



# LEAN APP

# Instagram Architecture at Scale

Frank Du

[frank@instagram.com](mailto:frank@instagram.com)

Jun 9, 2017

GMTC 全球移动技术大会, 北京 2017

**GMTC** 2017  
GLOBAL MOBILE  
TECH CONFERENCE  
全球移动技术大会

# What is Lean App?

Keep the app **small** and **compact**. Measurable metrics are:

1. Binary size
2. Build time
3. Java method count (Andorid only)

Today, we'll use Instagram Android app as example for discussions.





# BIO: FRANK DU

1. 2011- 2012, One of Old Guys on FB Android App
2. 2012, Started & Shipped Native News Feed
3. 2013 - 2016, Built & Managed FB Growth Android Team
4. 2014-2015, Started FB Lite App with a Small Team
5. 2016 - now: Building Instagram Camera Team. *Just Shipped Face Filter on May 16, 2017*

# TODAY'S MENU

1 Brief History of Instagram Android

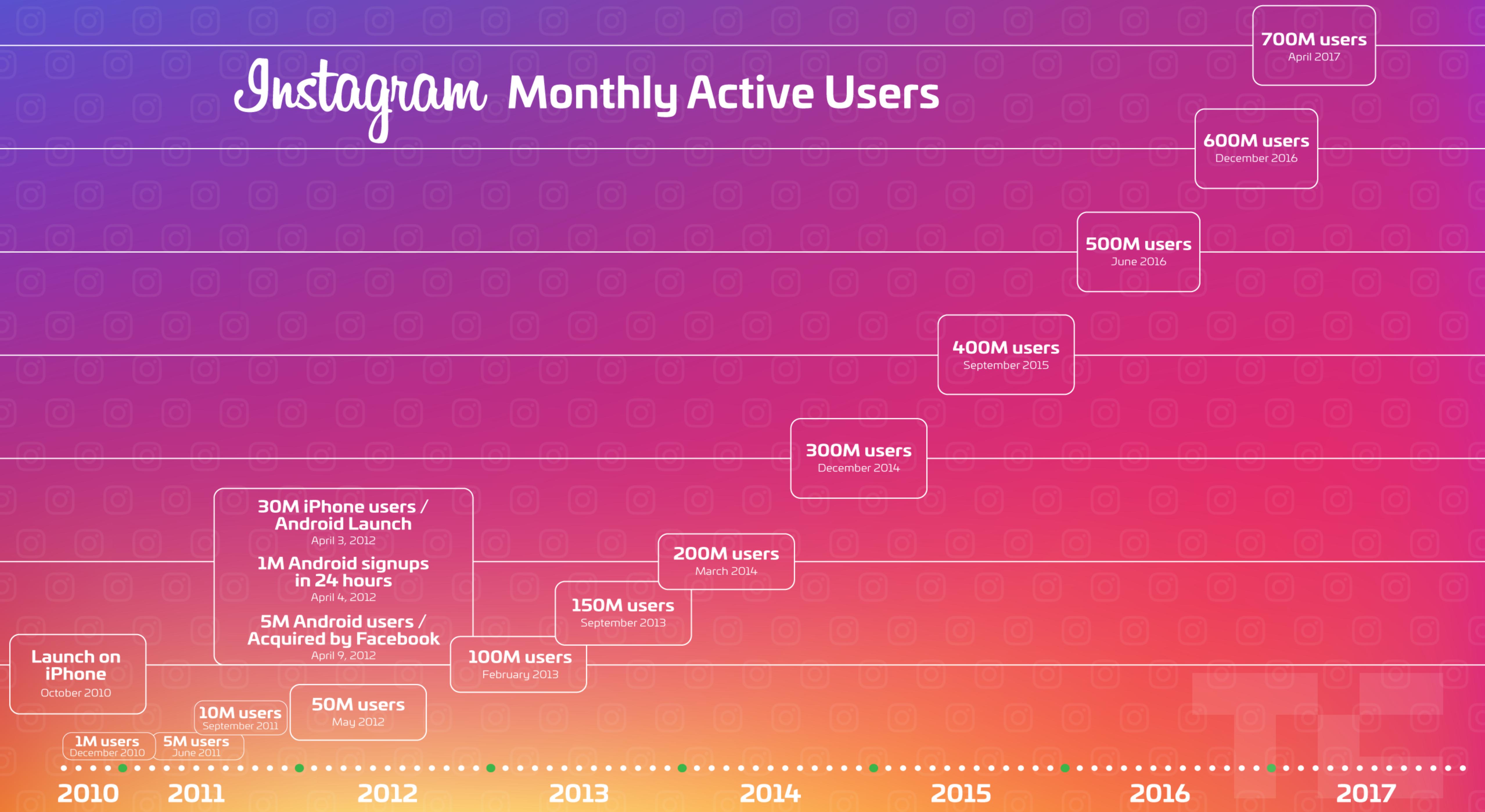
---

2 Challenges in Today's Android Development

---

3 Lean App Strategy

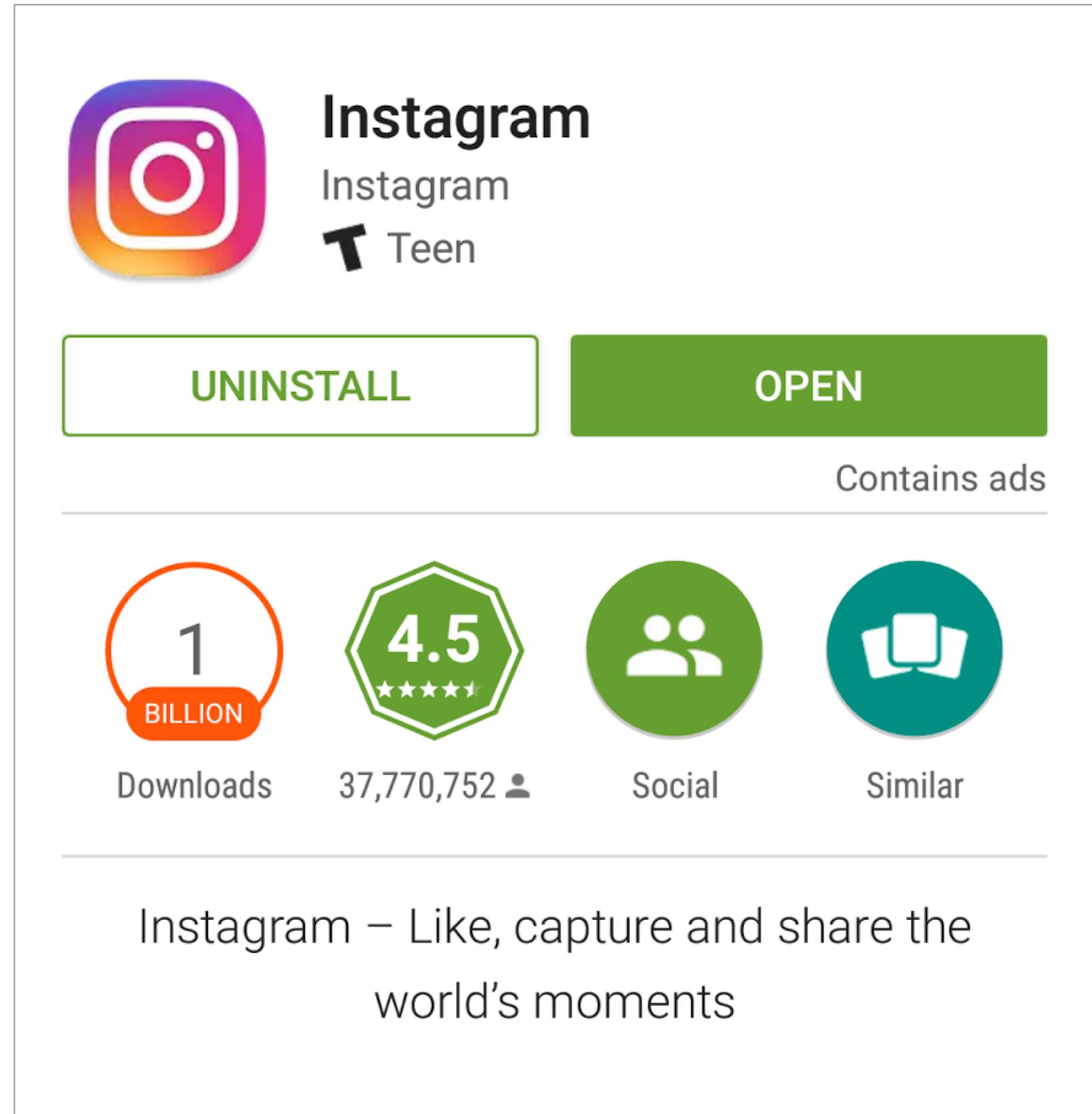
# Instagram Monthly Active Users



# HISTORY OF THE ANDROID APP

## A very successful product

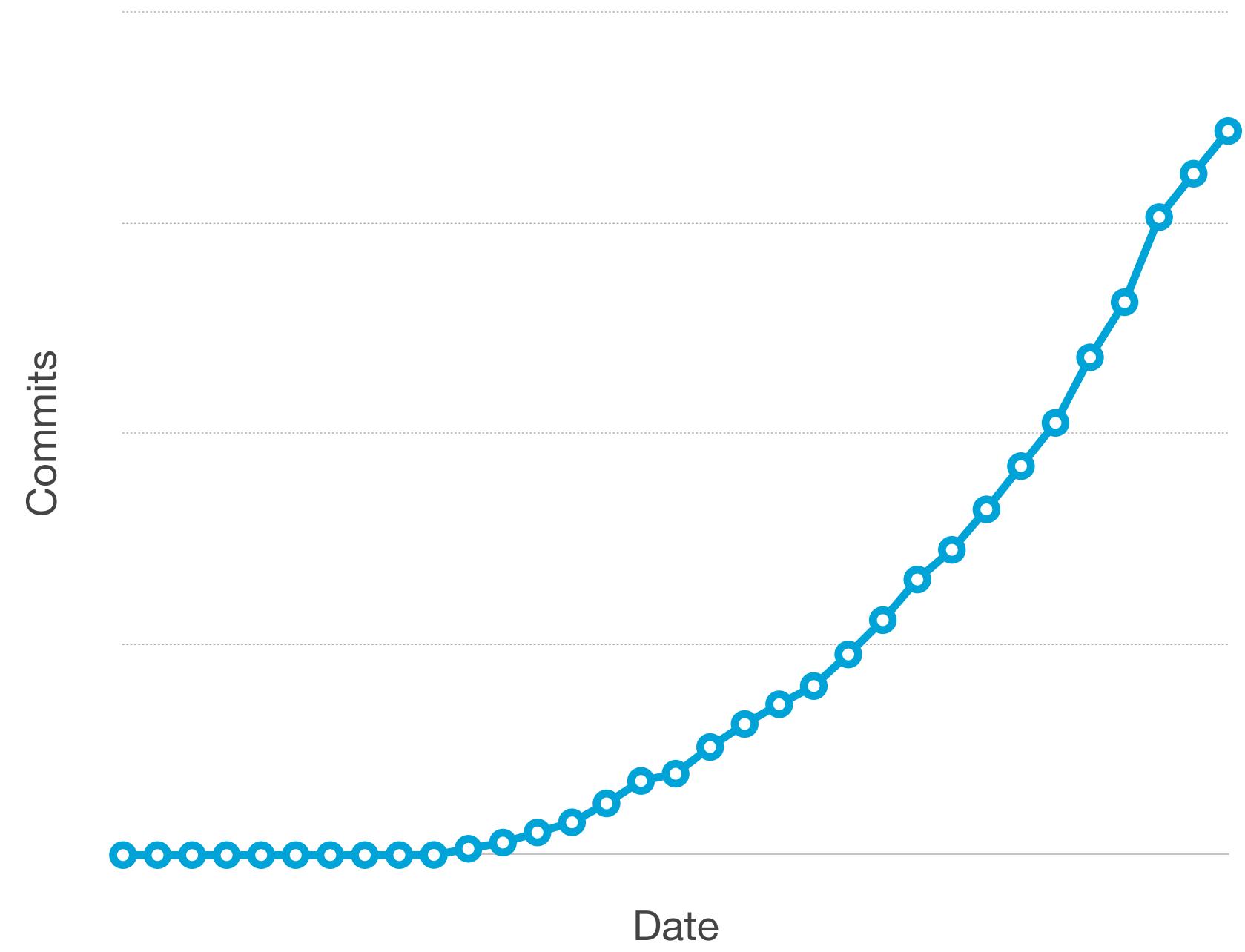
- First Android version released in 2012
- More than **700 Million** active users (iOS + Android)
- More than **one billion** downloads in Play Store
- **4.5 star** rating since day one



# ANDROID CODEBASE HISTORY

## A well-maintained codebase

- The codebase is 5 years old, with Java at the heart of it
- From 2 engineers to more than **30** engineers
- Exponential increase of commits
- Maintain 3 apps in PlayStore
  - Layouts - **55%** code is shared from main app.
  - Boomerang - **60%** code is shared from main app.



