500px

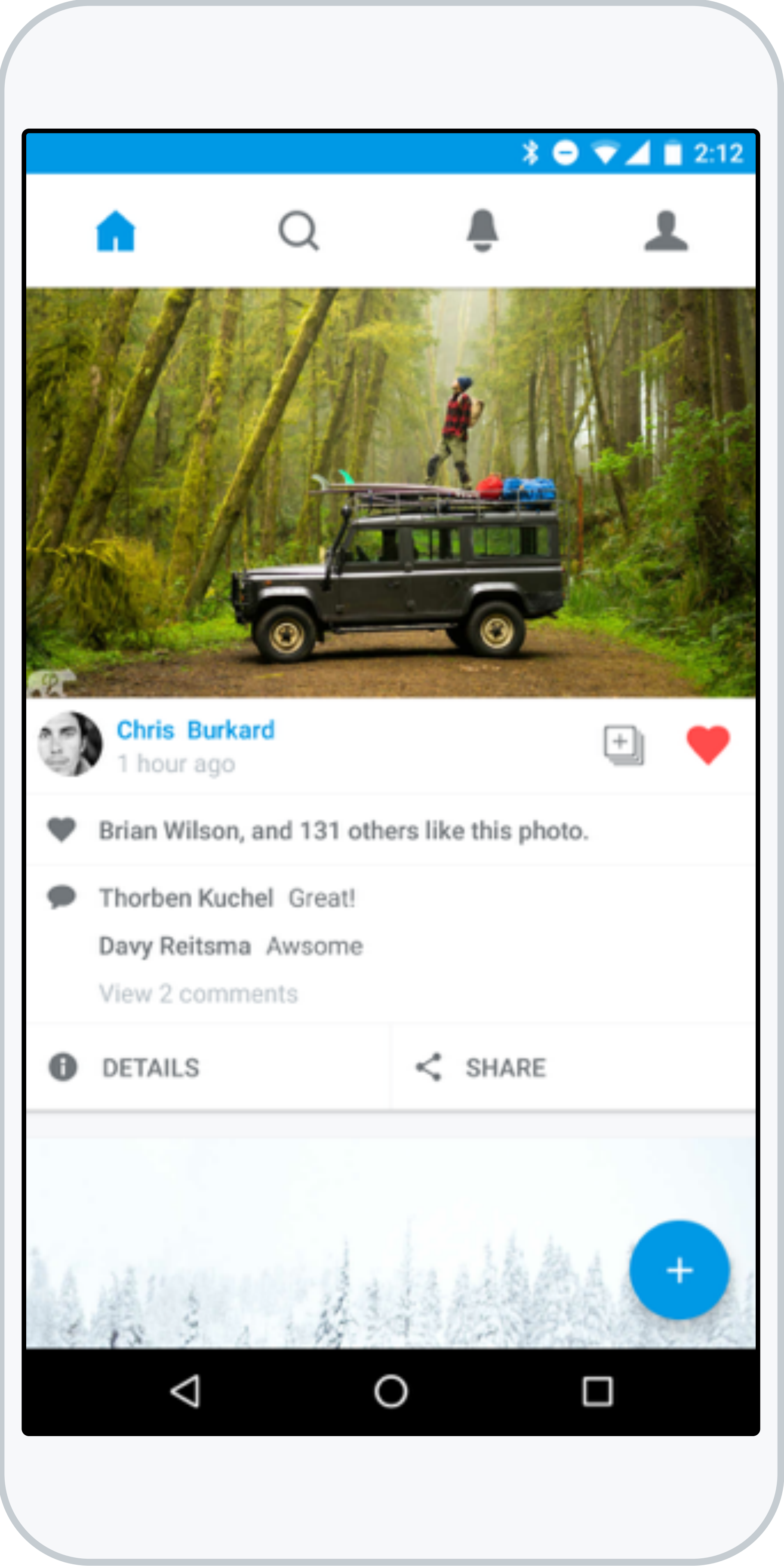
Mobile V4 Lessons Learned

What we learned building our latest iteration

Julian Villella
Software Developer, Mobile Team

Why a Mobile V4?

- Make the experience more approachable for all new and existing users
- Give more emphasis on the photos
- Give more emphasis on the photography *you* are interested in
- Simplify affection actions
- Work with an updated codebase!



