# Ordering Messages and Phone Calls in TextNow

By: Joshua Kalpin

June 12th 2013

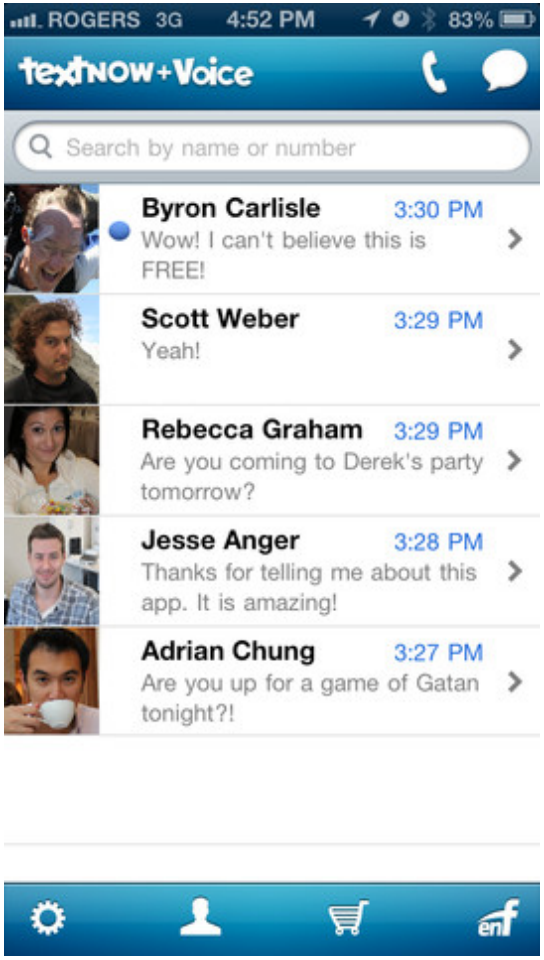
# Outline

- What is TextNow?
- Background on the Problem
- Requirements
- Overall Changes
- Failed Solutions
- The Chosen Solution
- Summary

# What is TextNow?

- Created By Enflick Inc. in 2009
- Started as a free Texting Application
- Calling features added later
- Cross-Platform including iOS, Android, Web and Windows Phone
- iOS version is currently receiving a complete rewrite and redesign

