

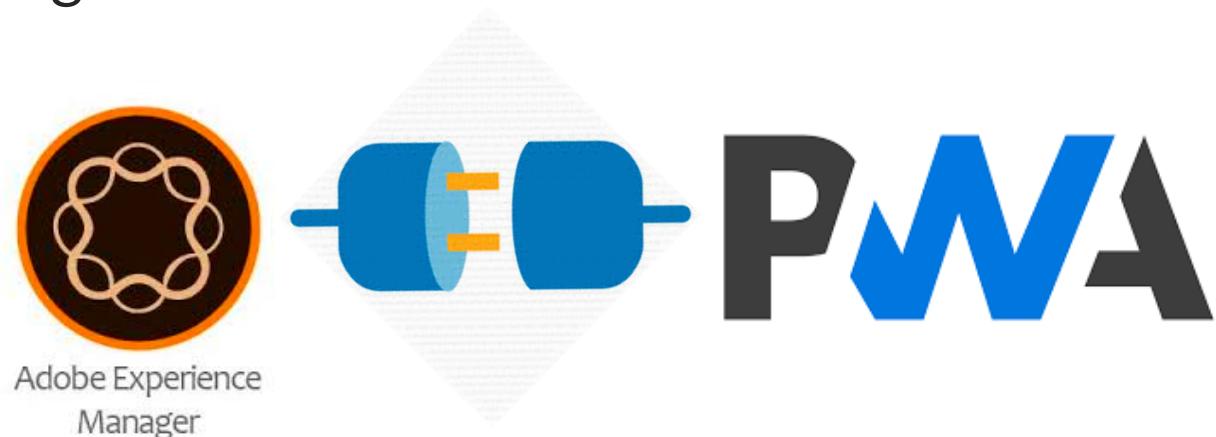
TL30 - Building a Progressive Web Application (PWA) with AEM

Olympe Kassa | Senior Consultant, Adobe | kassa@adobe.com
Rob Wilson | Solution Consultant, Adobe | rowilson@adobe.com



Agenda

- Mobile Marketing as a key for businesses
- The PWA to build
- Progressive Web Applications Building blocks
- Lab pre-requisites
- Lessons
 - Manifest
 - Service Worker
 - Caching content
 - Push Notifications (Bonus)
 - Background Synchronization (Bonus)
- Resources



Mobile Marketing as a key for businesses



Forbes

Flipkart The Flipkart logo consists of the brand name in a blue, bold, sans-serif font next to a yellow shopping bag icon with a blue 'f' on it.

FT
FINANCIAL
TIMES

LANCÔME
PARIS

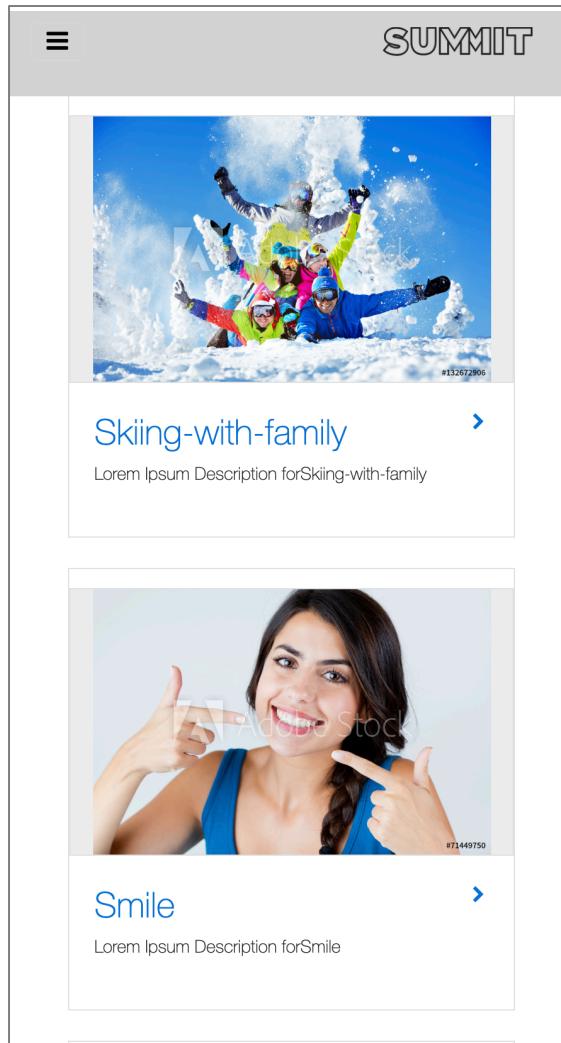
The Washington Post

 **america**

The Virgin America logo, featuring a red diamond shape with a white 'V' inside, followed by the word 'america' in a lowercase, sans-serif font.

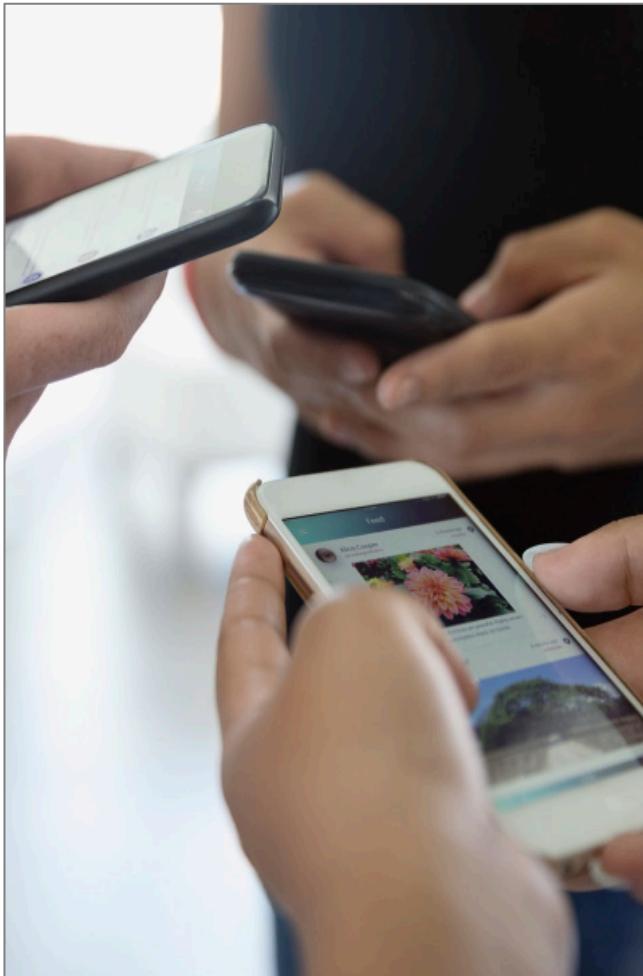
- **Why build Progressive Web Applications ?**
- Progressive Web Applications bring mobile-app-like experiences to end users without requiring them to install an app from a store.
- Deliver reliable, fast and engaging content to customers that drive customer engagement and accelerate transaction processing.
- Empower Marketing and Technical teams for reaching out to customers as those ones use their mobile devices for their daily activities like site searches and shopping.

