



eSpark

Personalized Learning on the iPad

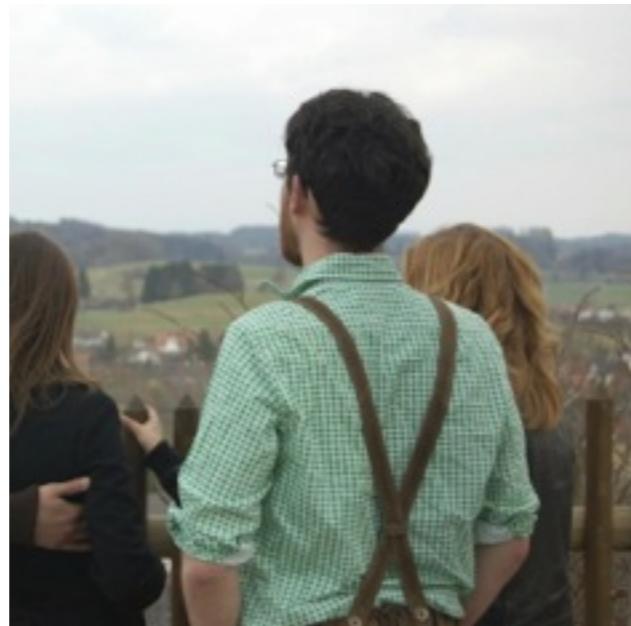
Deploying iPads at Scale

© eSpark Learning

Confidential – not for distribution without permission

This document is protected under the copyright laws of the United States and other countries as an unpublished work. This document contains information that is proprietary and confidential of eSpark, Inc. which shall not be disclosed outside the recipient's organization or duplicated, used or disclosed in whole or in part by the

Alex Koppel (@arsduo)



- Joined eSpark in 2013
- Ruby/iOS/JS since 2008
- Previously: Wunderlist, social marketing, healthcare IT