# A COMPACT APP

## A race car

- One of the fastest Android apps in top downloads
  - According to [blog.nimbledroid.com](http://blog.nimbledroid.com)
  - Instagram Android is still a **single DEX** app
- The APK most people download is comparable to the first release in 2012
  - 22Mb v.s. 13Mb (Sliced APK)

# TODAY'S MENU

1 Brief History of Instagram Android

---

2 Challenges in Today's Android Development

---

3 Lean App Strategy

# CHALLENGES IN TODAY'S ANDROID DEV

Lots of influencing factors, with Greater Tension

- Device fragmentation
- Large user base
- Infinite maintain cycle
- Rapid product changes

Smart Phone OS Market Share (Source: IDC)

Period	Android	iOS
2016Q3	86.8%	12.5%
2016Q2	87.6%	11.7%
2016Q1	83.5%	15.4%
2015Q4	79.6%	18.7%

Source: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

# TODAY'S MENU

1 Brief History of Instagram Android

---

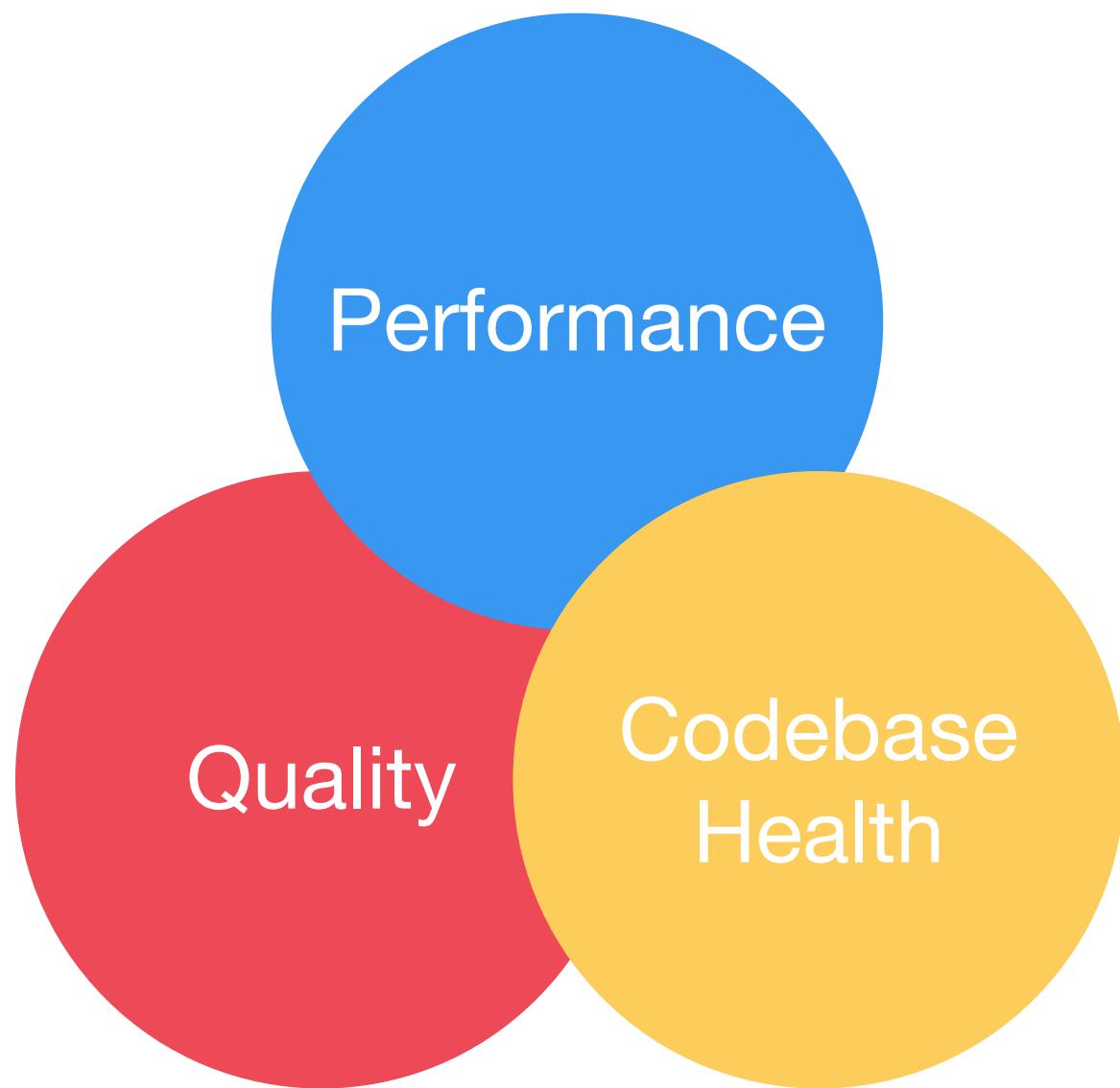
2 Challenges in Today's Android Development

