



flight



flight

Building SDKs

Ty Smith & Javier Soto

Code Monkeys | @tsmith & @javi

How to make great SDKs

EASY TO USE

STABLE

LIGHTWEIGHT

FLEXIBLE

**WELL
SUPPORTED**

Easy to Use

Easy To Integrate

```
Fabric.with(this, new Crashlytics());
```

Easy To Integrate: Builder

```
Crashlytics crashlytics = new Crashlytics.Builder()  
    .delay(1)  
    .listener(createCrashlyticsListener())  
    .pinningInfo(createPinningInfoProvider())  
    .build();  
  
Fabric.with(this, crashlytics);
```

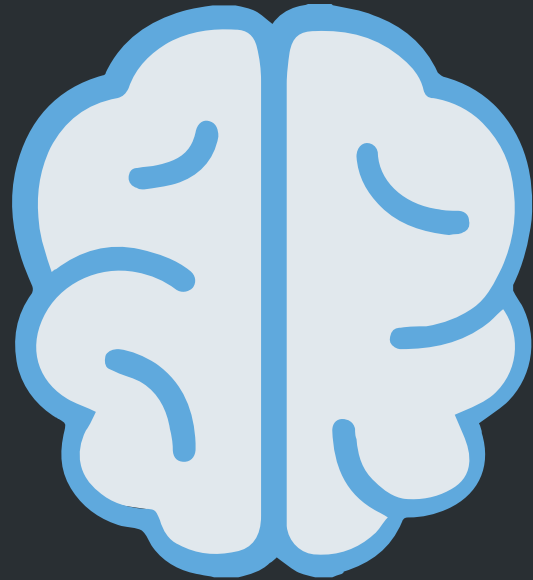
Easy To Integrate: Dependencies

```
<manifest package="com.example.SDK">  
  <application>  
    <meta-data android:value="01235813213455"  
      android:name="com.fabric.ApiKey" />  
  </application>  
</manifest>
```

Easy To Integrate: Dependencies

```
public static Fabric with(Context context) {  
    Context appContext = context.getApplicationContext();  
  
    ApplicationInfo ai = appContext  
        .getPackageManager()  
        .getApplicationInfo(  
            ac.getPackageName(),  
            PackageManager.GET_META_DATA);  
  
    String apiKey = ai.metaData.getString(API_KEY);  
    return new Fabric(appContext, apiKey);  
}
```