eSpark

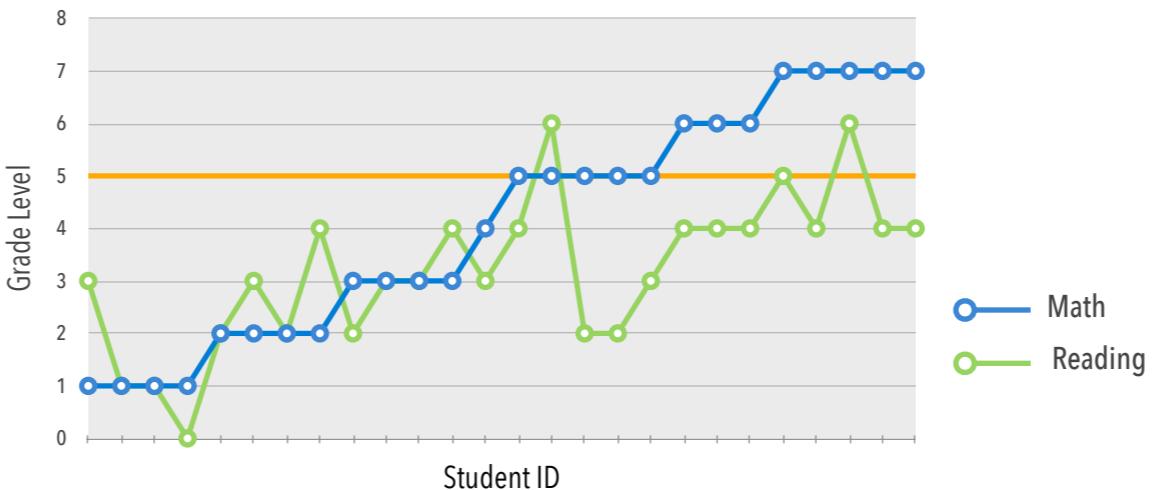


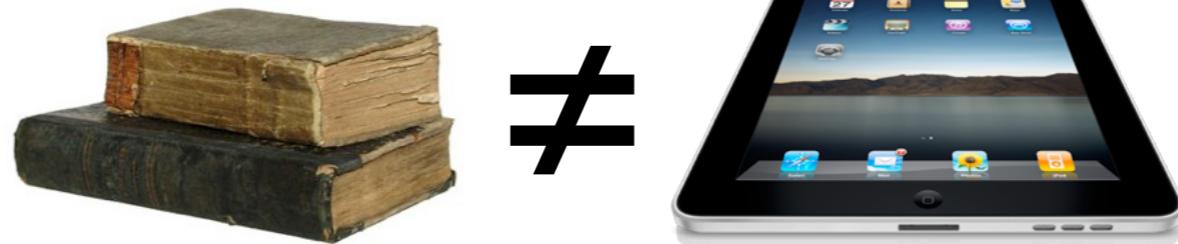
<http://flic.kr/p/6ryQsW>

Differentiated Learning Needs

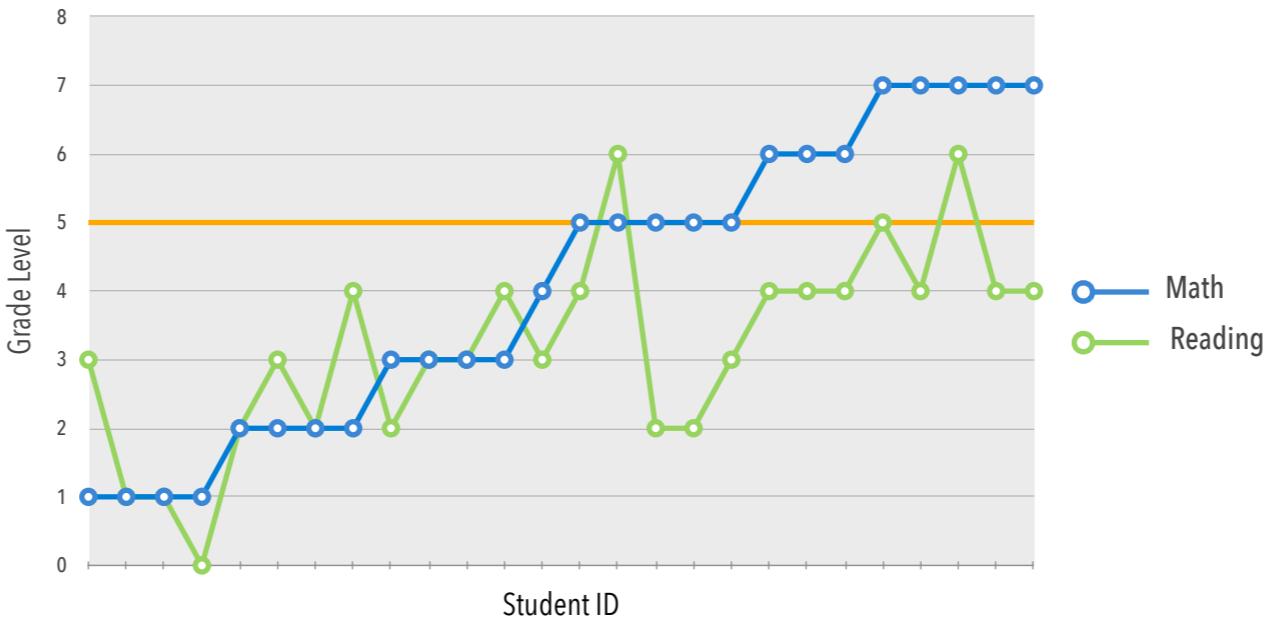
This graph displays the diverse learning needs of a real 5th grade classroom in Chicago. Only 6 students are working at grade level.

This is a common problem all teachers face.