2:12

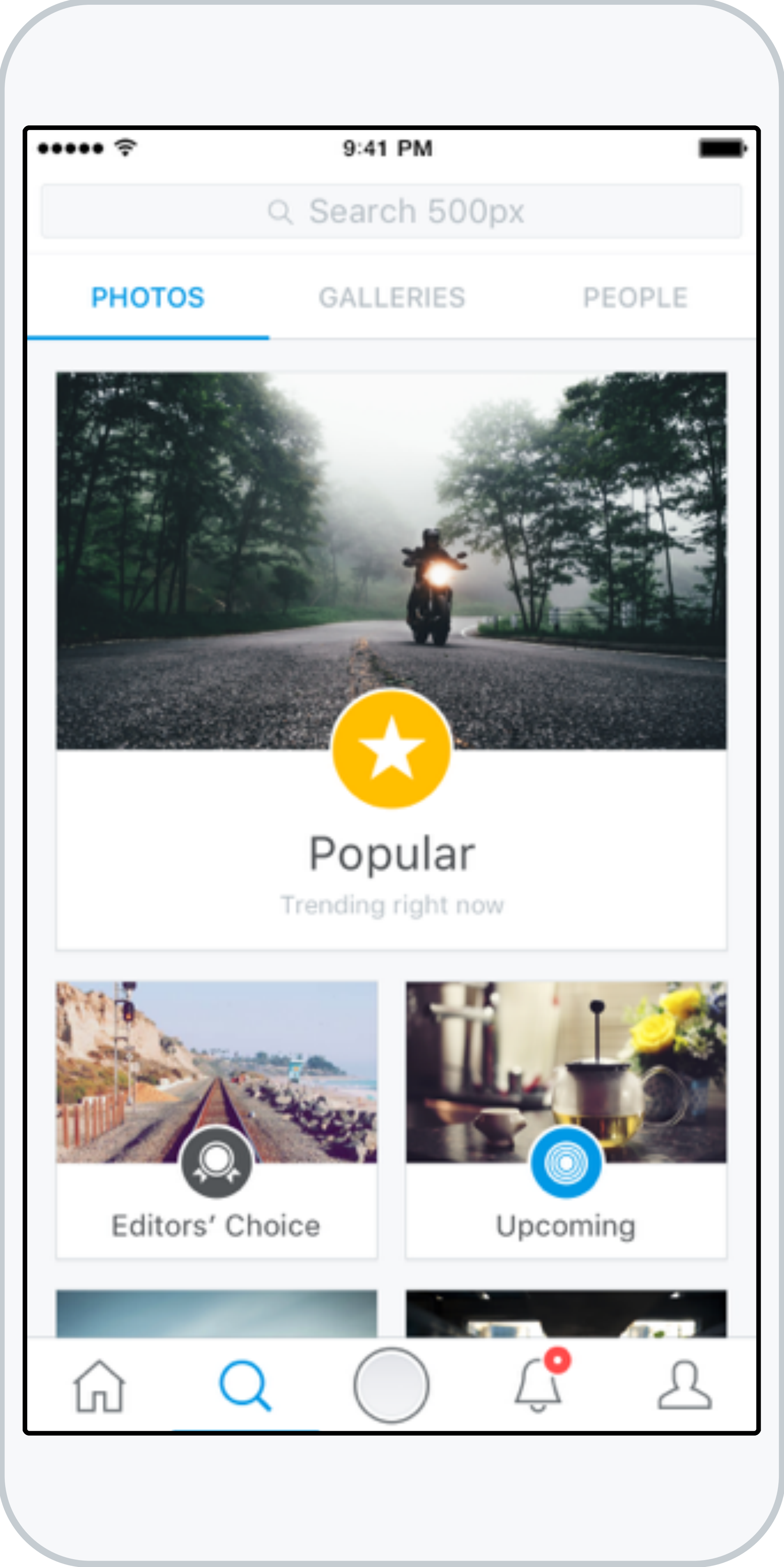
Chris Burkard
1 hour ago

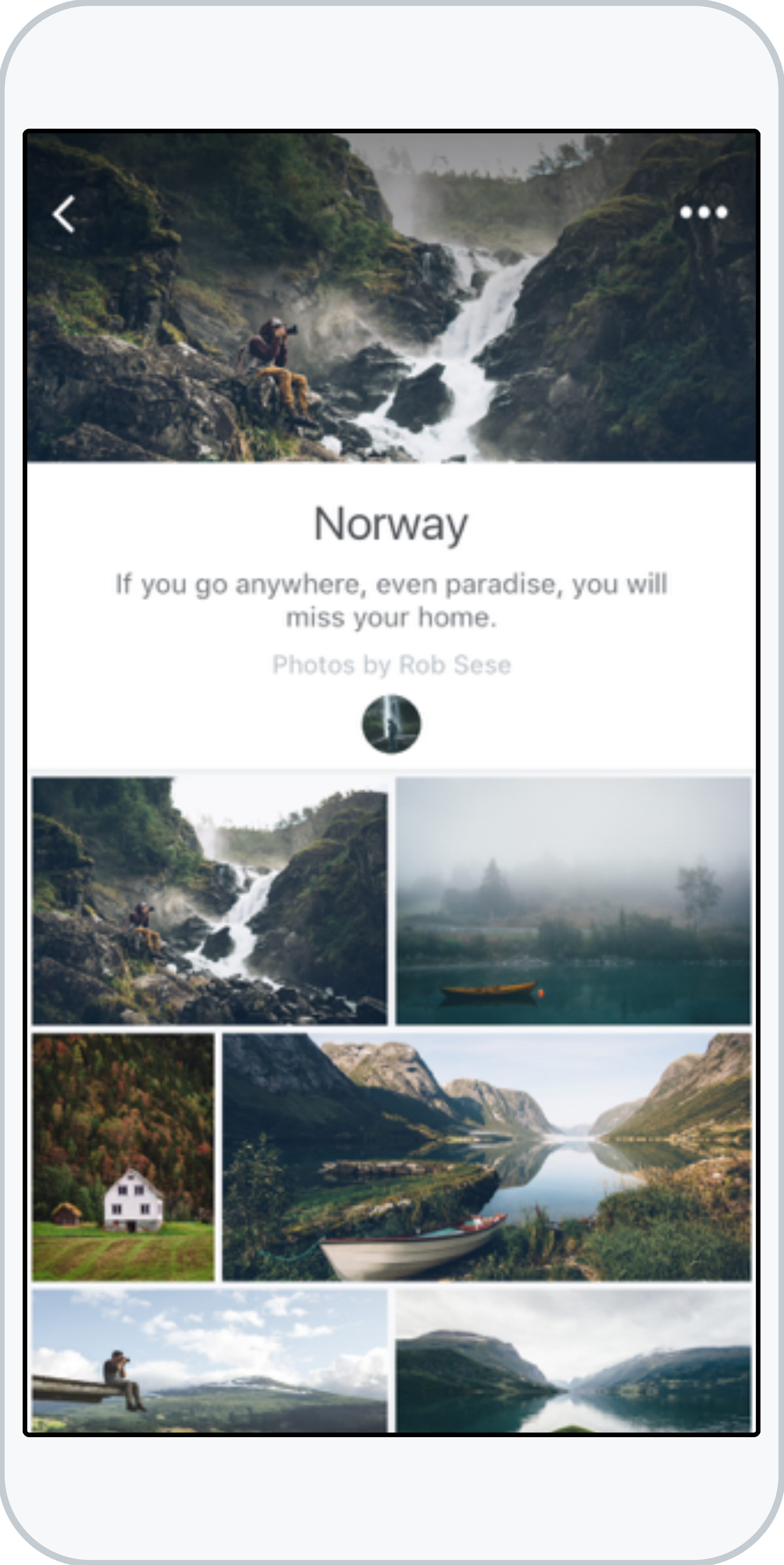
Brian Wilson, and 131 others like this photo.

Thorben Kuchel Great!

Davy Reitsma Awsome

View 2 comments




Norway

If you go anywhere, even paradise, you will miss your home.