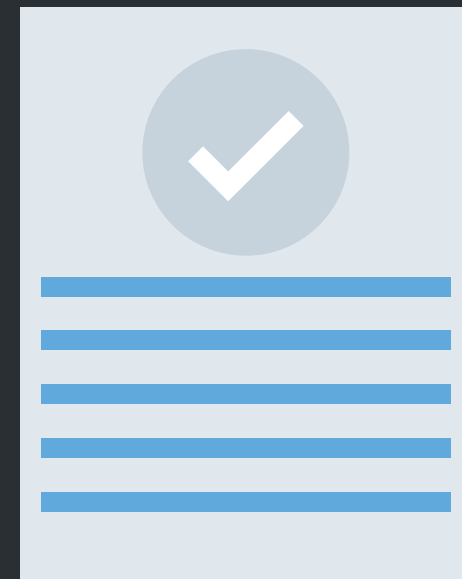

Great API Traits



INTUITIVE



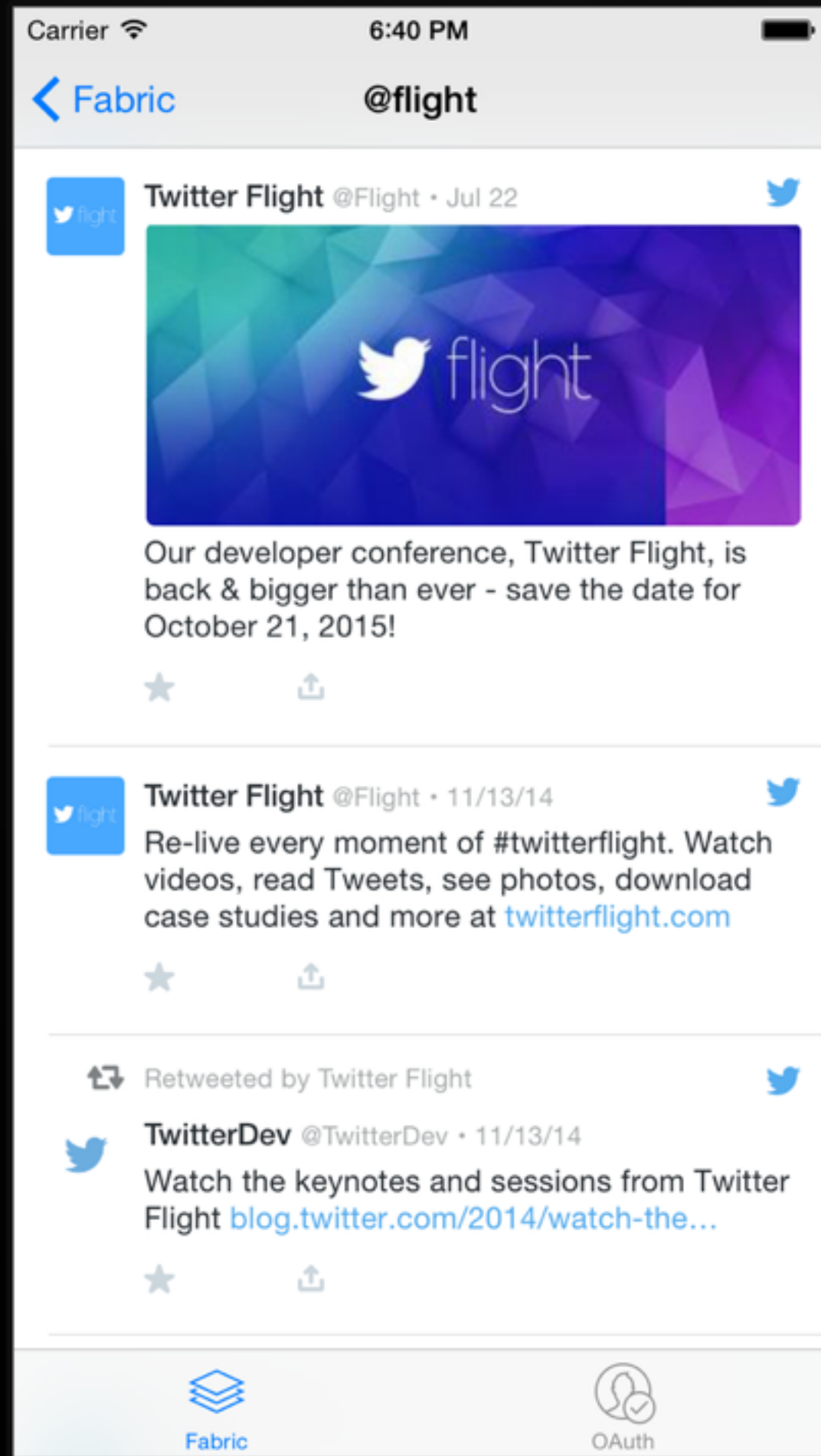
CONSISTENT



EASY TO USE,
HARD TO MISUSE

AN API EXAMPLE

TimelineViewController



Example: TimelineViewController

Consistent

```
class TWRTimelineViewController : UITableViewController {  
    init(dataSource: TWRTimelineDataSource)  
  
    @available(*, unavailable, message="Use init(dataSource:) instead")  
    init(style: UITableViewStyle)  
  
}
```


Example: TimelineViewController

Hard to misuse

```
class TWRTimelineViewController : UITableViewController {  
    init(dataSource: TWRTimelineDataSource)  
  
    @available(*, unavailable, message="Use init(dataSource:) instead")  
    init(style: UITableViewStyle)  
  
}
```


API Intuitiveness: A Counter-Example

java.net.URL

```
URL url1 = new URL("http://foo.example.com");  
URL url2 = new URL("http://example.com");  
  
url1.equals(url2)
```


How We Design APIs

LET ME EXPLAIN YOU CAT SDKS!



Inbox x

Me x



Mr. Manager

to me ▾

2:33 PM (19 hours ago)



Hi team,

Let me explain you how important cat SDKs are going to be. The board has dictated we need one by the end of the week.

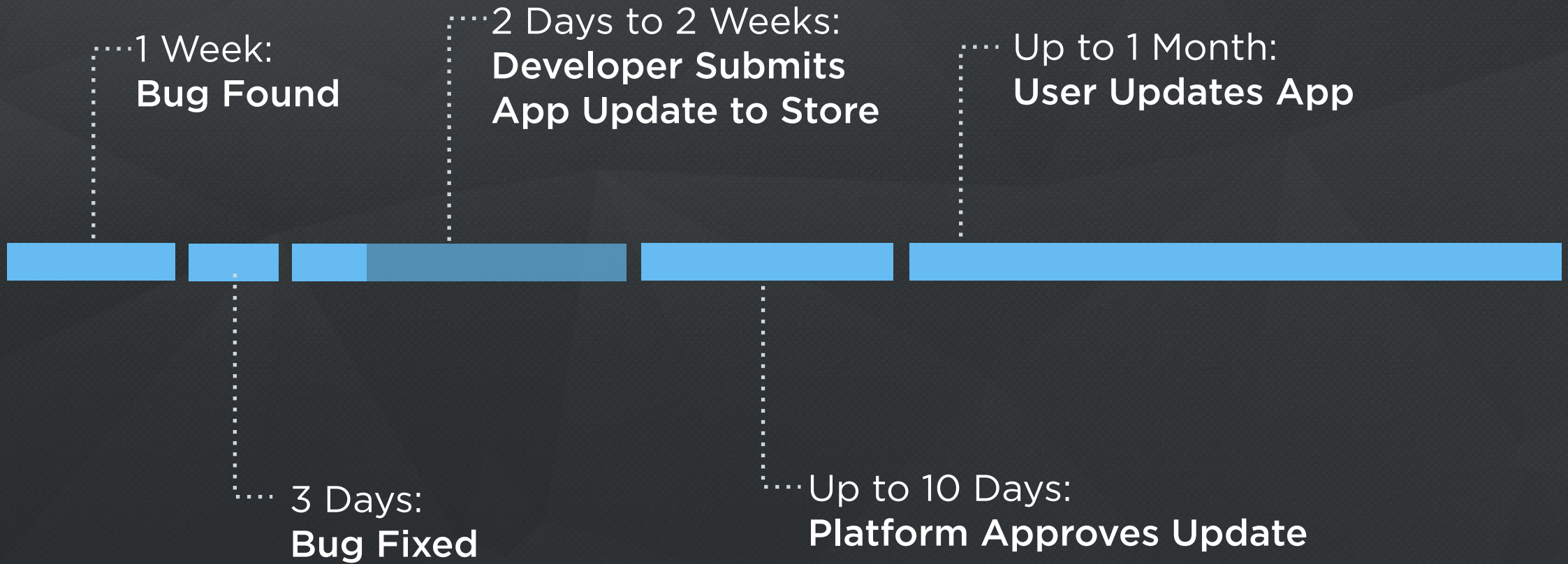
Thanks,
The Management

How We Design APIs

- Should have team alignment
- An API is like a baby
- Should be native to the platform