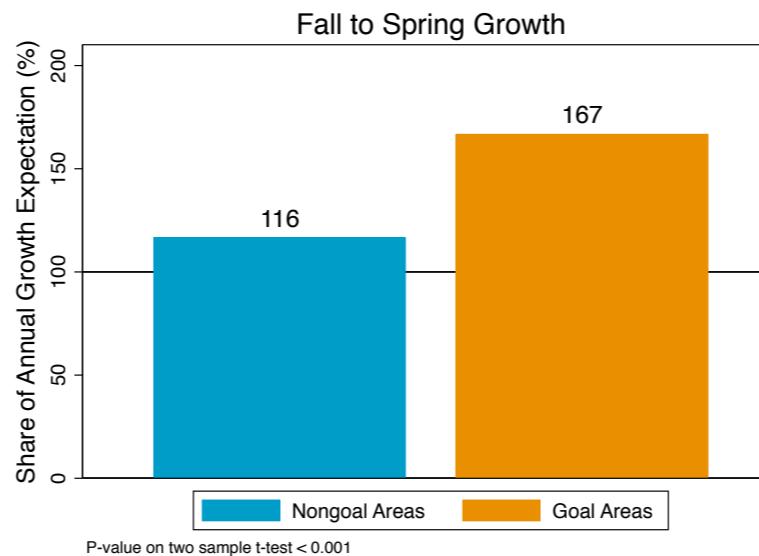




This can be an opportunity



Effective

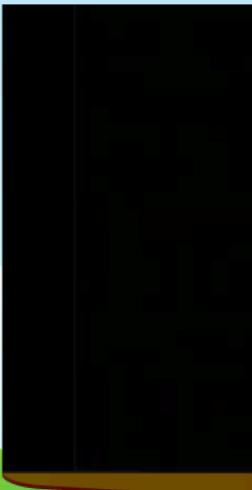




Personalized education for K-8
Curriculum based on 3rd-party educational content
Delivered by iPads

LOG
OUT

1. _____
2. _____
3. _____
4. _____



LOG
OUT







100,000+
educational
resources for iPad

Personalized education for K-8
Curriculum based on 3rd-party educational content
Delivered by iPads



**Mr. Babai's
4th grade
Reading**



Matt



Mr. Babai's
4th grade
Reading

Sara





Matt



2nd grade level

Mr. Babai's
4th grade
Reading

Sara



6th grade level



Matt



2nd grade level

Mr. Babai's
4th grade
Reading

Sara



6th grade level



Conjunctions



Oliver Twist

How do we make it
happen?

