



TESTING OFFLINE FIRST MOBILE APPLICATIONS

The TestCon logo, which includes a stylized 'T' icon made of vertical bars of varying heights, followed by the word "TestCon" in a blue, sans-serif font.

TestCon Europe 2019

[slido.com](https://www.slido.com)

#testcon2019

The logo for TestCon, featuring a stylized 'T' icon composed of vertical bars in blue and orange, followed by the text 'TestCon' in a bold, blue, sans-serif font.

TestCon Europe 2019

Why Offline First Mobile?



Making Users

Wait is *BAD*

**Networks are
NOT reliable**

What is “Offline”?

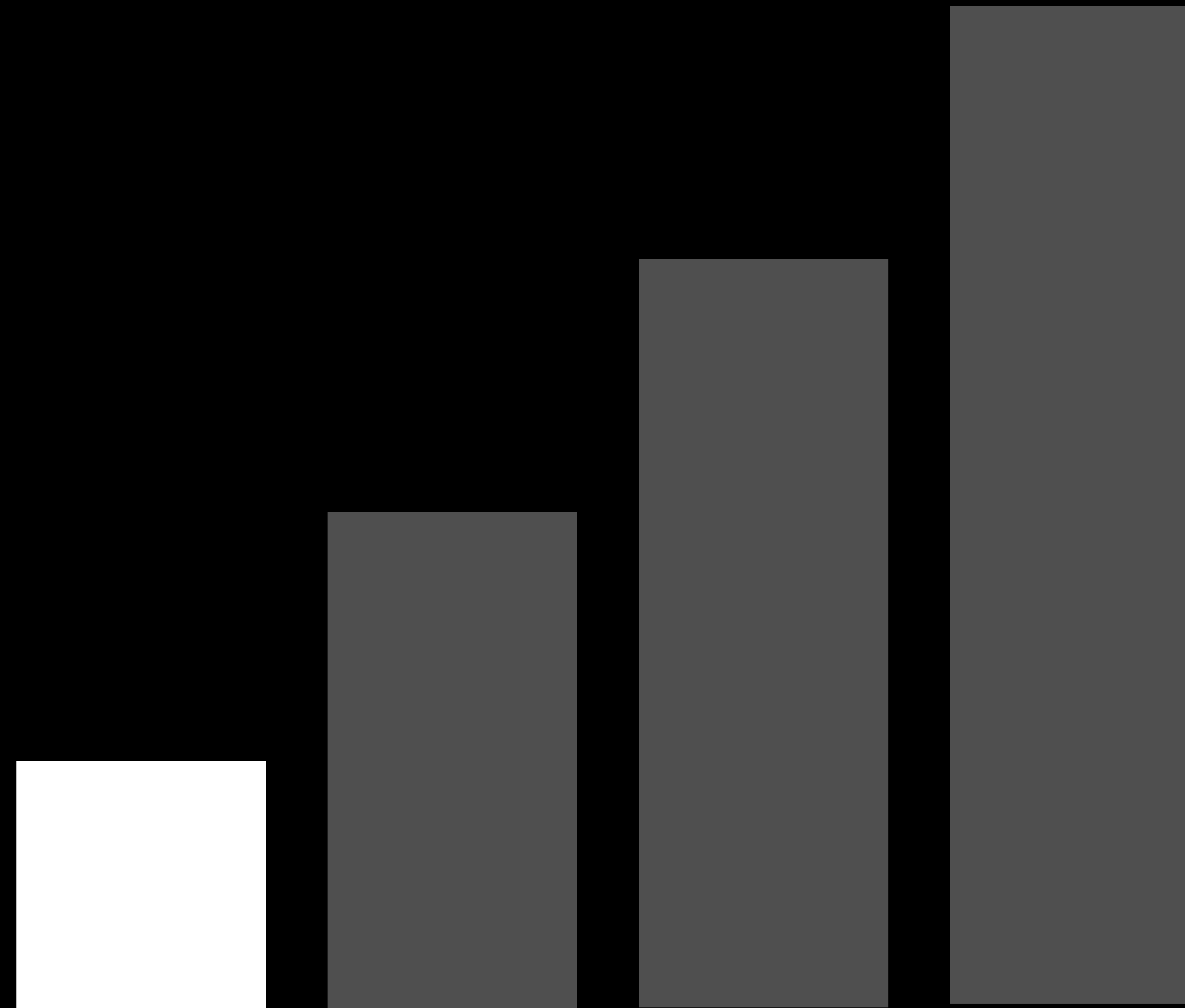


What is Offline?





What is Offline?





What is Offline?

The screenshot shows a browser window titled "Join '@Hyatt_WiFi'". The page features the Hyatt logo in the top left and a "Language" dropdown menu in the top right. The main content is a white box with the text "YOU'RE ALMOST CONNECTED TO free WI-FI." and a prominent yellow "Connect Now" button. Below this box, the word "OR" is centered between two horizontal lines. The footer contains "Help | © 2019 Hyatt Corporation". The browser's address bar shows "bap.aws.opennetworkexchange.net" with a lock icon and a "Cancel" button on the right.

What is Offline First Mobile?