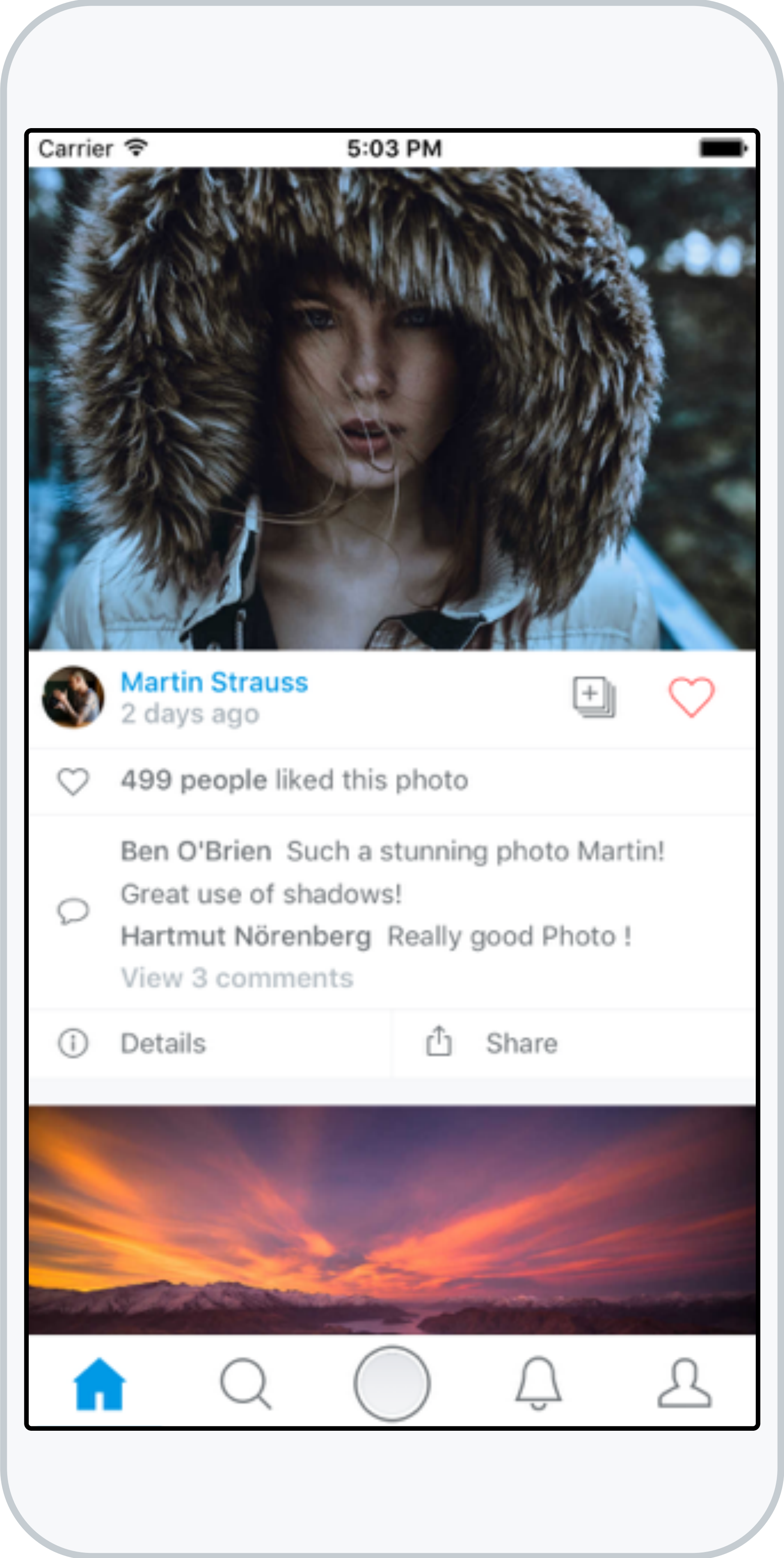
Photos by Rob Sese

Rewriting the iOS Client

Starting Fresh

iOS V4

- **Were,**
  - Using Core Data for caching
  - Universal binary but no UI code sharing between iPhone and iPad
- **Now,**
  - Moved data layer to PF500, including in-memory caching
  - More adaptable UI with AutoLayout
- **But,**
  - A problem started to emerge in regards to model consistency



Martin Strauss
2 days ago

499 people liked this photo

Ben O'Brien Such a stunning photo Martin!
Great use of shadows!
Hartmut Nörenberg Really good Photo !
View 3 comments

