

# Lessons Learned from Making Mobile Apps Resilient

Matthew Soucoup  
Sr Cloud Developer Advocate  
Microsoft

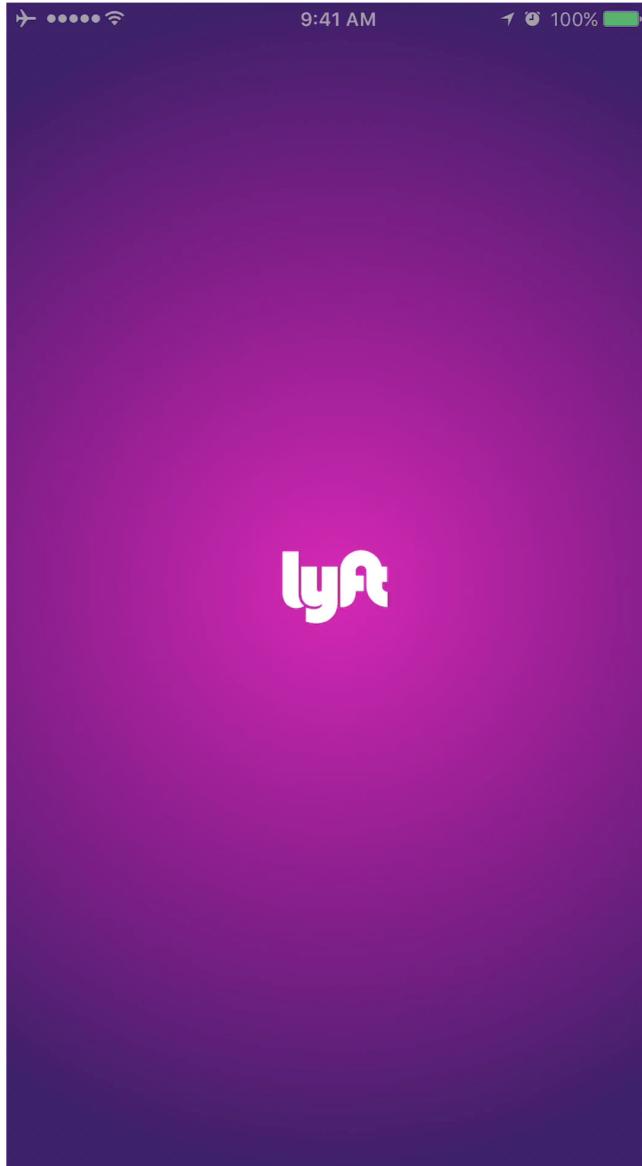
Level: Intermediate



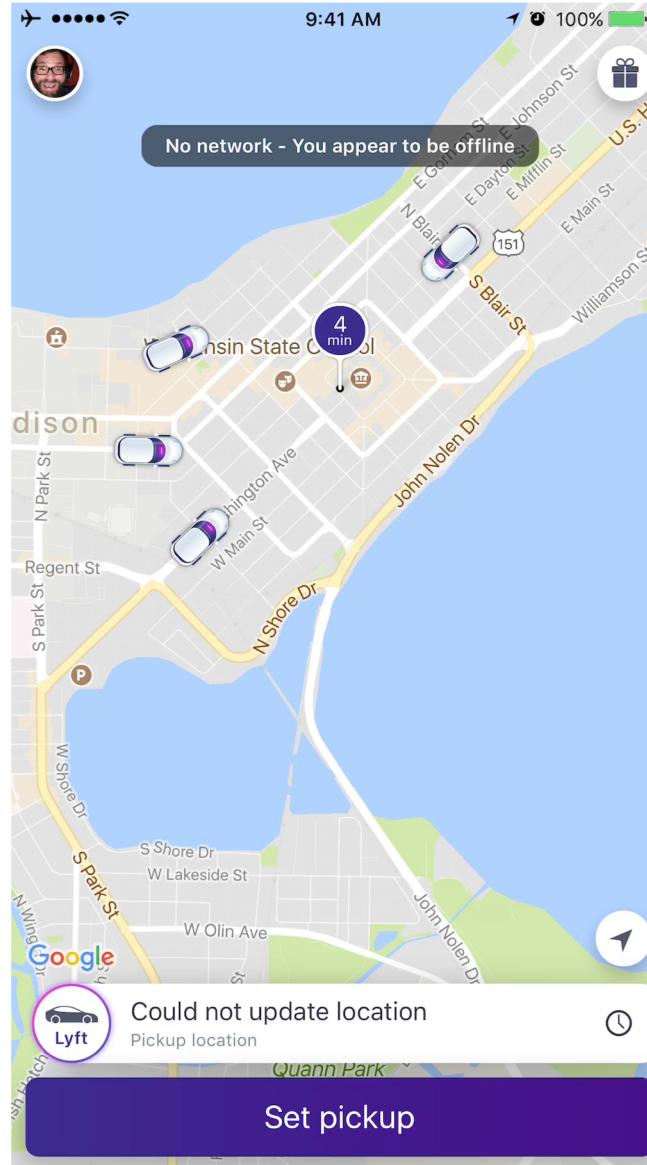
Code Again for  
the First Time!