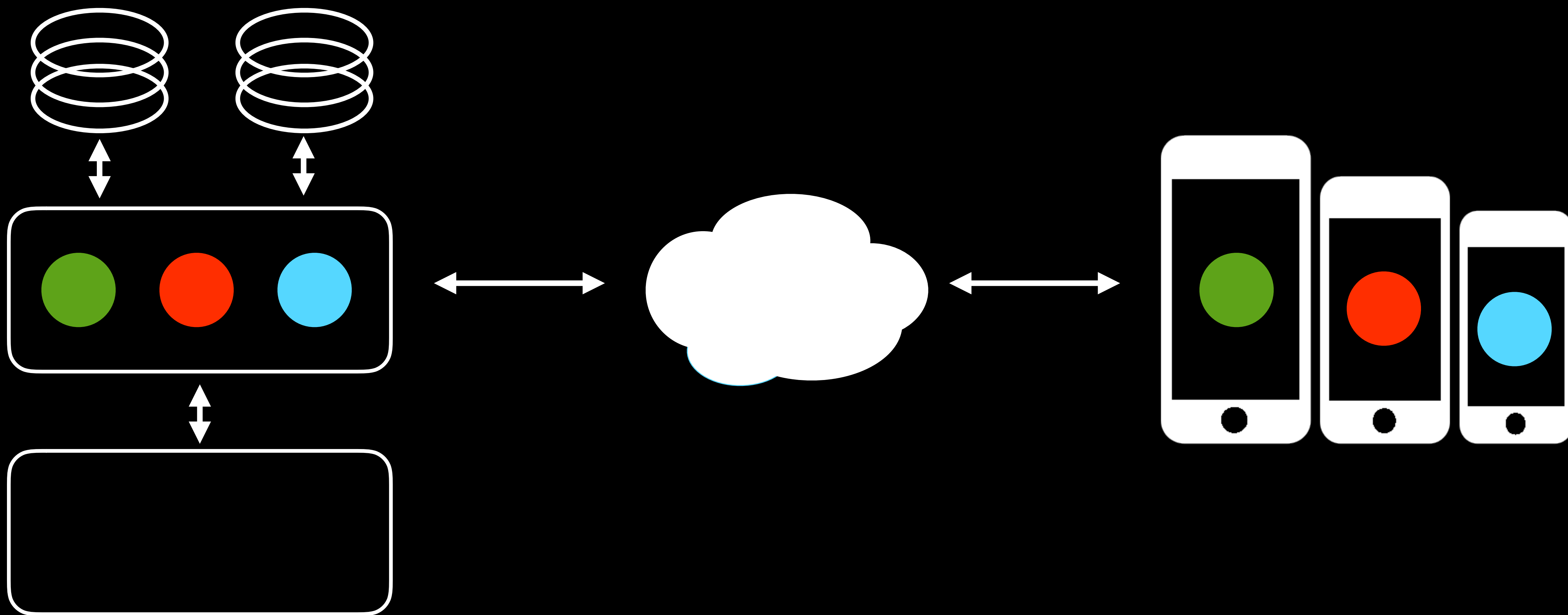


Offline First Mobile

- Handle All (or at least most) User Stories Offline
- Show All Available Data *quickly*
- Be Responsive When Loading Remote Data or Sending User Generated Data
- Report Network Status / Last Synced Time
- When Network becomes Available, *automatically* pull new data / push local data
- Note: it's *not* easy - but it's worth it



Typical Network Connected Mobile App





Typical Network Connected Mobile App

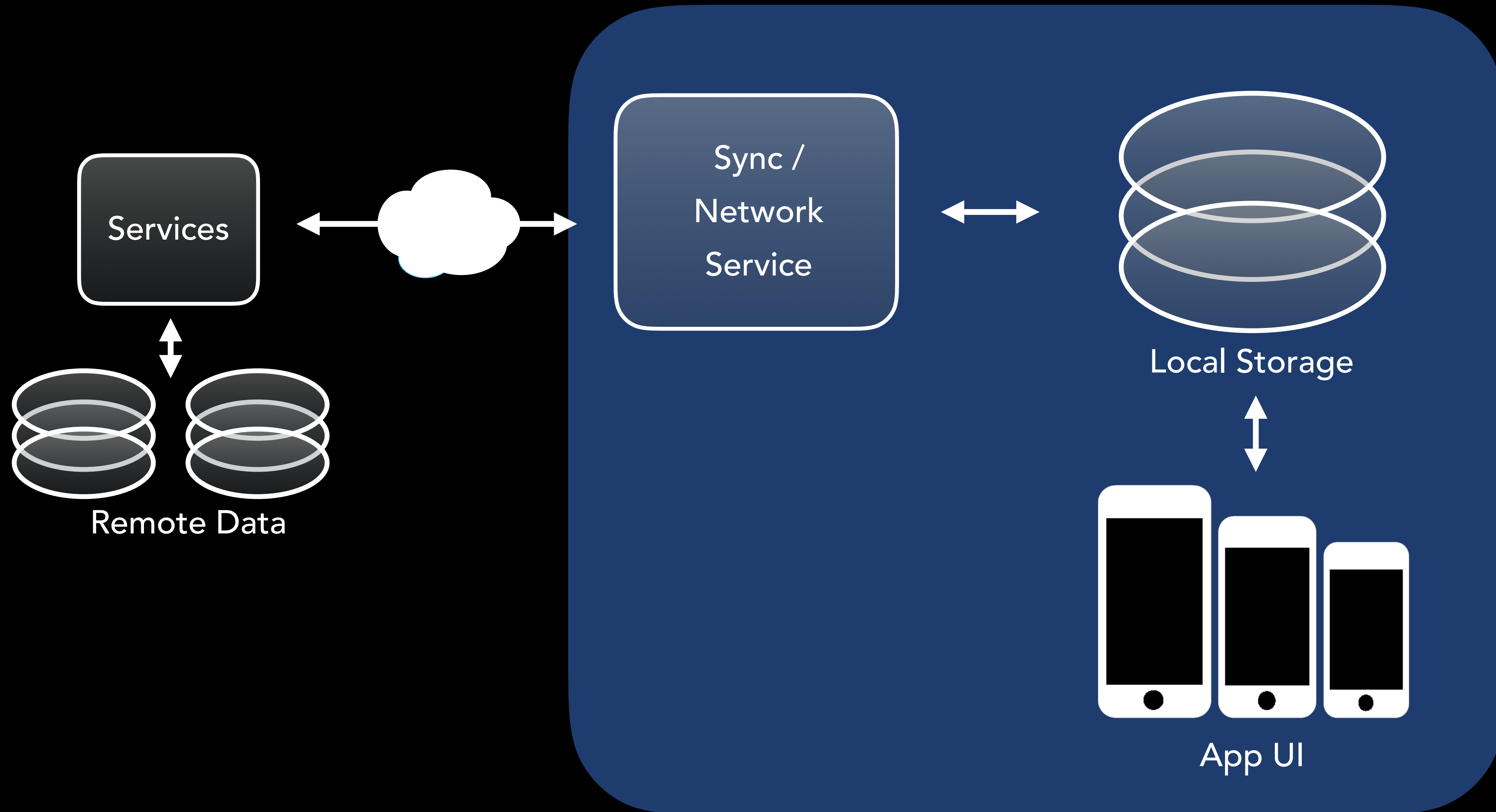
- Process:
 - Make API calls - receive lists of objects or individual objects
 - “Ingest” or process objects as needed; display directly or add to local storage
 - Update UI with processed objects