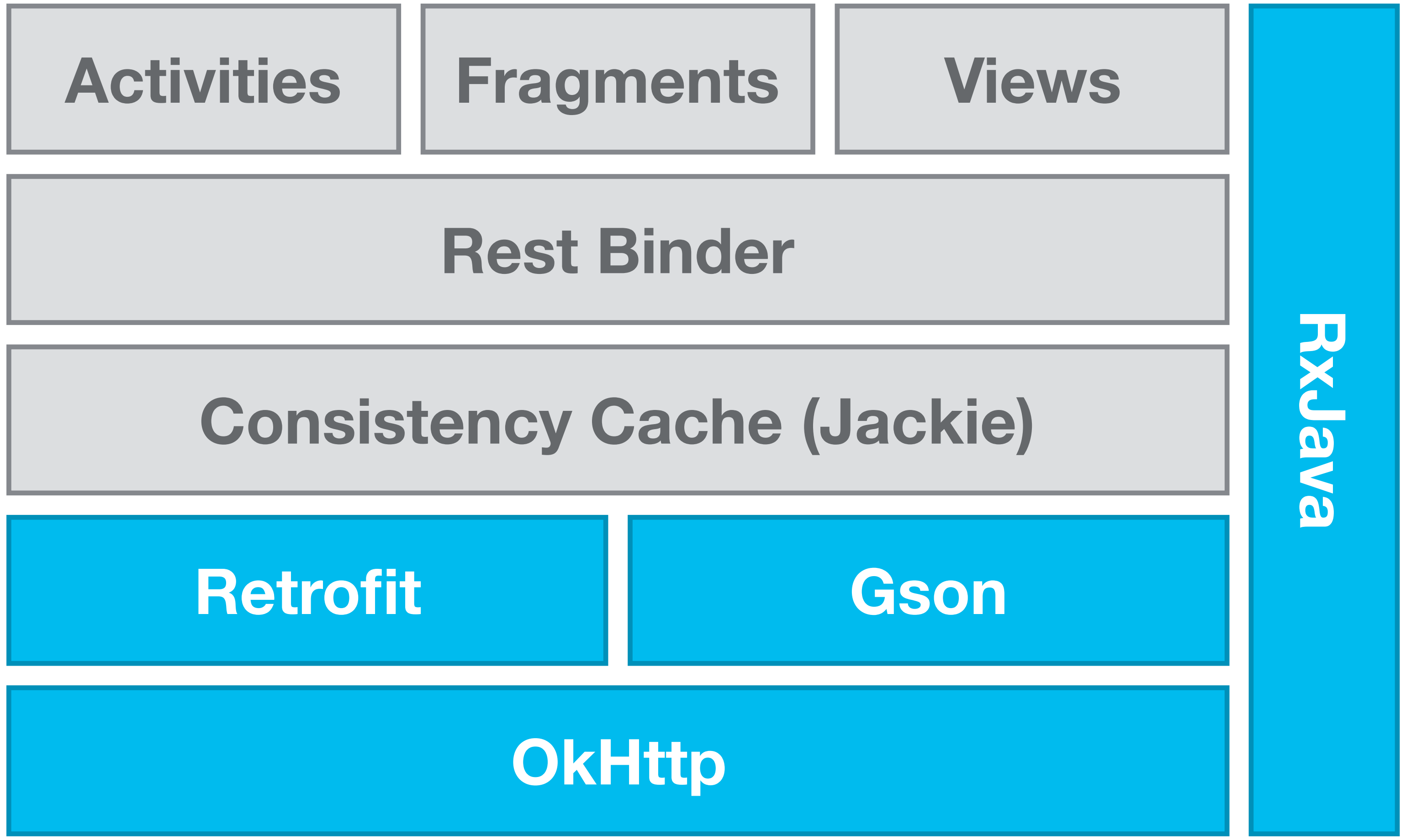
On to Android

Lessons Learned

Let the *foundation* do the heavy-lifting.

