

C++ heavy Apps from Dropbox

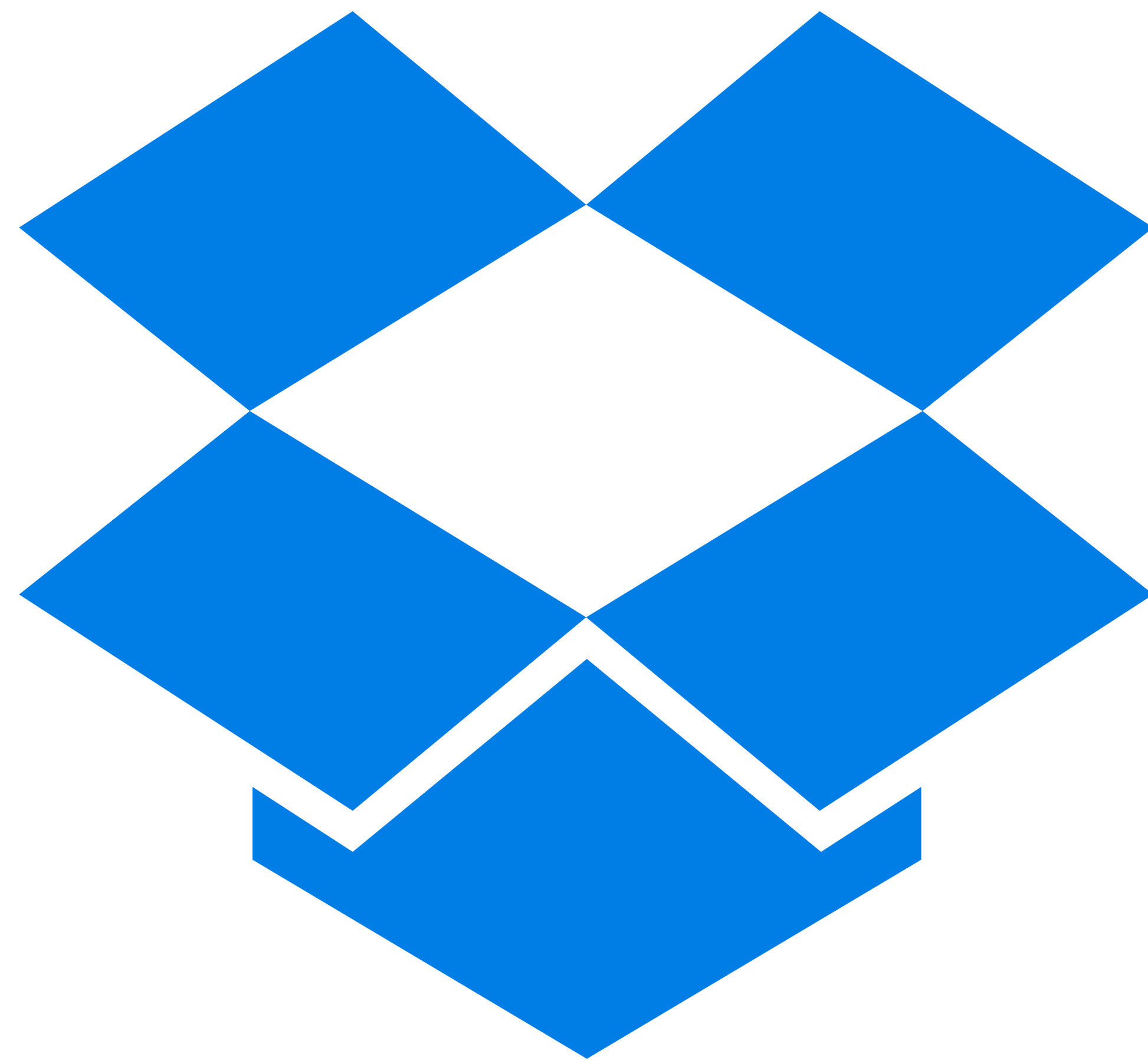


Carousel



Mailbox

Check them out on the
Play Store or the App Store



From the Dropbox Trenches

A Deep Dive into Two Cross-Platform
Mobile Apps Written in C++
CppCon 2014

Us

Tony Grue
tony@dropbox.com



Steven Kabbes
steven@dropbox.com
@kabb



You?

- ✓ Android
- ✓ iOS
- ✓ Other mobile
- ✓ Desktop / Web
- ✓ libraries



Dropbox mobile circa 2012

- ✓ iOS - Obj-C & UI Kit
- ✓ Android - Java & the Android Framework



What was wrong?

- ✓ Camera Uploads
 - ✓ Complicated business logic; edge case city
 - ✓ Shipped 4 months apart
 - ✓ People constrained, ~3 engineers
 - ✓ Two tiny teams, yet communication is hard, encountering same bugs months apart: “oh yes, I hit that”
 - ✓ Envious of desktop: In Python w/ shims to different UI frameworks



Possible solutions?

- ✓ HTML5 / CSS / Javascript
- ✓ QT Quick
- ✓ Kivy
- ✓ Xamarin
- ✓ Roll our own UI Framework (OpenGL Backed)

Rich App Requirements

- ✓ **ability to intercept and control any part of the rendering pipeline (aka getting the good perf)**
- ✓ touch / input / physics
- ✓ text rendering
- ✓ image rendering
- ✓ composing text + image into controls (buttons, list views, etc)
- ✓ **edit controls (interaction with system cut & paste)**
- ✓ web controls (issues here on mobile)
- ✓ generic access to composer surfaces (3d, camera previews etc)
- ✓ layout support
- ✓ animation support & rendering efficiency (don't want a tight game loop)
- ✓ **control flow integration (intents on Android, multi-task w/ backstack model on Android)**
- ✓ **API's that require platform integrations that require interaction through the UI framework or lifecycle**



Nothing Met the
Requirements =(

Except for building it ourself;
but we're too small of a team.



Mission: Build a Photos App

- ✓ Cross-Platform: Many existing users on Android; highly engaged users and much of the company on iOS
- ✓ **All the photos features of Dropbox + a cloud gallery that feels local**
- ✓ Really, really don't want to implement this twice