Typical Network Connected Mobile App

- Pros:
 - Simple, easy to understand process
 - Works well with good network connectivity
- Cons:
 - Poor user experience with bad network conditions
 - Poor user experience with captive portals
 - May not work at all offline

Offline First Mobile App





Offline First Mobile App

- Process:
 - App UI queries local storage / displays objects
 - Sync / Network service refreshes objects
 - Can be kicked off from App UI, or
 - Can be kicked off before needed
 - Sync / Network service can notify local storage / App UI new data is available
 - App UI queries local storage / displays objects



Offline First Mobile App

- Pros:
 - Much more responsive to user; no waiting for initial display or loading from network
 - Clear separation of concerns in code
 - Works well offline, with bad network conditions, and good network connectivity
- Cons:
 - More complex process
 - May require more “edge case” handling
 - May require extra UI to illuminate sync status
 - More challenging to implement and get right

**What About My Network-
Dependent Social App?**



Network Dependent Offline Considerations

- Fail Gracefully
 - Don't make the user wait (long)
 - Show what you have (corollary: you should have **something** to show)
- Let users access the content they have created
- Let users create content offline

Offline First Testing Considerations

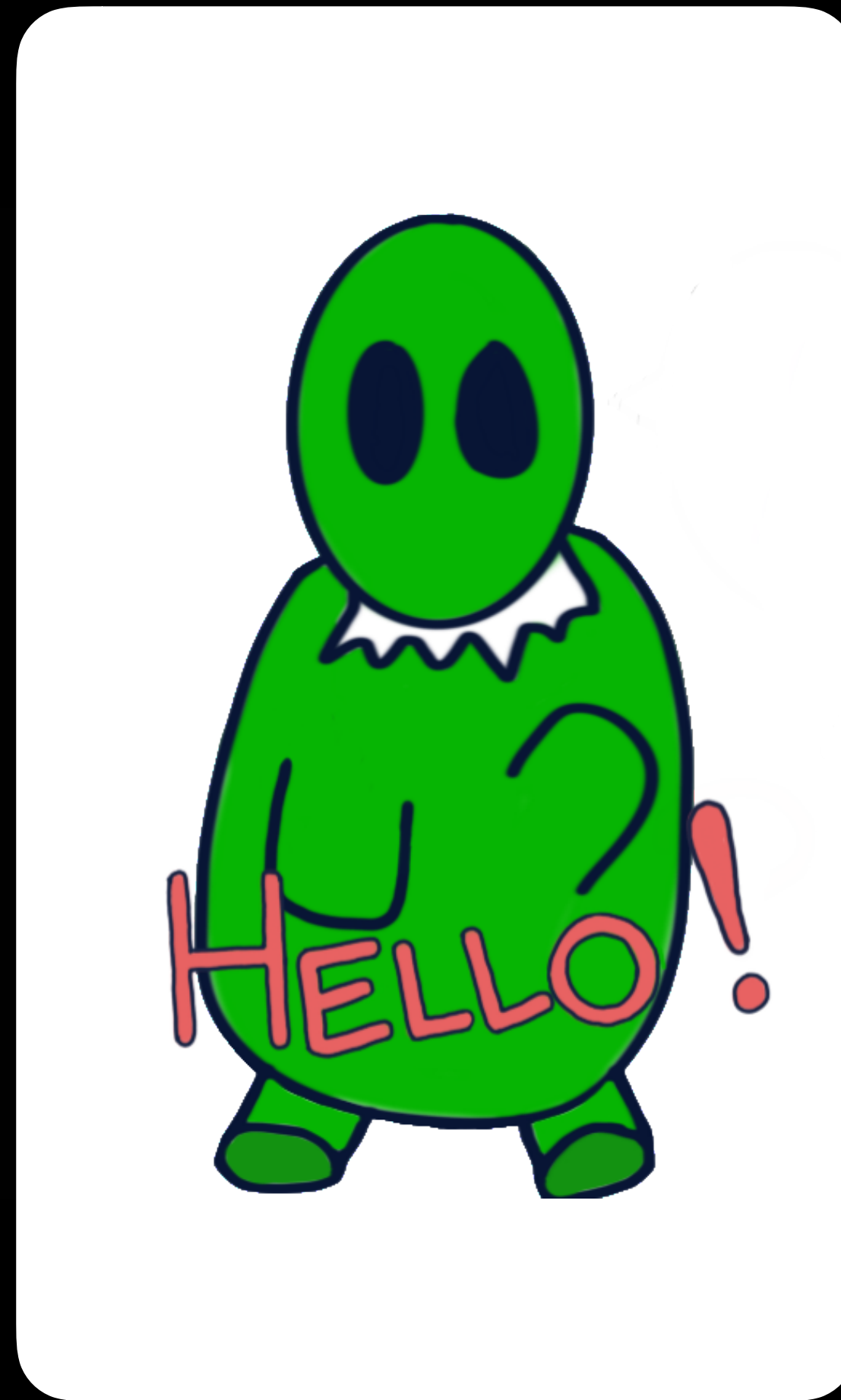


Offline First Testing Considerations

- Authentication
- Downstream Data Syncing
- User Generated Data Syncing
- Image Loading
- UI Indications

Authentication

Authentication



Authentication

User Name

Password

Sign in!

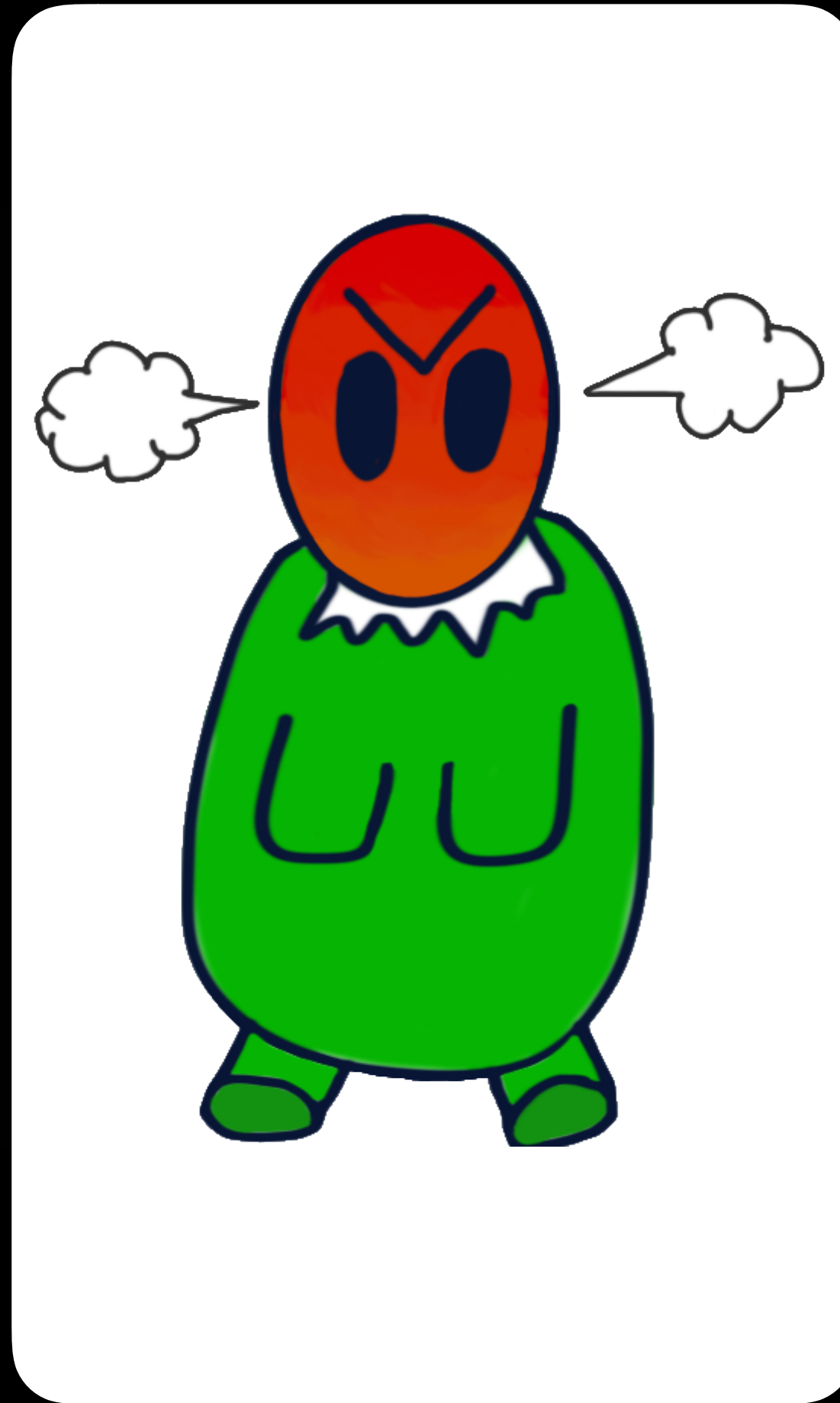
Authentication

NO RORI
FOR YOU!!!

HAHA YOU
ARE OFFLINE...

Sign in!