# Not so great offline...



# Not so great offline...



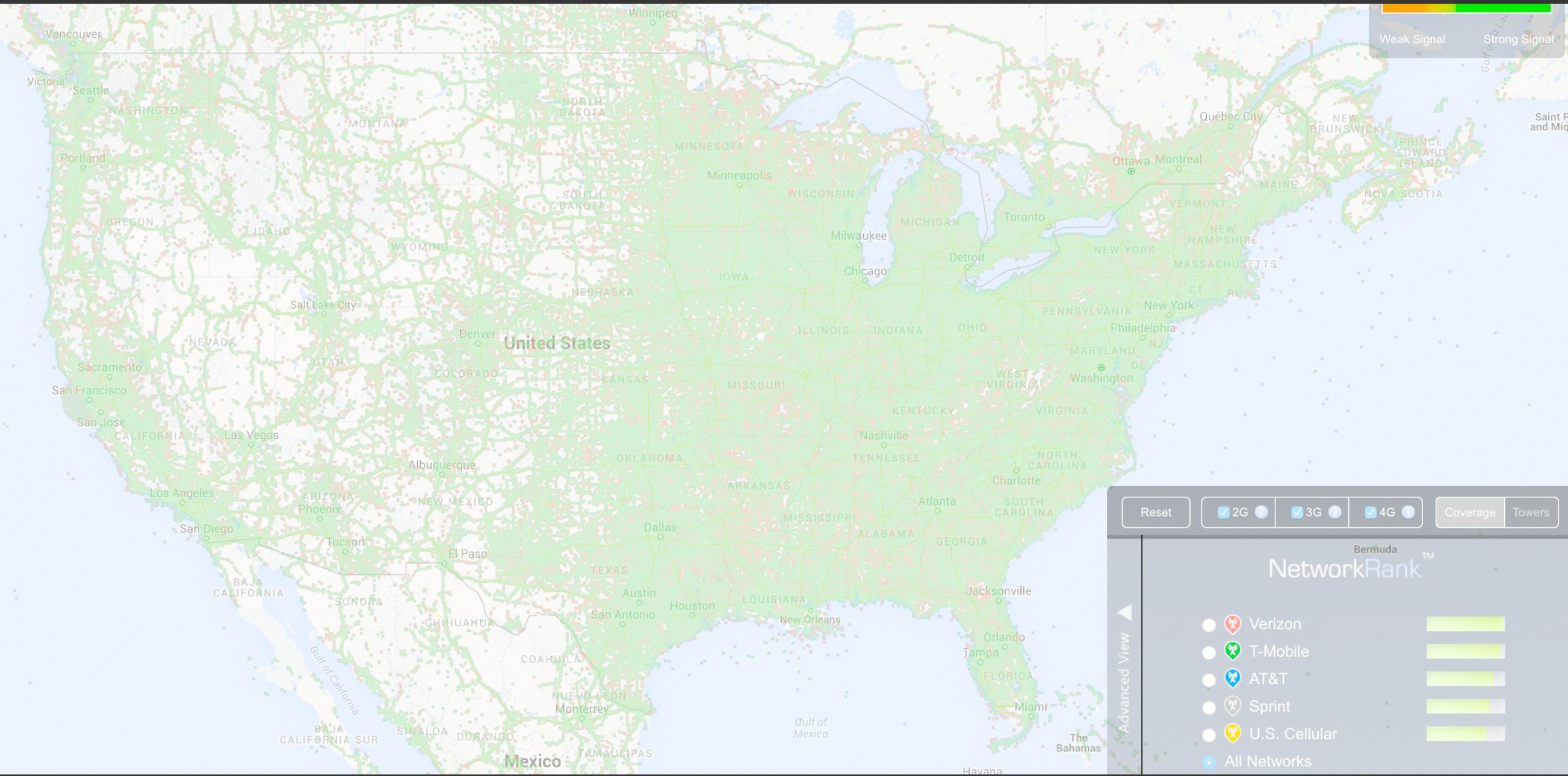
@codemillmatt

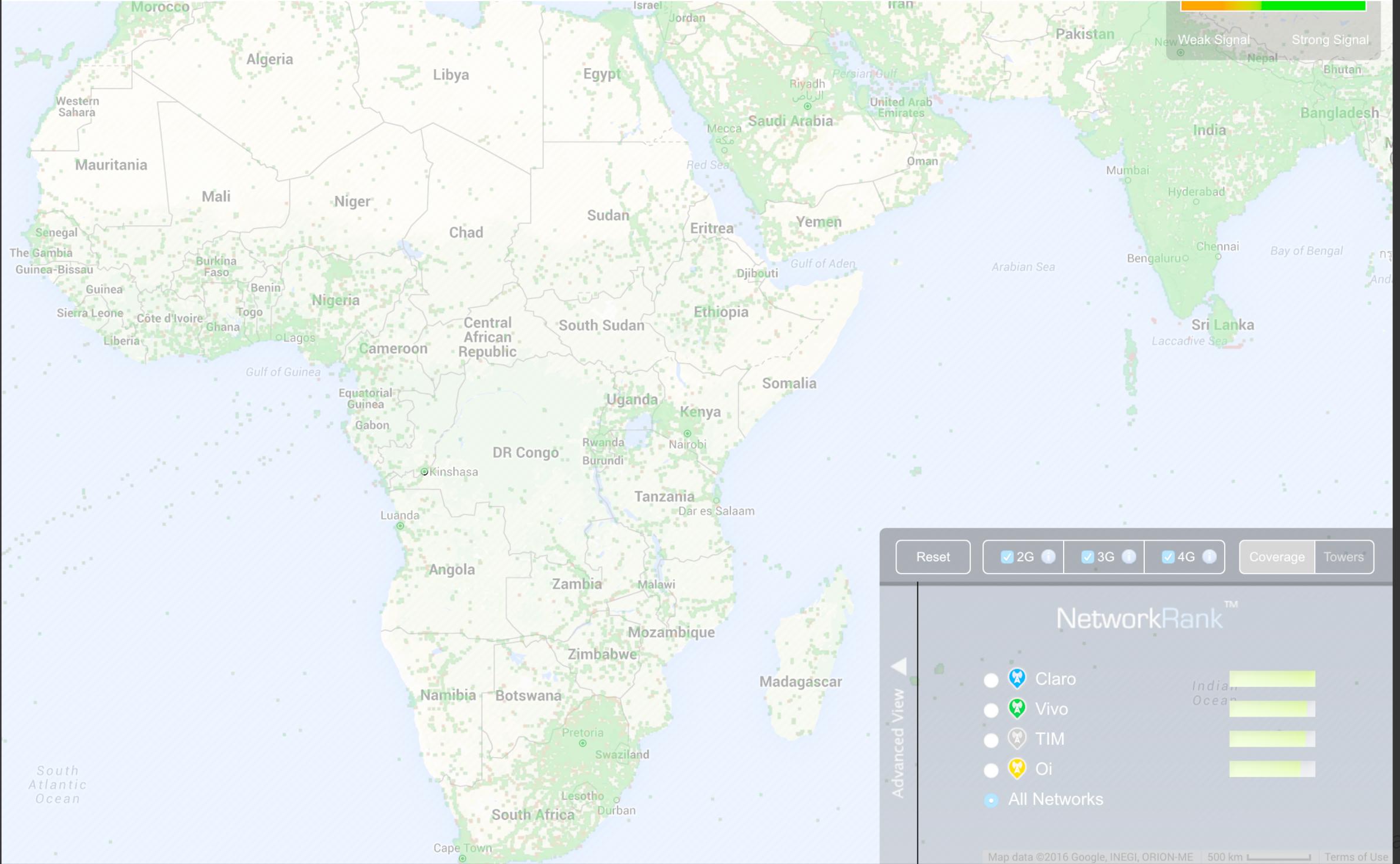
# Disconnected - Usable

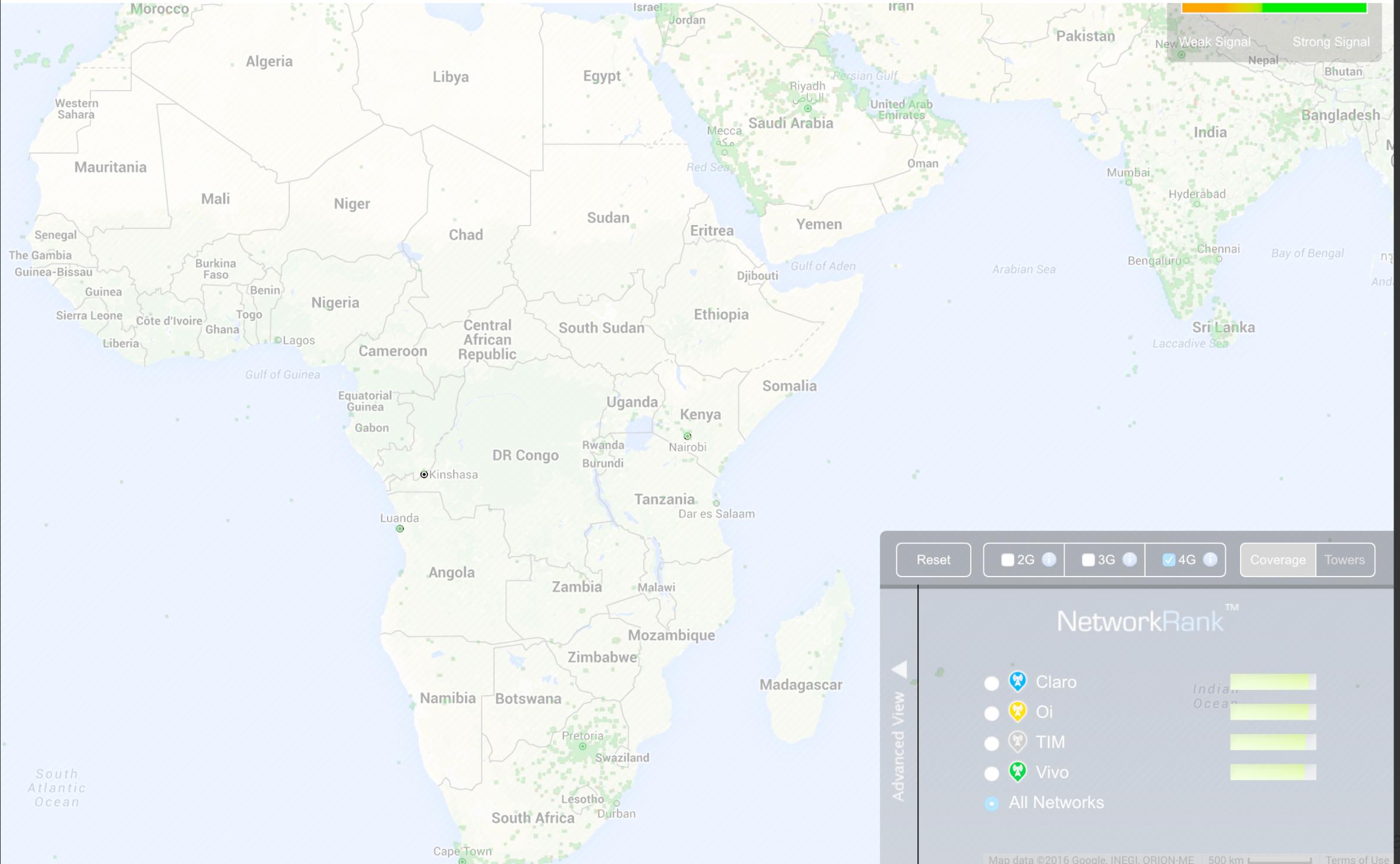


# Disconnected -> Reconnected









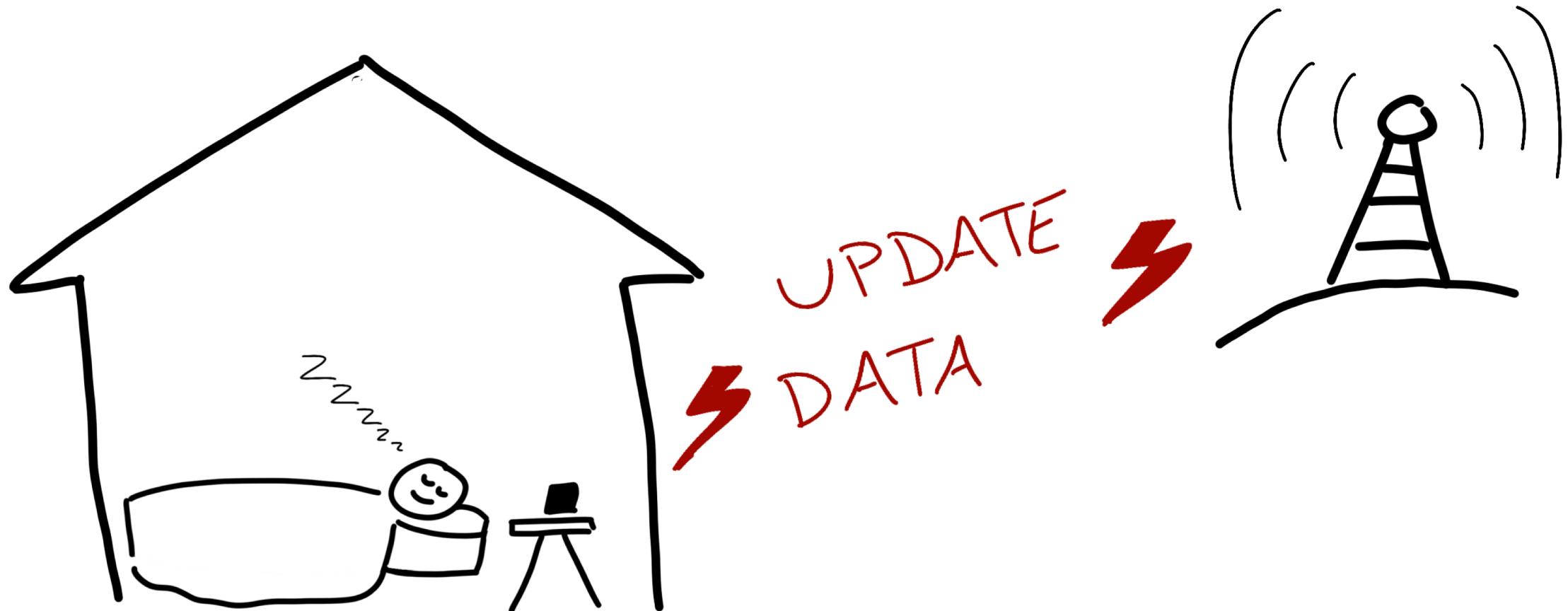
# Conflicts...