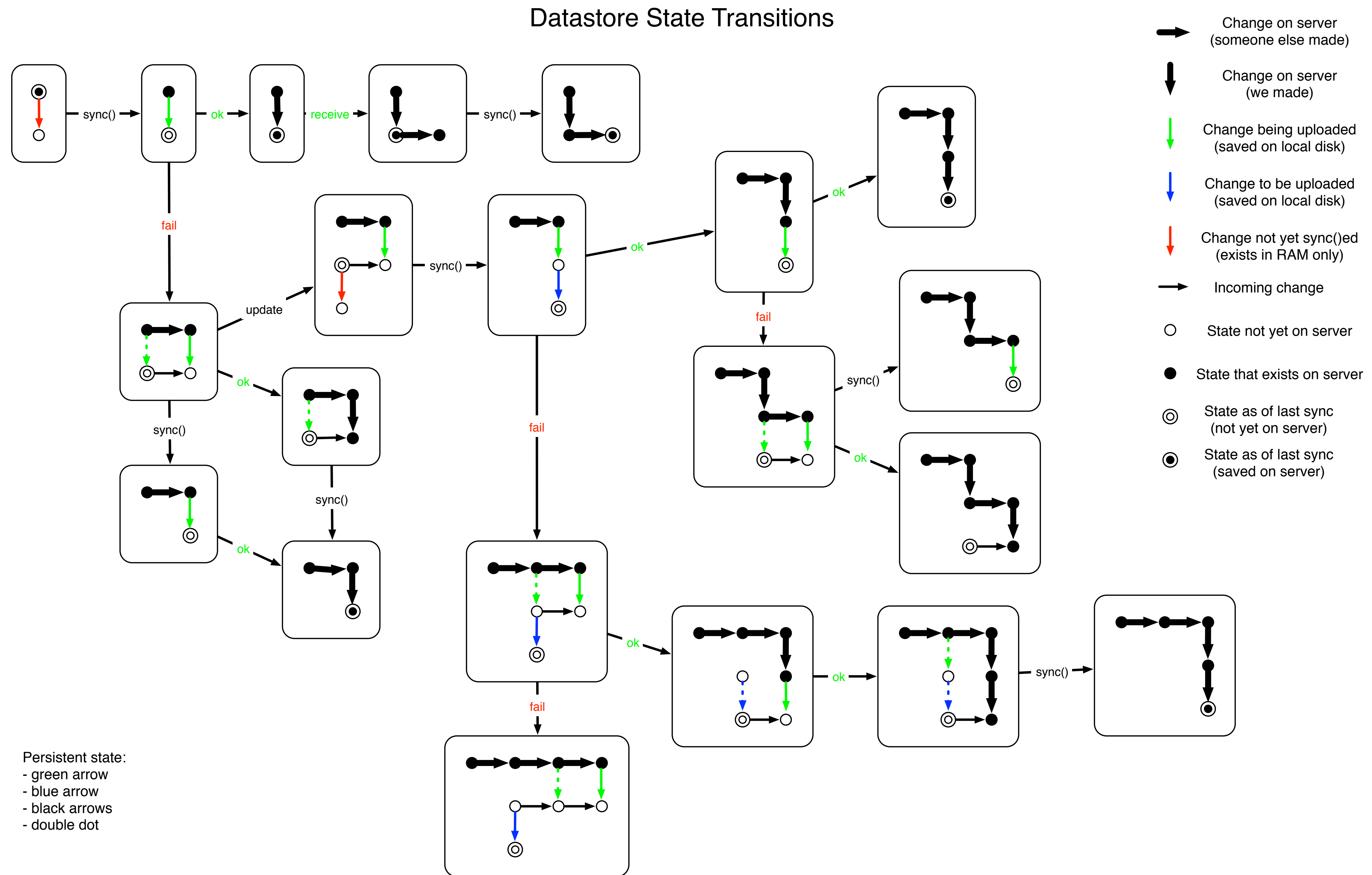
Mailbox Joins!

Mission 2: Welcome Mailbox

- ✓ Mailbox teams up with Dropbox!
- ✓ iOS only app
- ✓ We want Android, we want desktop
- ✓ Write in Java? Port Obj-C to desktop? What happens when we go to Windows?
- ✓ Very lean team
- ✓ Really, really don't want to implement this thrice



Sync API & Datastores



Sync API & Datastores

- ✓ Multi-platform and language API
 - ✓ C++, Python, Android (Java), iOS (Obj-c)
 - ✓ Future: Ruby, Haskell, Etc
- ✓ It's an API but all the hard logic is local and offline and needs storage interaction etc
- ✓ Prototyped in Java on Android
- ✓ We need a better solution



Sync API & Datastores

- ✓ C implementation
- ✓ Platform specific wrappers
- ✓ Every language can interface with C
- ✓ Write complex logic once, fix bugs once!