The PWA to build



- A very simple blog website. This website will allow the end users to achieve the following tasks :
 - List pictures on the PWA home page pulled from an AEM DAM Folder using fetch request and a Sling Model Exporter.
 - Create a post containing the following items :
 - a picture
 - a title
 - some tags : coma separated list
 - Authenticate with Sylvia Burke's credentials. These credentials are provided in the lab guideline document.
 - Subscribe to web push notifications and receive them from AEM.

PWA Building blocks : Key elements

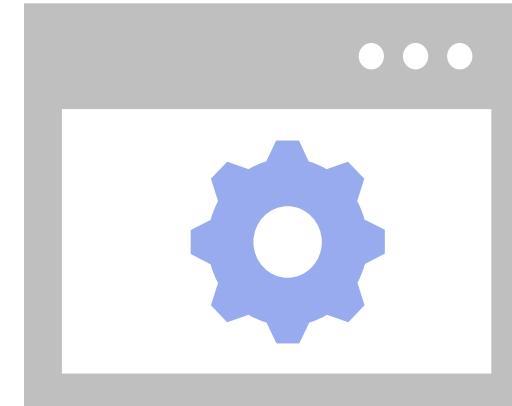


App Shell



User Experience

Service Worker



Accessing and
processing the content

Manifest



Defining a mobile
App-Like experience

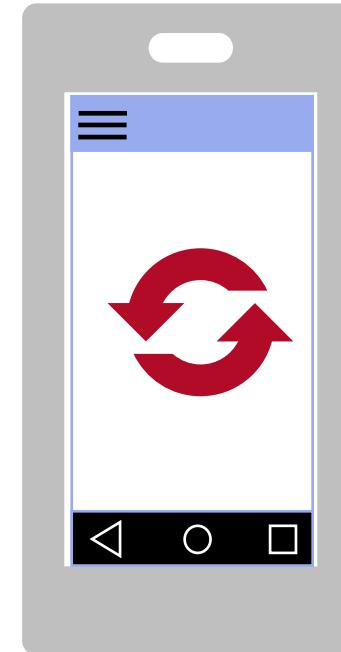
PWA Building blocks : Additional features