Authentication





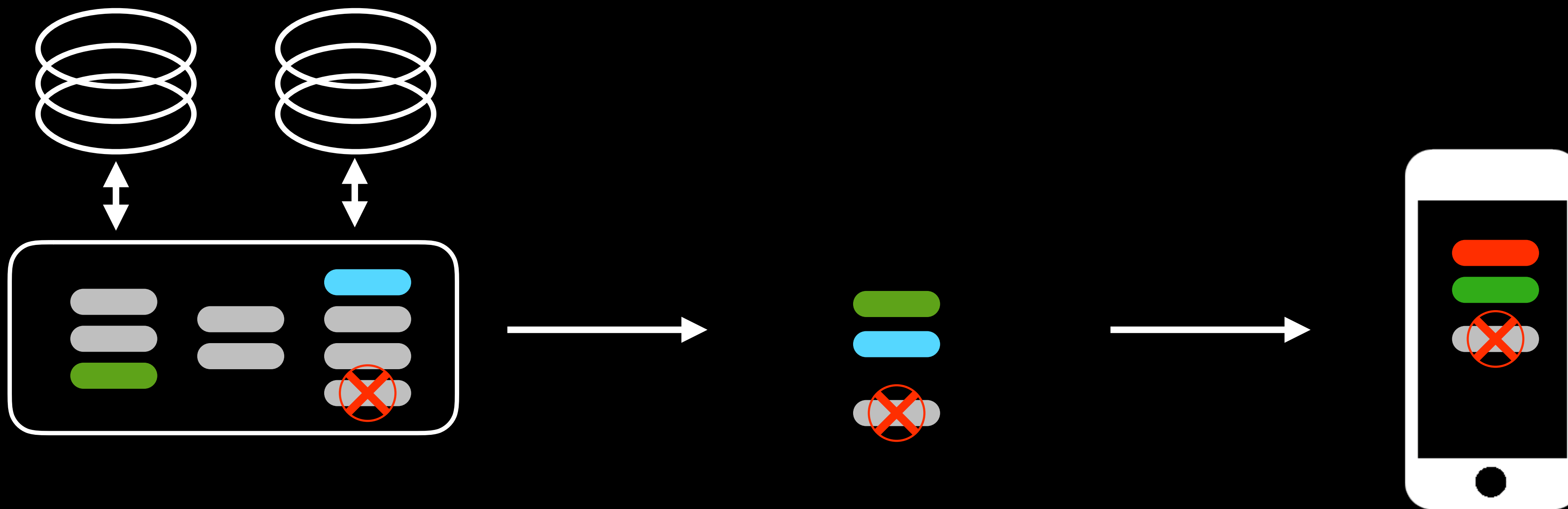
Authentication Testing Considerations

- Make sure Authentication method handles offline
- New Account / Not Logged In (offline / online)
- Logged Out State / Logging In (offline / online)
- Logged In State / Logging Out (offline / online)