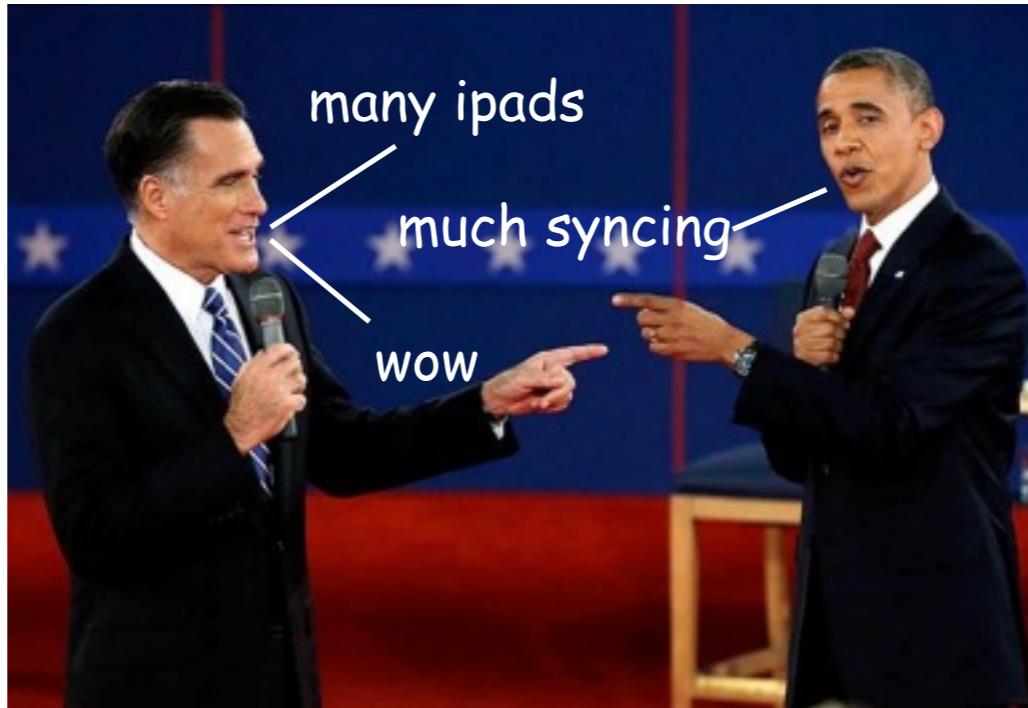


Fall 2012: 2,000 iPads



© eSpark Learning

Fall 2012: 2,000 iPads

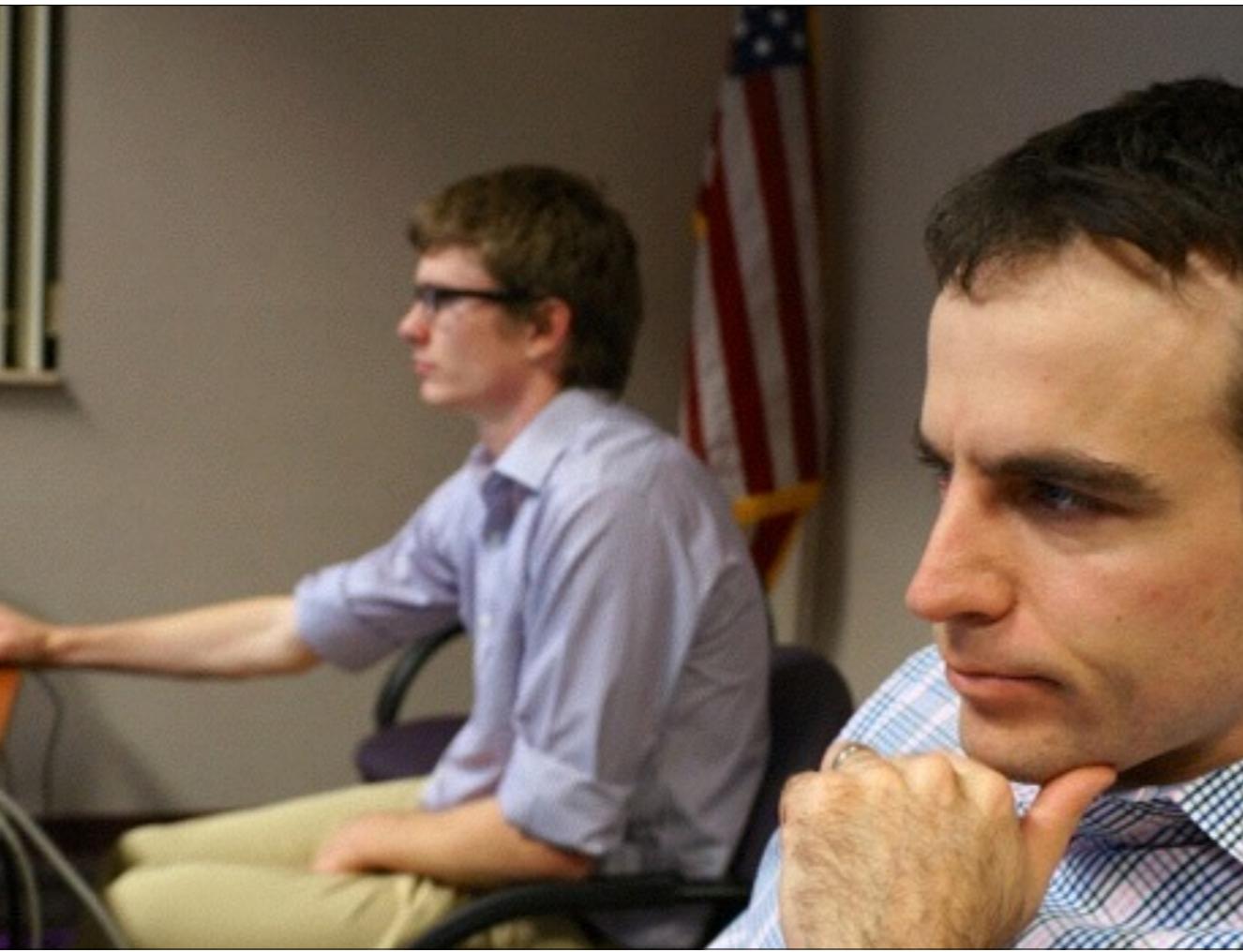












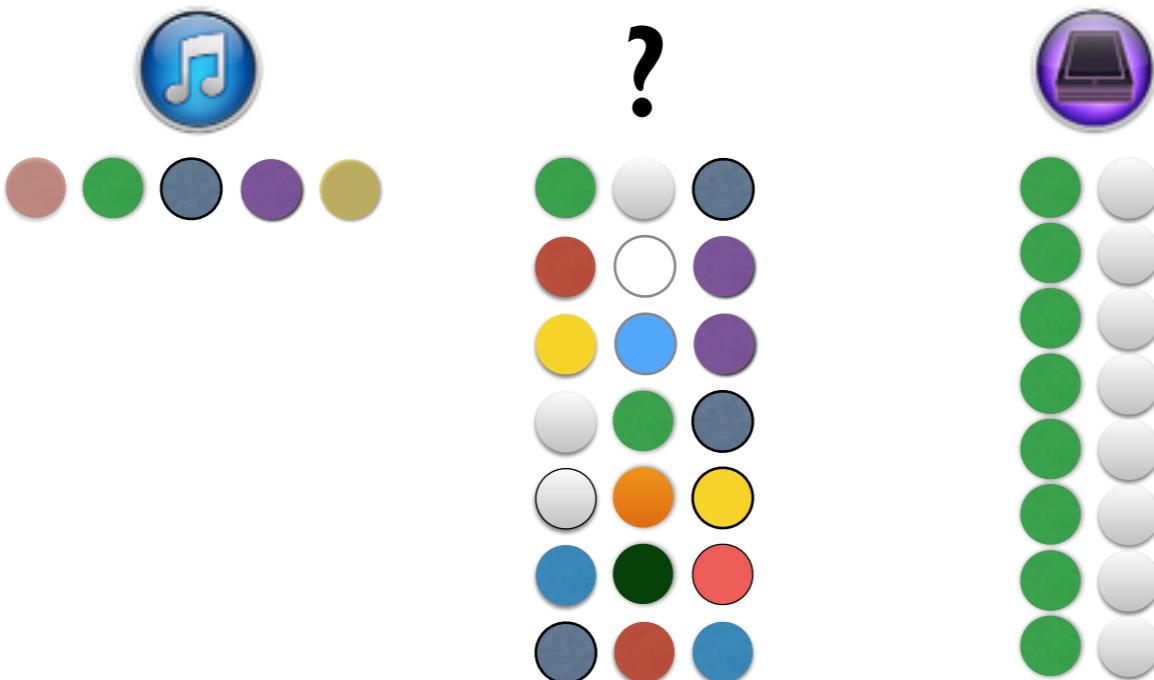
The Conundrum



© eSpark Learning

iTunes very manual and error-prone
Configurator very rigid
Ugh

The Conundrum