Push Notifications

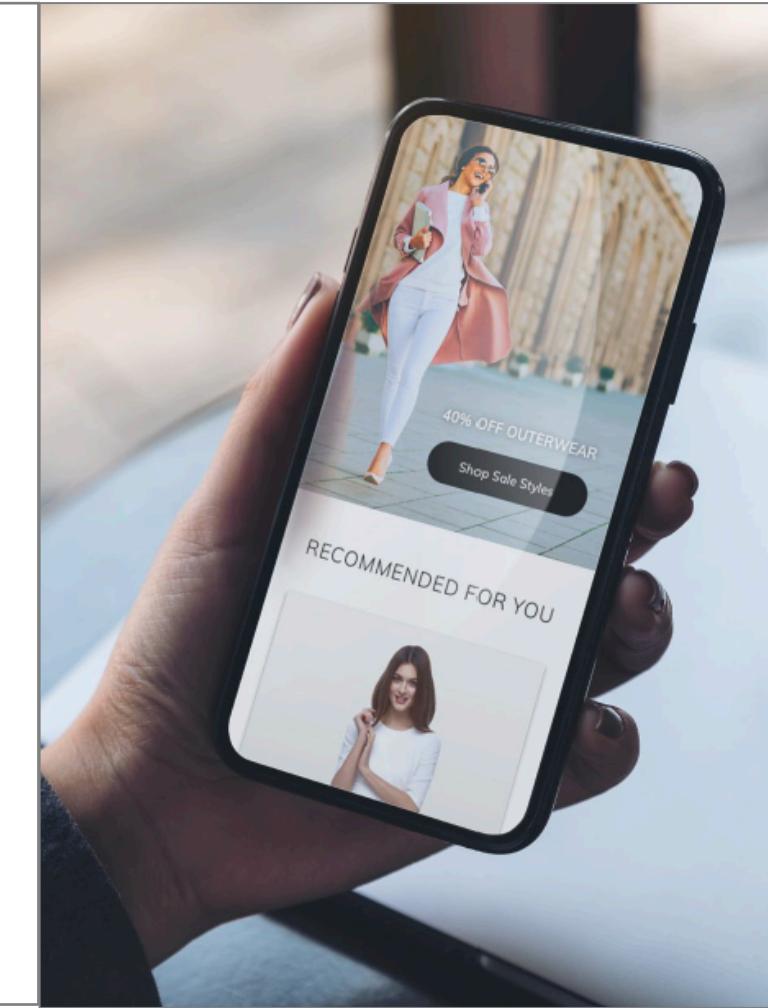


Customer engagement

Background Synchronization

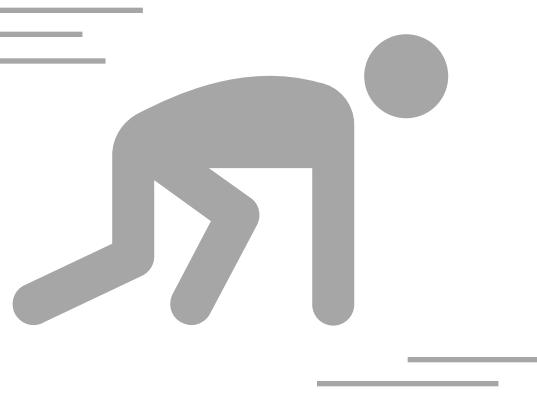


Offline mode support

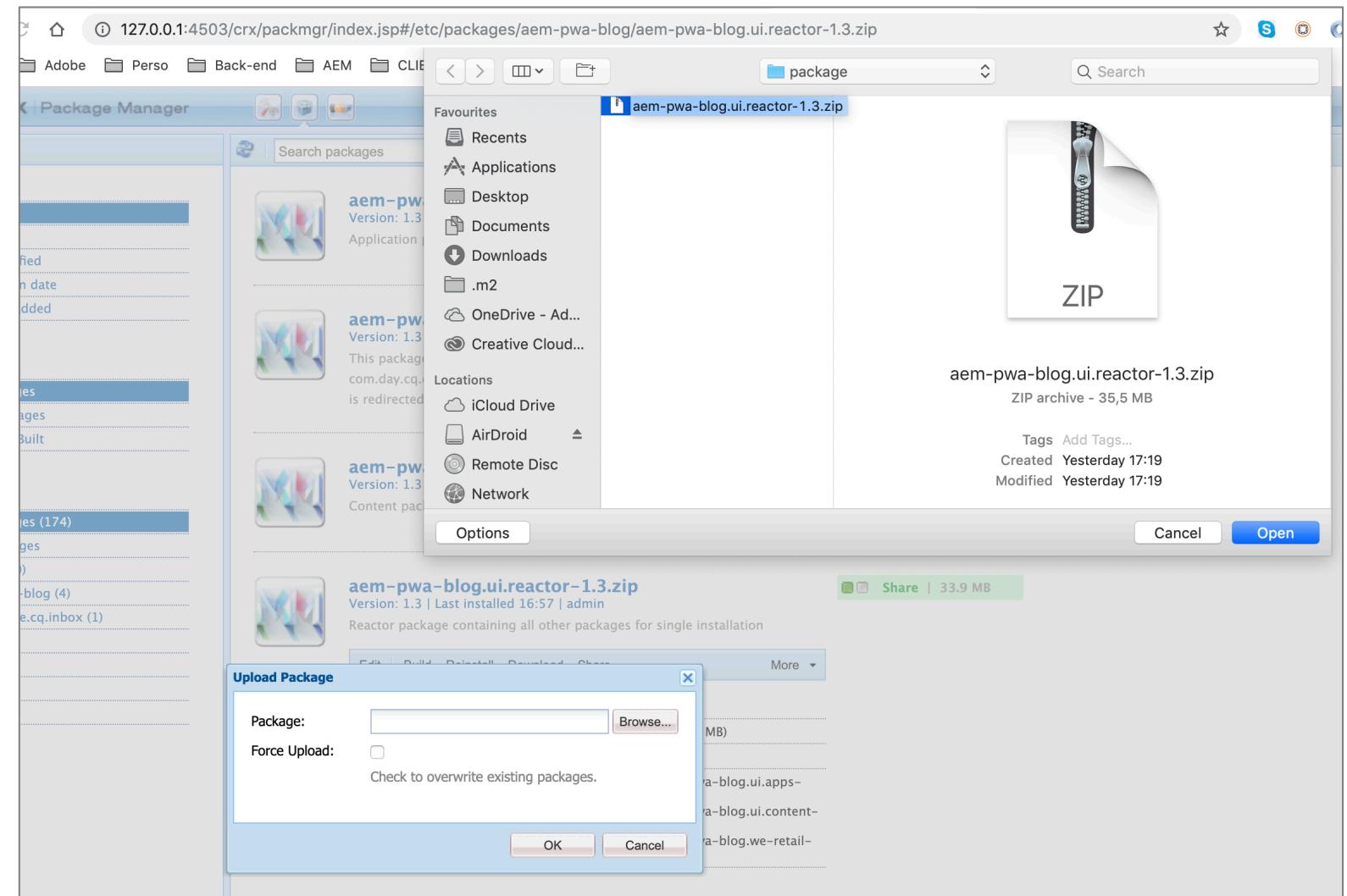


Lab pre-requisites (1/2)

- This lab focuses on PWA features - The source code will directly be updated on the publish instance using CRXDE and the admin user (**admin/admin**).
- Lab guide location : <http://emeatechlabs.com/lab30-final.pdf>
- End user tests will be executed with Sylvia Burke's credentials (**sburke/sburke**) shipped with the lab package.
- PWA mobile testing will be executed with a Nexus Android emulator.
- The AEM instance is exposed on the internet using <http://ngrok.io>. The url of the instance can be retrieved using this link <http://localhost:4040/status>
- **When updating the service worker source code, make sure you increase the VERSION variable value into /etc/clientlibs/aem-pwa-blog/sw.js**



Lab pre-requisites (2/2)