Stable

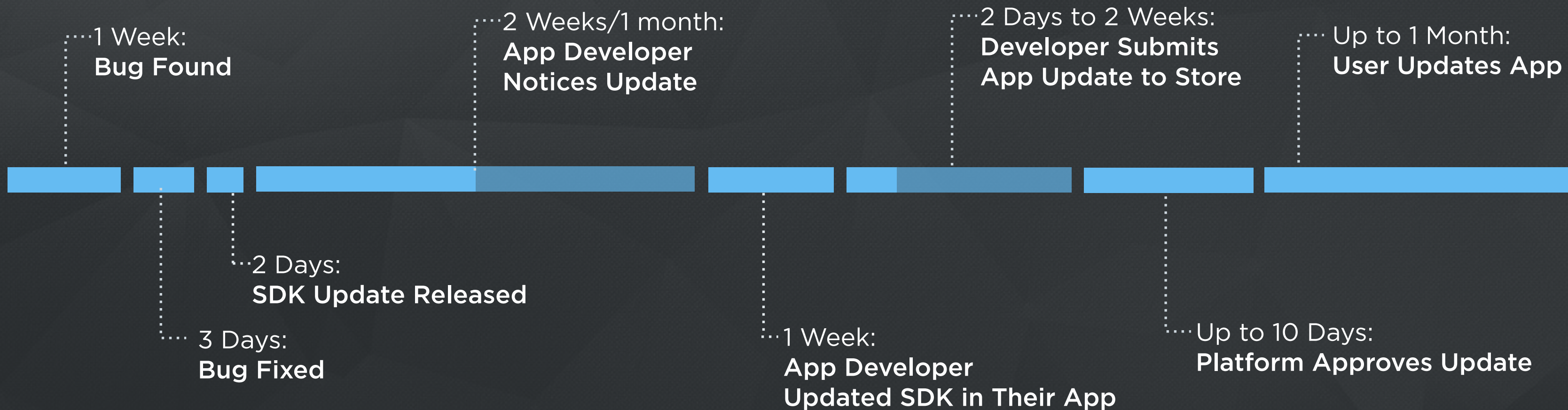
Timeline Of Fixing A Critical App Bug



TOTAL: ~ > 1 **month** or more for average user



Timeline Of Fixing A Critical SDK Bug



TOTAL: ~ > 2 months or more for average user

How Can You Ensure Stability?

Unit tests are **incredibly** useful.

Code Reviews

Well Tested

Testable

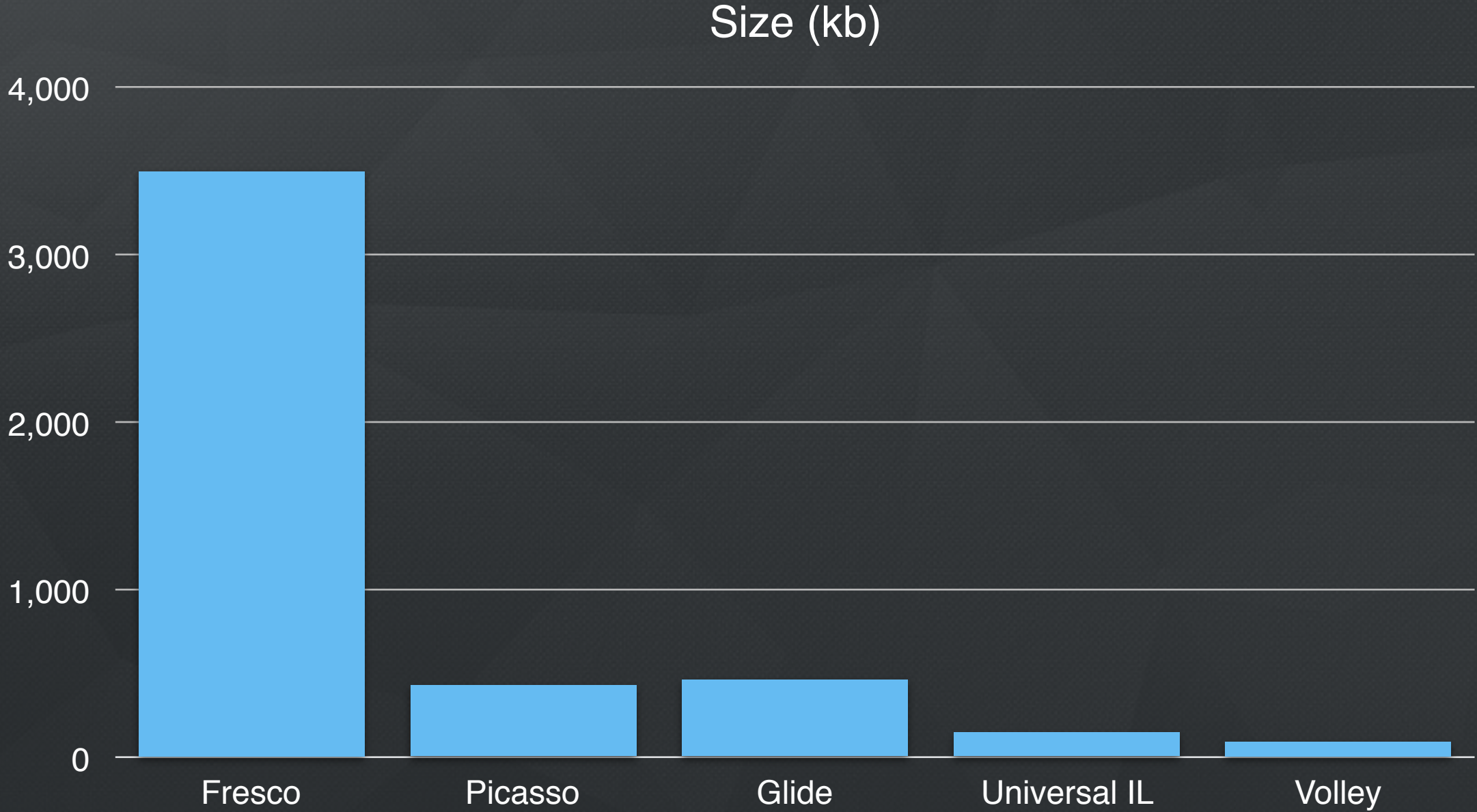
Continuous Integration +
Dog Fooding

“What can go **wrong** with this code?”