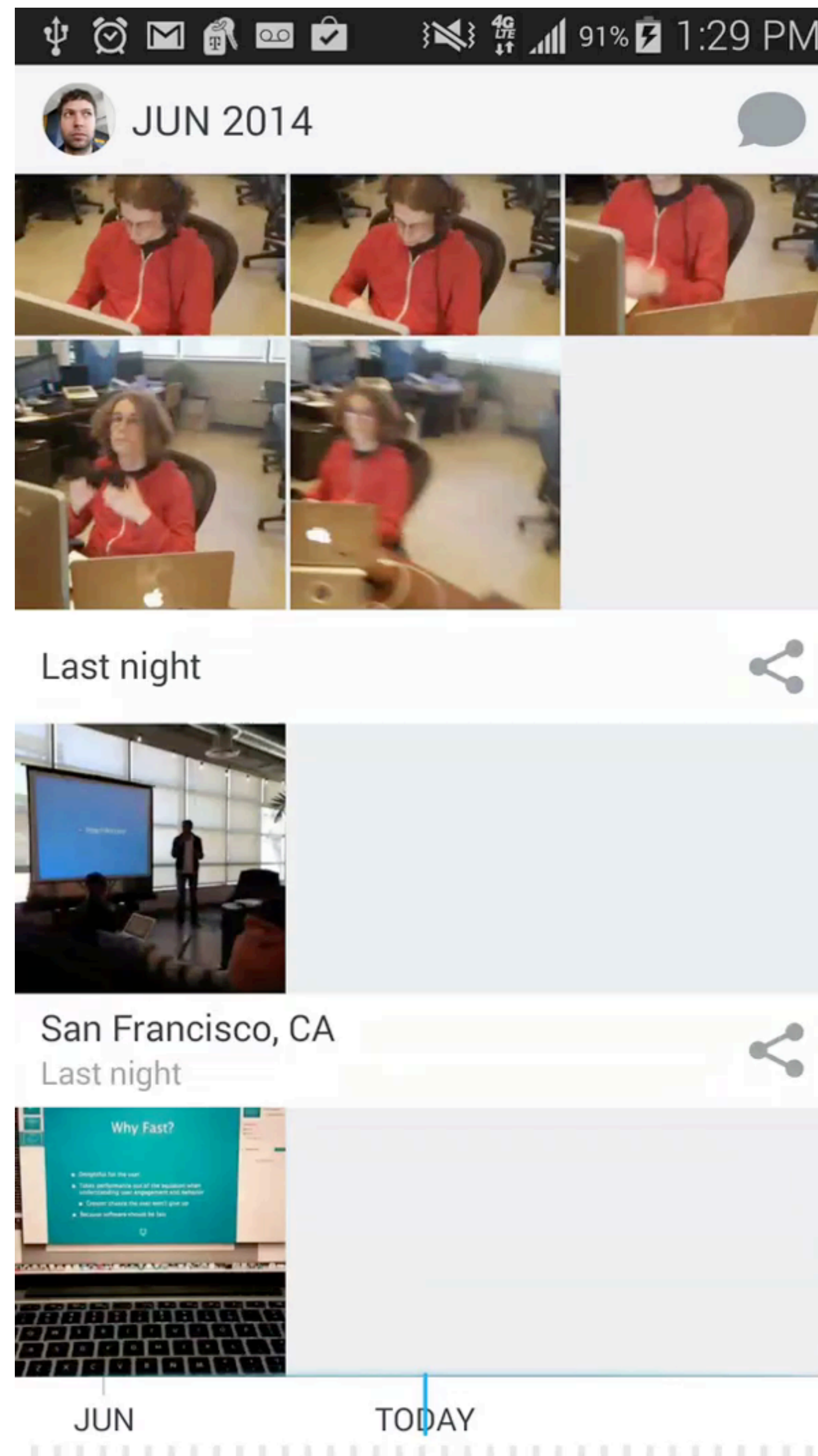


Solution: Leverage C++

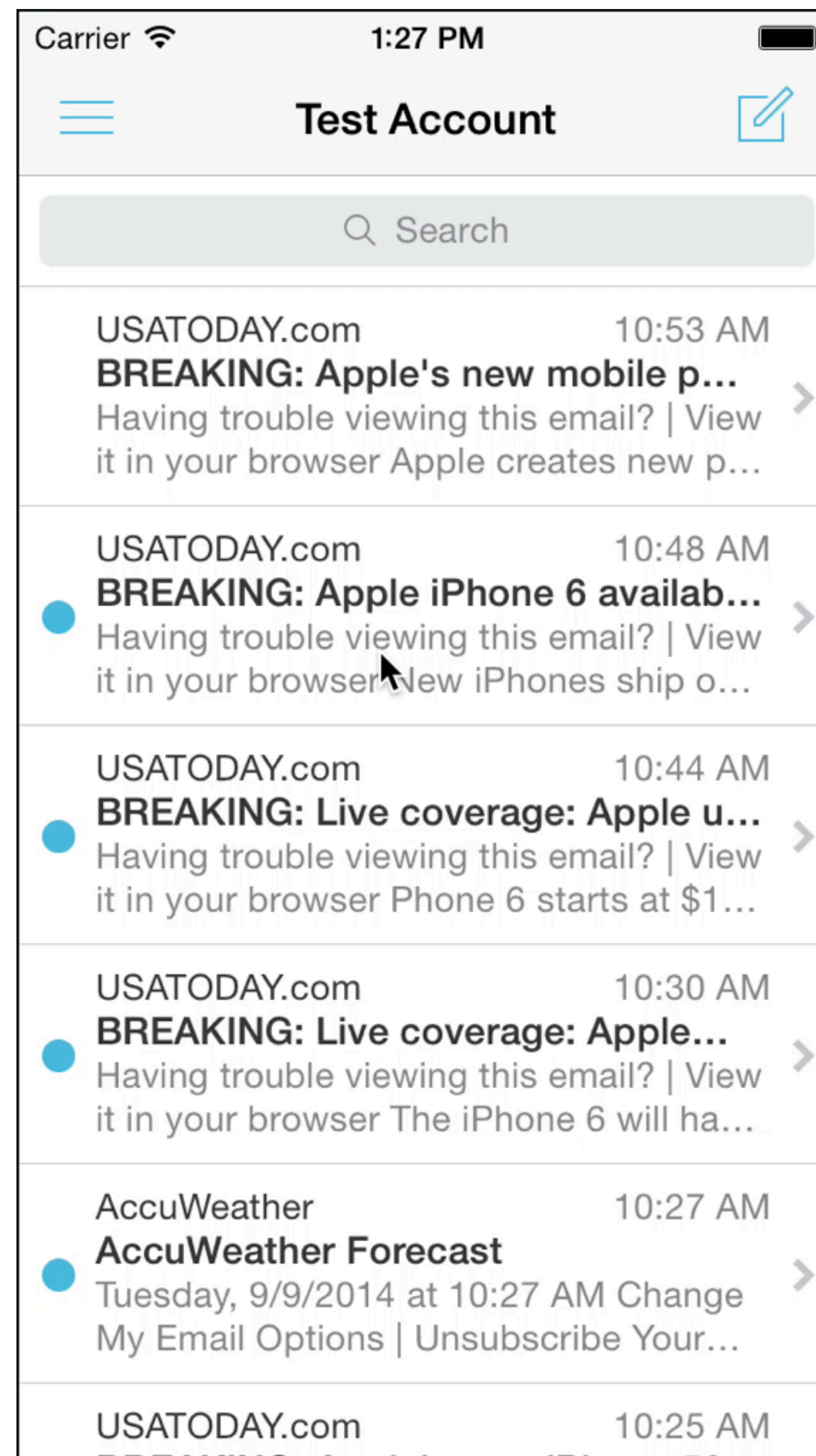
- ✓ Sync API already built in C and in use on mobile:
 - ✓ Networking abstraction
 - ✓ Database and storage abstraction
 - ✓ User model (storage, state, credentials)
- ✓ Dropbox API already partially implemented
- ✓ In fact we started playing with photos