Lessons

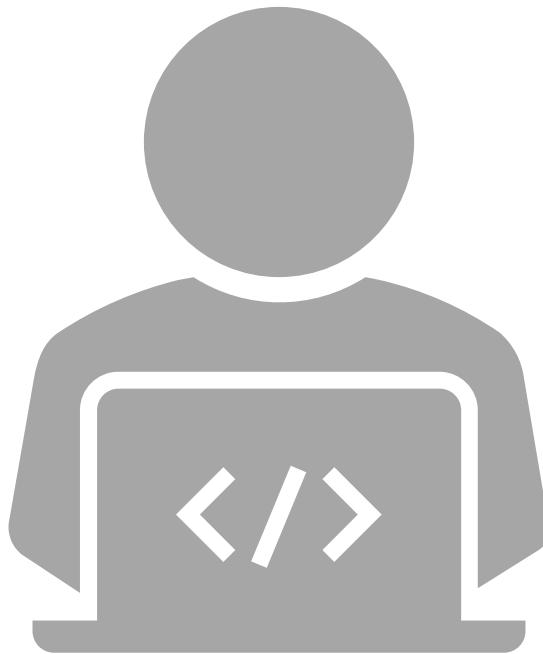


Lesson 1 : The manifest file

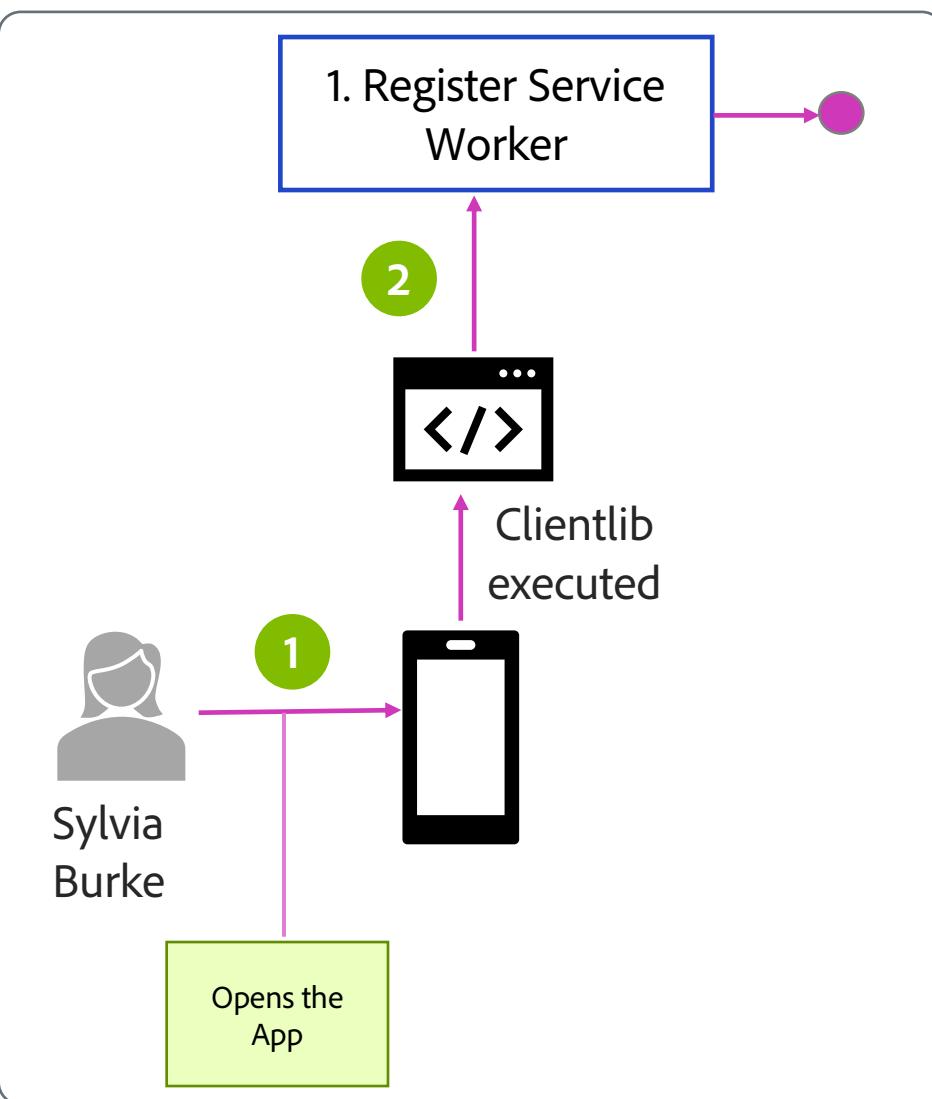
```
{  
  gcm_sender_id: "103953800507",  
  orientation: "portrait",  
  theme_color: "#003c7f",  
  display: "fullscreen",  
  start_url: "/content/aem-pwa-blog/home.html",  
  description: "PWA-TL30 is a progressive web application powered by Adobe Experience Manager",  
  dir: "ltr",  
  - icons: [  
    - {  
      sizes: "64x64",  
      src: "/etc/clientlibs/aem-pwa-blog/icons/summit-icon.png"  
    },  
    - {  
      sizes: "48x48",  
      src: "/etc/clientlibs/aem-pwa-blog/icons/summit-icon-48x48.png"  
    },  
    - {  
      sizes: "96x96",  
      src: "/etc/clientlibs/aem-pwa-blog/icons/summit-icon-96x96.png"  
    },  
    - {  
      sizes: "144x144",  
      src: "/etc/clientlibs/aem-pwa-blog/icons/summit-icon-144x144.png"  
    },  
    - {  
      sizes: "192x192",  
      src: "/etc/clientlibs/aem-pwa-blog/icons/summit-icon-192x192.png"  
    },  
    - {  
      sizes: "256x256",  
      src: "/etc/clientlibs/aem-pwa-blog/icons/summit-icon-256x256.png"  
    }  
  ],  
  - serviceworker: {  
    update_via_cache: "none",  
    src: "/content/sw.js",  
    scope: "/content/aem-pwa-blog/home.html"  
  },  
  background_color: "#003c7f",  
  scope: "/content/aem-pwa-blog/",  
  name: "PWA-TL30",  
  gcm_user_visible_only: true,  
  short_name: "PWA-TL30",  
  lang: "en"  
}
```

- The manifest file is a static JSON file, this file will be exposed by AEM at the following path /content/manifest.json
- W3C File structure of the Manifest
 - **name** : Name of the application installed on the homescreen
 - **gcm_sender_id** : Firebase ID for receiving push notifications
- A meta attribute to add into the head tags