Downstream Data Syncing



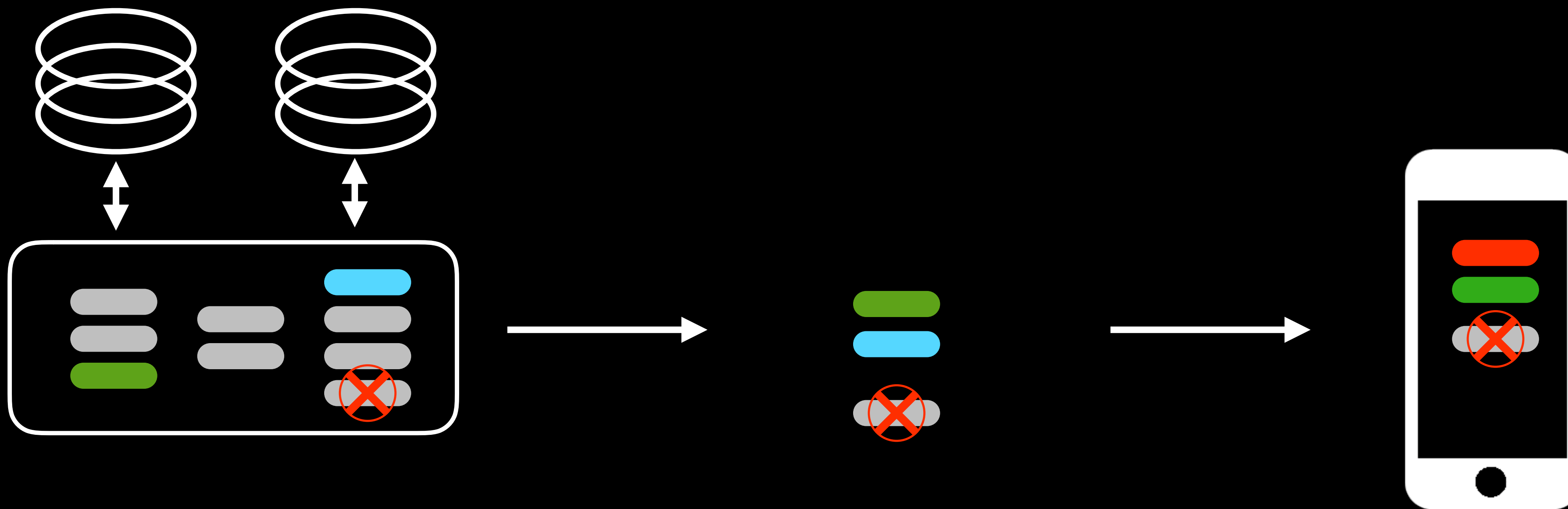
Downstream Data Syncing



<https://sync.server.com/allmystuff?since=20180211>



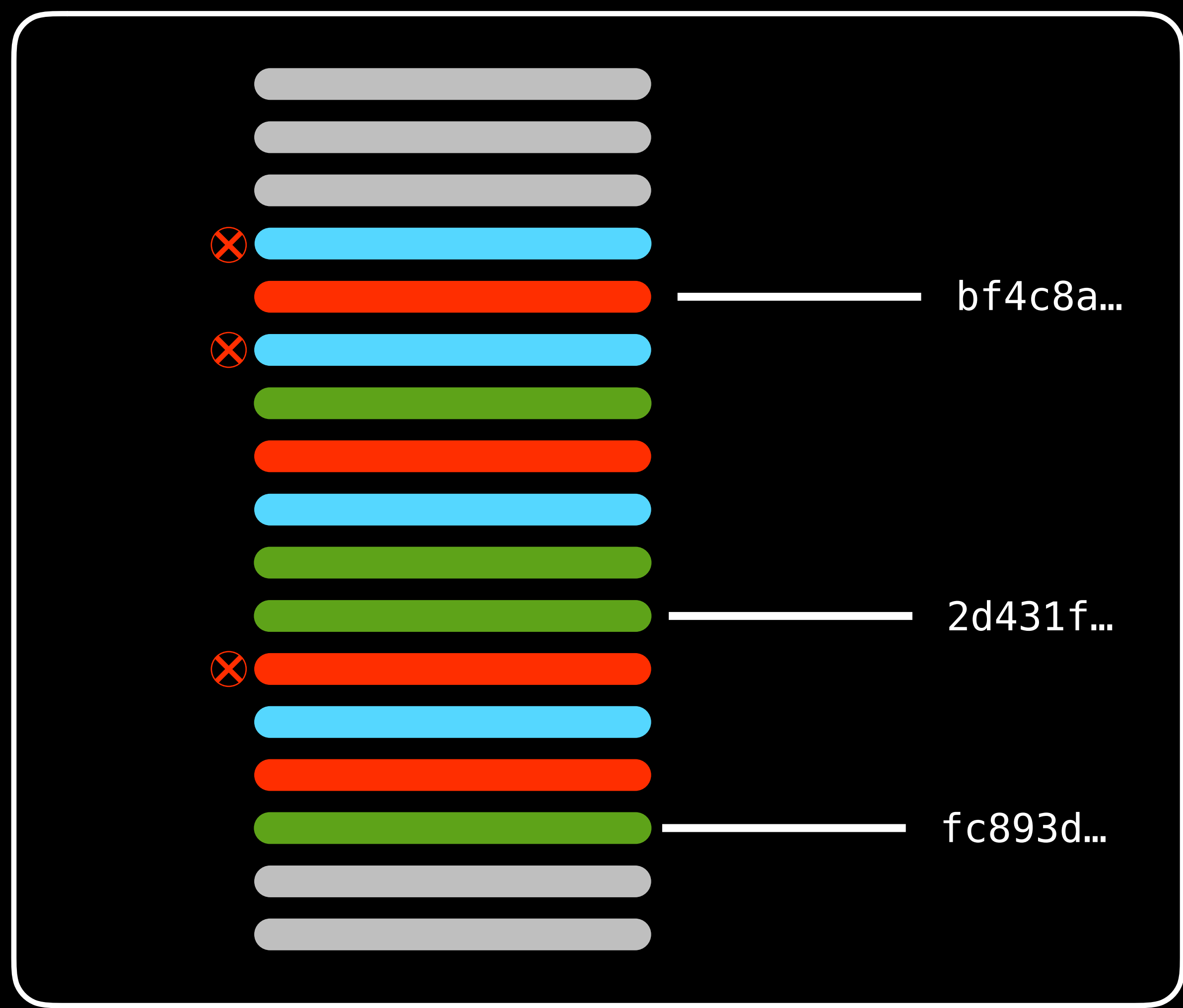
