vidéo, etc.

- Faire « comme si » c'était une application native :
  - Lancement depuis une icône sur l'écran d'accueil du mobile
  - Exécution en plein écran
  - Possibilité de s'exécuter hors-ligne

Fonctionnalités-clés des PWA

- **Progressive** — They work for every user, regardless of the browser chosen because they are built at the base with progressive improvement principles.
- **Responsive** — They adapt to the various screen sizes: desktop, mobile, tablet, or dimensions that can later become available.
- **App-like** — They behave with the user as if they were native apps, in terms of interaction and navigation.
- **Updated** — Information is always up-to-date thanks to the data update process offered by service workers.
- **Secure** — Exposed over HTTPS protocol to prevent the connection from displaying information or altering the contents.
- **Searchable** — They are identified as « applications » and are indexed by search engines.
- **Reactivable** — Make it easy to reactivate the application thanks to capabilities such as web notifications.
- **Installable** — They allow the user to “save” the apps that he considers most useful with the corresponding icon on the screen of his mobile terminal (home screen) without having to face all the steps and problems related to the use of the app store.
- **Linkable** — Easily shared via URL without complex installations.
- **Offline** — Once more it is about putting the user before everything, avoiding the usual error message in case of weak or no connection. The PWA are based on two particularities: first of all the ‘skeleton’ of the app, which recalls the page structure, even if its contents do not respond and its elements include the header, the page layout, as well as an illustration that signals that the page is loading.

Source : https://medium.com/iquii/progressive-web-app-pwa-what-they-are-...

Limitations des PWA

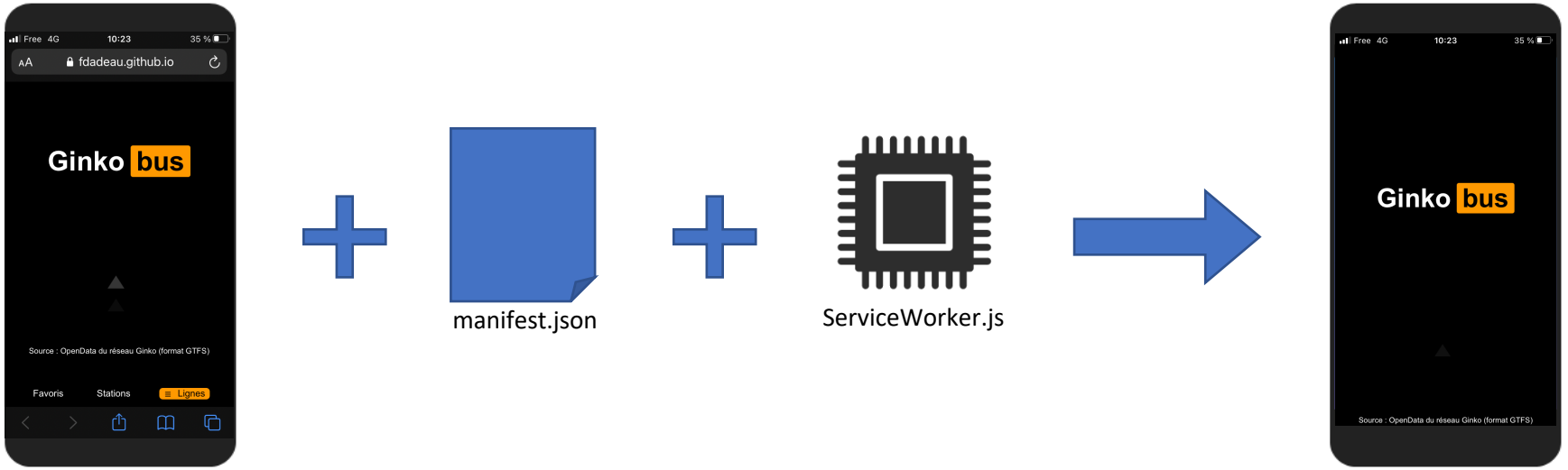
- Liées aux limitations du navigateur
  - Pas d'accès au système de fichier, mais stockage interne au navigateur (LocalStorage, IndexedDB) ou cloud
  - Accès plus restreint au matériel qu'avec un code natif (face ID, NFC, bluetooth, etc.)
  - Quelques soucis de prise en charge sur iOS (>= 11.3)

Pour vous aider : https://whatwebcando.today et http://caniuse.com

- Liées au mode de diffusion (pas de store) :
  - Pas de connaissance des « téléchargements »
  - Pas de contrôle des applis diffusées
  - Pas de vente…

D’une application web à une PWA

- Appli web (responsive)

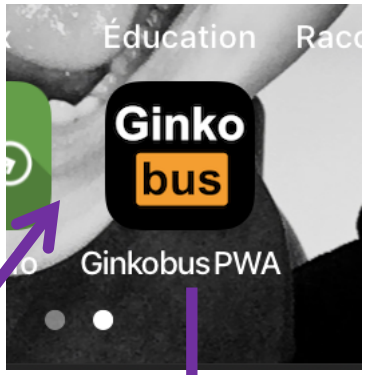


- hébergée en HTTPS
- rendue installable, avec un fichier *manifest*
- rendue utilisable hors-ligne avec un *Service Worker*

Ressource-clé : Chrome avec l’extension Lighthouse (audit PWA)