# Conflicts...



# Silent push...



# Azure Mobile App Services

# Why Use Azure Mobile App Services?

Extremely Powerful

Flexible

Cross-platform  
client SDK

Open source C#



Xamarin, iOS, Android  
and Windows SDKs

Offline sync



R  
E  
S  
T  
A  
P  
I

Offline Sync



Data Connections



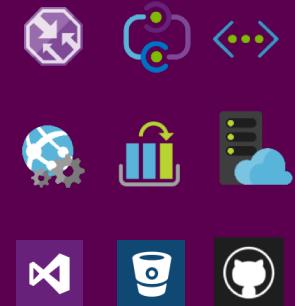
User Authentication



Push Notifications



Backend



# Agenda

Client-side schemas

Offline data

Conflict resolution

Silent data push

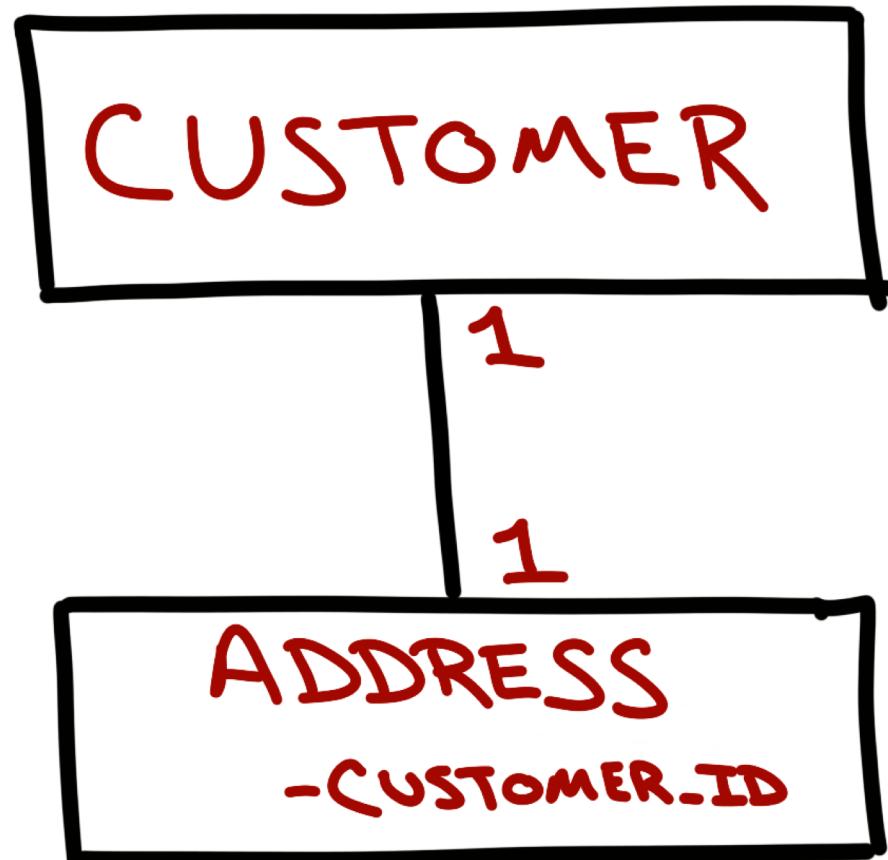
# Client-side schemas



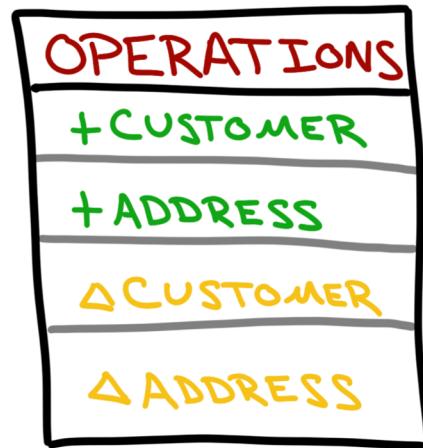
“Keep your client-side schema flat.”

