Practical ERSync

David Aspinall
Global Village Consulting Inc.

Outline

Introduction

- Who am I
  - David Aspinall
  - Developer / Consultant for Global Village Consulting Inc.

Why ERSync?

- Why not REST/SOAP/...
- Framework to simplify data distribution to mobile apps
- Contribute to the WOCommunity
- Leverage for my clients and projects

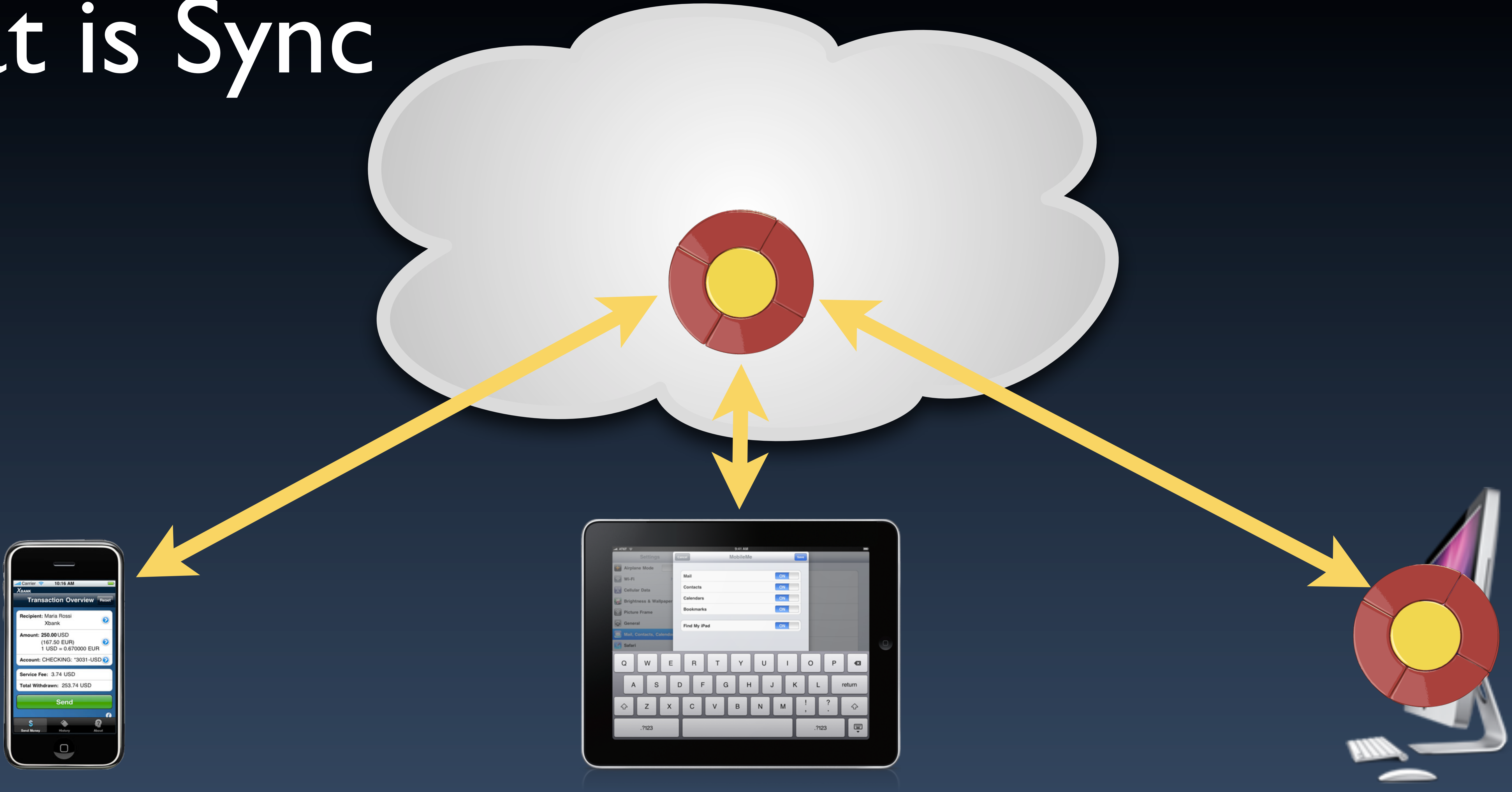
Objective

The goal is to make 2 disconnected sets of data match

.. as quickly as possible

.. in a way the user expects

What is Sync



Transaction Overview

Recipient: Maria Rossi Xbank

Amount: 250.00 USD (167.50 EUR) 1 USD = 0.670000 EUR

Account: CHECKING: *3031-USD

Service Fee: 3.74 USD

Total Withdrawn: 253.74 USD

Send

MobileMe

Mail ON

Contacts ON

Calendars ON

Bookmarks ON

Find My iPad ON

What is that cloud?