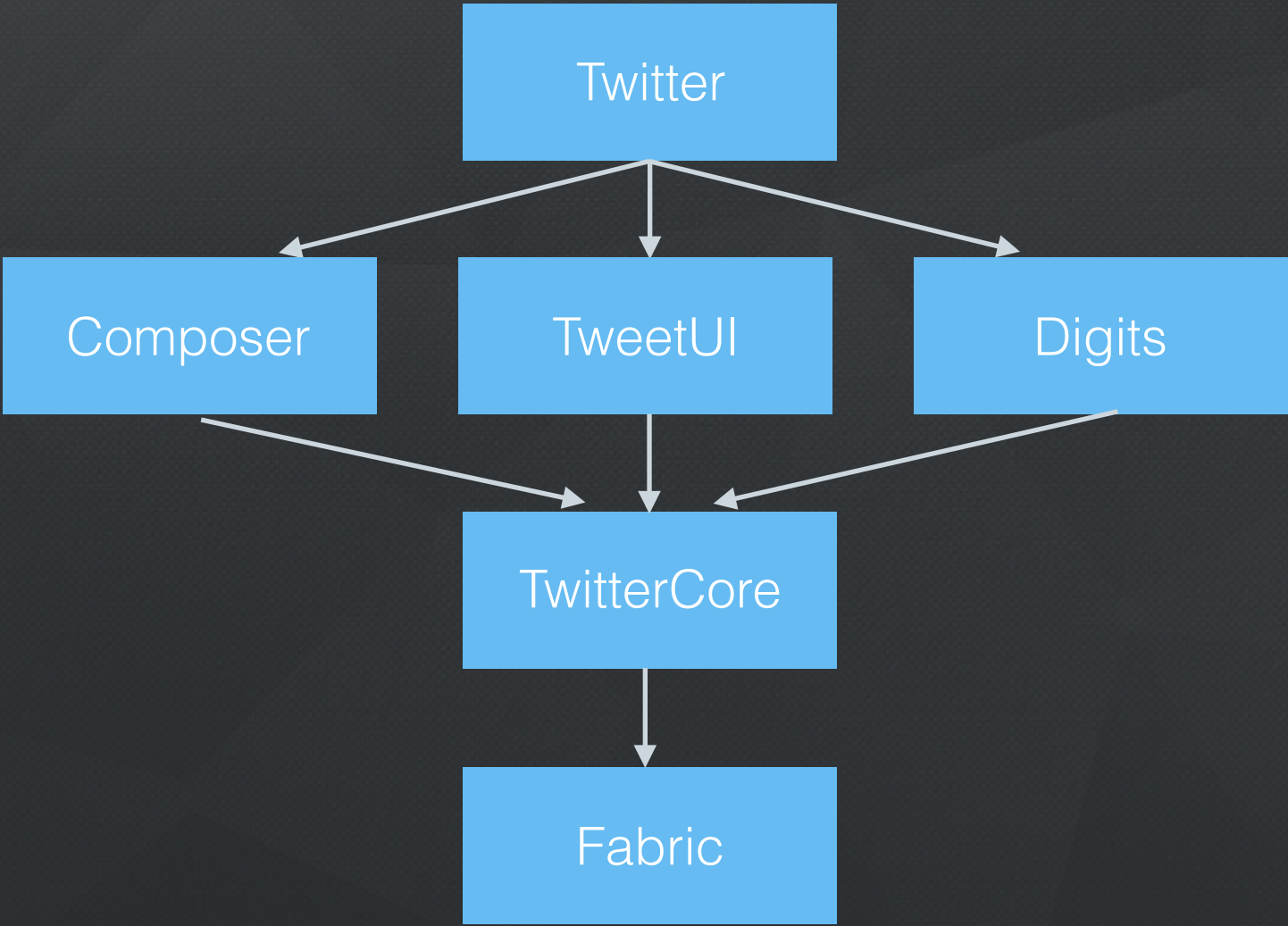
Run some SDK inside your **users'**
testing environments.