-Matt ... just now

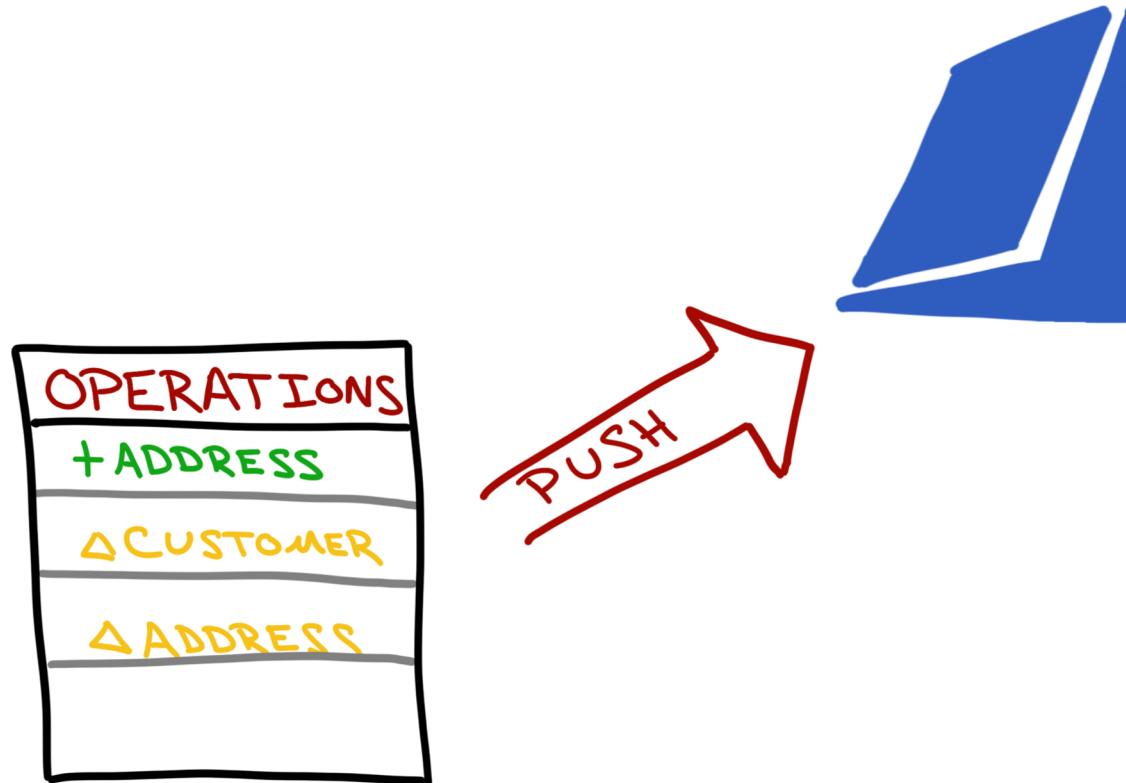
# Flat schemas – 1:1



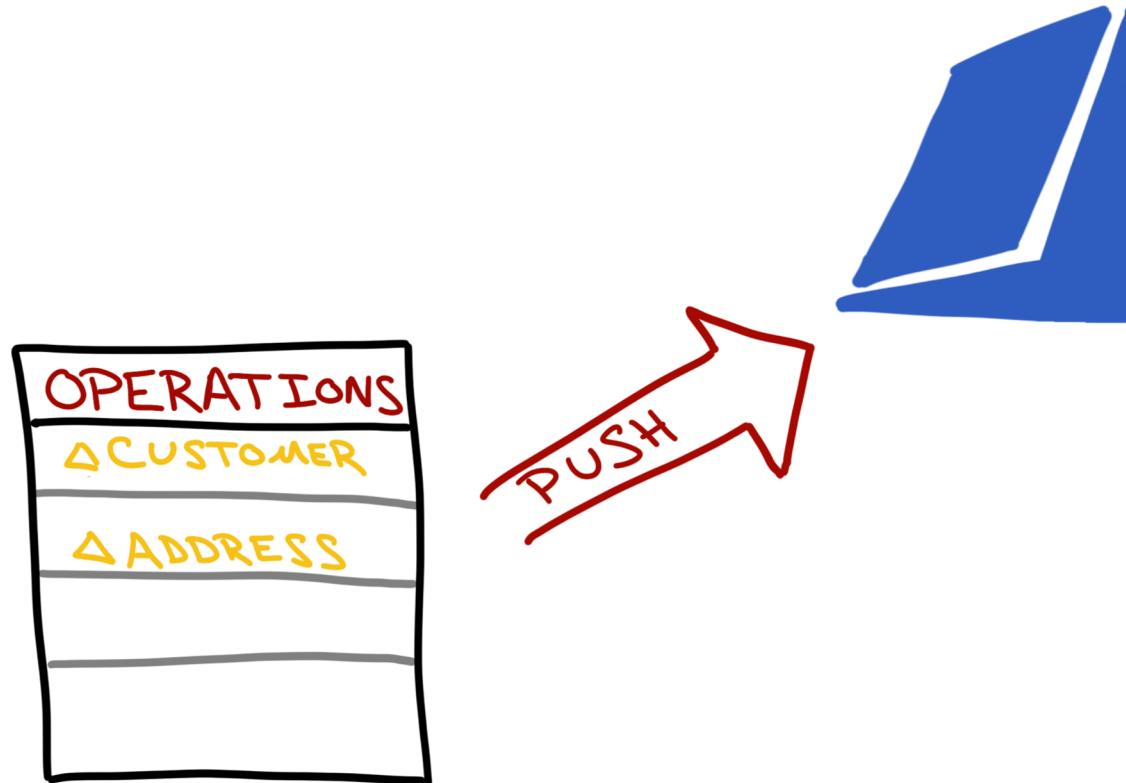
# Operations table – Offline sync



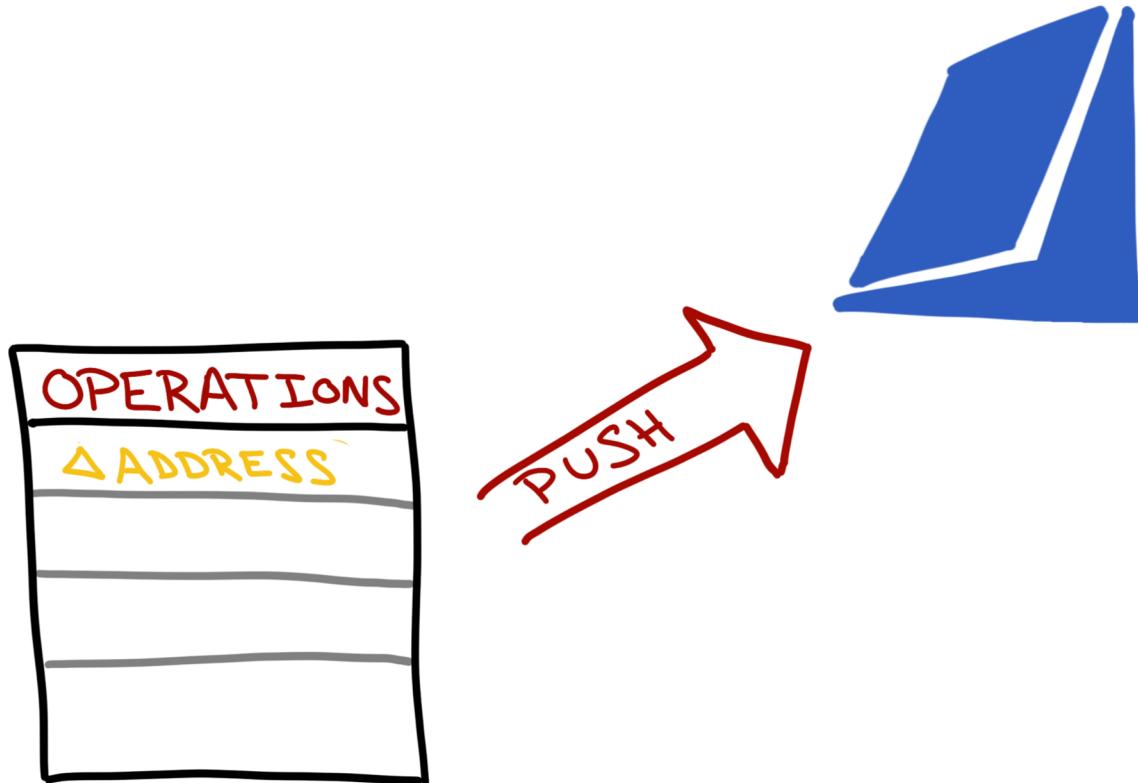
# Operations table – Offline sync



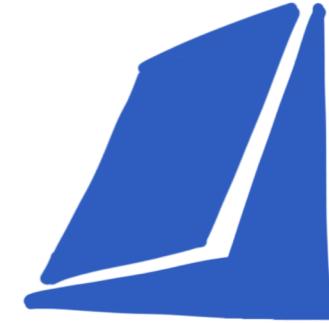
# Operations table – Offline sync



# Operations table – Offline Sync



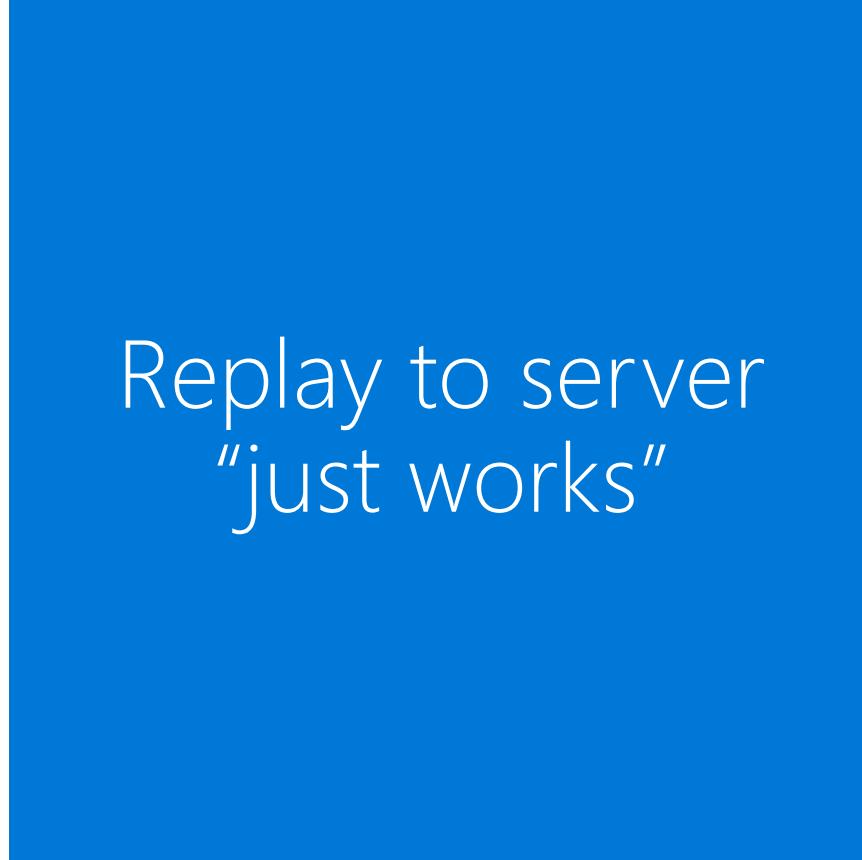
# Operations table – Offline Sync



# Offline sync with 1:1



Data relationships  
managed on client



Replay to server  
“just works”