Downstream Data Syncing



<https://sync.server.com/allmystuff?since=af5eb3c9>



Downstream Data Syncing





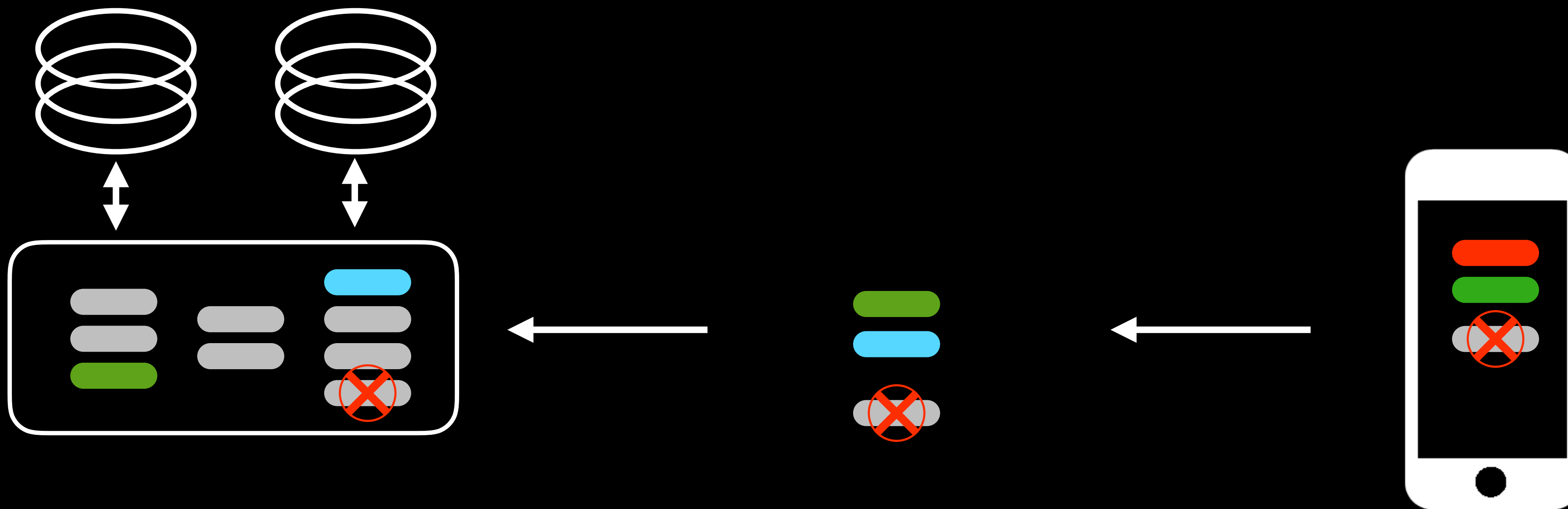
Downstream Data Syncing Testing Considerations