Lightweight

Third Party Dependencies

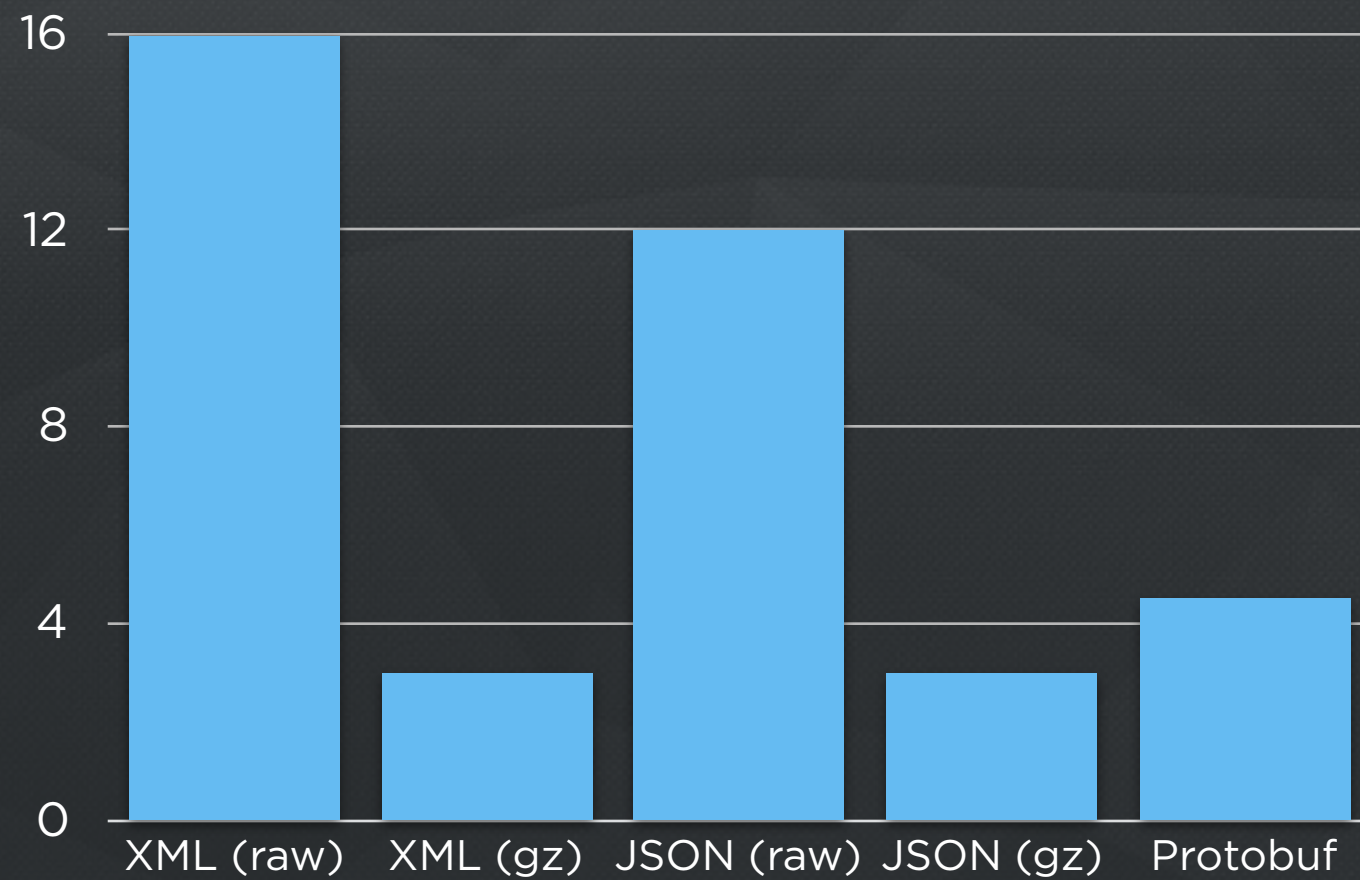


Modularity

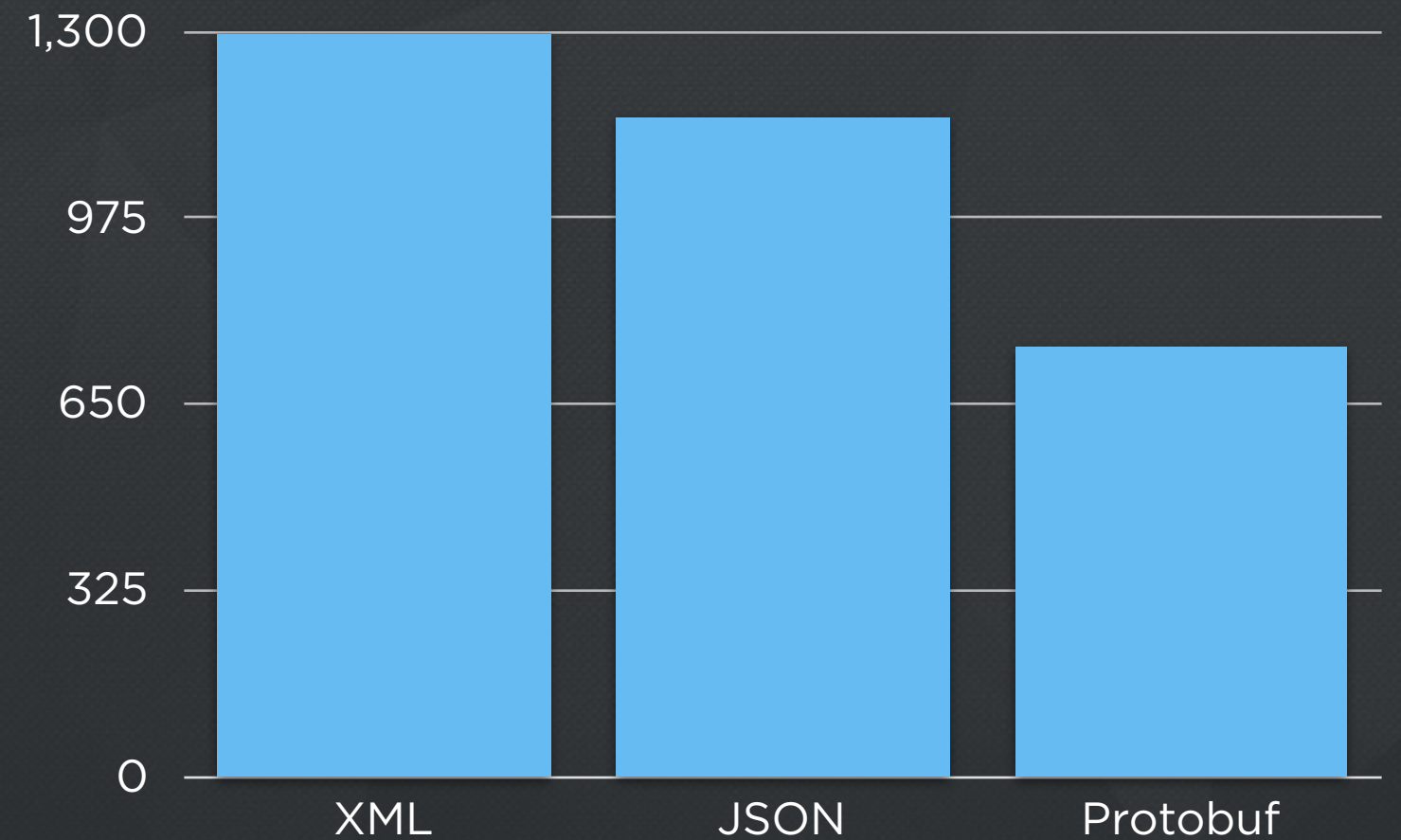