Dropbox

iCloud

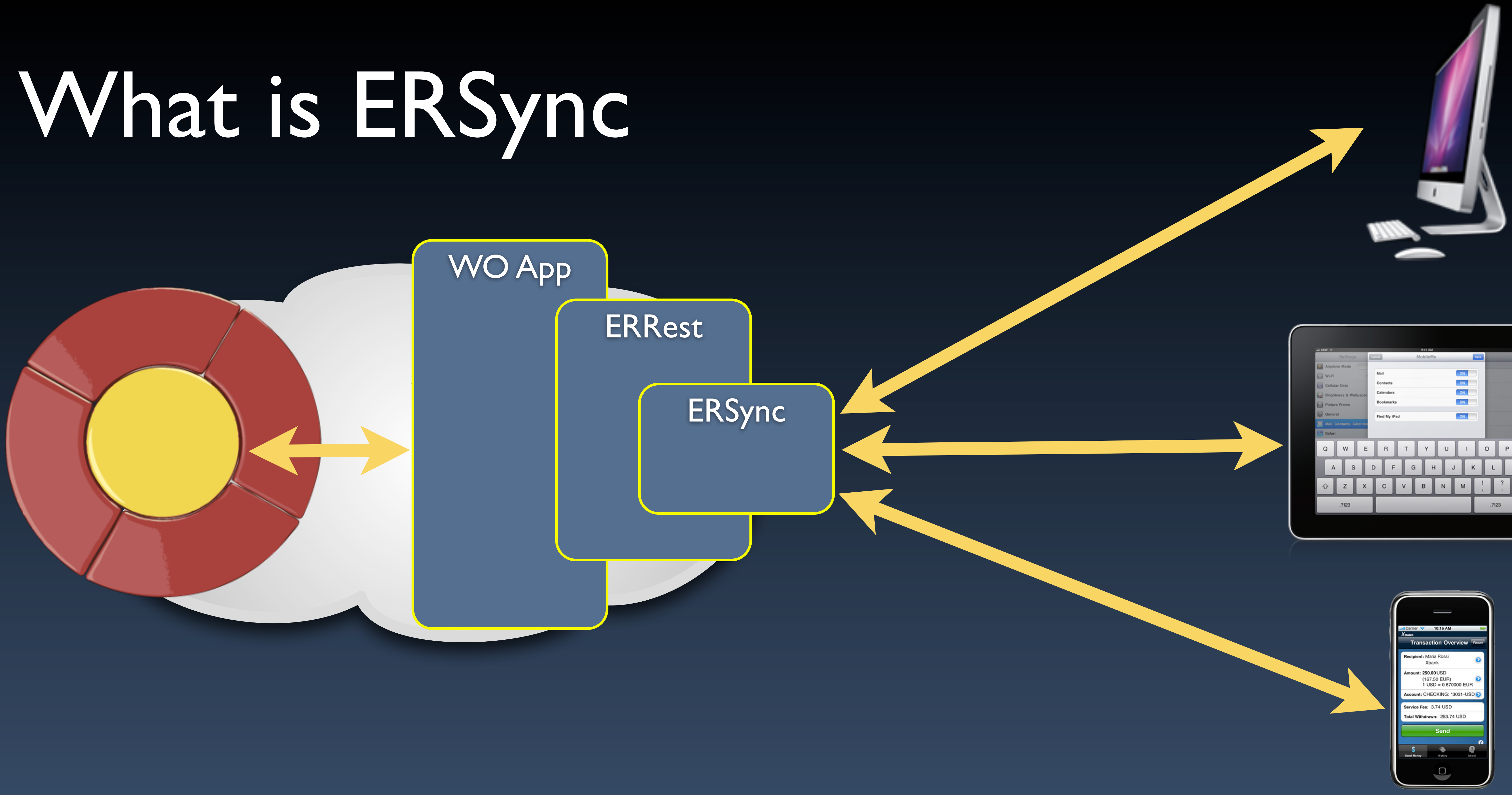
Miracles

We are Wonder-ful

Design Objectives

- Leverage our WO experience, products and services.
- Minimally impact our existing code base.
  - No server database changes in existing business system.
  - No sprinkling of interfaces or special logic in current system
- Simple client protocol and supporting library

What is ERSync



Integrating with WebObjects

- Built on ERRest (routes, transport ...)
- EOObjectStoreCoordinator.ObjectsChangedInStoreNotification
- No model changes required in existing system
- ERSync database can be different schema, host or database
- All relevant data changes must notify the sync engine
    - framework is linked in, turned on
    - distributed change notification (ERChangeNotificationJMS)

The Easy Part

```
public class Application extends ERXApplication {
public static void main(String[] argv) {
    ERXApplication.main(argv, Application.class);
}

public Application() {
    ERXApplication.log.info("Welcome to " + name() + " !");
    /* ** put your initialization code in here ** */
     ERXDatabaseContextMulticastingDelegate.addDefaultDelegate(new ERXEntityDependencyOrderingDelegate());

    ERXSyncHandler syncHandler = new ERXSyncHandler();
    syncHandler.setSyncAuthenticator(new SyncAuthenticationProvider());

    ERXSyncHandler.register(syncHandler);
}
}
```

What did that do?