Fichier Manifest

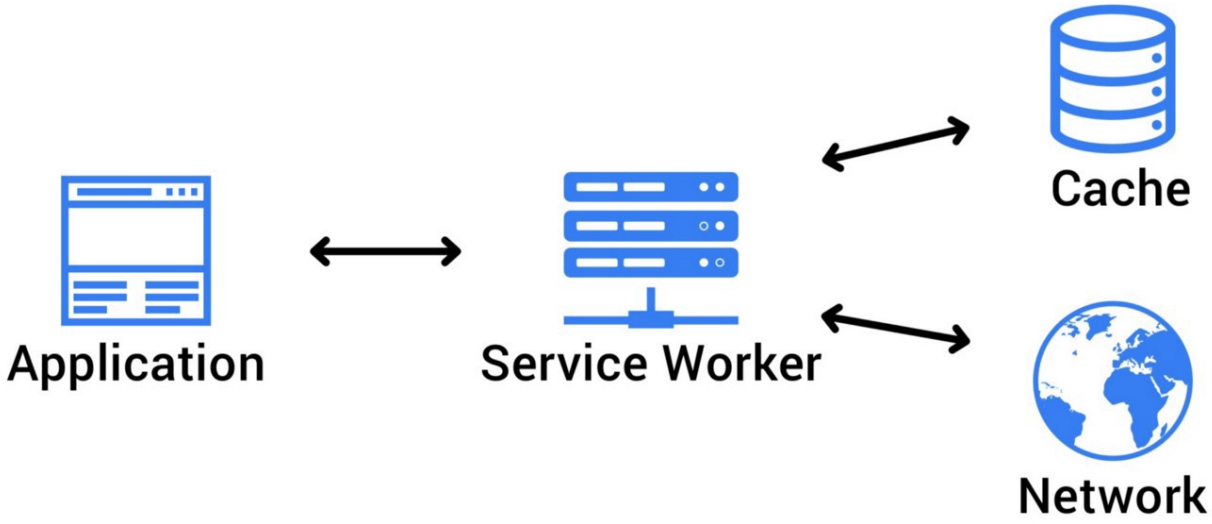
- Fichier JSON décrivant les paramètres de la PWA
  - pour contrôler son apparence pour l'utilisateur :
    - nom,
    - nom court,
    - icônes,
    - etc.
  - pour définir son apparence au lancement :
    - page de départ,
    - orientation de l'appareil,
    - couleurs,
    - etc.



https://developers.google.com/web/fundamentals/web-app-manifest

Service Worker : késako?

- Script qui s’exécute en tâche de fond (séparément de la page - web worker)

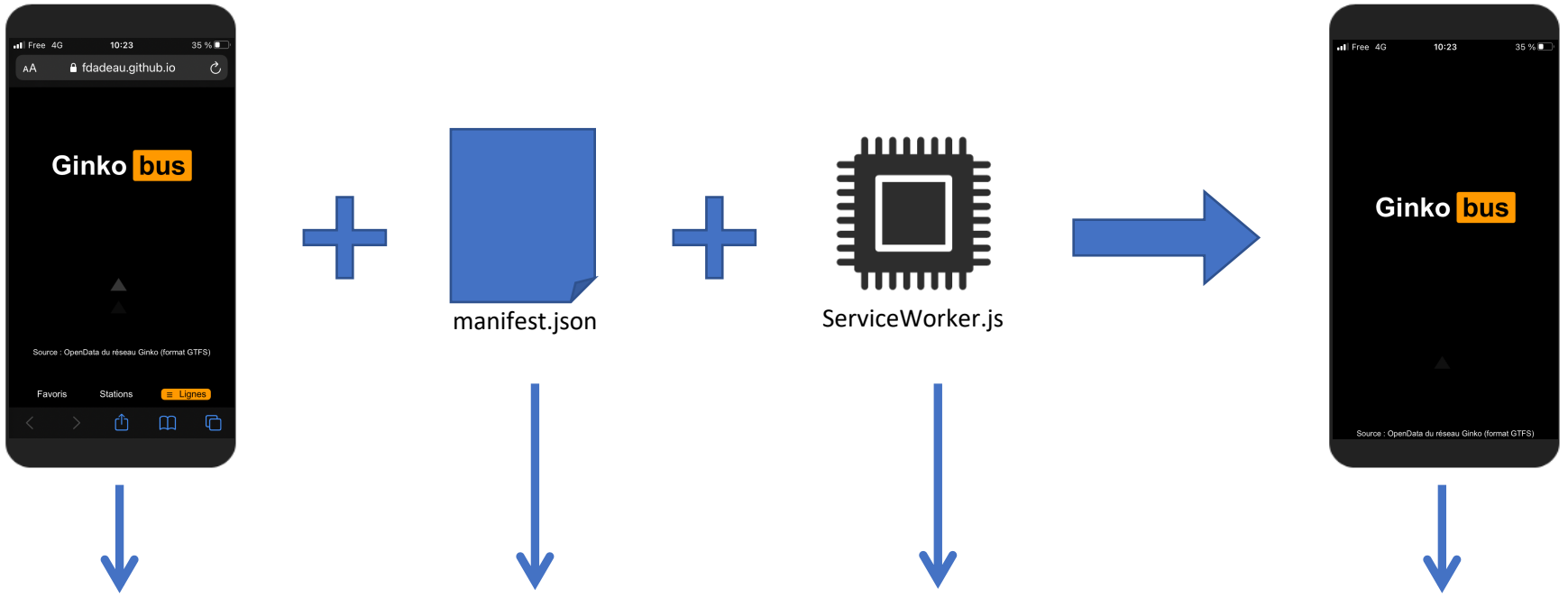


- Usage principal :
  - mise en cache des ressources et données de l’application (événement *install*)
  - dispatch des requêtes au réseau (événement *fetch*)
  - nettoyage des caches (événement *activate*)

https://developers.google.com/web/fundamentals/instant-and-offline/offline-cookbook/

Programme du TP

Application fournie :
Ginkobus Web App



A vous de jouer :	à héberger en HTTPS	à écrire	à écrire	enjoy!