Hands-on activity

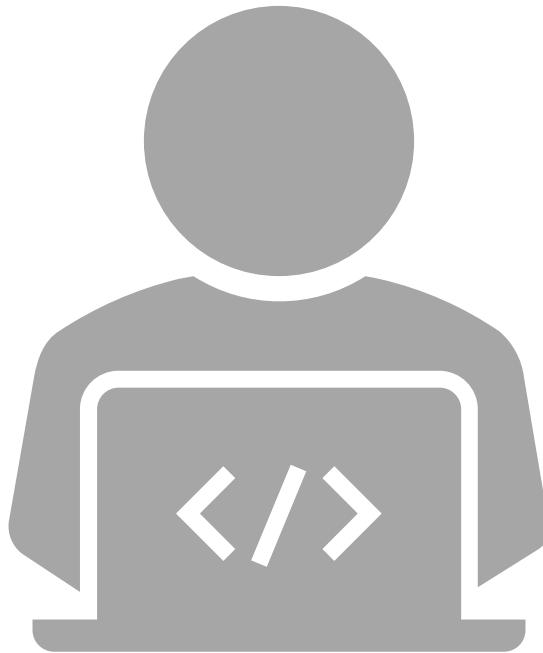


Lesson 2 : Registering a service worker

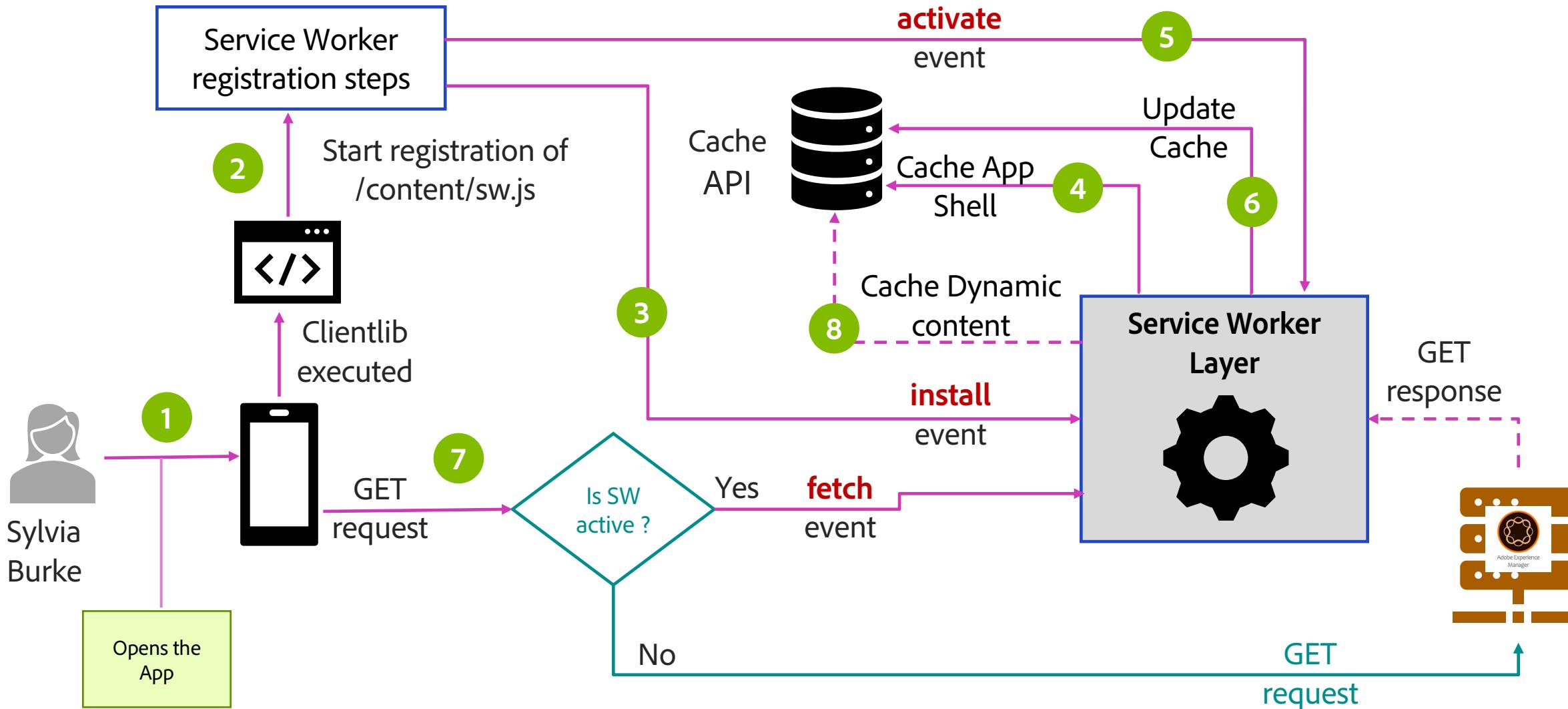


- **What is a service worker ?**
 - A javascript file, which is executed by the web browser in the background, separate from a web page.
- **What is the difference" between a web page js file and a service worker ?**
 - Service Workers run in a different context, thus have no access to DOM elements or Javascript variable.
 - Service Workers should be exposed at the root location of the web app.
- **How do Service Workers and Web Page client libraries communicate between each other ?**
 - Using HTML5 postMessage API
 - Using the BroadcastChannel API