- ERXSyncHandler extends ERXRouteRequestHandler
  - creates REST routes
  - adds change notification observer
- Sync Authenticator
  - this is the gateway class between ERSync and your Application

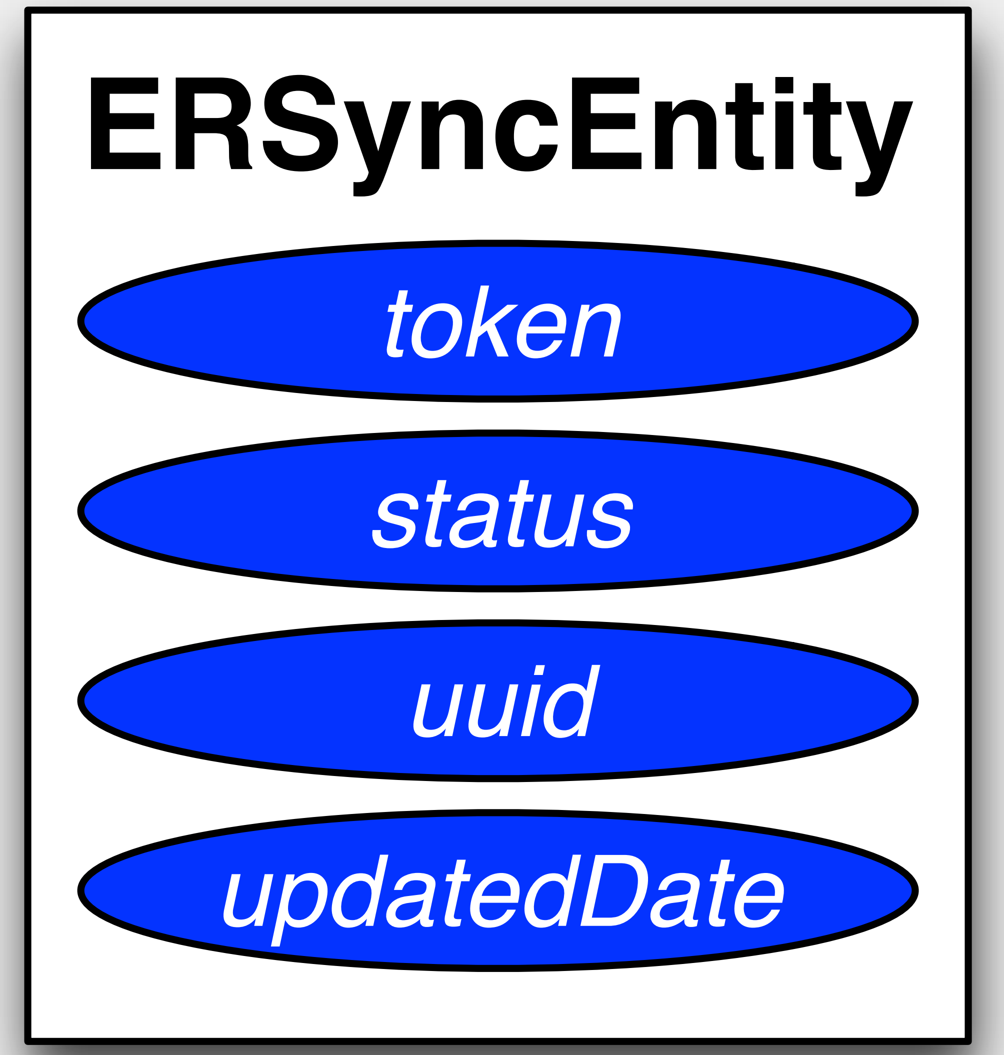
Sync Authenticator

- authenticates a user by username and password
  - does NOT implement authentication, it should call your logic
- provides list of Sync'able Entity Names
- provide all EOKeyGlobalID's for a given user
- basically CRUD processor

Sync Authenticator

```
   public interface ERXSyncAuthenticator
{
   public ERXSyncUser userForCredentials(String nme,String pwd, EOEditingContext ec);

   public NSArray<String> syncEntityNames();

   public NSArray<EOKeyGlobalID> syncObjectsForEntityUser(String entityName,
            ERXSyncUser usr, EOEditingContext ec);

   public EOEnterpriseObject syncInsertObject(EOEditingContext editingContext,
           EOEntity eoEntity, NSDictionary dict, ERXSyncUser user);

   public void syncUpdateObject(EOEnterpriseObject eo,
           NSDictionary dict, ERXSyncUser user);

   public void syncDeleteObject(EOEnterpriseObject eo, ERXSyncUser user);

}
```