- Pre-seeded / included data
 - Test initial state correct
 - Test updates to included data
- Post Installation Data
 - Adds / Changes / Deletes*
- Test version upgrades with changes
 - Included data
 - Schemas

User Generated Data Syncing

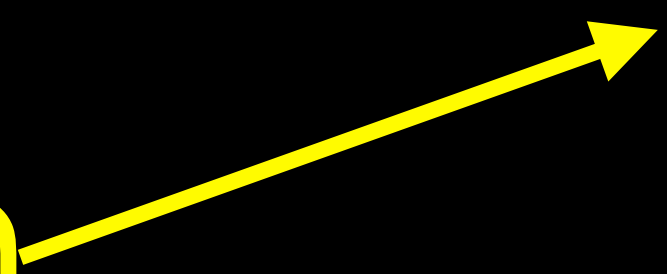
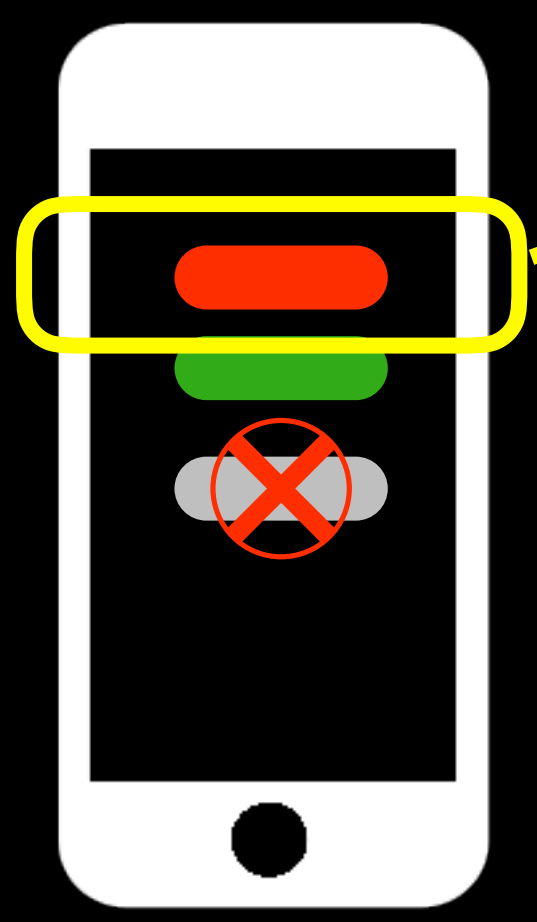


User Generated Data Syncing





User Generated Data Syncing



“Great Pic!”

```
{ "imageName" : "Pretty Mountain", ... }
```



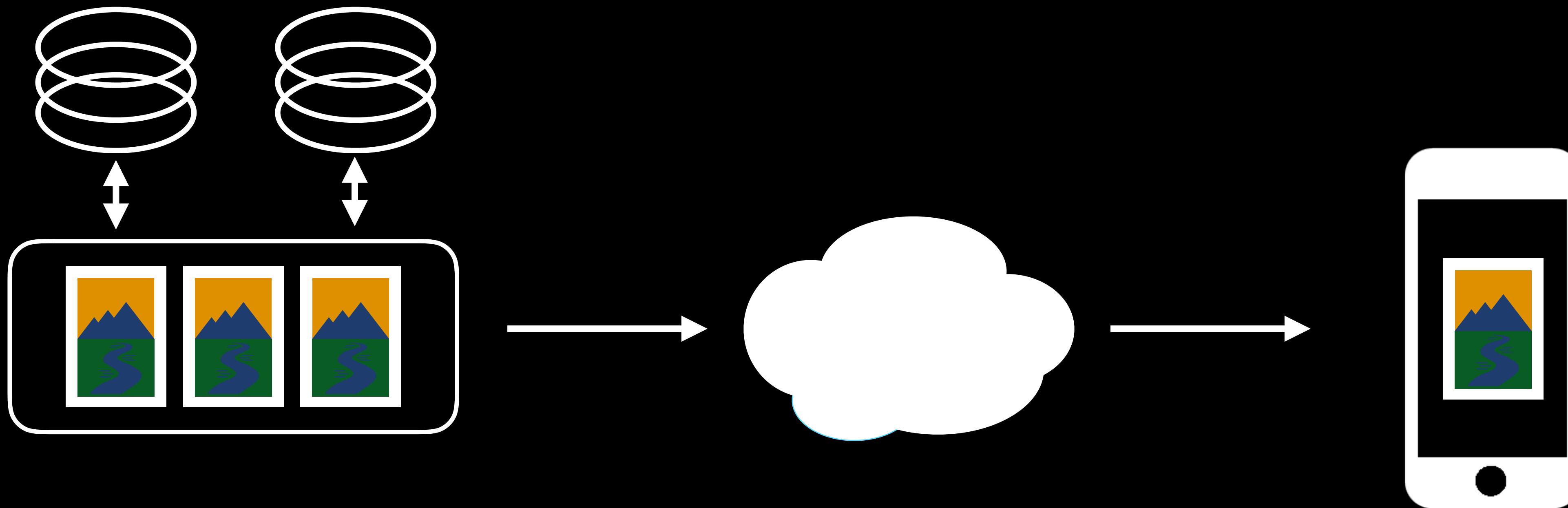
User Gen'd Data Syncing Testing Considerations