Carousel Demo

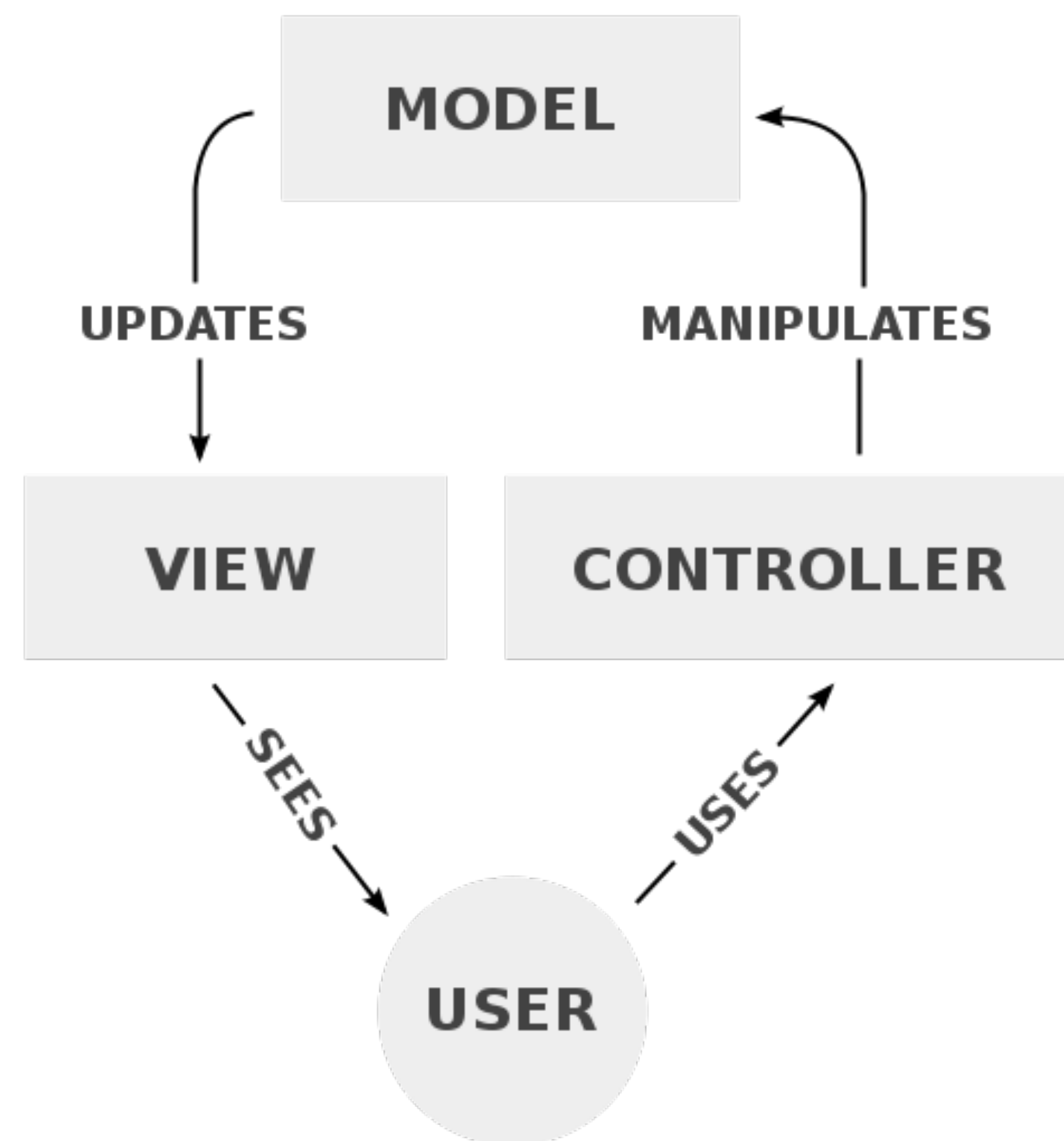


Mailbox Demo

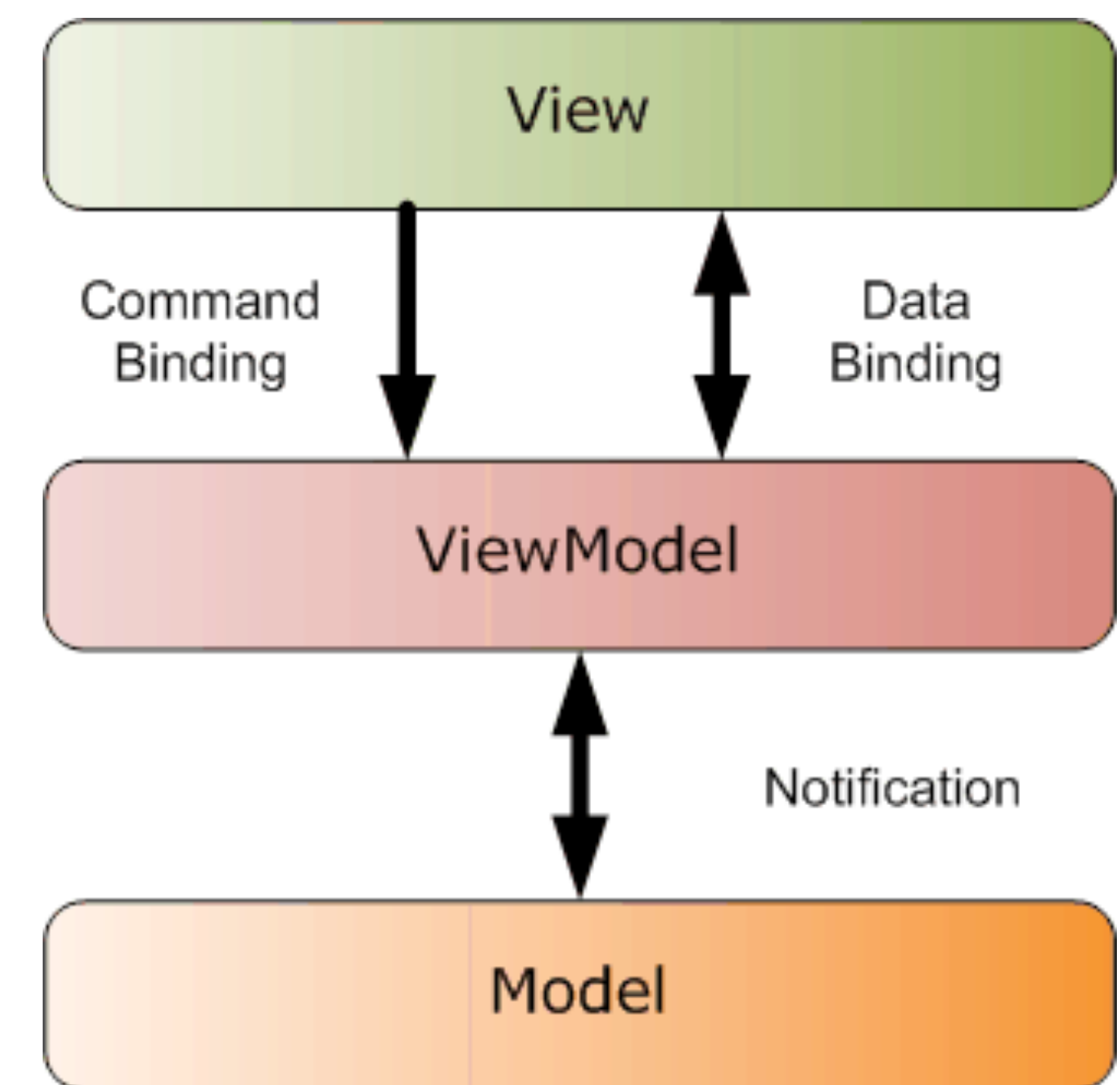


Architecture

MVC

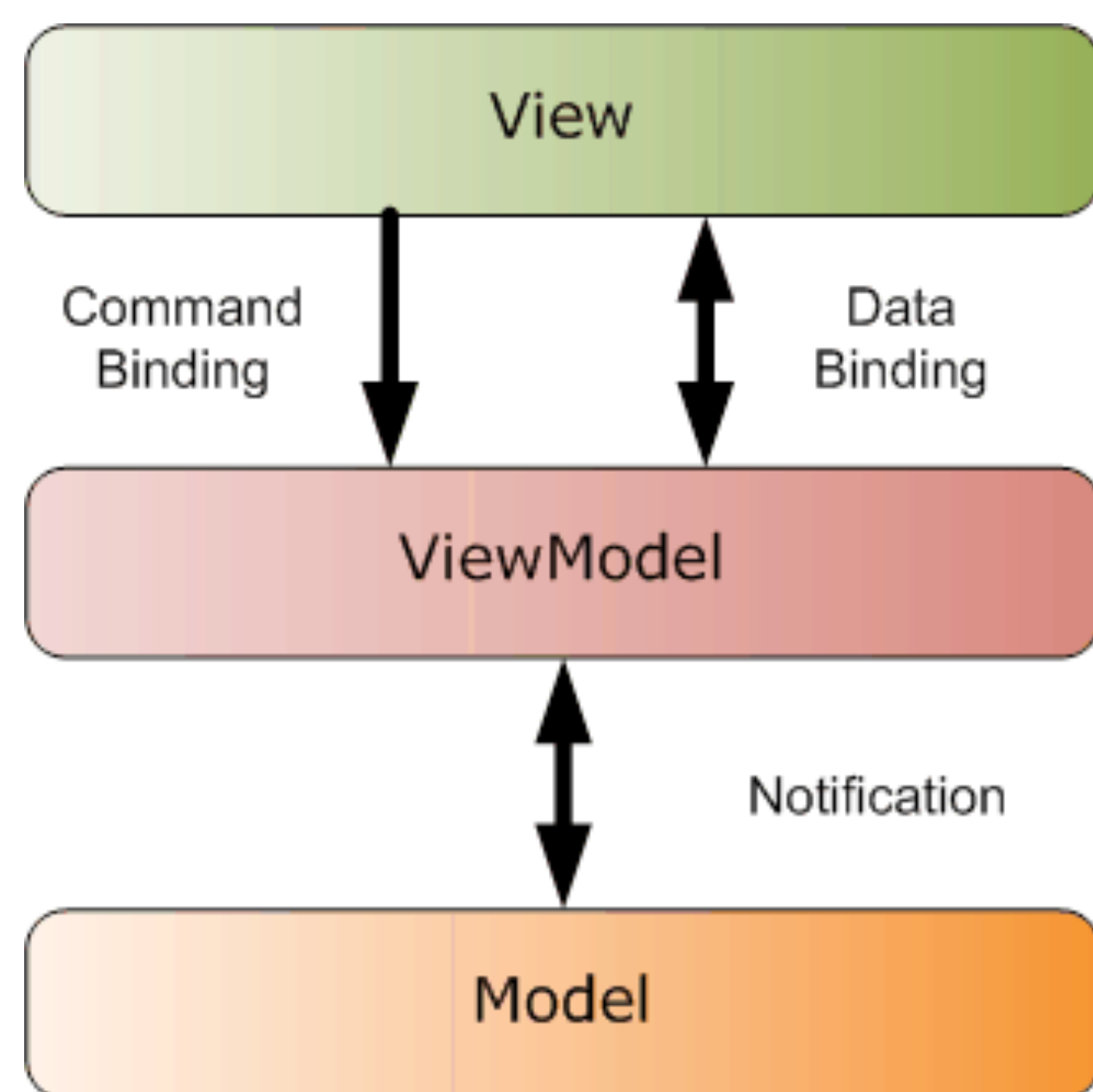


MVVM



Architecture

MVVM



Java or Obj-C

C++

Platform Specific UI x2

Platform Specific ViewModel x2

Java Wrapper or ObjC++

(JNI on Android)

Biz Logic C++ Header

Biz Logic C++ Impl

File System

Sqlite

Network

Camera Roll

POSIX

C libraries

Platform Specific Libs

Java Wrapper or ObjC

(JNI on Android)



Java JNI, C JNI, C wrapper, Obj-C, C++ header oh my!

- ✓ 1 function impl: besides the call site and the implementation, you need to add code in 5 places!!!
- ✓ **you can imagine no one wants to refactor, cleanup, or rename once something is implemented**

Platform Specific UI (xN)

Platform Specific ViewModel (xN)

Java/ObjC++ Wrappers

JNI (on Android)

C Interface to C++

C++ Header

Biz Logic C++ Impl

Platform Specific Libs

Java/ObjC++ Wrappers

JNI (on Android)

File System | Sqlite | Network | Camera Roll

POSIX

C libraries



Platform Specific UI (xN)

Platform Specific ViewModel (xN)

Platform Specific Libs

Language Bridge

Biz Logic C++ Impl

File System

Sqlite

Network

Camera Roll

POSIX

C libraries



Platform Specific UI (xN)

Platform Specific ViewModel (xN)

Platform Specific Libs

Language Bridge

Biz Logic C++ Impl

File System

Sqlite

Network

Camera Roll

POSIX

C libraries



Platform Specific UI (xN)

Platform Specific ViewModel

Platform Specific Libs

Language Bridge

Biz Logic C++ Impl

File System

Sqlite

Network

Camera Roll

POSIX

C libraries



Lesson Learned

- ✓ Strong encapsulation
- ✓ Names should look the same through every layer
- ✓ Use a tool for this layer!
 - ✓ A number of options, but we rolled our own, and we love it. Let us know what you think:
<https://github.com/dropbox/djinni>

There is a lot of code to write in the ViewModel that could be shared

- ✓ Photo Model
 - ✓ `vector<PhotoInfo>`
- ✓ ViewModel
 - ✓ `vector<Row>`
 - ✓ Row {
 type {HEADER | PHOTO_ROW},
 EventInfo,
 optional<vector<PhotoInfo>>
- ✓ Model / ViewModel Pair on each platform
 - ✓ MainGrid, Hidden Photos, Conversation



There is a lot of code to write in the ViewModel that could be shared

- ✓ ViewModels in an eventual consistency app can be complicated
- ✓ Performance is crucial: Model -> ViewModel
- ✓ ViewModels on different platforms end up looking similar

Platform Specific UI (xN)

Platform Specific Libs

Service Layer / Cross Language Interface

Platform Agnostic View Model

Biz Logic C++ Impl

File System

Sqlite

Network

Camera Roll

POSIX

C libraries



Lesson Learned

- ✓ Shared View Models
 - ✓ Saves time
 - ✓ Better x-plat consistency
 - ✓ Optimize performance once



Lesson Learned: Embracing C++11/14 makes replacing Java/Obj-C approachable

- ✓ embracing RAI and exceptions
- ✓ shared_ptr, unique_ptr
- ✓ new containers
- ✓ atomics and threading
- ✓ well, not a standard: optional
- ✓ Lots existed in libraries; but standards eased learning.

Diving In



Mailbox++

- ✓ Path to support Android