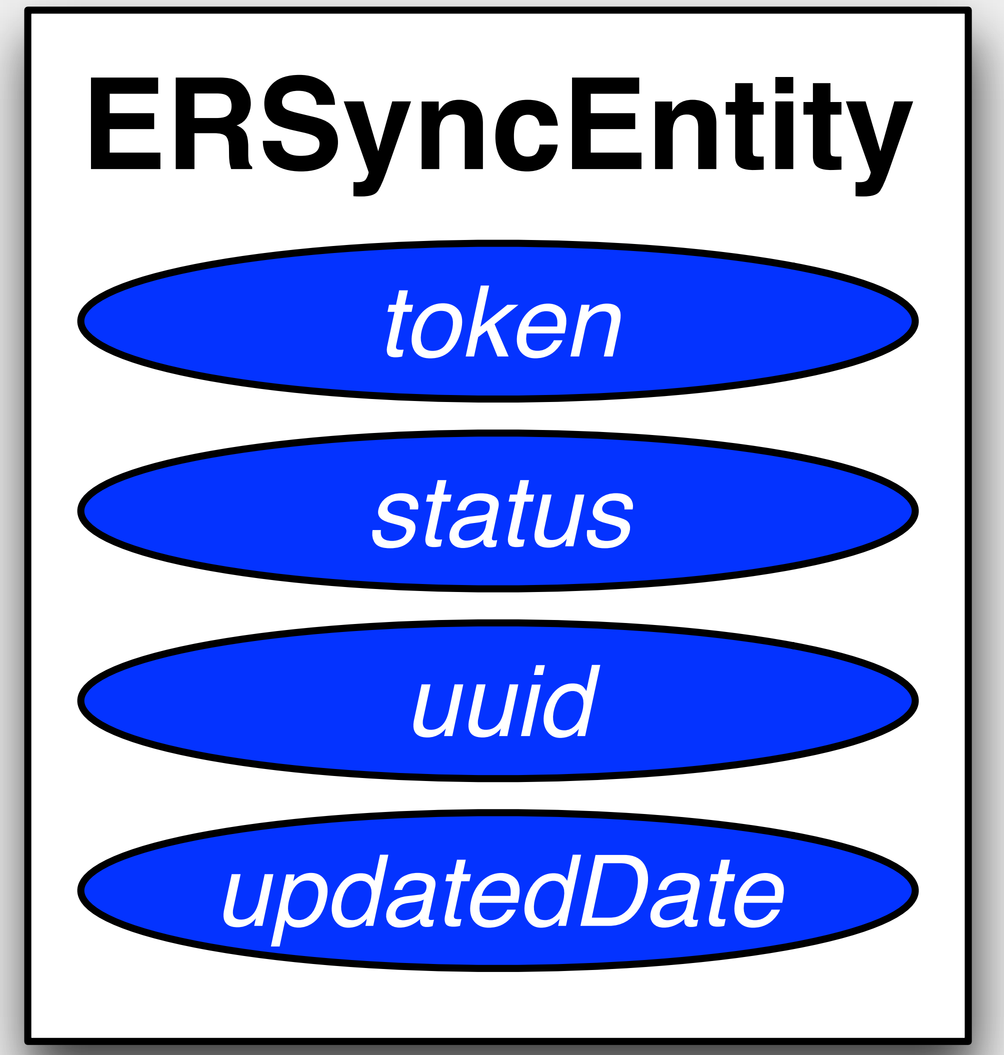
ERSync tracking



Token

- Must be able to reconstruct the EO
- Cannot be a FK because we need to track deletes
- Currently using
  - EntityName:pk[-pk*]
  - Note:1000001
- Planning to change it to URI
  - ersync://EntityName/pk[/pk*]
  - ersync://Note/100001
  - ersync://Compound/10001/4432