Compress Data

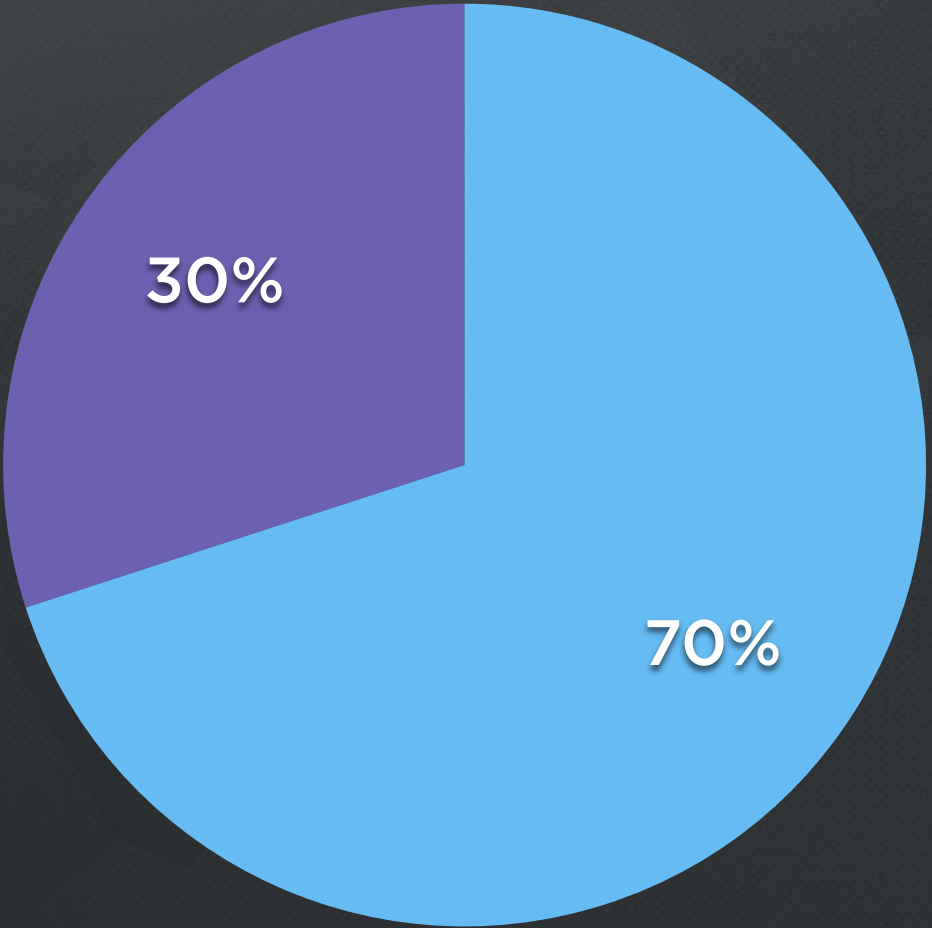
Size (Kb)



Serialization Speed (ms)