 1. Port legacy Obj-C to c++ shared lib
 2. (re)launch iOS with no regressions
 3. Build Android on top of c++ lib



~~CoreData~~

- ✓ Data persistence
- ✓ Data relationships
- ✓ NSManagedObjectContextObjectsDidChange Notification

Concepts

✓ DataView

```
template<typename DataType>
class DataView {
public:
    virtual size_t count() = 0;
    virtual const DataType& get(size_t index) = 0;
};
```



Concepts

- ✓ DataView
- ✓ DataQuery

```
template<typename DataViewType>
class DataQuery {
    public:
        virtual DataViewType operator()() = 0;
};
```

Concepts

- ✓ DataView
- ✓ DataQuery
 - ✓ UnreadEmails()
 - ✓ EmailsInThread(thread_id)
 - ✓ ContactsMatchingSearch(query_string)

Concepts

- ✓ DataView
- ✓ DataQuery
 - ✓ UnreadEmails()
 - ✓ EmailsInThread(thread_id)
 - ✓ ContactsMatchingSearch(query_string)
- ✓ ChangeSet - UI Diff



```
struct ChangeSet final {  
    // refers to _old_ index  
    vector<size_t> deletes;  
  
    // refers to _new_ index  
    vector<size_t> inserts;  
  
    // refers to things that have changed but did not move  
    // refers to _new_ index  
    vector<size_t> updates;  
};
```



```
enum class ChangeType {  
    INSERT,  
    DELETE,  
    MOVE_INSERT,  
    MOVE_DELETE  
};  
  
struct ChangeSet final {  
    // a pair of ((MoveDelete | Delete), old position)  
    vector< pair<ChangeType, size_t> > deletes;  
  
    // a pair of ((MoveInsert | Insert), new position)  
    vector< pair<ChangeType, size_t> > inserts;  
  
    // things which were updated (but not reordered)  
    vector< pair<size_t, size_t> > updates;  
};
```



Concepts

- ✓ DataView
- ✓ DataQuery
 - ✓ UnreadEmails()
 - ✓ ContactsMatchingSearch(query_string)
 - ✓ EmailsInThread(thread_id)
- ✓ ChangeSet - UI Diff



How they relate

1. $\text{DataQuery}_1 \Rightarrow \text{DataView}_1$
2. (time passes, something changes)
3. $\text{DataQuery}_1 \Rightarrow \text{DataView}_2$
4. $\text{diff}(\text{DataView}_1, \text{DataView}_2) \Rightarrow \text{ChangeSet}_{12}$
5. $\text{DataView}_1 + \text{ChangeSet}_{12} \Rightarrow \text{DataView}_2$



How they relate

1. $\text{DataQuery}_1 \Rightarrow \text{DataView}_1$
2. (time passes, something changes)
3. $\text{DataQuery}_1 \Rightarrow \text{DataView}_2$
4. **$\text{diff}(\text{DataView}_1, \text{DataView}_2) \Rightarrow \text{ChangeSet}_{12}$**
5. $\text{DataView}_1 + \text{ChangeSet}_{12} \Rightarrow \text{DataView}_2$

How they relate

1. $\text{DataQuery}_1 \Rightarrow \text{DataView}_1$
2. (time passes, something changes)
3. $\text{DataQuery}_1 \Rightarrow \text{DataView}_2$
4. $\text{diff}(\text{DataView}_1, \text{DataView}_2) \Rightarrow \text{ChangeSet}_{12}$
5. $\text{DataView}_1 + \text{ChangeSet}_{12} \Rightarrow \text{DataView}_2$

How they relate

1. $\text{DataQuery}_1 \Rightarrow \text{DataView}_1$
2. (time passes, something changes)
3. $\text{DataQuery}_1 \Rightarrow \text{DataView}_2$
4. $\text{diff}(\text{DataView}_1, \text{DataView}_2) \Rightarrow \text{ChangeSet}_{12}$
5. **$\text{DataView}_1 + \text{ChangeSet}_{12} \Rightarrow \text{DataView}_2$**

Did it work?