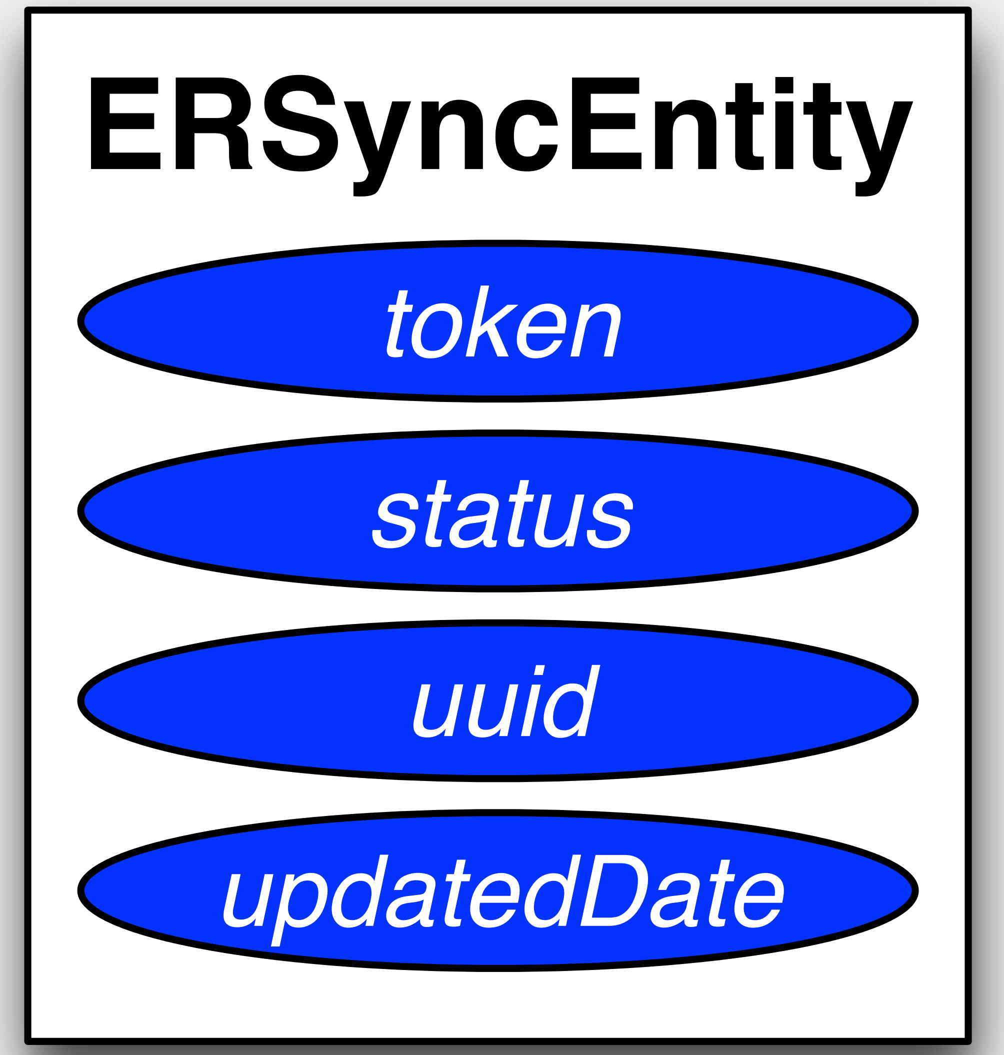
ERSync tracking



Status

- V - Virgin - never by sync'd
- I - Inserted
- U - Updated
- D - Deleted

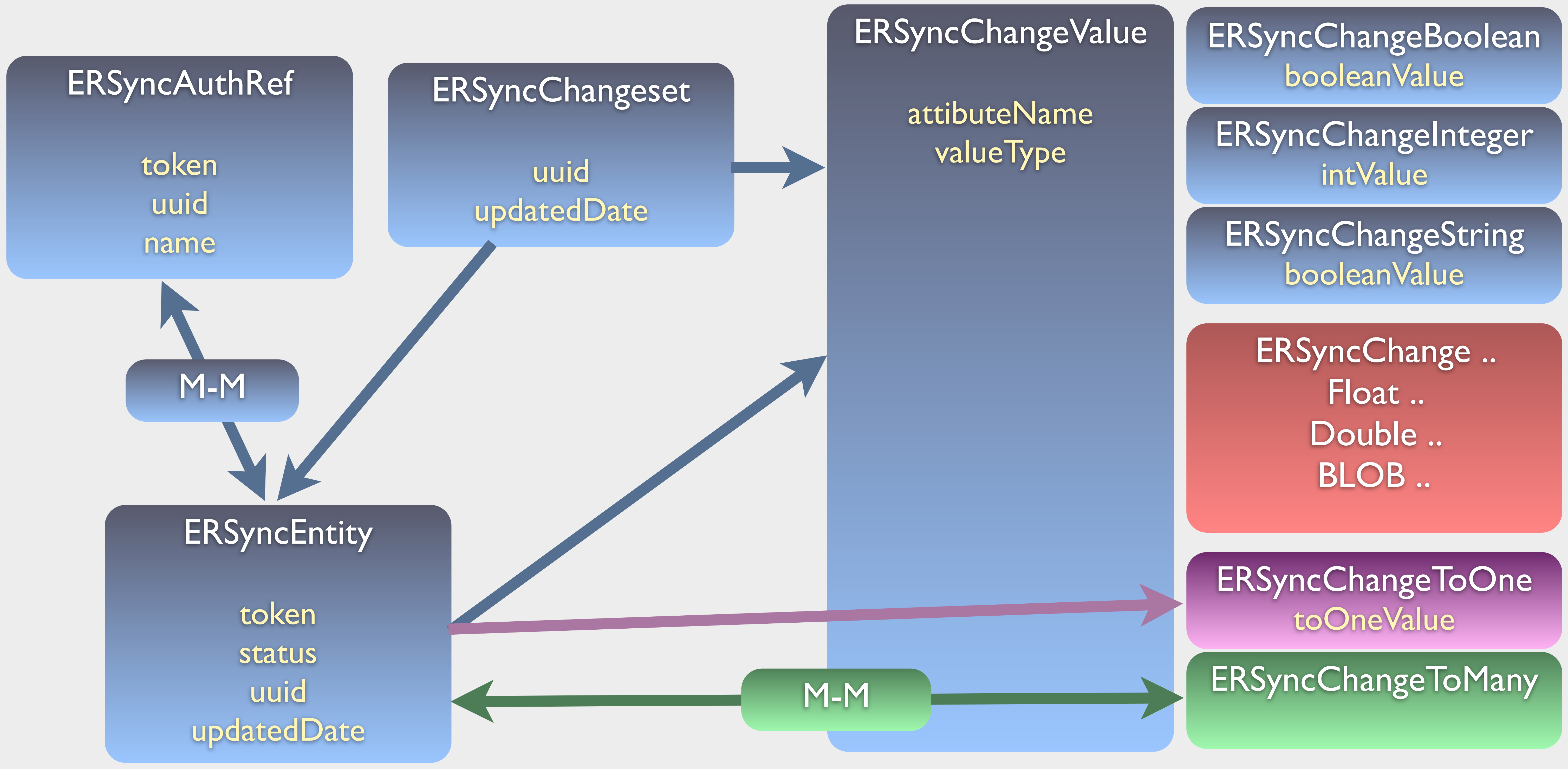
ERSync tracking



UUID

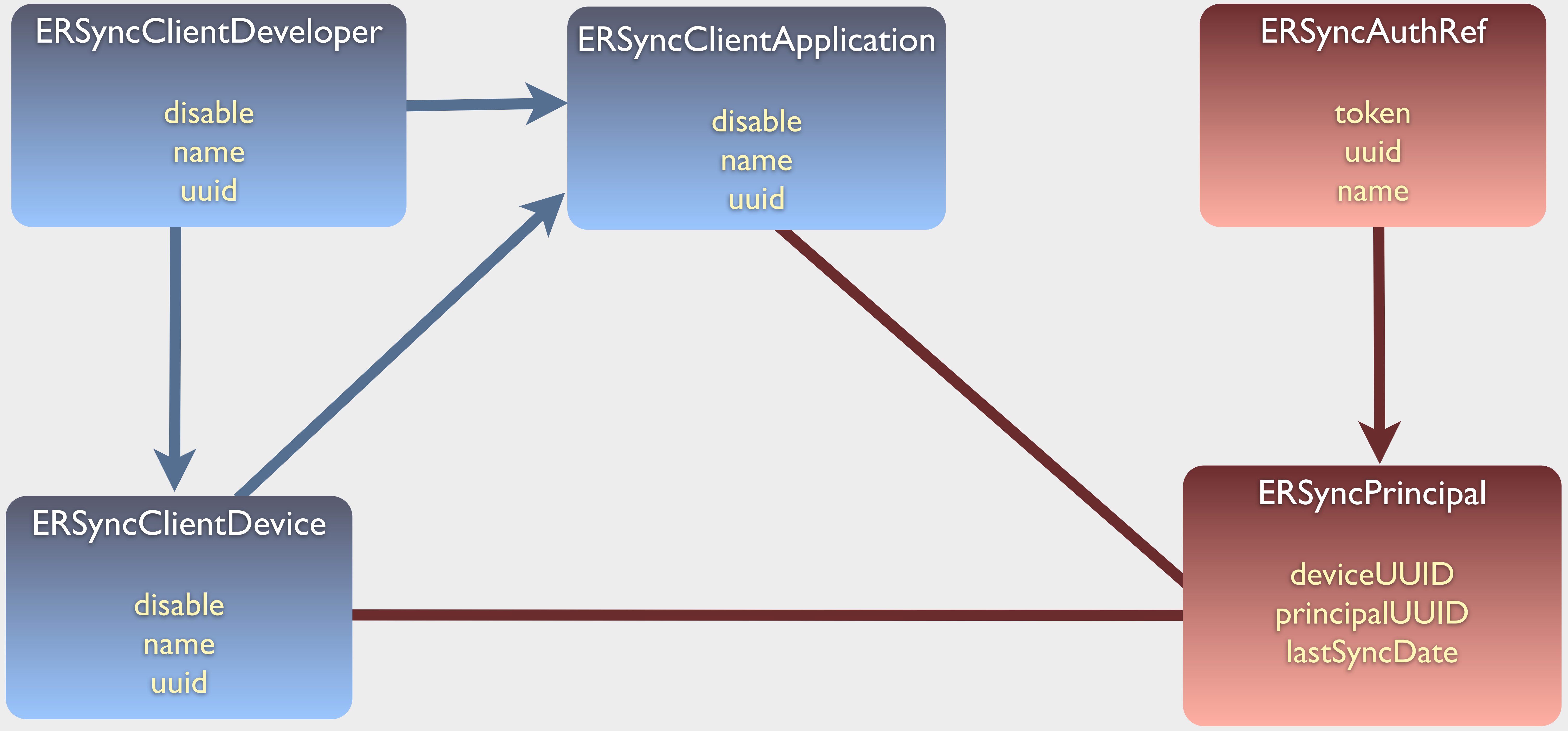
- The Database agnostic, universally unique id
- Clients will ALWAYS provide a UUID
  - usually the UUID assigned by the server
  - where the client inserts, it assigns the UUID and leaves the token blank
- Removes primary key distribution and collision problems

Change Notification Process



WOWODC ’012

ERSync API Security



Integrating with iOS