# Problem Background



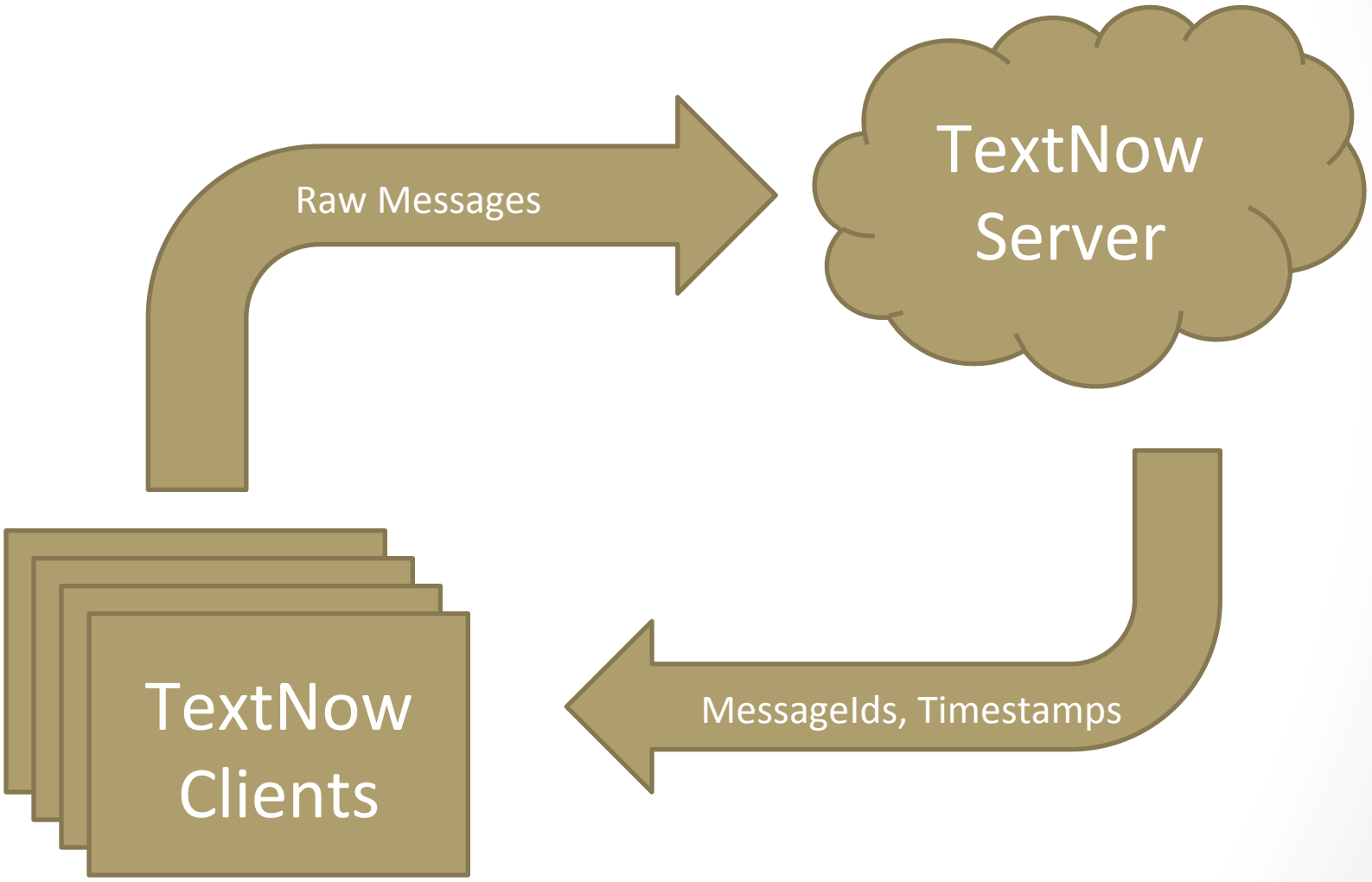


https://itunes.apple.com/us/app/textnow-unlimited-free-texting/id314716233?mt=8

# Problem Background Cont.

- Interactions between users are shown in the Conversation Screen
- Conversations contain messages and call history
- Displayed from Core Data
- Initially, only messages were implemented
- Problems occurred when trying to implement call history