Search...



Mailbox 156

Test Account 71

Dropbox 85

Later 4

Lists

CppCon 14

To Buy 20

To Watch 50

To Read 2

UIKonf 5

Receipts 3

license 1

Bonfire 3

Blankets 1

Test List 123 1

Archive

Trash

Spam

Sent

Drafts

LivingSocial 8:13 AM

Car Detailing Package

And buy and redeem your deal on the go with our mobile app. Shop easy, shop happy. San Jose

LivingSocial Sep 4

House Cleaning

And buy and redeem your deal on the go with our mobile app. Shop easy, shop happy. San Jose

AccuWeather Sep 5

AccuWeather Forecast

Friday, 9/5/2014 at 3:28 PM Change My Email Options | Unsubscribe Your 5-Day Weather

USATODAY.com Sep 5

BREAKING: U.S. military charter order...

Having trouble viewing this email? | View it in your browser A U.S. military charter aircraft flying

USATODAY.com Sep 5

BREAKING: Biggest-ever Alibaba IPO...

Having trouble viewing this email? | View it in your browser The initial price range of the shares

USATODAY.com Sep 5

BREAKING: S&P 500 notches record c...

Having trouble viewing this email? | View it in your browser The government said employers

USATODAY.com Sep 5

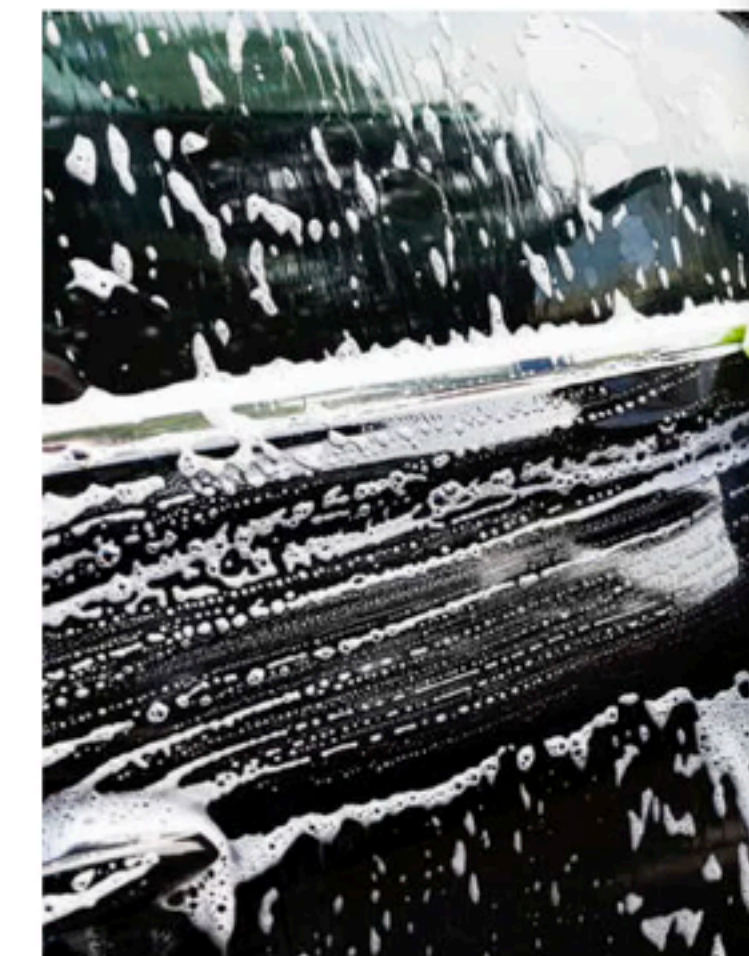
BREAKING: 'We will defend every ally,'...

Having trouble viewing this email? | View it in your browser President. in statement. also says

LivingSocial

livingsocial

home local escapes



BACK TO SCHOOL
YOU



Carrier

6:18 PM



Test Account



LivingSocial 8:13 AM

Car Detailing Package

And buy and redeem your deal on the go with our mobile app. Shop easy, shop...

LivingSocial Sep 4

House Cleaning

And buy and redeem your deal on the go with our mobile app. Shop easy, shop...

AccuWeather Sep 5

AccuWeather Forecast

Friday, 9/5/2014 at 3:28 PM Change My Email Options | Unsu...

USATODAY.com Sep 5

BREAKING: U.S. military charter o...

Having trouble viewing this email? | View it in your browser A U.S. military chart...

USATODAY.com Sep 5

BREAKING: Biggest-ever Alibaba I...

Having trouble viewing this email? | View it in your browser The initial price rang...

USATODAY.com Sep 5

BREAKING: S&P 500 notches reco...

Having trouble viewing this email? | View it in your browser The government sai...

C++ Wins on Android

- ✓ Better awareness of what's going on
- ✓ Circumventing the GC
 - ✓ Lots of objects
 - ✓ Quantity and size of allocations
- ✓ Hiding memory

Win 1: Better Awareness

- ✓ Much of Android is implemented in C++
- ✓ SQLiteCursor
 - ✓ Copies data across the JNI boundary in 2MB pages
 - ✓ Sometimes in lazy init, or when boundary crossed reruns **on UI Thread**
 - ✓ Paging based **on row position**, yikes!



Win 1: Better Awareness

```
@Override
public boolean onMove(int oldPosition, int newPosition) {
    // Make sure the row at newPosition is present in the window
    if (mWindow == null || newPosition < mWindow.getStartPosition() ||
        newPosition >= (mWindow.getStartPosition() + mWindow.getNumRows())) {
        fillWindow(newPosition);
    }

    return true;
}
```



Win 1: Better Awareness

```
private void fillWindow(int requiredPos) {
    clearOrCreateWindow(getDatabase().getPath());

    try {
        if (mCount == NO_COUNT) {
            int startPos = DatabaseUtils.cursorPickFillWindowStartPosition(requiredPos, 0);
            mCount = mQuery.fillWindow(mWindow, startPos, requiredPos, true);
            mCursorWindowCapacity = mWindow.getNumRows();
            if (Log.isLoggable(TAG, Log.DEBUG)) {
                Log.d(TAG, "received count(*) from native_fill_window: " + mCount);
            }
        } else {
            int startPos = DatabaseUtils.cursorPickFillWindowStartPosition(requiredPos,
                mCursorWindowCapacity);
            mQuery.fillWindow(mWindow, startPos, requiredPos, false);
        }
    } catch (RuntimeException ex) {
        // Close the cursor window if the query failed and therefore will
        // not produce any results. This helps to avoid accidentally leaking
        // the cursor window if the client does not correctly handle exceptions
        // and fails to close the cursor.
        closeWindow();
        throw ex;
    }
}
```



Win 1: Better Awareness

- ✓ SQLite Reality
 - ✓ Execute Query; Read Results; 1 Query @ A Time
- ✓ By moving to C++
 - ✓ Small/Medium Queries
 - ✓ Read everything into memory at once
 - ✓ Large Queries
 - ✓ Paging via unique sortkey (actually correct!)