- Very similar to the WO approach
- Built on GVC Open frameworks

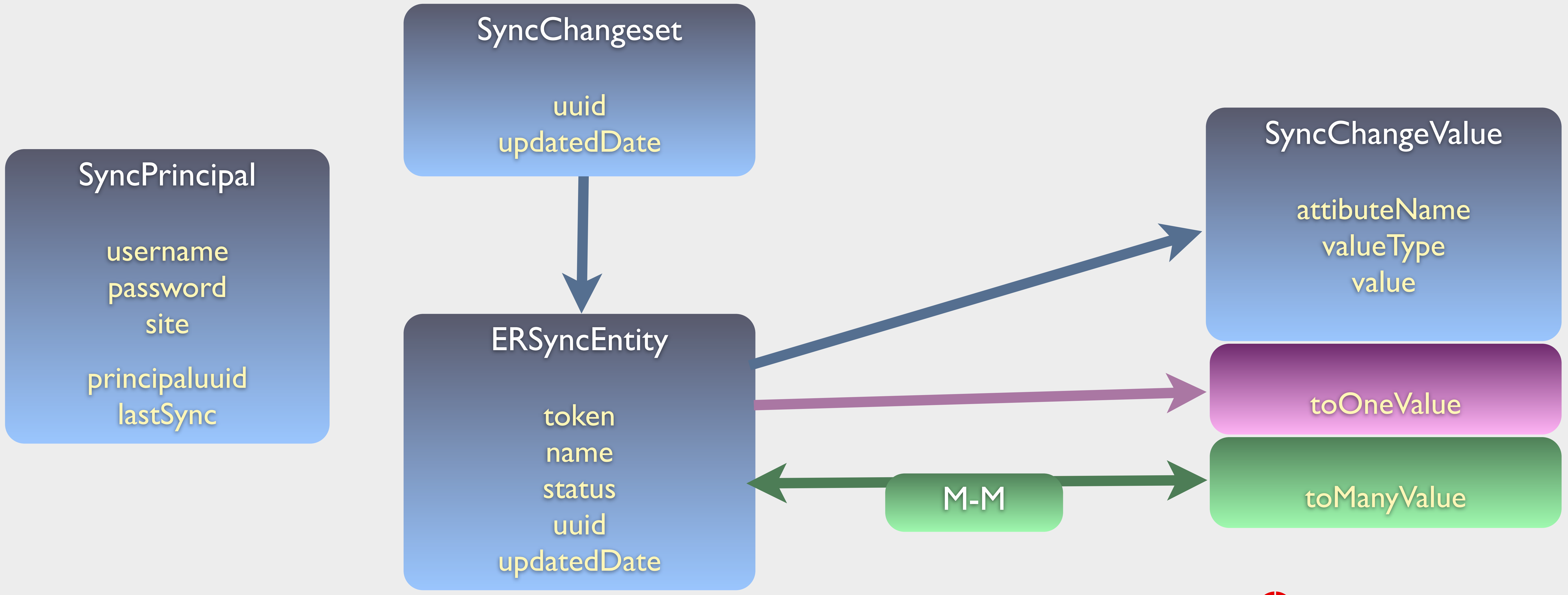
The Easy Part

```
    - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
   [super application:application didFinishLaunchingWithOptions:launchOptions];

   [self setEngine:[[SyncEngine alloc]
        initWithEditingContext:[self managedObjectContext]]];

   [[self engine] addSupportedEntity:[Note entityName]];
   [[self engine] addSupportedEntity:[Category entityName]];

   [self setPrincipal:[SyncPrincipal
        pseudoSingletonInContext:[self managedObjectContext]]];

    return YES;
}
```