- Confirm basic data upload
- Confirm dependencies
- Test failures / retries
 - Look for duplications or other unexpected artifacts
- Check visibility offline/pre-sync and post-sync
 - Look for weirdness in refresh after sync
 - Ensure sorting (or other data-driven presentation) consistent pre-sync and post-sync

Image Loading



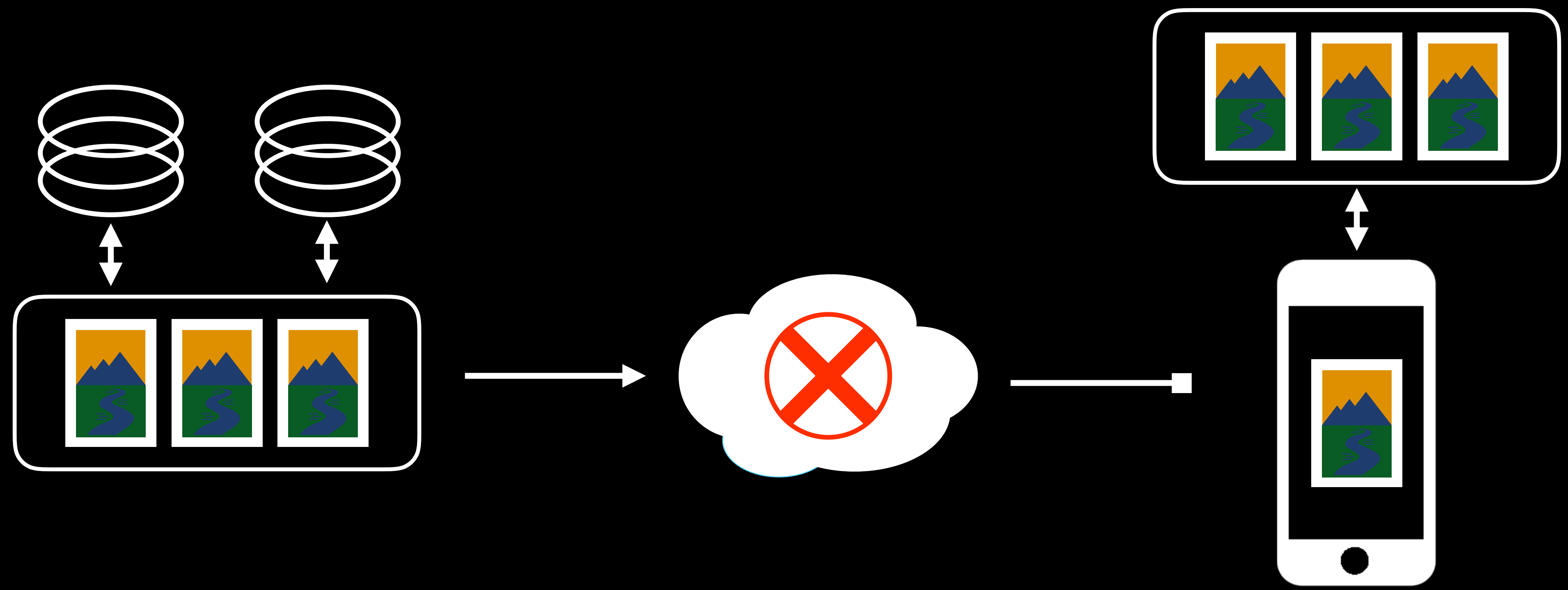
Image Loading



<https://sync.server.com/images?name=mountain.jpg>



Image Loading



<https://sync.server.com/images?name=mountain.jpg>



Image Loading

- Leverage HTTP standards for remote image loading
 - eTag
 - Expiration
- On iOS, URLCache/NSURLCache
- On Android, HTTPResponseCache
- You **can** customize to load cached first if present, then attempt to load remote (overriding HTTP standard)



Image Loading Testing Considerations

- Remote Loading, connected
- Loading, offline from cache
- Check “expired” images, ensure don’t load from cache
- Check changed images (different eTag), ensure remote image loads
- Check unchanged images (same eTag), ensure cached image loads / no network traffic (other than HEAD call)

UI Indications



UI Indications

- Show connection status *when it matters*
 - When user expects new data to load
 - Entering screen that shows new content
 - Pull to refresh
 - When user creates new data / expects to upload
- Don't make user wait for data to load
- Show progress *as accurately as possible*
- Show last synced time when it matters



UI Indication Test Considerations

- Test each connection state reflected correctly
- Test *transitions* between states
 - Slow transitions (simulate turning airplane mode on/off)
 - Fast transitions (simulate train going in and out of tunnels/stations - the “BART” test)

Mobile Test Techniques



Mobile Test Techniques

- Connected
 - Wifi
 - Cellular
- Offline
 - Airplane Mode
 - Wifi on, but not connected
 - Cellular on but signal not available (basements, some buildings, remote valleys, tunnels, microwave)



Mobile Test Techniques

- Poor Connection
 - iOS: Network Link Conditioner
 - Android: Emulator Extended Controls: Cellular
 - Android: 3rd Party App
 - Charles or similar
 - If all else fails...
 - Coffee can under your desk in the basement!



Mobile Test Techniques

- Captive Portal
 - Hotels / Coffee Shops / Conferences ;) / Airports
 - Set gateway address incorrectly in router
 - Parental Controls
 - Charles - null route

questions?



@jwkeeley



THANK YOU

martiancraft.com



TestCon Europe 2019