# How TextNow Works



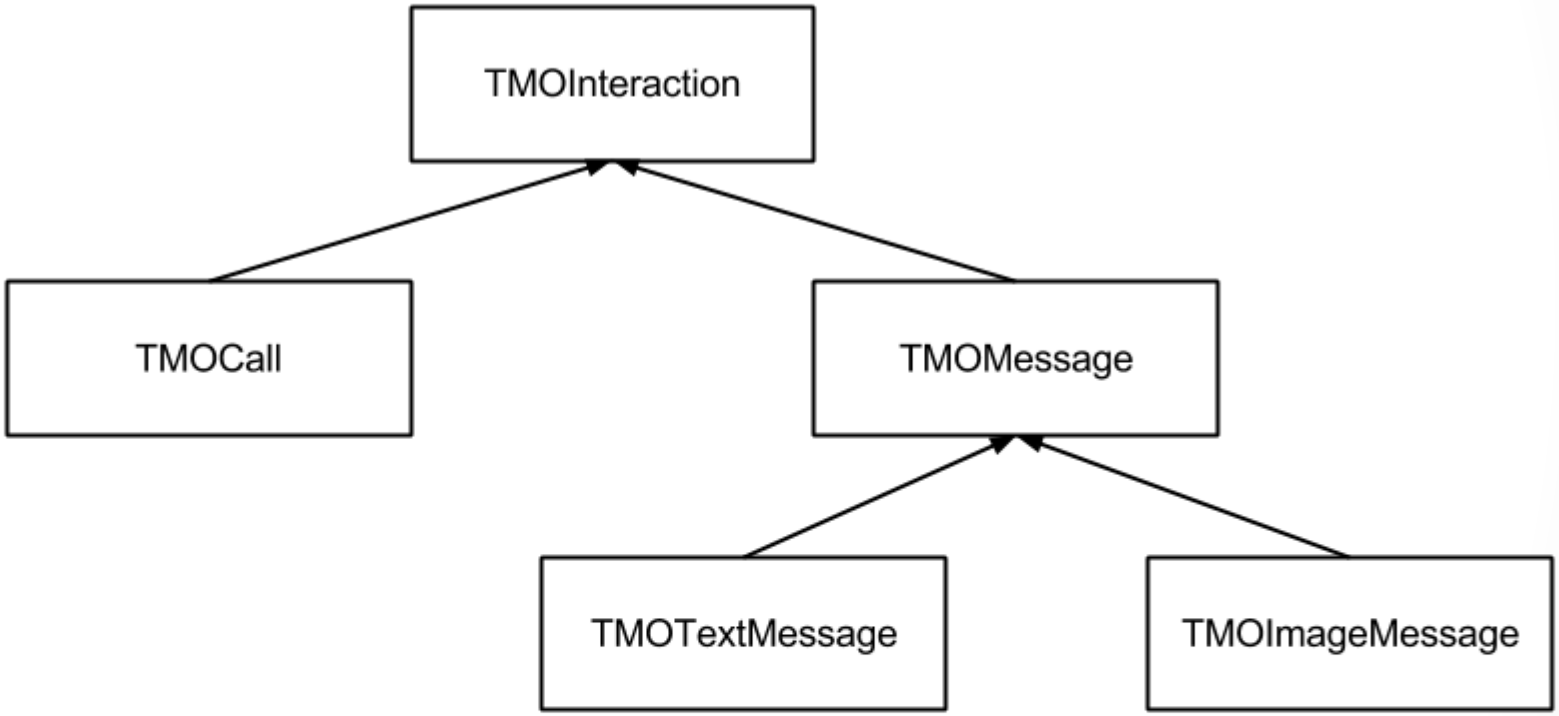
# Requirements

- The order of interactions should be identical to the order they were created
- The displayed timestamp should be correct
- The order should be consistent across clients
- The solution should be easy to implement and maintain

# Overall Changes

- New TMOCall object to represent phone calls
- Both messages and calls inherit from a new abstract TMOInteraction object
- Most properties that were exclusive to messages moved to the interactions

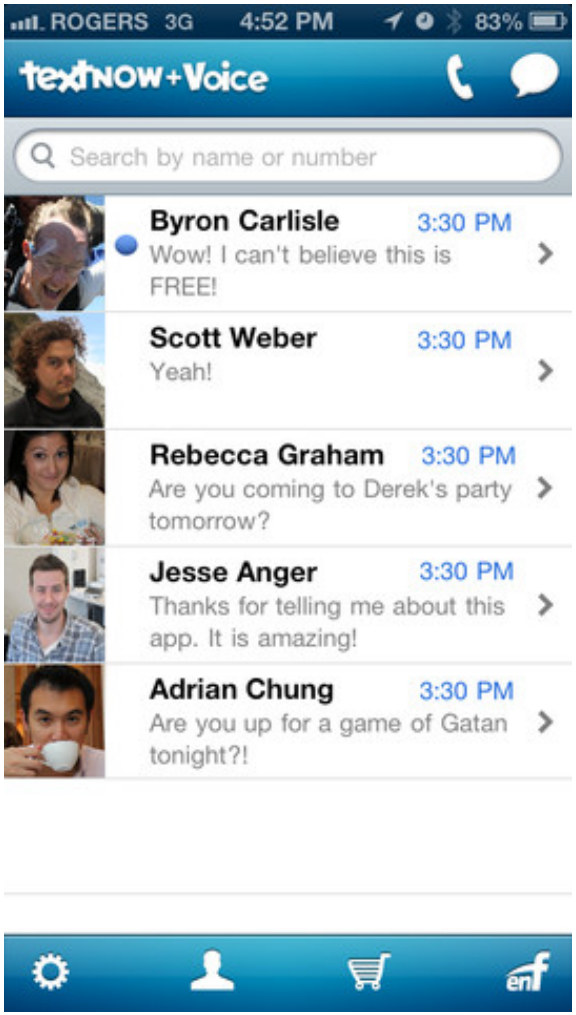
# Class Structure



# First Attempt

- Every interaction had a local timestamp
- Used for ordering and display
- Ordering was always correct
- Displayed timestamps did not support older messages

# Problem With First Attempt



# Second Attempt

- Every interaction had an ordering timestamp
- Used for ordering and display
- Messages used their server timestamp
- Calls used a local timestamp

# Second Attempt Problems

- Time is inconsistent across servers and devices
- Ordering was different on all devices
- Interactions were displayed out of order

# Third Attempt

- All interactions had an ordering timestamp and a local timestamp
- Ordering timestamps were used for display
- Messages are the same as before
- Calls used the latest server timestamp
- Interactions sorted first by the ordering timestamp then by the local timestamp

# Third Attempt Problems

- The latest server timestamp eventually became stale
- As a result, the displayed timestamps were incorrect on calls

# The Chosen Solution

- All interactions had an ordering id and local timestamp
- Set to the message id for messages and their remote timestamp for display
- Calls used the latest ordering id from the server and local timestamp for display
- Ordering was based on the ordering id and local timestamp

# The Chosen Solution Cont.

- Did not rely on time as primary ordering
- Led to accurate display times and correct ordering across devices
- Was relatively simple to implement; only required a few changes

# Summary

- Needed a simple method to combine elements across multiple clients
- Relying on only local time did not support older messages
- Time should not be used to primarily order elements regardless of the situation
- Instead rely on a unique identifier
- Store all elements in one place!

# Thank You!