What did that do?

- SyncEngine is the client version of ERXSyncHandler
    - adds change notification observer for main context
- SyncPrincipal
    - Core Data record to store configuration information.

iOS Model



WOWODC '012

Client Process

1. Registration
2. Initial / Full / Slow Sync
3. Delta / Fast Sync

Client Process - Registration

- Links the client application and device to a user on the server
- must provide server assigned
  - application UUID
  - device Type UUID
- provide client assigned and locally stored
  - device UUID
  - user credentials

```
<registration>
  <appid>
    5AC343C6-2C35-4BB0-9A00-CE2938A12260
  </appid>
  <deviceType>
    743E2D47-DDA4-4827-A164-0C61547CD4D5
  </deviceType>
  <deviceUUID>
    D9781163-2A97-4E90-B978-DE2B9F86A9D5
  </deviceUUID>
  <user>david</user>
  <password>tester</password>
</registration>
```

Client Process - Registration

- server response provides
  - principal UUID
  - last Sync date (only if previously sync'd)

```
<sync>
   <principalUUID>
     dce87db1-0e87-44b6-9680-19dcd672eadb
   </principalUUID>
   <lastSync type = "datetime">
     2012-06-30T10:25:10Z
   </lastSync>
</sync>
```

Client Process - Sync

- Client initiates communication
  - sends principal UUID and last Sync
  - data in the Insert / Update / Delete order

```
<sync>
  <principalUUID>
    dce87db1-0e87-44b6-9680-19dcd672eadb
  </principalUUID>
  <lastSync>2012-06-30T09:39:11Z</lastSync>

  <data>
    <Note id="EA3E9977-8B58-40FE-85CF-4E4027723DF8"
      status="update">
      <subject>My new subject</subject>
      <category>
        <Category id="Category:1000000" />
      </category>
...
```

Client Process - Sync

- Server response
  - echo principal UUID
  - provides new last Sync date
  - data in the Insert / Update / Delete order

```
<sync>
  <principalUUID>
    dce87db1-0e87-44b6-9680-19dcd672eadb
  </principalUUID>
  <lastSync>2012-06-30T09:55:11Z</lastSync>

  <data>
    <Reminder id="EA3E9977-8B58-40FE-85CF-4E4027723DF8"
      status="update">
      <name>Get Siri off my back</name>
      <type>
        <ReminderType id="ReminderType:active"/>
      </type>
...
```

We are Wonder-ful

Design Objectives

- ✓ Leverage our WO experience, products and services.
- ✓ Minimally impact our existing code base.
  - ✓ No server database changes in existing business system.
  - ✓ No sprinkling of interfaces or special logic in current system
- ✓ Simple client protocol and supporting library

Development Plan - WO

- Add support for Additional Change Value Types
  - float, double, char .. scalar types
  - BLOB, CLOB, LOB .. SLOBs

Development Plan - WO

- Add support for Additional Change Value Types

- Track the SyncEntity status *per* ***Principal***
  - a principal represents a client/device combo
  - entity status can be different on each device

Development Plan - WO

- Add support for Additional Change Value Types
- Track the Entity status *per* ***Principal***

- Purge Changesets once all ***Principals*** have sync'd
  - if all the registered client/devices have the change then it is not needed
  - new registrations will be 'Virgin' and require the *current* record anyway. (ERSync is not a History engine)

Development Plan - WO

- Add support for Additional Change Value Types
- Track the Entity status *per* ***Principal***
- Purge Changesets once all ***Principals*** have sync'd

- Make SyncEntity a composite of several EOEntity types
  - Related but De-normalized data
  - virtual entity for non-relational data (images/thumbnails)

Development Plan - WO

- Add support for Additional Change Value Types
- Track the Entity status *per* ***Principal***
- Purge Changesets once all ***Principals*** have sync'd
- Make SyncEntity a composite of several EOEntity types

- Allow the client to sync a ***subset*** of data
  - date range (example bank transactions in 30 day window)
  - data group (example Toronto address book as a window into a larger address book)

Development Plan - iOS

- Clean up the iOS ERSync framework
  - ARC and Block compatible
  - ERBranding
  - Sync related UI components?