# Troubles with 1:Many - Server

Server-side  
ICollection<T>

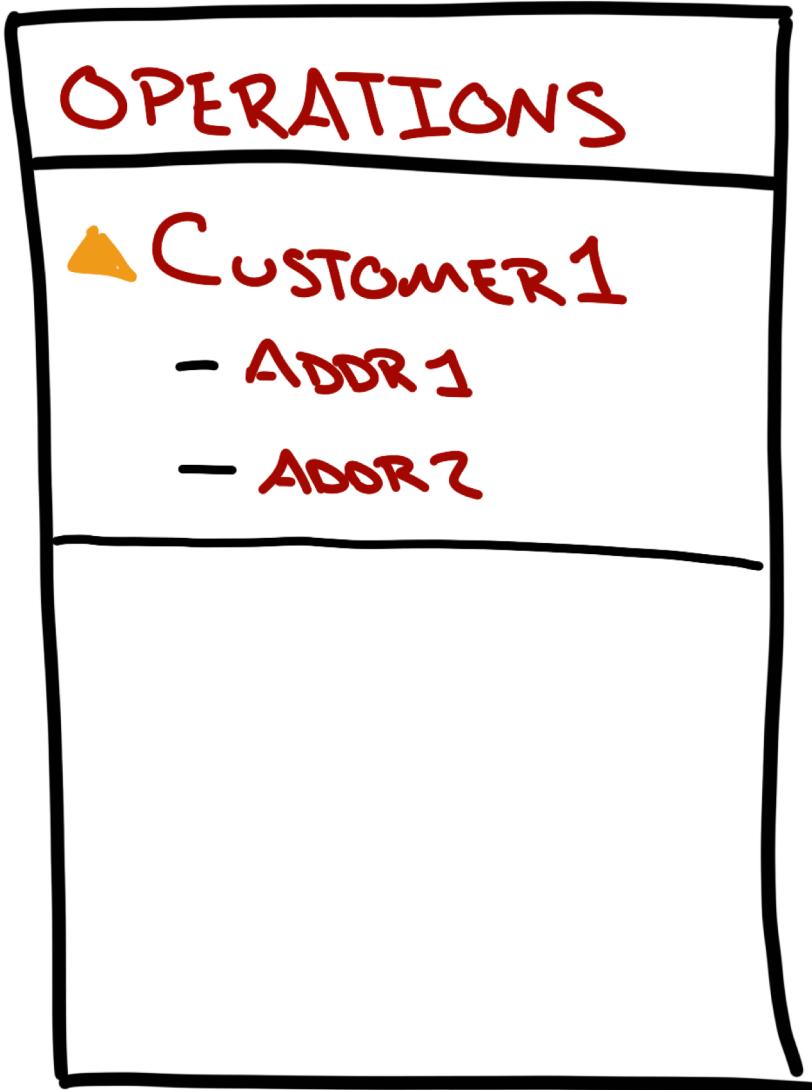
Obey OData

\$expand must  
be used

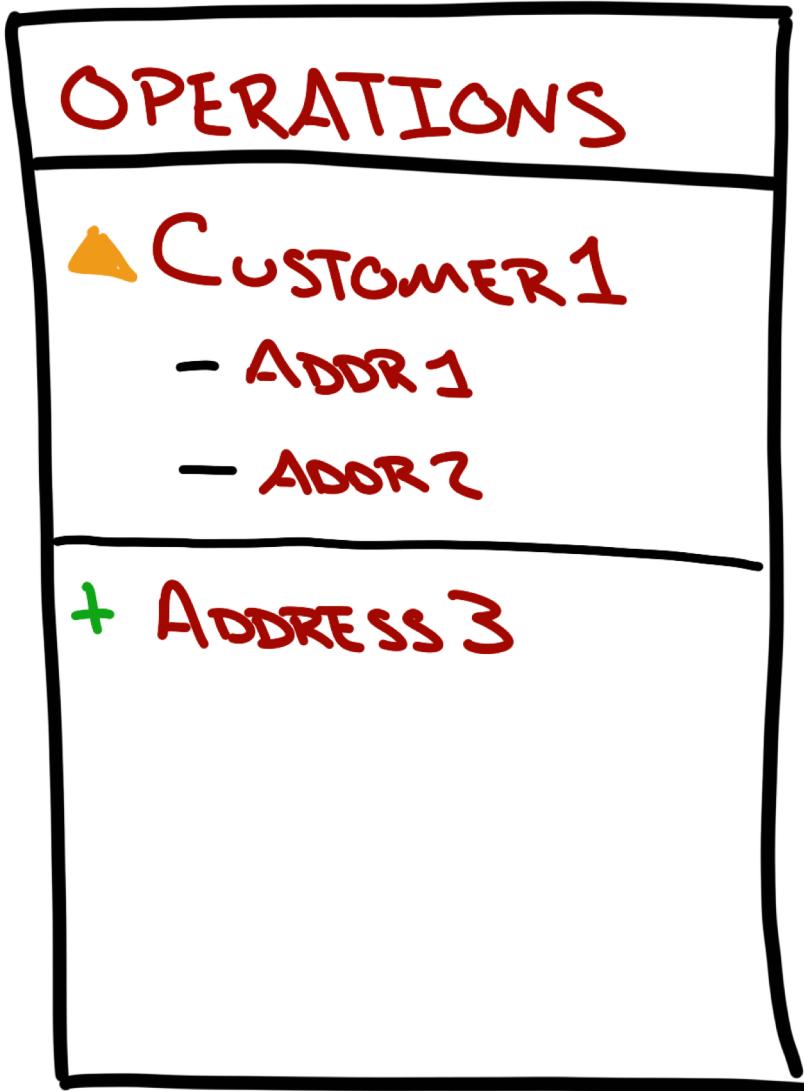
TableController<T>  
doesn't \$expand

Read-only  
results

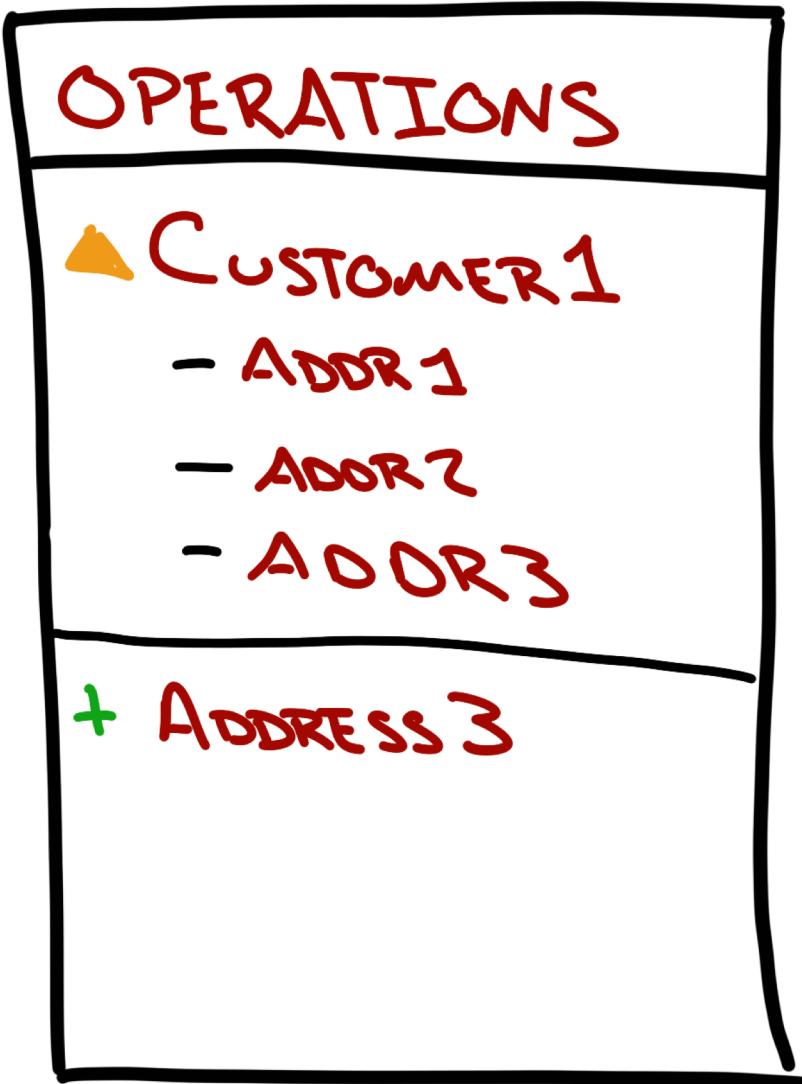
# Troubles with 1:many - Client



# Troubles with 1:many - Client



# Troubles with 1:many - Client



# 1:many – How to work around

New tables on  
server to model  
1:1 relationships

Use the Mapped  
Entity Domain  
Manager

Customized  
Web APIs

# Lessons learned

Flat data (1:1)