Win 2: Defeat the GC #1

- ✓ Nexus 4:
- ✓ ~10k objects: 2-5MS
- ✓ ~50k objects: 10-30 MS
- ✓ ~500k objects: **500 MS GC!**

Win 2: Defeat the GC #1

- ✓ Hold data in C++, retrieving only as needed

```
class photo_model_snapshot {  
    int get_count();  
  
    dbx_photo_item get_photo_by_id(int photo_id);  
    dbx_photo_item get_photo_by_index(int index);  
}
```

- ✓ 30k photos, ~10 objects each in C++
- ✓ **300k objects disappear from the GC's view!**

Win 3: Defeat the GC #2

✓ /vm/alloc/Heap.cpp

```
182. /* Try as hard as possible to allocate some memory.
183.  */
184. static void *tryMalloc(size_t size)
185. {
186.     void *ptr;
187.
188. //TODO: figure out better heuristics
189. //     There will be a lot of churn if someone allocates a bunch of
190. //     big objects in a row, and we hit the frag case each time.
191. //     A full GC for each.
192. //     Maybe we grow the heap in bigger leaps
193. //     Maybe we skip the GC if the size is large and we did one recently
194. //         (number of allocations ago) (watch for thread effects)
195. //     DeflateTest allocs a bunch of ~128k buffers w/in 0-5 allocs of each other
196. //         (or, at least, there are only 0-5 objects swept each time)
197.
```



Win 3: Defeat the GC #2

- ✓ Every bitmap creation or JPEG load is a 'big' allocation
- ✓ Heap Growth: Stop the World
- ✓ 10-20ms pause, 16MS for whole frame
- ✓ 100's during startup, leads to be startup time

Win 3: Defeat the GC #2

- ✓ You can reuse buffers, still not perfect
- ✓ Instead:
 - ✓ Buffer JPEGs into memory in C++
 - ✓ Allocate bare minimum amount of bitmaps in Java
 - ✓ One JPEG buffer to ferry data between C++ and Java
- ✓ GC better in Android L, but what new gotchas?



Win 4: Hiding Memory

- ✓ Android restricts apps to different memory consumptions (set by OEM) in Java VM
- ✓ Overall process space (i.e. C++ allocs) however are only managed from running away by system
- ✓ Can use 1GB instead of 100MB on a 2GB device
- ✓ Amp up the JPEG pre-fetching on high end devices!
- ✓ But be careful not to abuse this!

X-Platform for App Devs

(not for kernel devs)



Leaving this to library devs

```
#undef PLATFORM_IS_LITTLE_ENDIAN
#if defined(OS_MACOSX)
    #include <machine/endian.h>
    #if defined(__DARWIN_LITTLE_ENDIAN) && defined(__DARWIN_BYTE_ORDER)
        #define PLATFORM_IS_LITTLE_ENDIAN \
            (__DARWIN_BYTE_ORDER == __DARWIN_LITTLE_ENDIAN)
    #endif
#elif defined(OS_SOLARIS)
    #include <sys/isa_defs.h>
    #ifdef _LITTLE_ENDIAN
        #define PLATFORM_IS_LITTLE_ENDIAN true
    #else
        #define PLATFORM_IS_LITTLE_ENDIAN false
    #endif
#elif defined(OS_FREEBSD)
    #include <sys/types.h>
    #include <sys/endian.h>
    #define PLATFORM_IS_LITTLE_ENDIAN (_BYTE_ORDER == _LITTLE_ENDIAN)
#elif defined(OS_OPENBSD) || defined(OS_NETBSD) || \
    defined(OS_DRAGONFLYBSD)
    #include <sys/types.h>
    #include <sys/endian.h>
```



Cross Platform UI Service

- ✗ filesystem
- ✗ networking
- ✓ list views
- ✓ performance & GC
- ✓ “push” registration and background fetch
- ✓ account and settings integration



Library choices

- ✓ one json library
- ✓ one persistence library
- ✓ one image library
- ✓ no library available? delegate to platform

Lean on me!

- ✓ sqlite
- ✓ leveldb
- ✓ openssl
- ✓ bitmagic
- ✓ json11 ([github.com/dropbox](https://github.com/dropbox/json11))
- ✓ miniutf ([github.com/dropbox](https://github.com/dropbox/miniutf))



Mailbox Mac

Search...