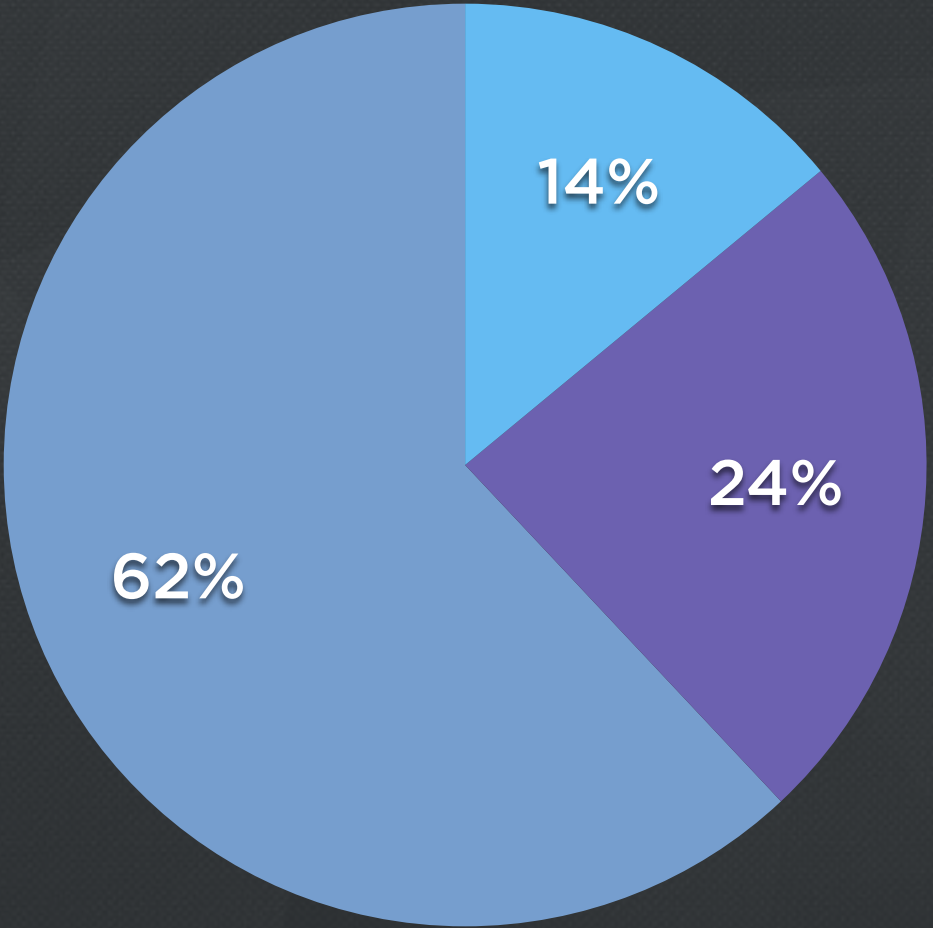


Batch Requests



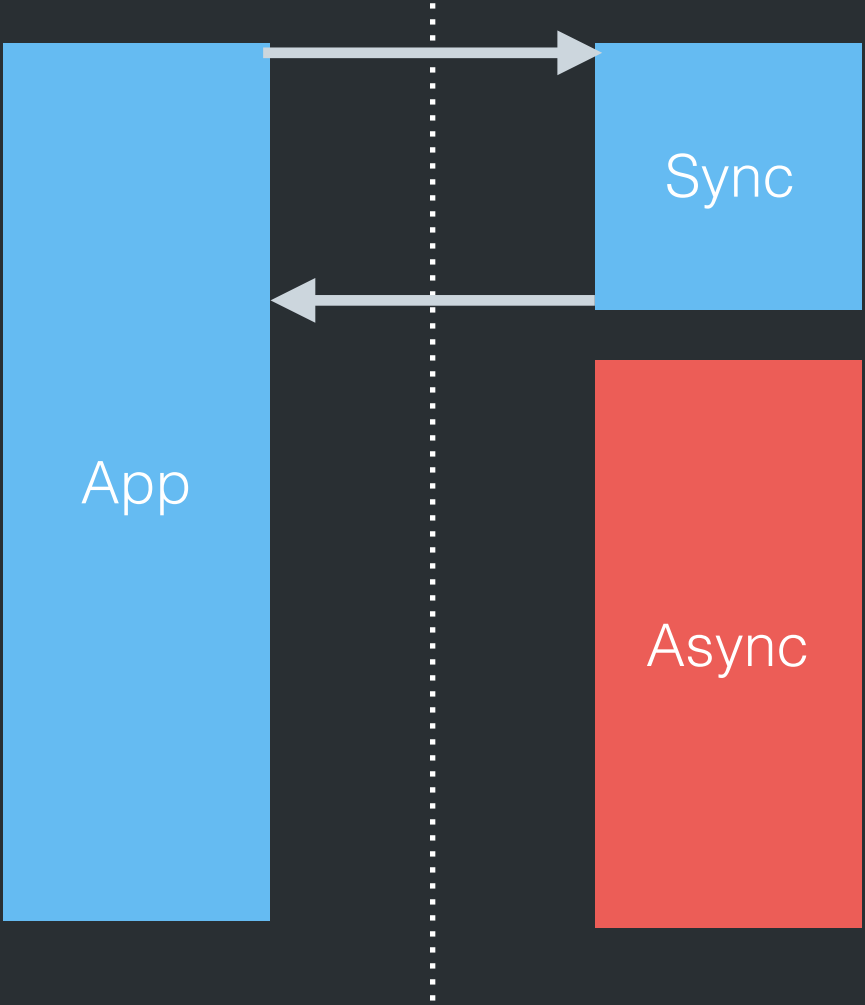
● High Power

● Low Power



● Idle

Asynchronous Tasks



● Main Thread

● Background Thread

Flexible

Build Tools

- Cocoapods support!
- CLI Plugins for Gradle/Maven/Ant
- Plugins for Android Studio

Choosing Your Minimum OS Version

COMPLEXITY

ADOPTION



OS Version Detection

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.FROYO) {  
    return formatID(Build.HARDWARE);  
}  
return null;
```


Enforcing minimum OS Version

```
#if __IPHONE_OS_VERSION_MIN_REQUIRED < 70000  
#error "Digits doesn't support iOS 6.x and lower. Please, change your min. deployment target to iOS 7.0"  
#endif
```


Swift Support

- To provide iOS 7 compatibility, our SDKs are delivered as static libraries, inside .framework packages.
- For Swift to allow `import Fabric`, you must provide a module map file:

Swift support: .modulemap

```
framework module DigitsKit {  
    umbrella header "DigitsKit.h"  
  
    export *  
    module * { export * }  
}
```


Better Swift Support: Nullability

```
#if !__has_feature(nullability)
    #define nullable
    #define nonnull
    #define null_unspecified
    #define null_resettable
    #define __nullable
    #define __nonnull
    #define __null_unspecified
#endif
```


Better Swift Support: Generics

```
#if __has_feature(objc_generics)
    #define CLS_GENERIC_NSARRAY(T) NSArray<T>
    #define CLS_GENERIC_NSDICTIONARY(K,V) NSDictionary<K, V>
#else
    #define CLS_GENERIC_NSARRAY(T) NSArray
    #define CLS_GENERIC_NSDICTIONARY(K,V) NSDictionary
#endif
```

```
– (void)recordCustomExceptionName:(NSString *)name
    reason:(nullable NSString *)reason
    frameArray:(CLS_GENERIC_NSARRAY(CLSStackFrame *) *)frameArray;
```


Better Swift Support

```
import Crashlytics
```

```
func recordCustomExceptionName(name: String!, reason: String!, frameArray: [AnyObject]!)
```

```
func recordCustomExceptionName(name: String, reason: String?, frameArray: [CLSStackFrame])
```


Android Manifests

- One Manifest per APK
- One Manifest per AAR
- Manifests are merged at build time

Dynamically Detect Permissions

```
<manifest>  
  <application>  
  </application>  
  <uses-permission  
    android:name="android.permission.ACCESS_WIFI_STATE" />  
</manifest>
```


Dynamically Detect Permissions