---

3 Lean App Strategy

# LEAN APP STRATEGY

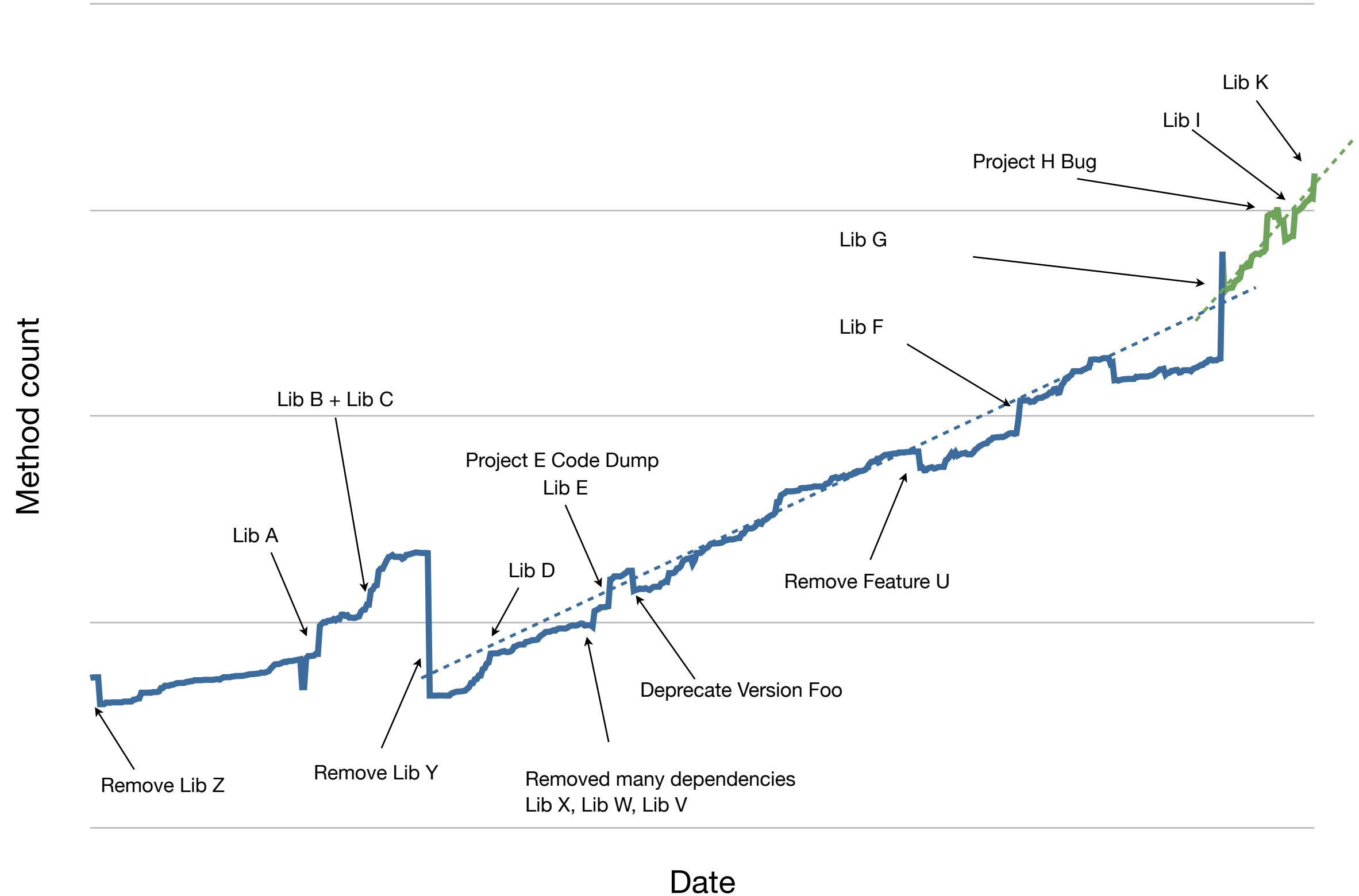
Our Approach to Performance, Quality and Codebase Health



# PERFORMANCE

## 1. Lean App Architecture

- Dependency test from 2013. Whitelist for external dependencies. Monitor DEX method count since 2013.
- Lean Infrastructure
  - ImageCache only has 10 files, including interface and data models. Super efficient, compact and highly customized. It supports progressive jpeg, blur image preview.
  - No dependency injection
  - Product code is very close to vanilla Android

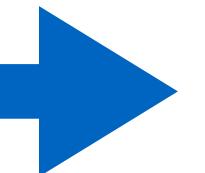


# PERFORMANCE

## 2. Pragmatic - optimize what matters

Example: We do streaming JSON parsing since the first release in 2012, to minimize reflection and allocations.

```
class User {  
    @JsonField("name")  
    String name;  
}  
  
User user =  
    objectMapper.readValue(jp, User.class);
```