Mobile specific APIs

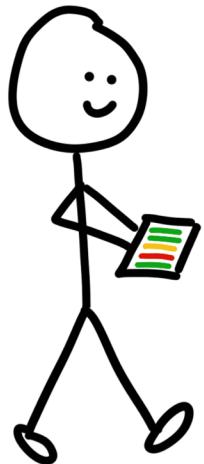
Optimize for bandwidth

Optimize for MVVM

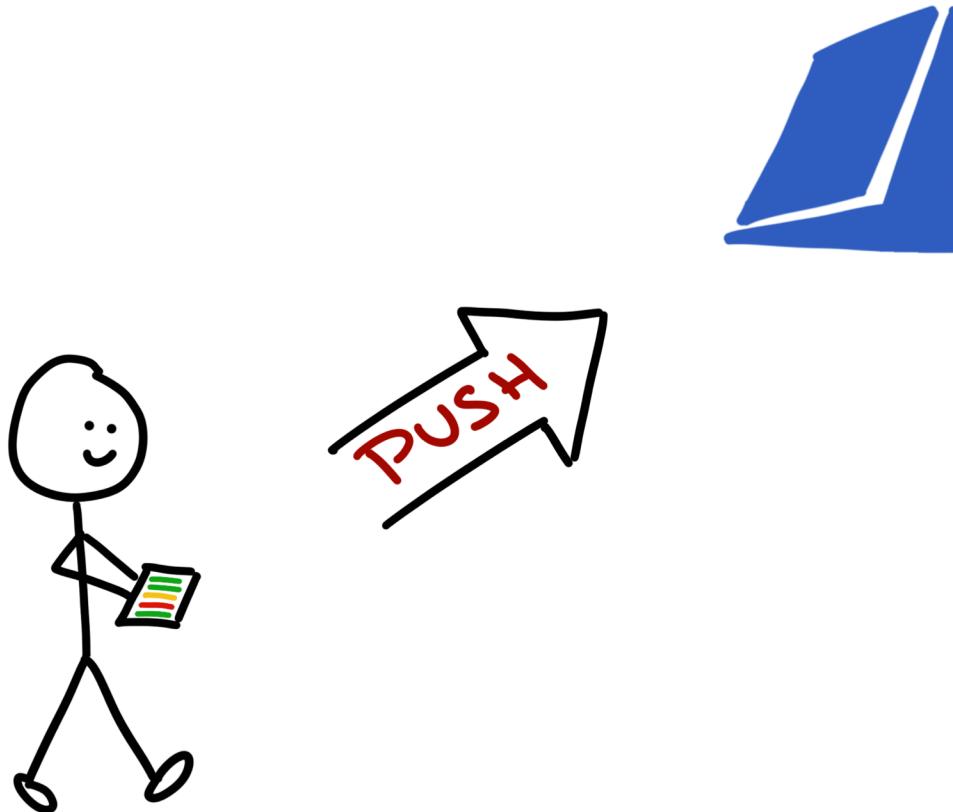
# Offline data



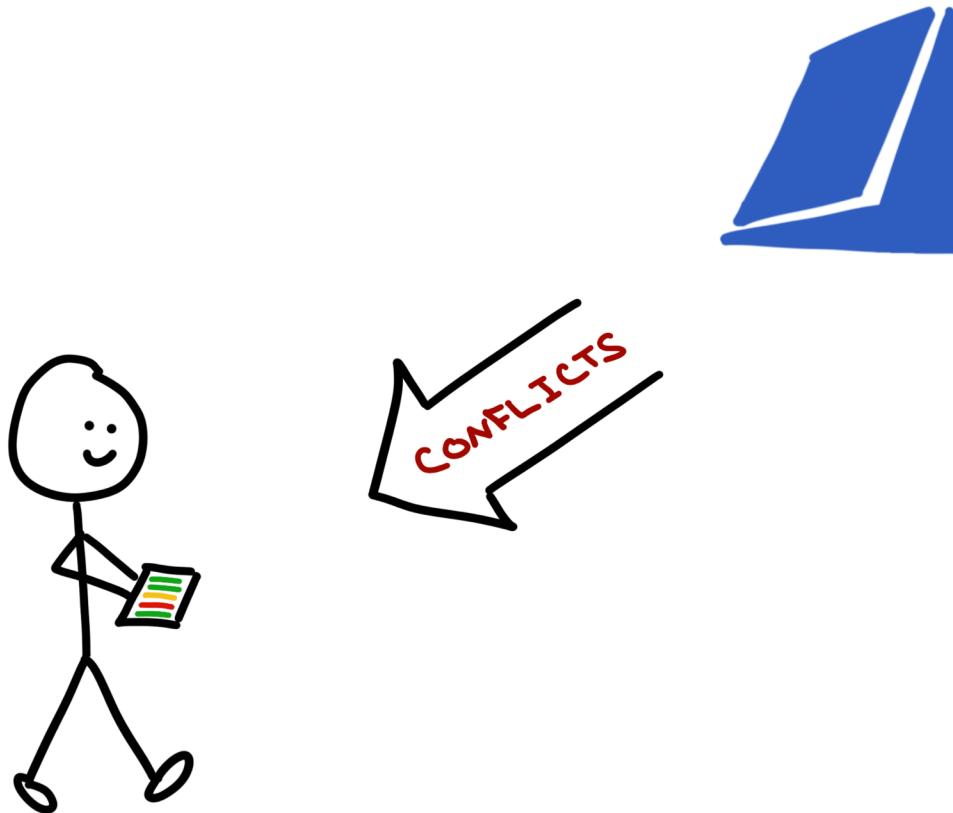
# Offline data flow



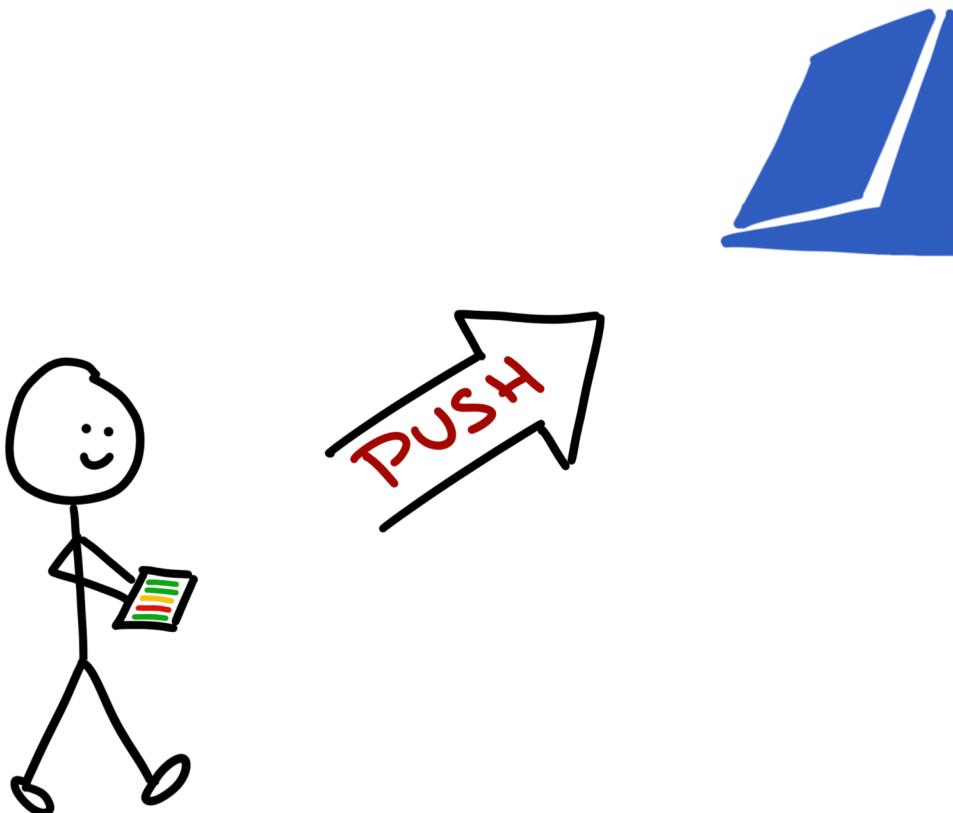
# Offline data flow



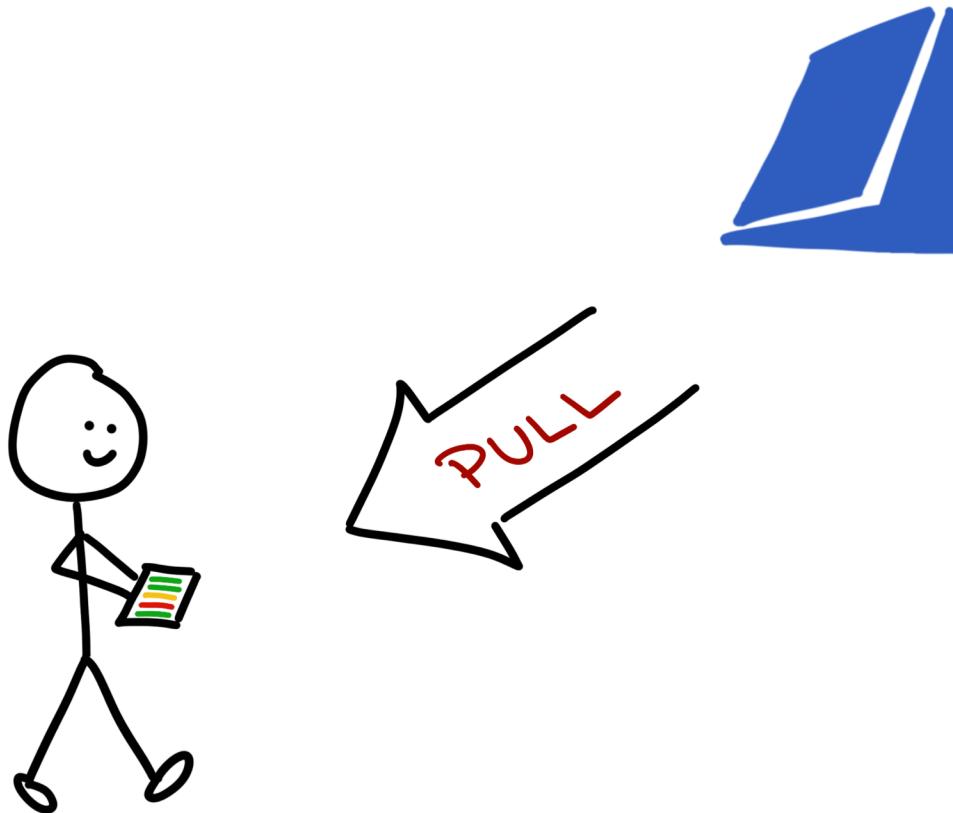
# Offline data flow



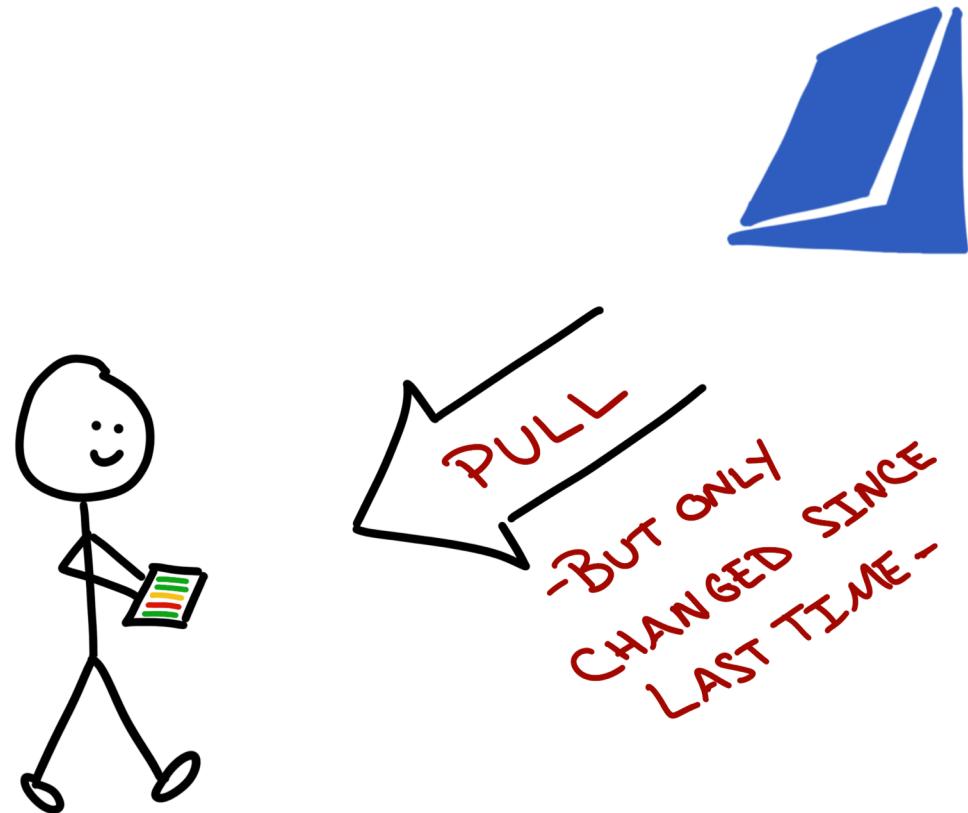
# Offline data flow



# Offline data flow



# Offline data flow



# Demo



# Lessons learned

No need to push  
data immediately

Only do large  
pushes on WiFi

Only store data  
offline when it  
makes sense

Incremental sync  
is your friend

SyncContext has  
pending ops

# Conflict resolution



# Conflicts...



# Conflicts...



# How can Zumo tell?

Id

CreatedAt

UpdatedAt

Deleted

Version

# Version column

Byte array

Maintained on  
the server

On push, Zumo  
takes care of  
updating value

# 3 possible conflict outcomes

Server wins

Client wins

Hybrid – combo  
of both win

# Demo



# Lessons learned

Push shared data ASAP

Never mess with the  
version column (normally)

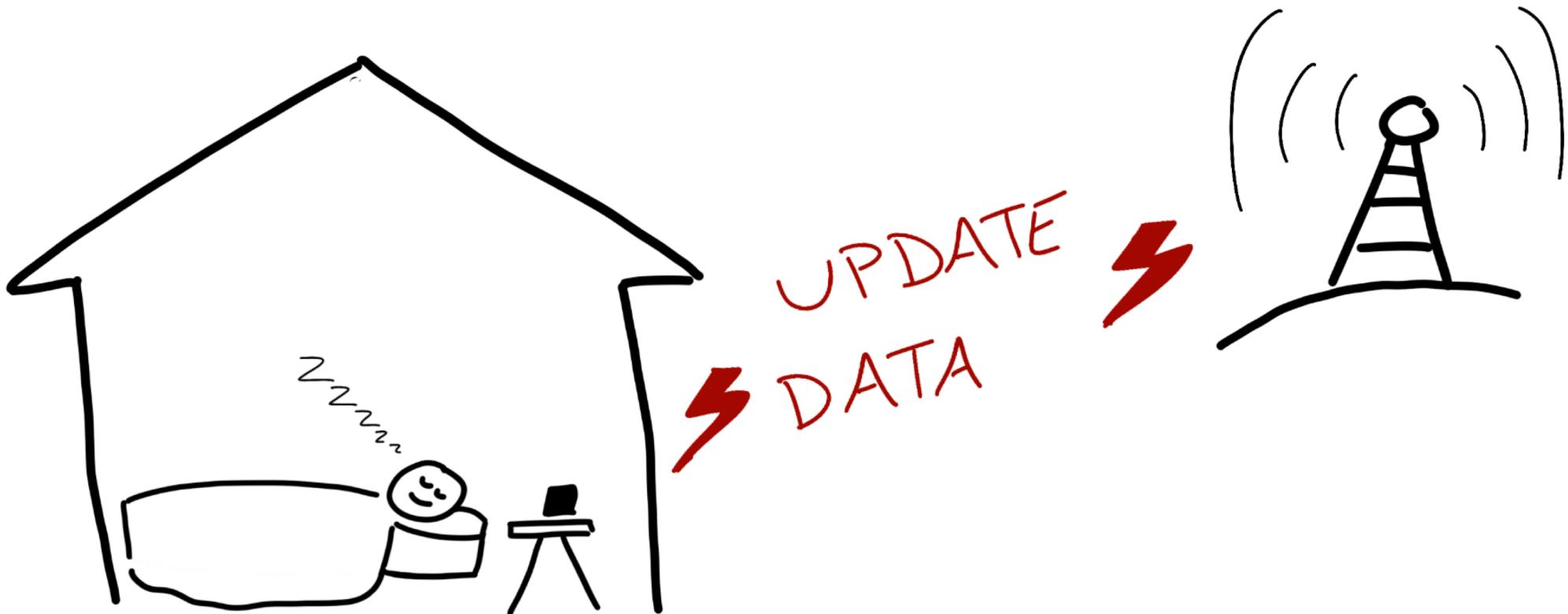
Different exceptions for  
offline vs online push