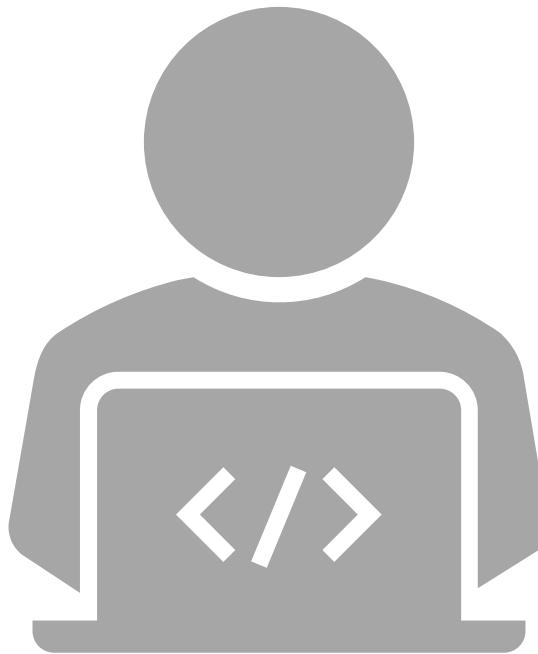
Hands-on activity



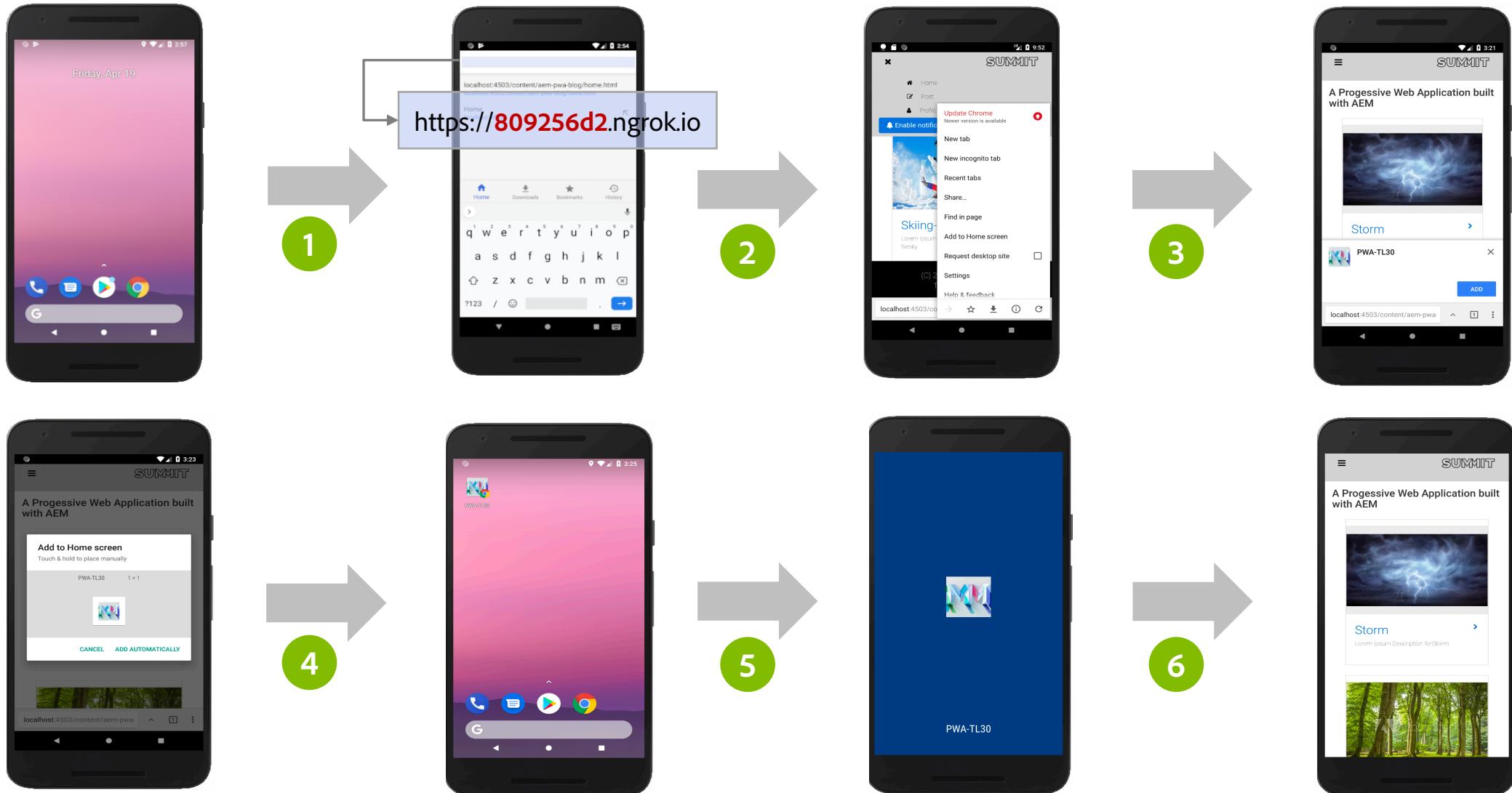
Lesson 3 : Caching content (1/2)