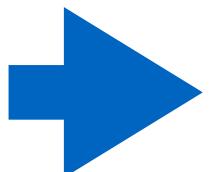
```
class User {  
    String name;  
}  
  
User user = new User();  
while(jp.hasNextToken()) {  
    if (jp.getFieldName().equals("name")) {  
        user.name = jp.value()  
    }  
    jp.nextToken();  
}
```

# PERFORMANCE

## 3. Simplicity - do the simple thing first

Example: JSON parsers have been hand written until 2014.

```
class User {  
    String name;  
}  
  
User user = new User();  
while(jp.hasNextToken()) {  
    if (jp.getFieldName().equals("name")) {  
        user.name = jp.value()  
    }  
    jp.next();  
}
```



```
@JsonType  
class User {  
    @JsonField("name")  
    String name;  
}  
  
User user = User__JsonHelper.parse(jp);
```

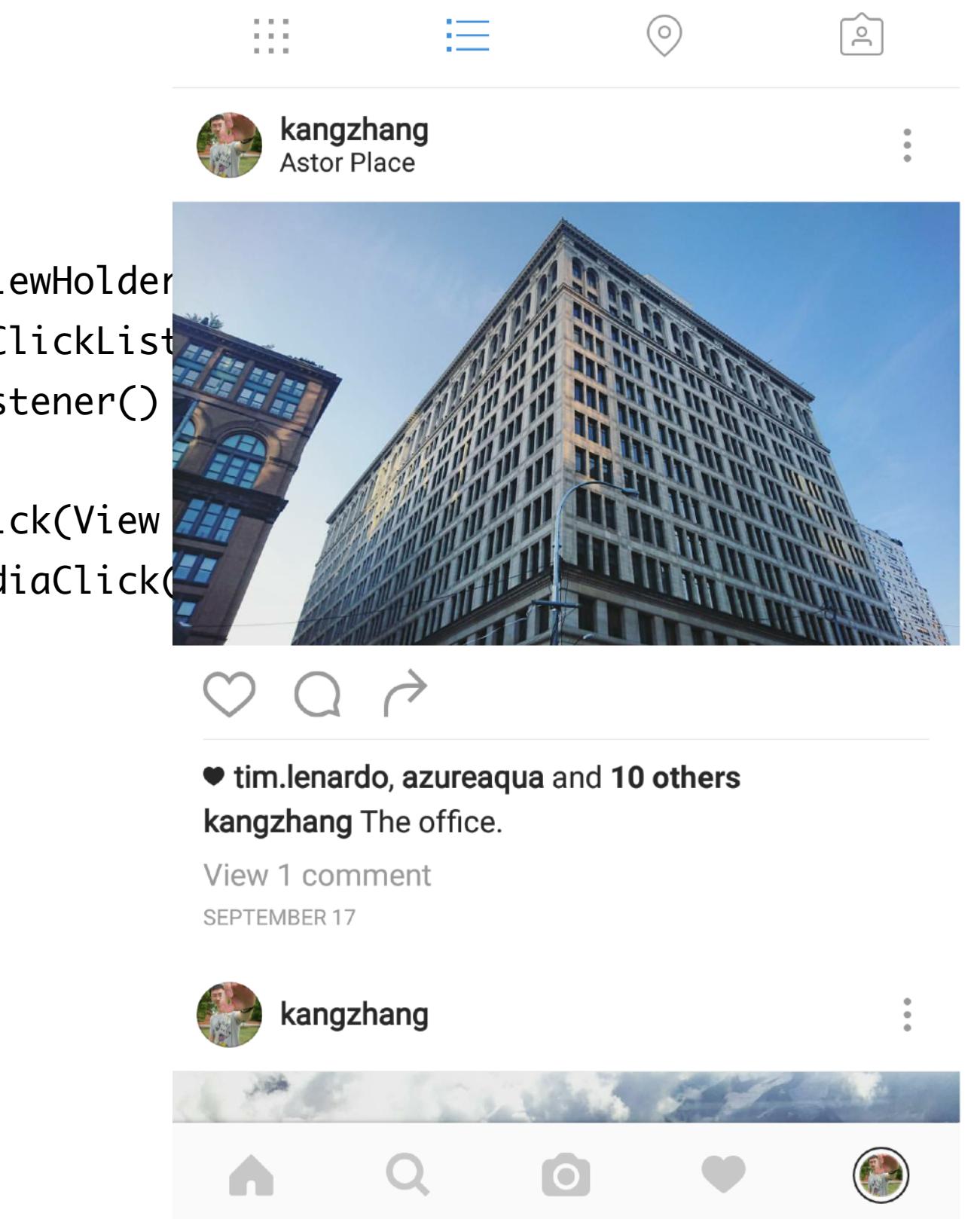
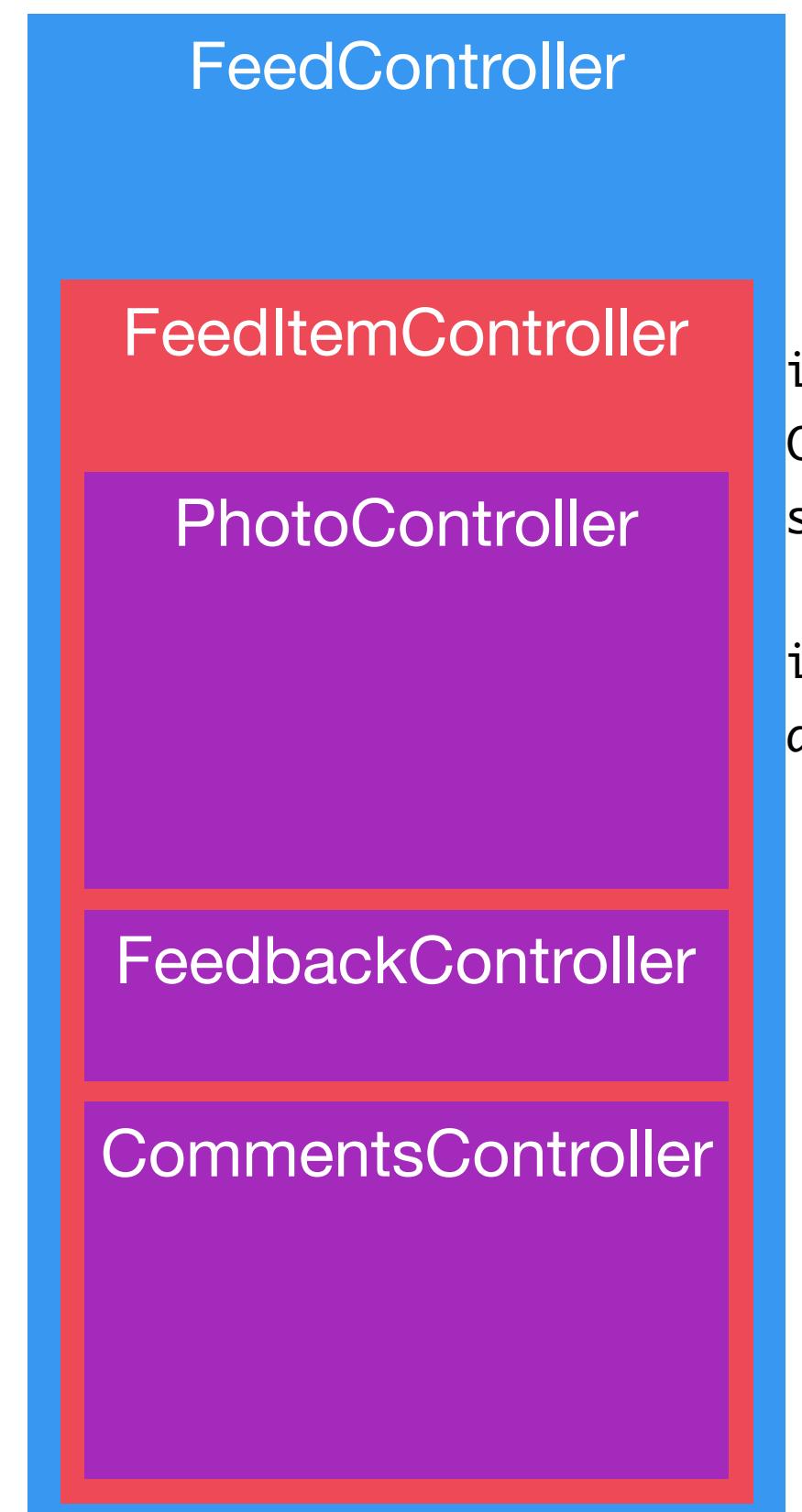