Don't ask the user if  
their data should win

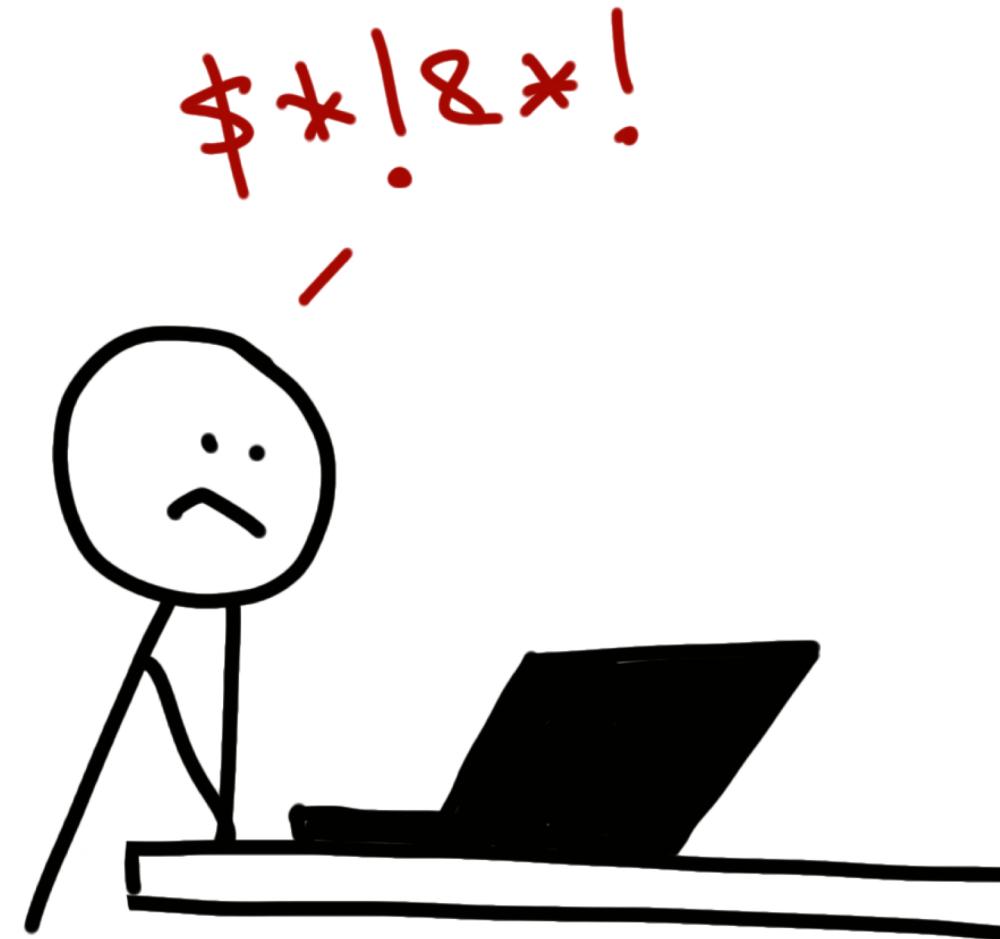
# Silent push



# Silent data push

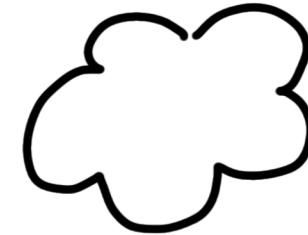


# Developing Push Enabled Apps

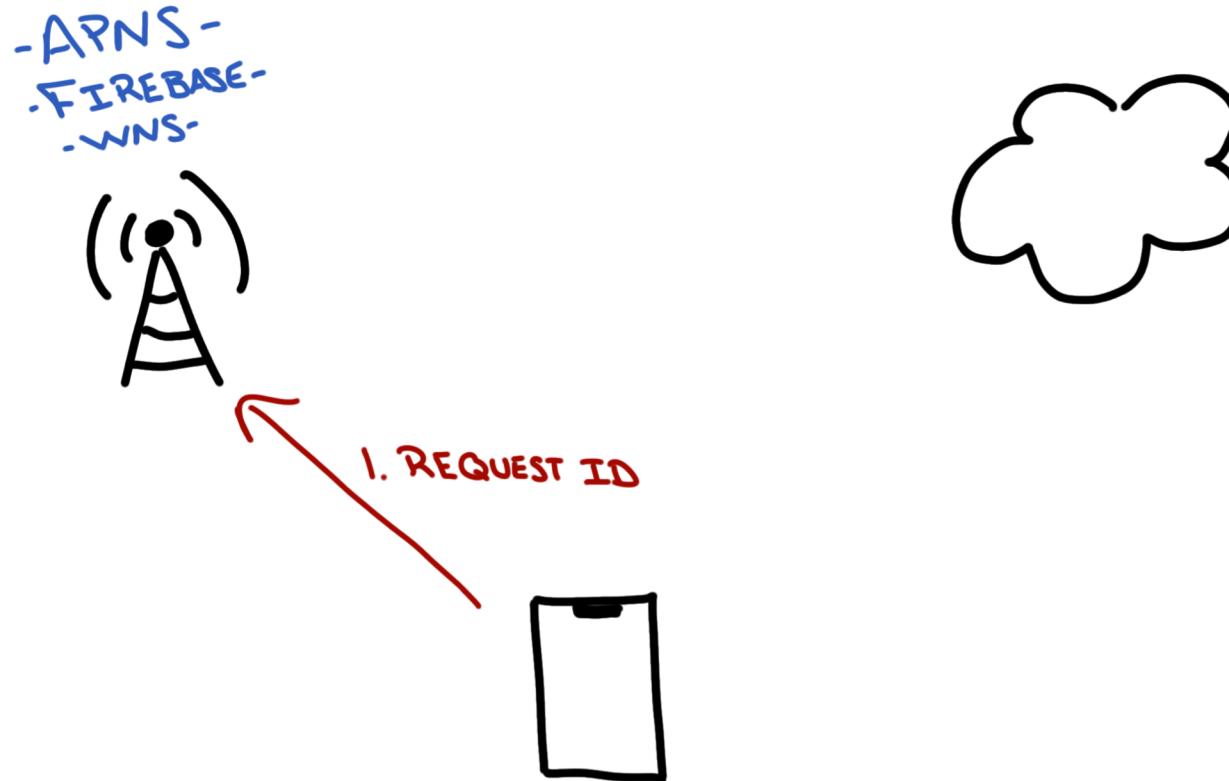


# Developing Push Enabled Apps

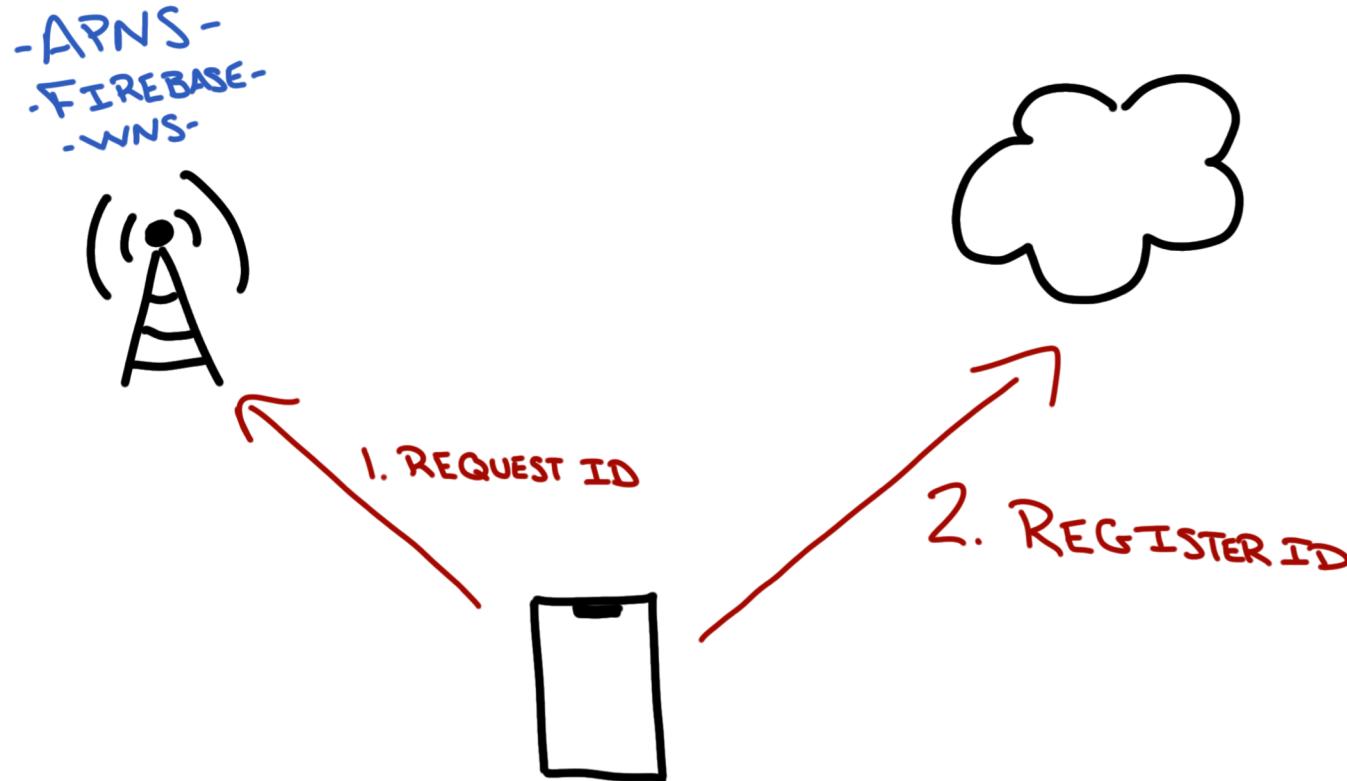
-APNS-  
-FIREBASE-  
-WNS-



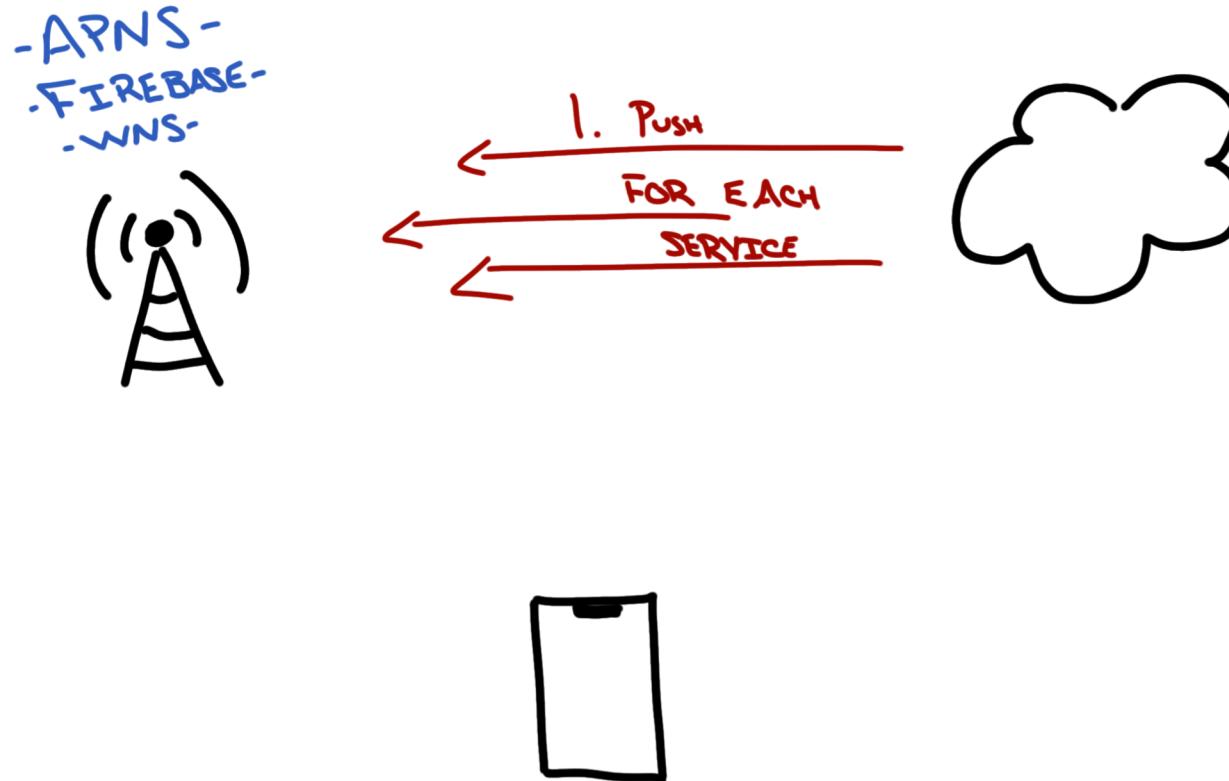
# Developing Push Enabled Apps



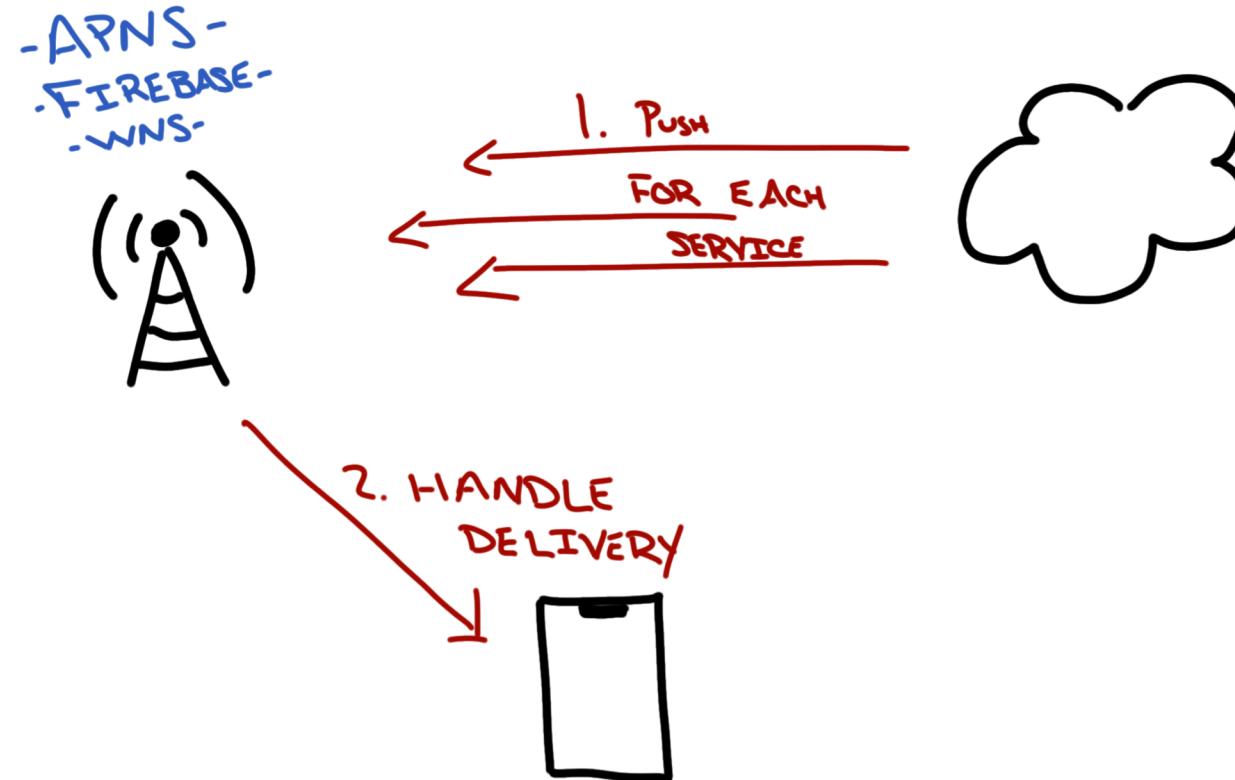
# Developing Push Enabled Apps



# Developing Push Enabled Apps

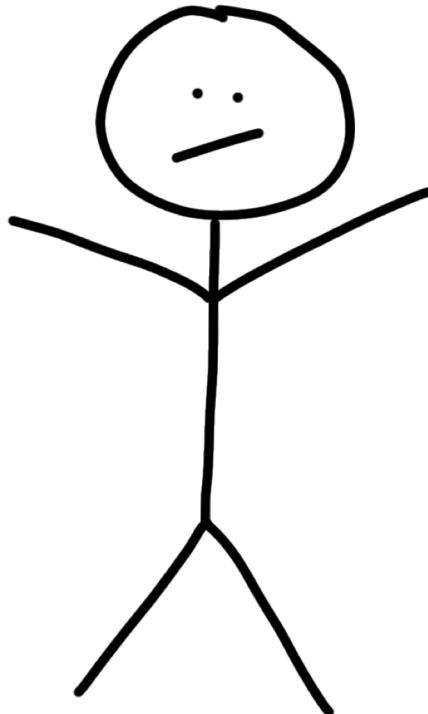


# Developing Push Enabled Apps



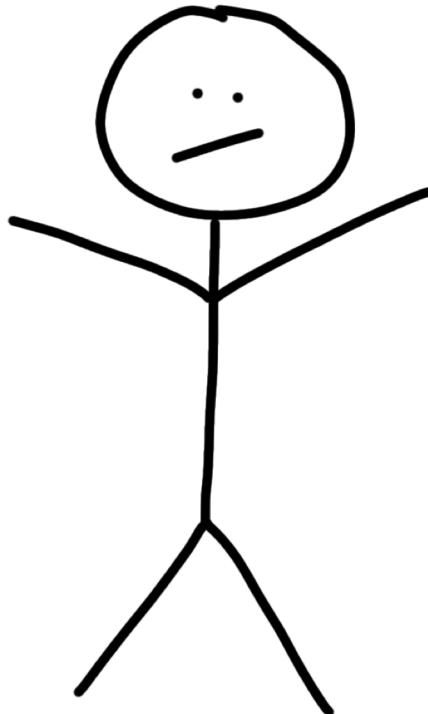
# Developing Push Enabled Apps

SERIOUSLY ?!?

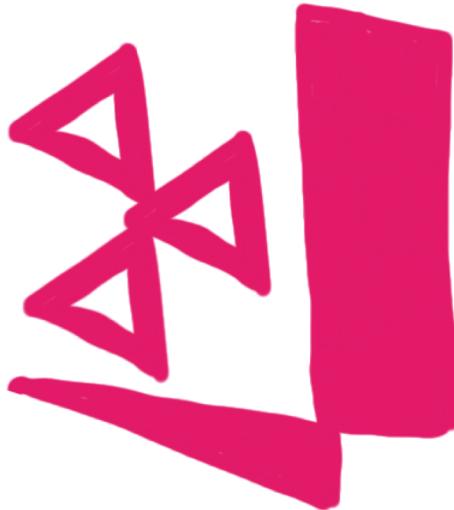