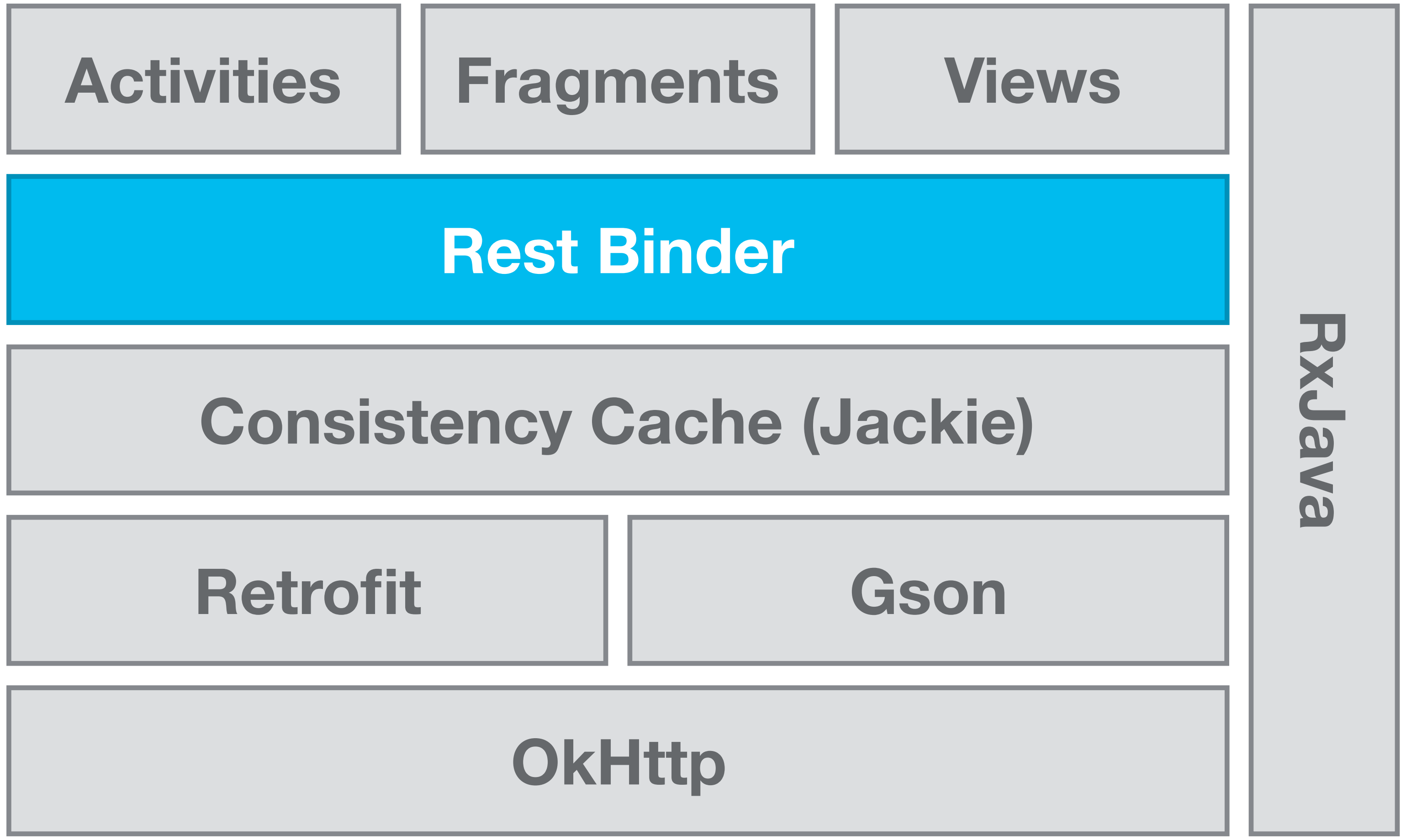
Android Data Layer

- **OkHttp serves** our **network requests**
  - Maintains connection pooling
  - Transparent gzip compression
  - Response caching
- **Retrofit**
  - Puts HTTP API behind a **Java interface**
  - **Typed responses** through Gson
- Gson - JSON ↔ POJO
- RxJava - **subscribe** to Retrofit responses