```
protected boolean canCheckNetworkState(Context context) {  
    String permission =  
    Manifest.permission.ACCESS_NETWORK_STATE;  
    int result = context.checkCallingOrSelfPermission(permission);  
  
    return (result == PackageManager.PERMISSION_GRANTED);  
}
```




Enrico's 2nd kindle 1

11:45

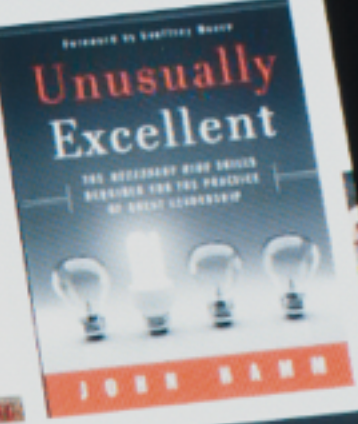
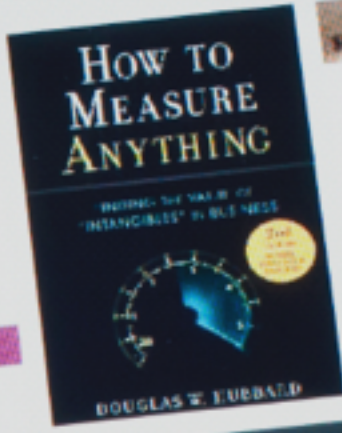
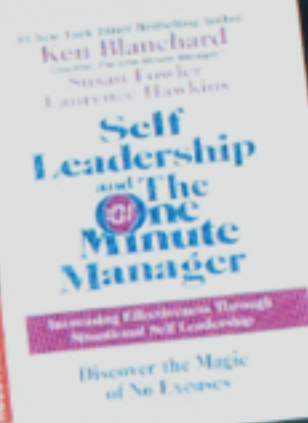
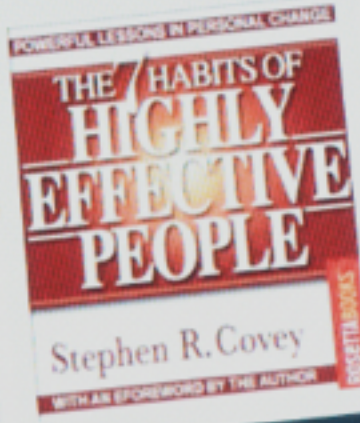
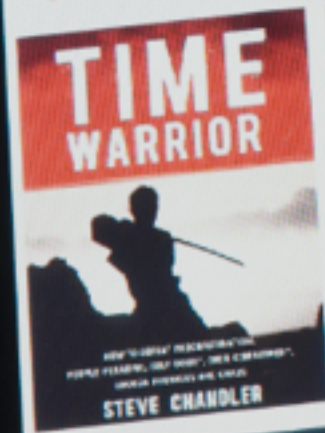


Library >

Search Book store

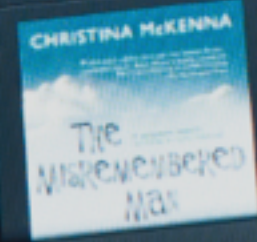
Books

Recommended for You



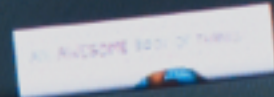
See All >

Top 100 Paid



See All >

Children's Picture Books

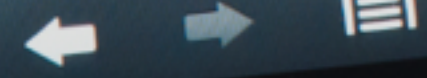


See All >

Browse Books

Browse Kindle Singles

Visit the



CCOLA BIBLI
La 1-
front

IN THE

Make Features Optional

```
<manifest>  
  <application>  
  </application>  
  <uses-feature  
    android:name="android.hardware.camera"  
    android:required="false"/>  
</manifest>
```


Make Features Optional

```
protected boolean hasCamera(Context context) {  
    PackageManager pm = context.getPackageManager();  
  
    String camera = PackageManager.FEATURE_CAMERA;  
  
    return pm.hasSystemFeature(camera);  
}
```


Classpath Detection

```
private boolean hasOkHttpOnClasspath() {  
    try {  
        Class.forName("com.squareup.okhttp.OkHttpClient");  
        return true;  
    } catch (ClassNotFoundException e) { }  
    return false;  
}
```

```
provided 'com.squareup.okhttp:okhttp:2.0.0'
```


Classpath Detection

```
public class DefaultClient implements Client {
    private final Client client;
    public DefaultClient() {

        if (hasOkHttpOnClasspath()) {
            client = new OkHttpClient();
        } else {
            client = new UrlConnectionClient();
        }

    }
}
```


Intent Detection

```
try {  
    startActivityForResult (new  
        Intent (RecognizerIntent.ACTION_RECOGNIZE_SPEECH), 0);  
} catch (ActivityNotFoundException e) { }
```


Intent Detection

```
PackageManager pm =  
    context.getApplicationContext().getPackageManager();  
  
List<ResolveInfo> activities = pm.queryIntentActivities(  
    new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH), 0);  
  
if (activities.size() > 0) {  
    //an app exists that can recognize speech  
}
```


Extensible Classes

```
class MyApiClient extends TwitterApiClient {  
  
    interface MyService {  
        @GET("/1.1/statuses/show.json")  
        Tweet getTweet(@Query("id") int id);  
    }  
  
    MyService getMyService() {  
        return getService(MyService.class);  
    }  
}
```


Extensible Interfaces

```
interface Logger {  
    void d(String tag, String text, Throwable throwable);  
    void v(String tag, String text, Throwable throwable);  
    void i(String tag, String text, Throwable throwable);  
    void w(String tag, String text, Throwable throwable);  
    void e(String tag, String text, Throwable throwable);  
}
```


Sane Defaults

```
public class DefaultLogger {  
  
    public DefaultLogger(int logLevel) {  
        this.logLevel = logLevel;  
    }  
  
    void d(String tag, String text, Throwable throwable)  
    {  
        if (isLoggable(tag, Log.DEBUG))  
            Log.d(tag, text, throwable);  
    }  
}
```


Well Supported

Support

- Documentation
- Sample Code
- Deprecation
- Versioning
- Communication

BUILDING SDKs: A RECAP

Easy To Use

Stable

Lightweight

Flexible

Well Supported

Thank You

@tsmith @javi