# Developing Push Enabled Apps

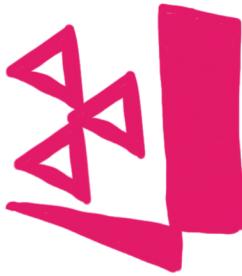
SERIOUSLY ?!?



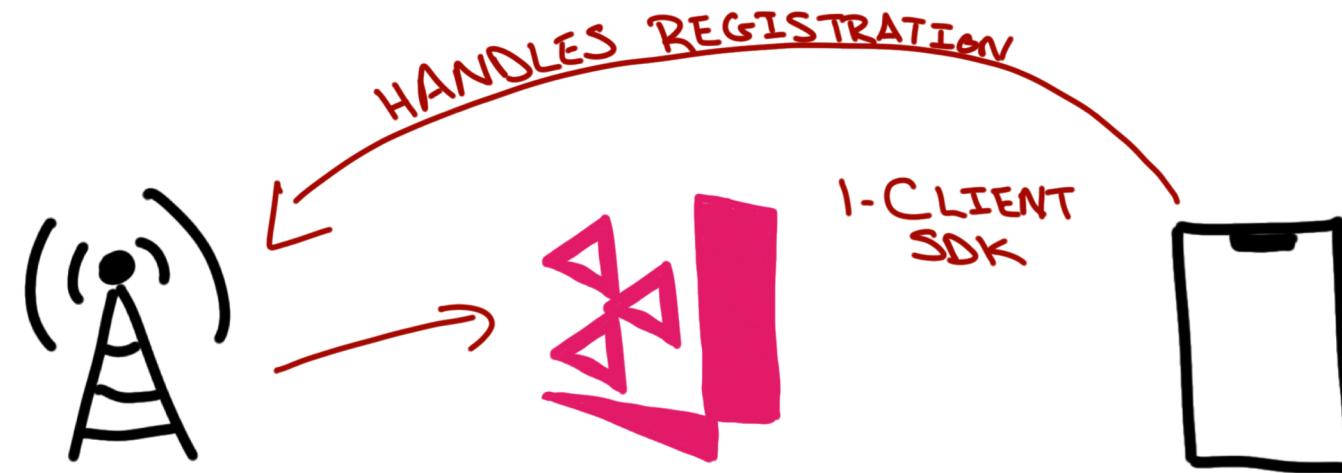
# Visual Studio App Center



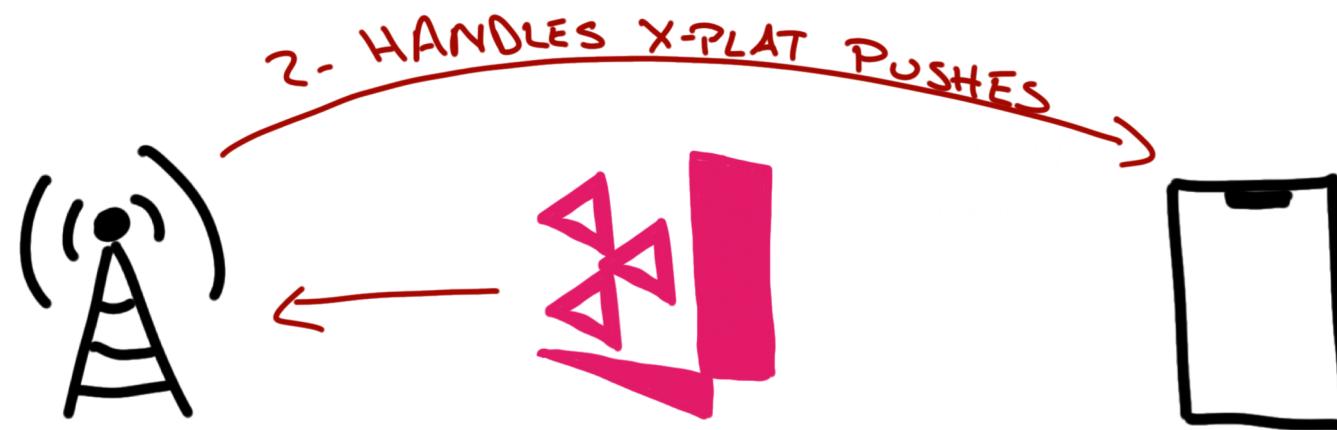
# Push with VS App Center



# Push with VS App Center



# Push with VS App Center



# Silent data push - benefits

Keeps data up to date  
no interactions

Can be targeted

# Demo



# Lessons learned

Spend time with  
APNS and Firebase

Need to account  
for message not  
arriving

App Center  
Notification Hub  
templates

# Summary

Client-side schemas

Offline data

Conflict resolution

Silent data push



# Thank you!

<https://msou.co/9m>

Matthew Soucoup  
Sr Cloud Developer Advocate  
@codemillmatt