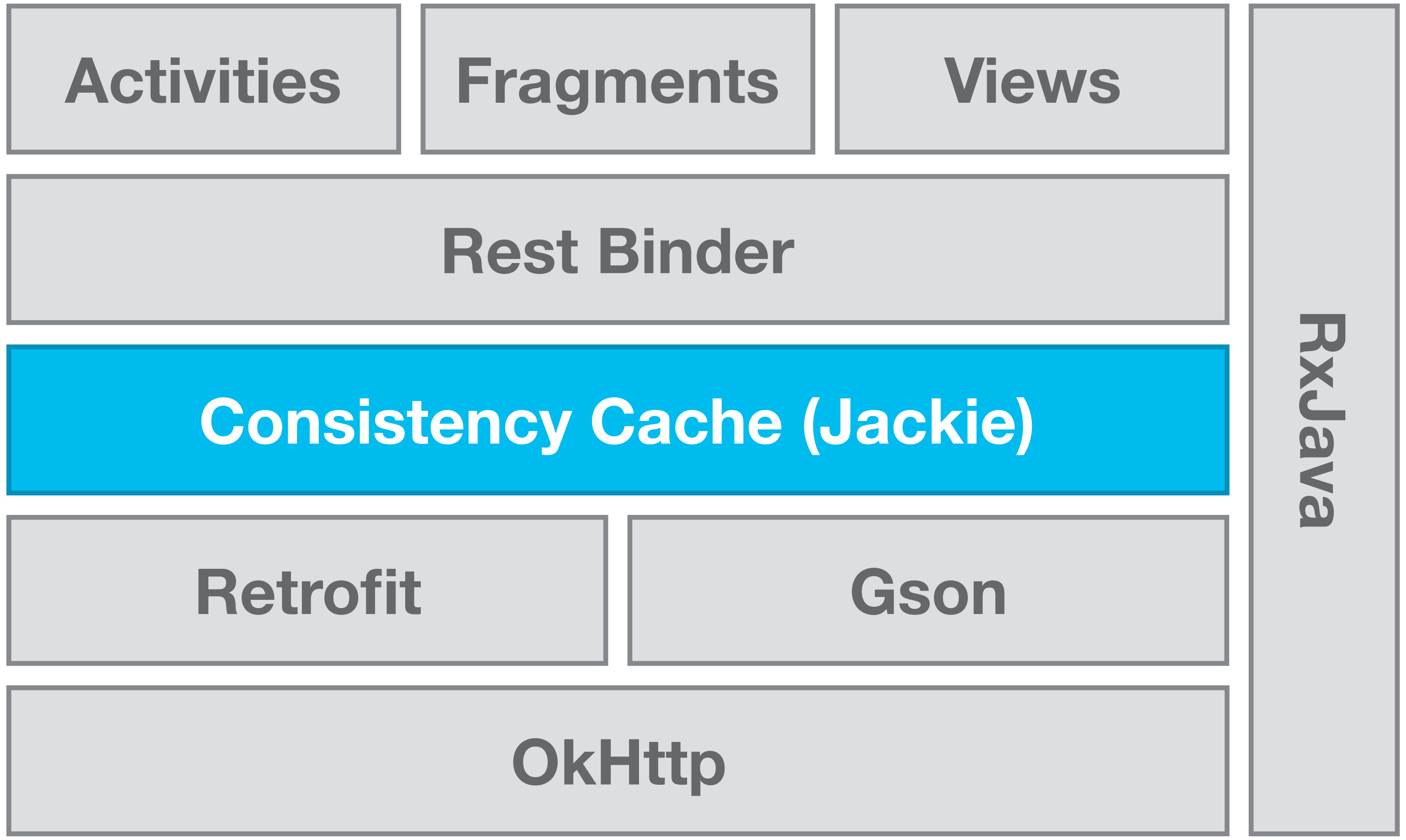
Android Data Layer - Endpoint Binding

- Fragments and activities **bind** to an API endpoint via the **Rest Binder**
- **Lifecycle** callbacks **(un)subscribe** to Rest Binder
- Rest Binder deals with **fetching**, **pagination, refresh**
- Rest Binder **delegates caching** responsibilities to **Jackie**