Development Plan - iOS

- Clean up the iOS ERSync framework

- ERSyncEngine
  - Does not need to process anything until registration
  - Manage operations automatically
  - detect errors/resets from server and perform full sync cycle

Development Plan - iOS

- Clean up the iOS ERSync framework
- ERSyncEngine

- Multiple CoreData model migration

1. merged models cannot be migrated automatically
2. Individual Stores cannot be migrated automatically

WOWODC '012

MONTREAL JUNE 30, JULY 1ST AND 2ND 2012

Q&A

Web Resources

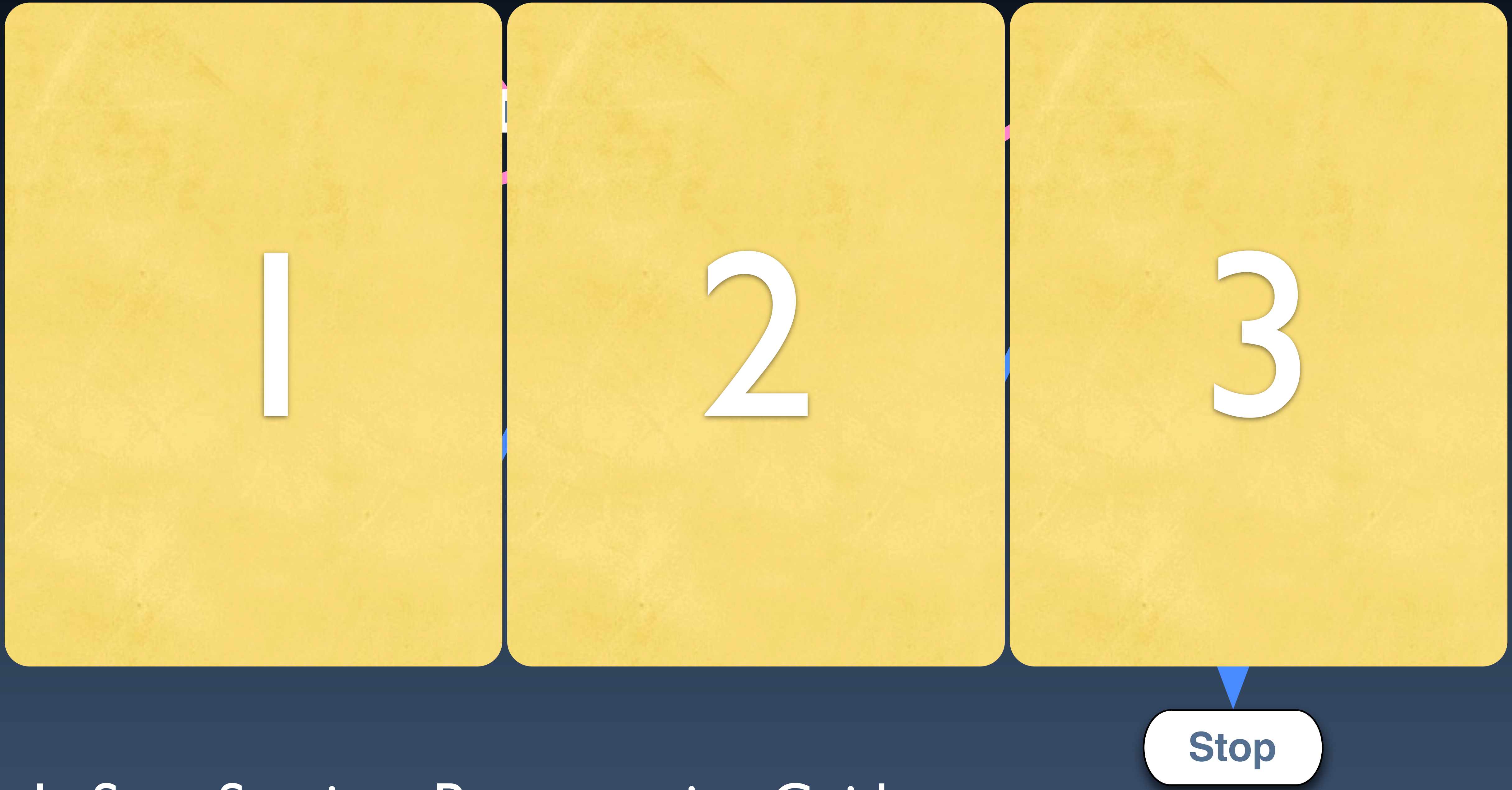
- Source is currently available at:

  https://github.com/davidAtGVC/RemoteSync

- The iOS projects have submodule references for the GVC Open kits:

  https://github.com/davidAtGVC/GVCFoundation
  https://github.com/davidAtGVC/GVCUIKit
  https://github.com/davidAtGVC/GVCCoreData

The sync process overview



Source: Apple Sync Services Programming Guide

Why is this important

- Data != Files
- Pass the TTC test