Mailbox161

Test Account76

Dropbox85

Later4

Lists

Sent Apartments1

To Buy20

To Watch50

To Read1

UIKonf5

Receipts3

CppCon13

license1

Bonfire3

Blankets1

Test List 1231

Archive

Trash

Spam

Sent

Drafts

Groupon Good

53% Off Kitch

Deferred until S

American Expr

INVOICE 014

Deferred until C

Concur Travel

Concur Itine

Deferred until C

TestD & Me

Smoke test

Deferred until C

New Message

steven@dropbox.com

To:deals

Subject

LivingSocial Deals deals@livingsocial.com

Homejoy Deals sales@em.homejoy.com

Homejoy Deals sales@homejoy.com

LivingSocial deals@livingsocial.com

LivingSocial Escapes deals@livingsocial.com

LivingSocial Events deals@livingsocial.com

Discard

Send

Aug 26

Aug 26

Aug 26

Aug 26

Aug 26

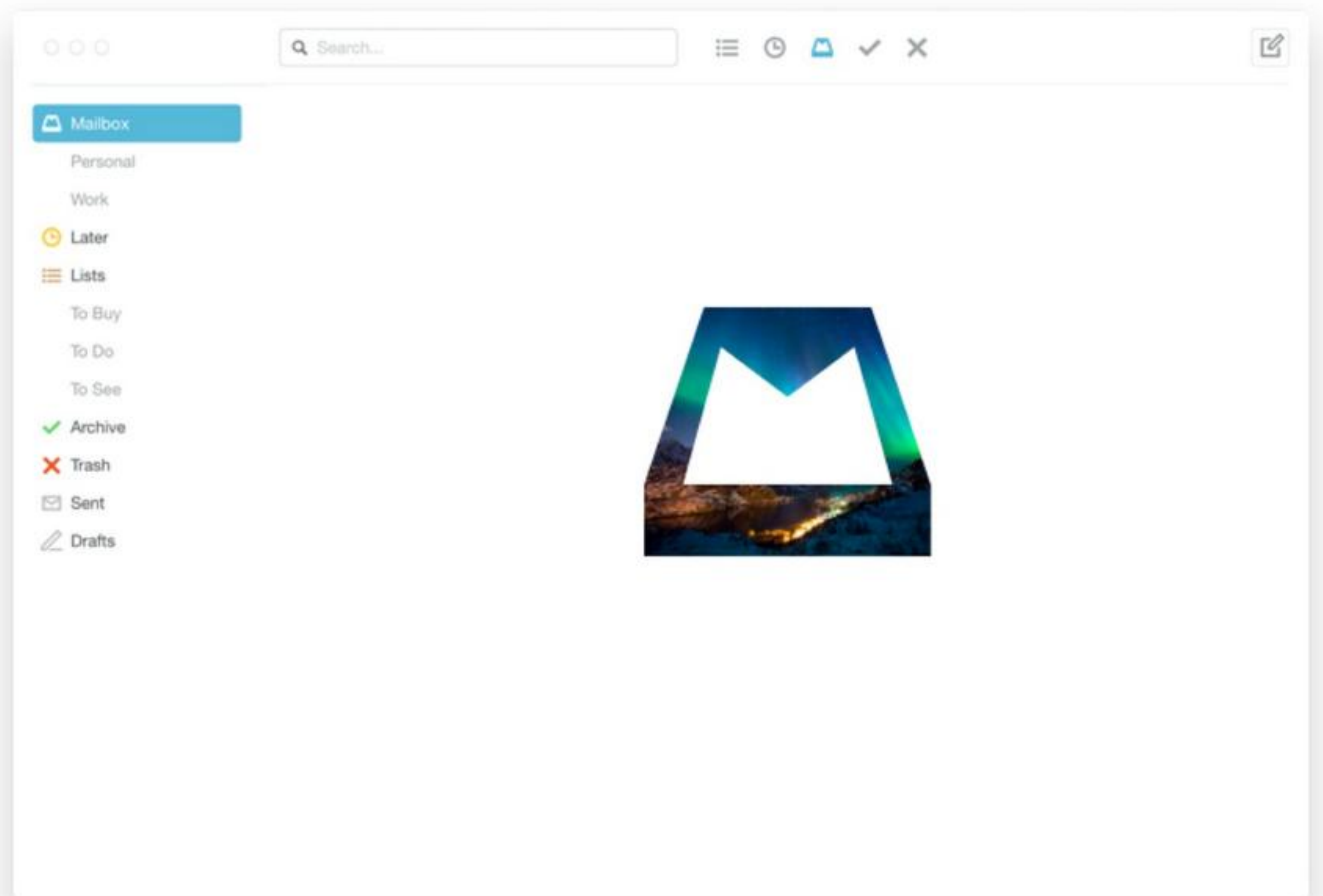


Did it work?

- ✓ if you do multi-repository, binding layer *needs* to be in shared lib
- ✓ make sure to track master
- ✓ educate the UI folk about templates, linking
- ✓ implicit API, usage differences
- ✓ **These are not architecture problems**

Features with no extra c++

- ✓ Autoswipe (gmail filters)
- ✓ Zero-Inbox
- ✓ Drafts
- ✓ ?



Roses have thorns

- ✓ Lots of tooling (make system, build tools, testing frameworks, new libraries)
- ✓ No great X-Plat C++ IDE
- ✓ Debugging on Android
- ✓ Crash Reporting & Callstacks (on Android)
- ✓ Bugs in Android's NDK and missing features
- ✓ Memory Safety (of course)



Roses have thorns

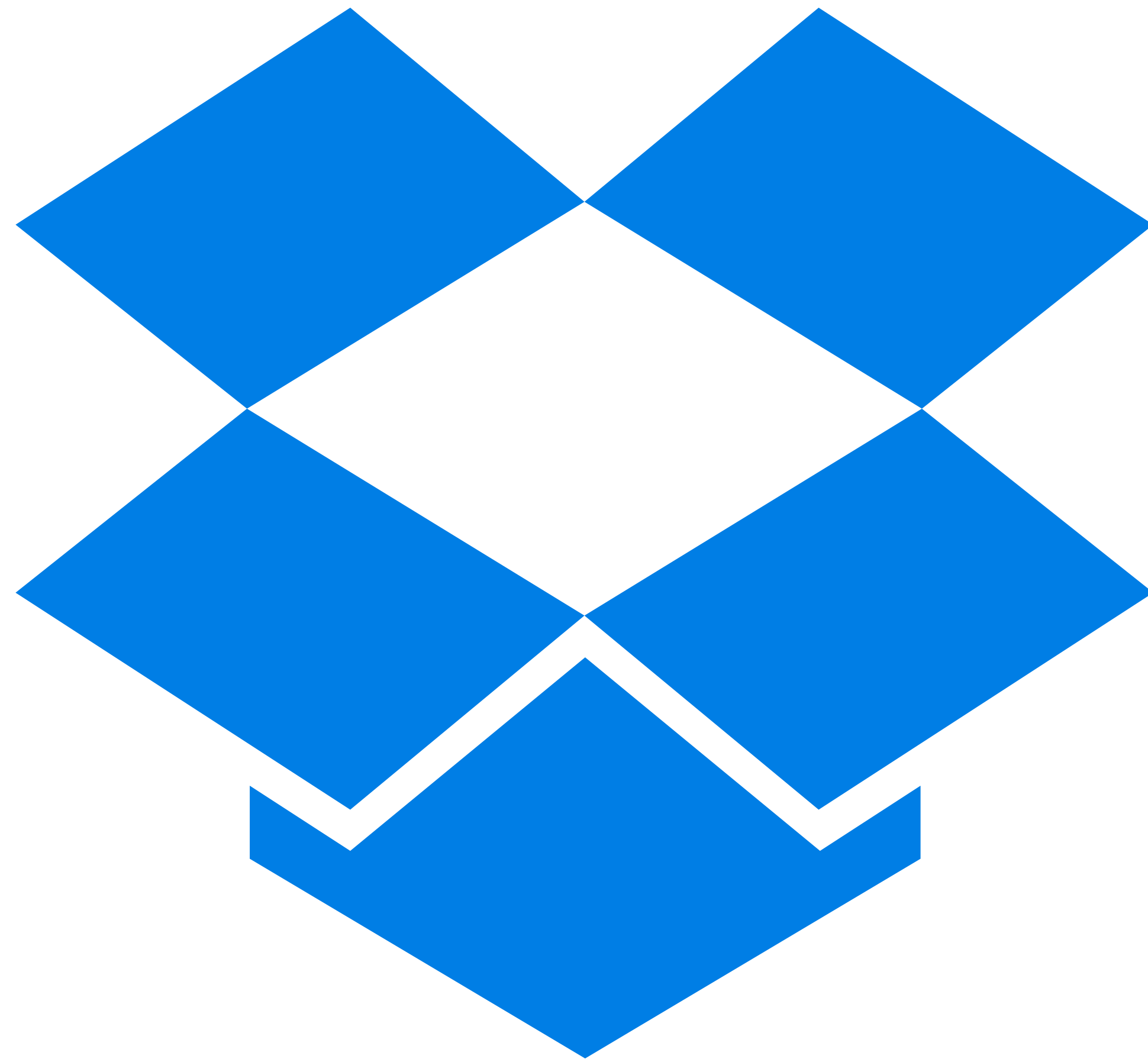
- ✓ Team Issues
 - ✓ 3 different languages for 2 clients
 - ✓ To Our C++ Devs: Writing App Not Kernel
 - ✓ Iterate on one platform first



Future work

- ✗ richer IDL, more platforms, more features!
- ✗ async framework / cleaner concurrency patterns (for app debs - used to message loops)
- ✗ build systems - build once
- ✗ better persistence abstraction





Questions?

Sound Interesting?

```
auto presenters = {  
    "tony@dropbox.com",  
    "steven@dropbox.com"  
};
```

```
auto platforms = {  
    ANDROID,  
    IOS,  
    OSX,  
    WINDOWS  
};
```

```
if (heart(CPP) &&  
    any_of(platforms.begin(), platforms.end(), heart)) {  
    send_resume(presenters);  
}
```