Android Data Layer - Consistency Cache

- Provides an interface to **subscribe** to **objects and lists by unique identifiers**
- Get **notified** of updates, **UI reacts** accordingly
- Subscription is **tied to activity/fragment lifecycle**
- Cache policies
- Handle unique cases with update interceptors
- Immutable copies with **Lombok's Wither**



Activities

Fragments

Views

Rest Binder

Consistency Cache (Jackie)

Retrofit

Gson

OkHttp

RxJava

The Critical Path

Shipping on time

The Critical Path

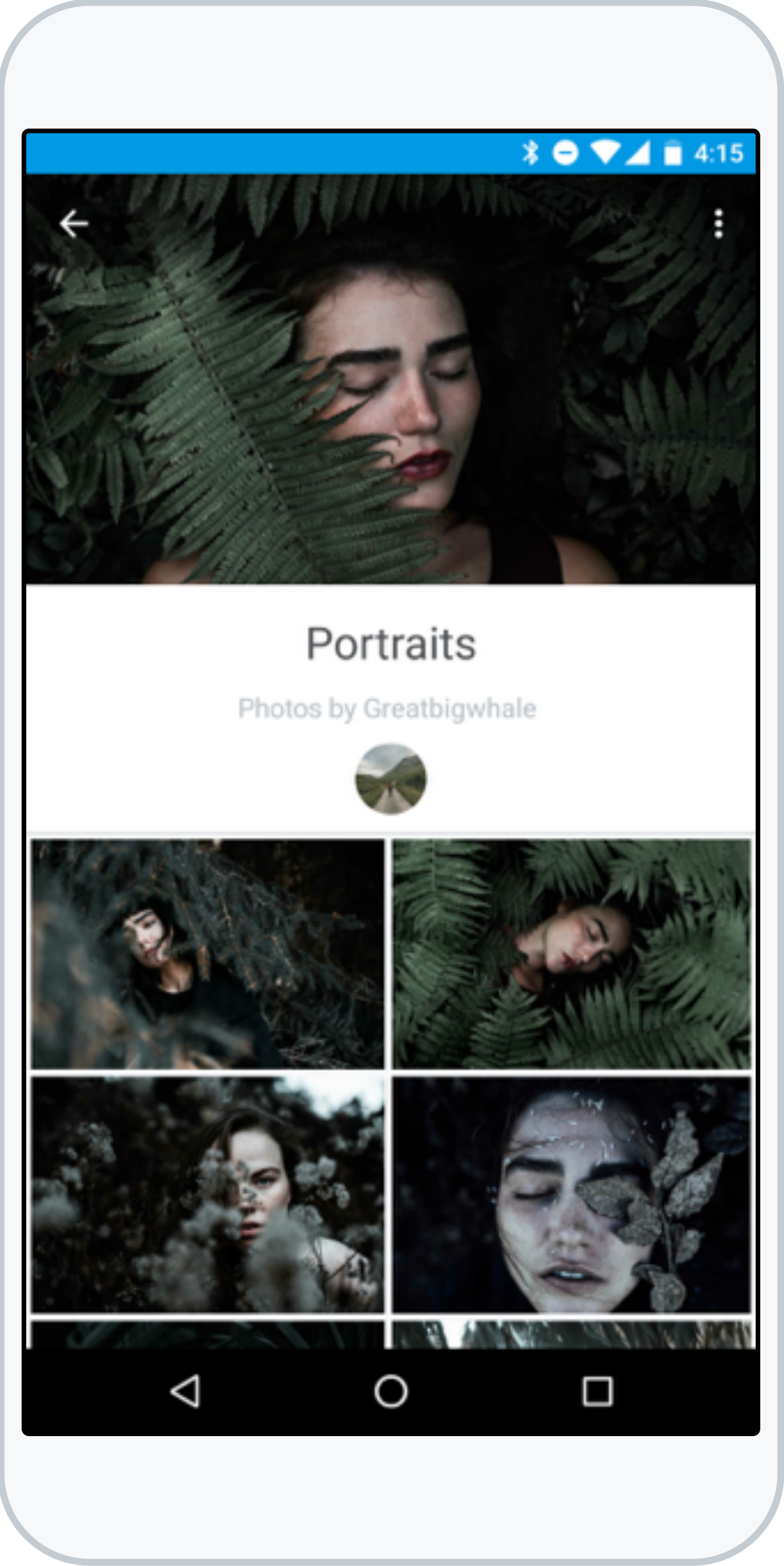
- War room
- Near-daily sprints
- Dog fooding
    - Daily internal company builds
    - Frequent external builds
- Internal bug bounty
    - Generated over 200 tickets
    - Leader with 50 tickets
- Positivity!