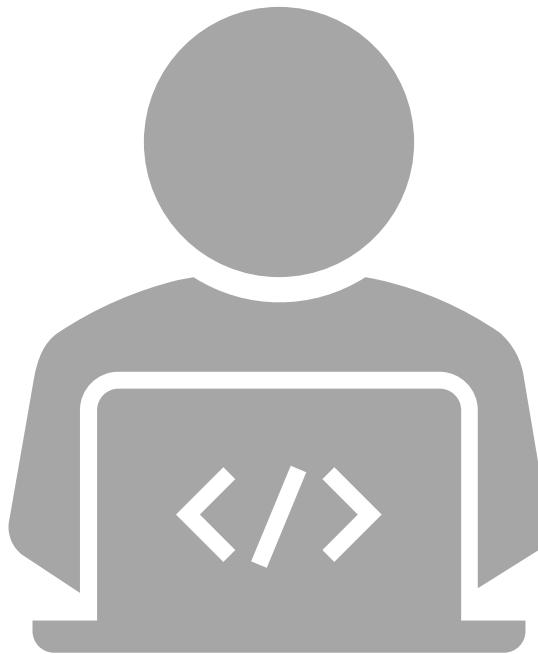
Hands-on activity



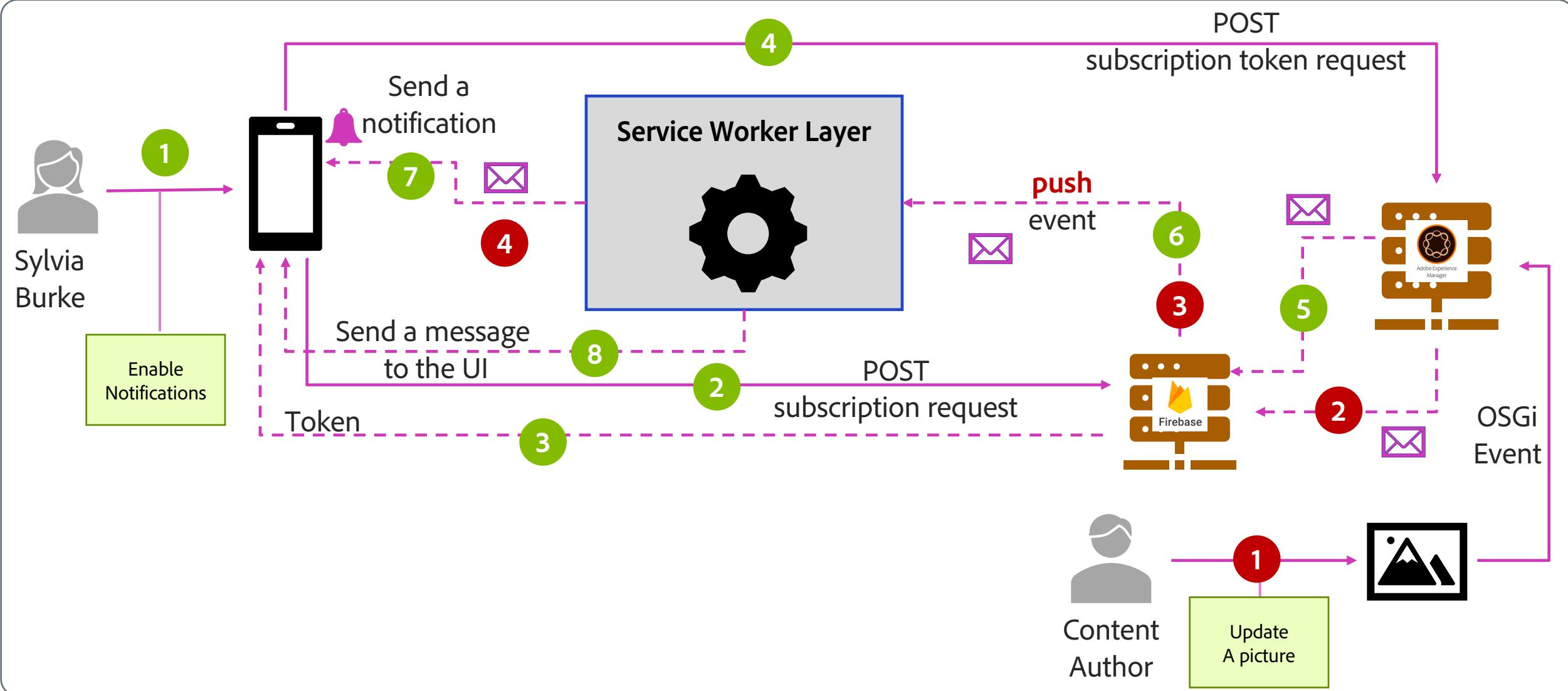
Lesson 3 : Installing the PWA on the emulator (2/2)



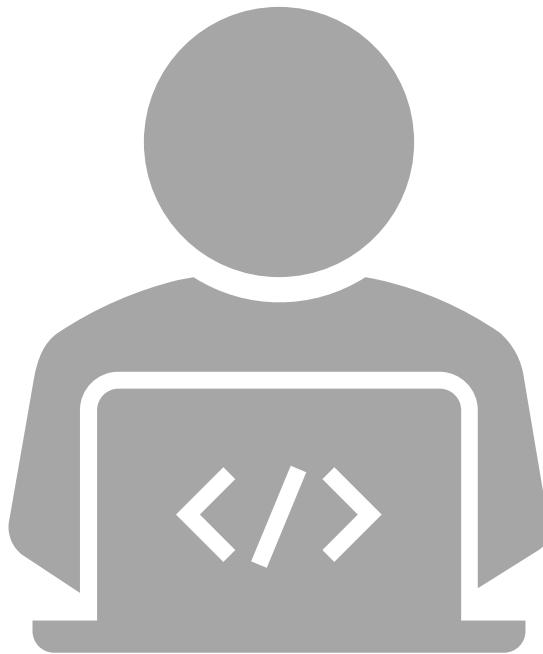
Hands-on activity



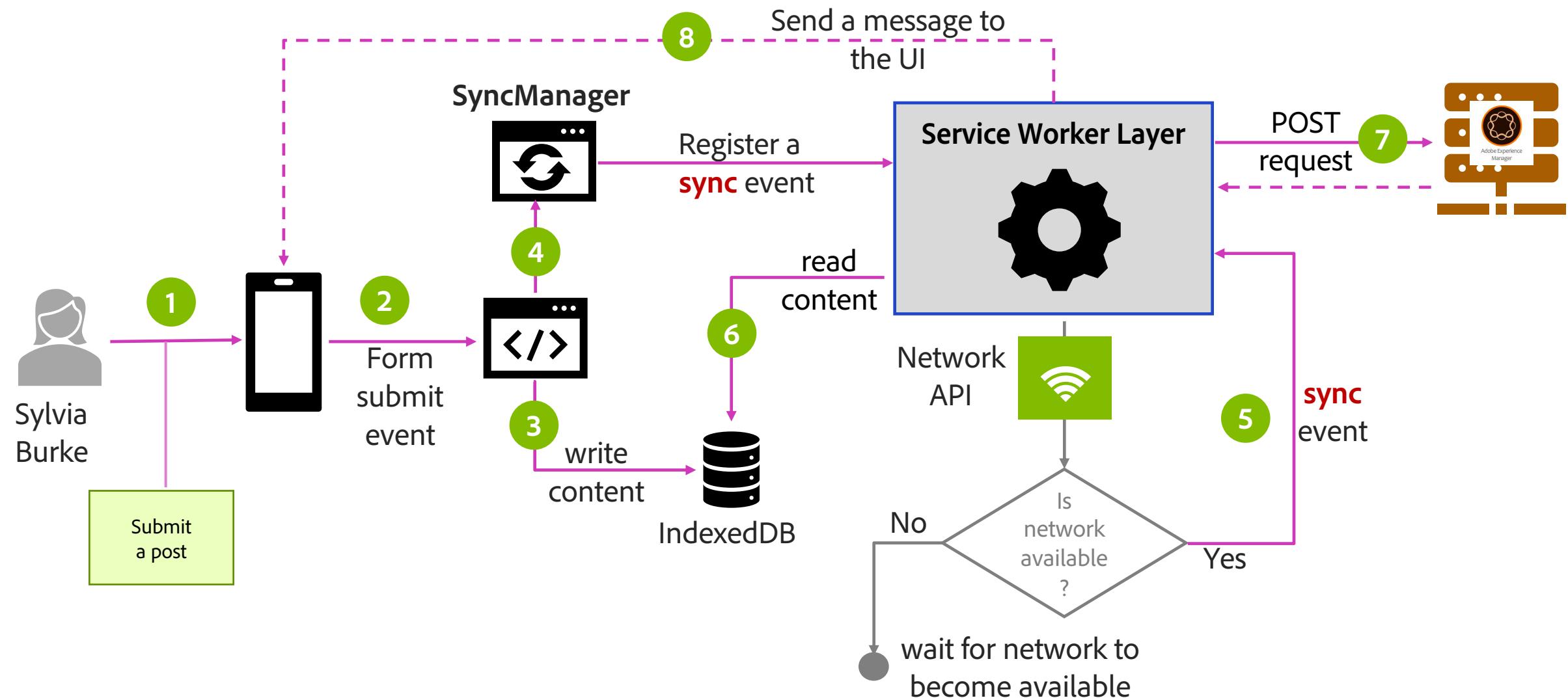
Lesson 4 : Push Notifications (Demo/Bonus)