Helper class generated by Annotation processor

# QUALITY

## 4. Fail-fast

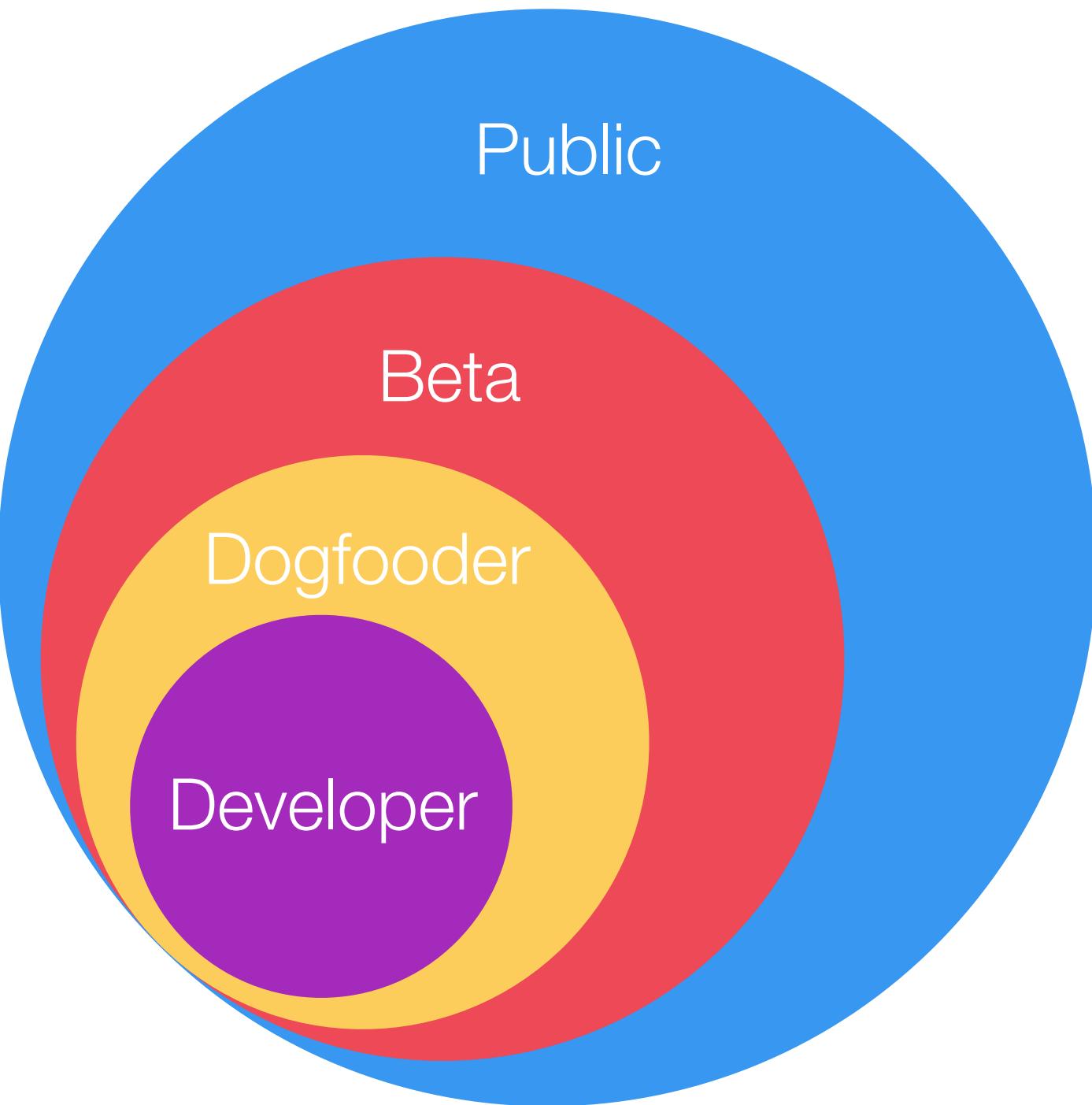
- Strongly typed API
  - Delegate pattern(similar to iOS respond chain)
- Prefer crash rather than swallowing bad input
- Very few product breakage because of unintentional client side change



# QUALITY

## 5. Fast and agile release process

- Developer (every diff)
- Employee Dogfooder (every 4 hours)
- Beta release (every two days)
- Public release (every week)
- Realtime crash monitoring



# CODEBASE HEALTH

## 6. Constant refactor, minimize legacy code

- We have an app toolkit that is composed by infrastructure code that is designed to share between apps.

## 7. Developer experience

- Sub 10s incremental build time

## 8. Mentorship

- On average engineers spend 20-40% of their day on code review and mentorship activities

Analytics	Media Encoding/Decoding
Image Cache	Gallery
Network Middleware	UI Components
AsyncTask	Video Cache
Error Reporting	Image Processing
Event Bus	Waterfall Logging
JSON Parsing	Security(SSL Pinning)
A/B Testing	Others ...

# SUMMARY

Keep lean app with measurable goals.

Do Simple Things First!

Invest on dev velocity tools.

Automation tools: let bots do the work.

Keep learning and revisiting assumptions.

No one-size-fits-all solution, choose what works best for your product and scale.

# Q&A

Thank you!

Frank Qixing Du

Email: [frank@instagram.com](mailto:frank@instagram.com)

Email: [frank@fb.com](mailto:frank@fb.com)