Where We Are Now

And what’s happened since then

Where We Are Now

- Double uploads since this time last year
- Addressed much user feedback from launch
  - Image loading times 1/5th the size (adaptive)
  - Streamlined release process
  - Automatic feed refreshing
  - Android 99.6% crash free
- Galleries was first synchronized release (Web, Android, iOS)
- Broadening our appeal to a wider audience
- Clean code base!



Open Source

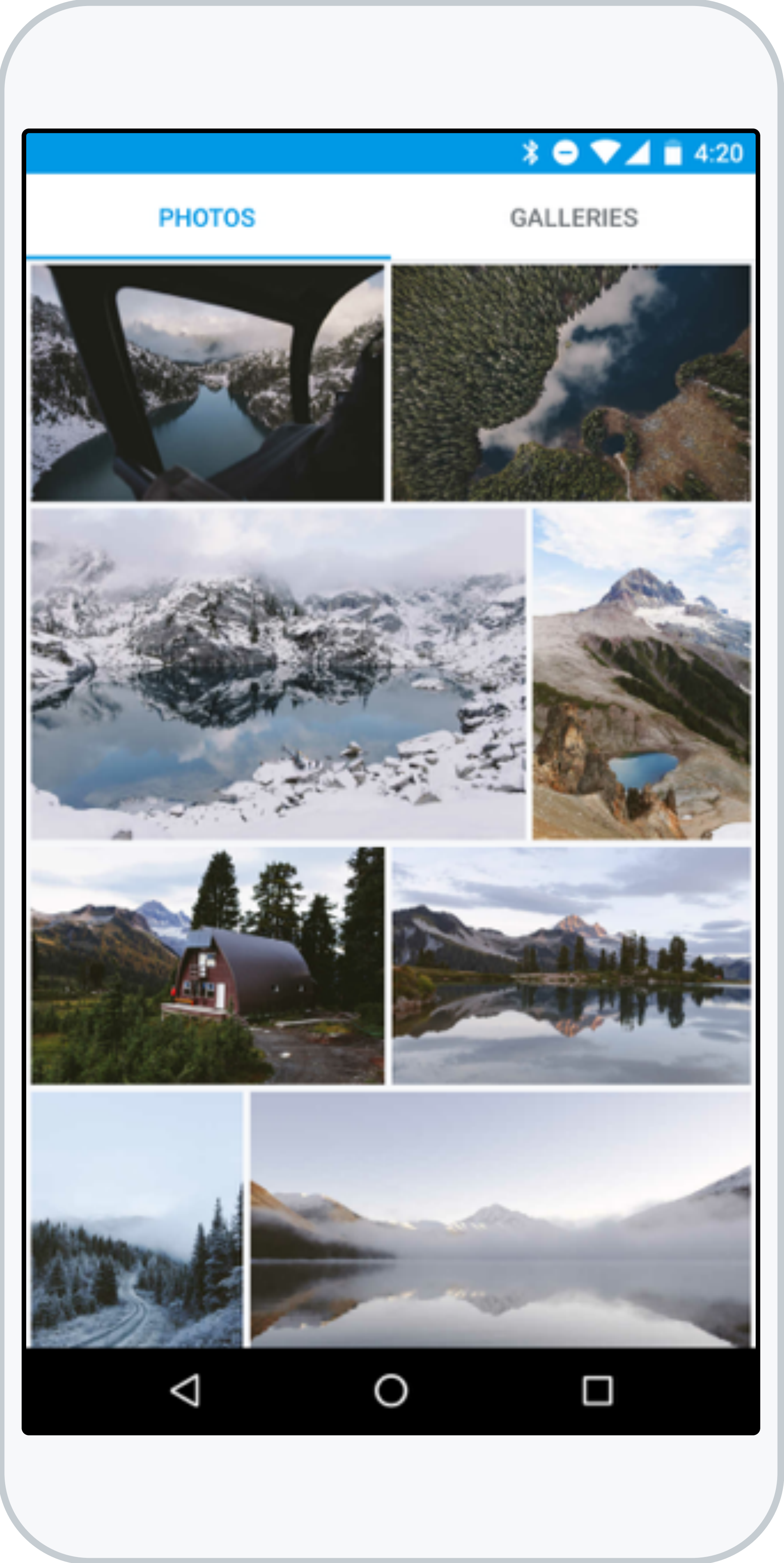
Beginning to open source some of the internal components we’ve been using

Updates to Android Blurring View

- Now a maven artifact you can include in your projects

Greedo Layout

- Today we are open sourcing the **aspect ratio grid layout** we use on **Android** and **iOS**
- https://github.com/500px/greedo-layout-for-android
- https://github.com/500px/greedo-layout-for-ios



Questions?