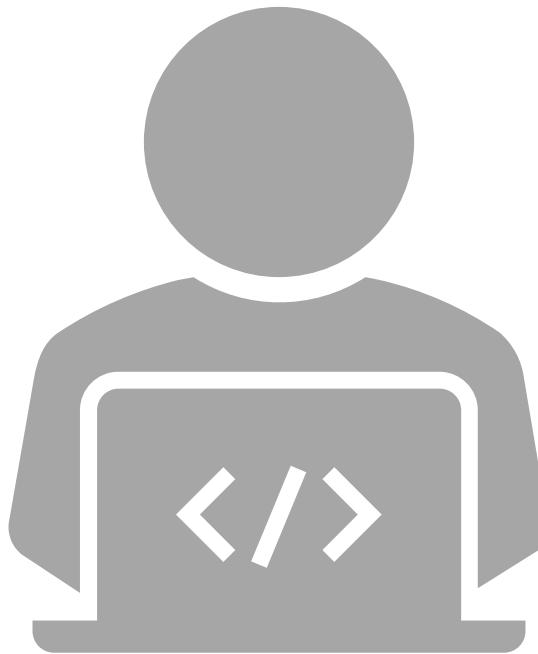
Hands-on activity



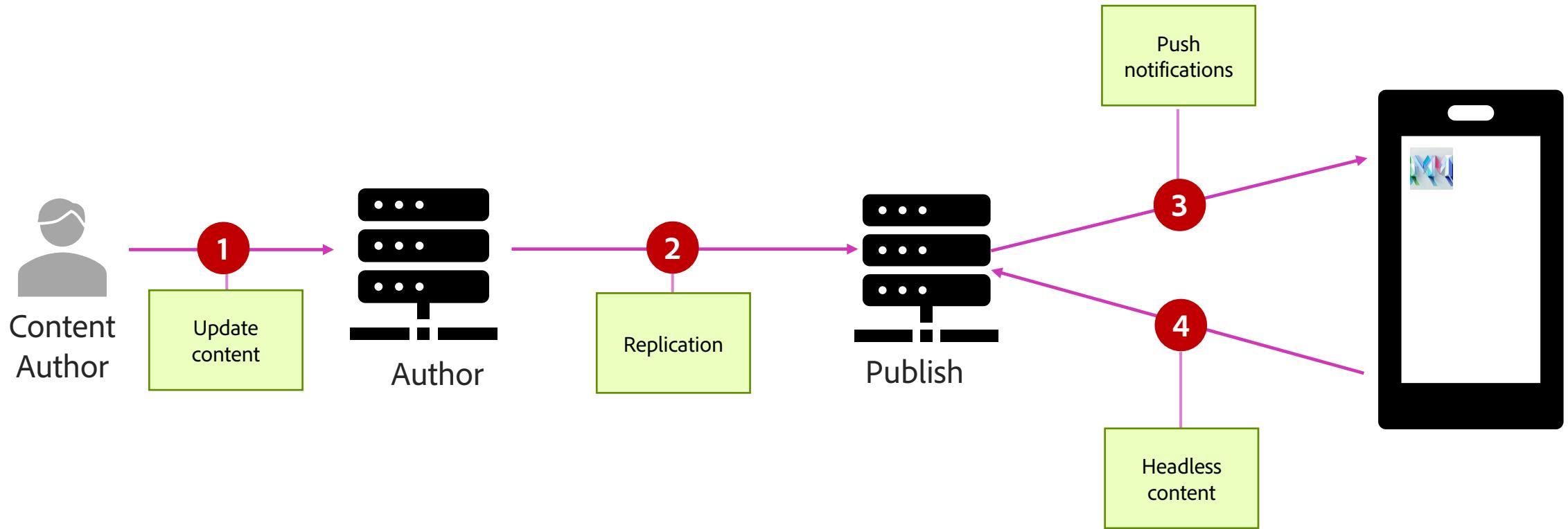
Lesson 5 : Background Synchronization (Demo/Bonus)



Hands-on activity



PWA as an AEM Channel



Resources



Links & Source code

- **Source code**
 - <https://github.com/okassa/TL30>
- **Links**
 - **Manifest**
 - <https://www.w3.org/TR/appmanifest/>
 - <https://developers.google.com/web/fundamentals/web-app-manifest/>
 - **Service Worker**
 - https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Workers
 - <https://www.w3.org/TR/service-workers-1/>
 - <https://jakearchibald.github.io/isserviceworkerready/#moar>
 - **Caching**
 - <https://developer.mozilla.org/en-US/docs/Web/API/Cache>
 - <https://developers.google.com/web/ilt/pwa/caching-files-with-service-worker>
 - **Push notifications**
 - https://developer.mozilla.org/fr/docs/Web/API/Push_API
 - https://developer.mozilla.org/fr/docs/Web/API/Push_API/Best_Practices



Submit your questions using the **Summit App**
Your question, or one similar, may already be in the queue

Q & A

Olympe Kassa

kassa@adobe.com

Rob Wilson

Email: rowilson@adobe.com



Experience Success expert zone at the Adobe booth.

- Talk with Adobe's top experts in digital experiences
- Learn about Adobe's Digital Performance Services
- Meet with customer support teams
- Check-in for your personalized assessment

Adobe.com/go/experience-success

VISIT US AND
Enter
^{TO} WIN
AN iPAD PRO
& OTHER PRIZES



Customer
Solutions