© eSpark Learning

iTunes very manual and error-prone
Configurator very rigid
Ugh

Fall 2013: 10,000 iPads

iOS 7

The mobile OS from a whole new perspective.

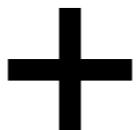


© eSpark Learning

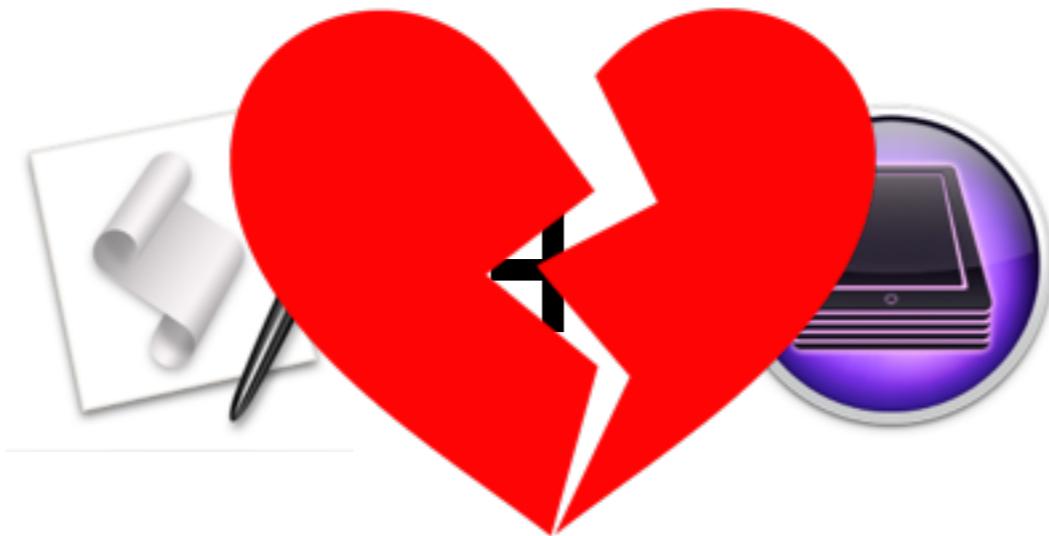


<http://www.plasticpals.com/?p=17235>

First try: use what we had

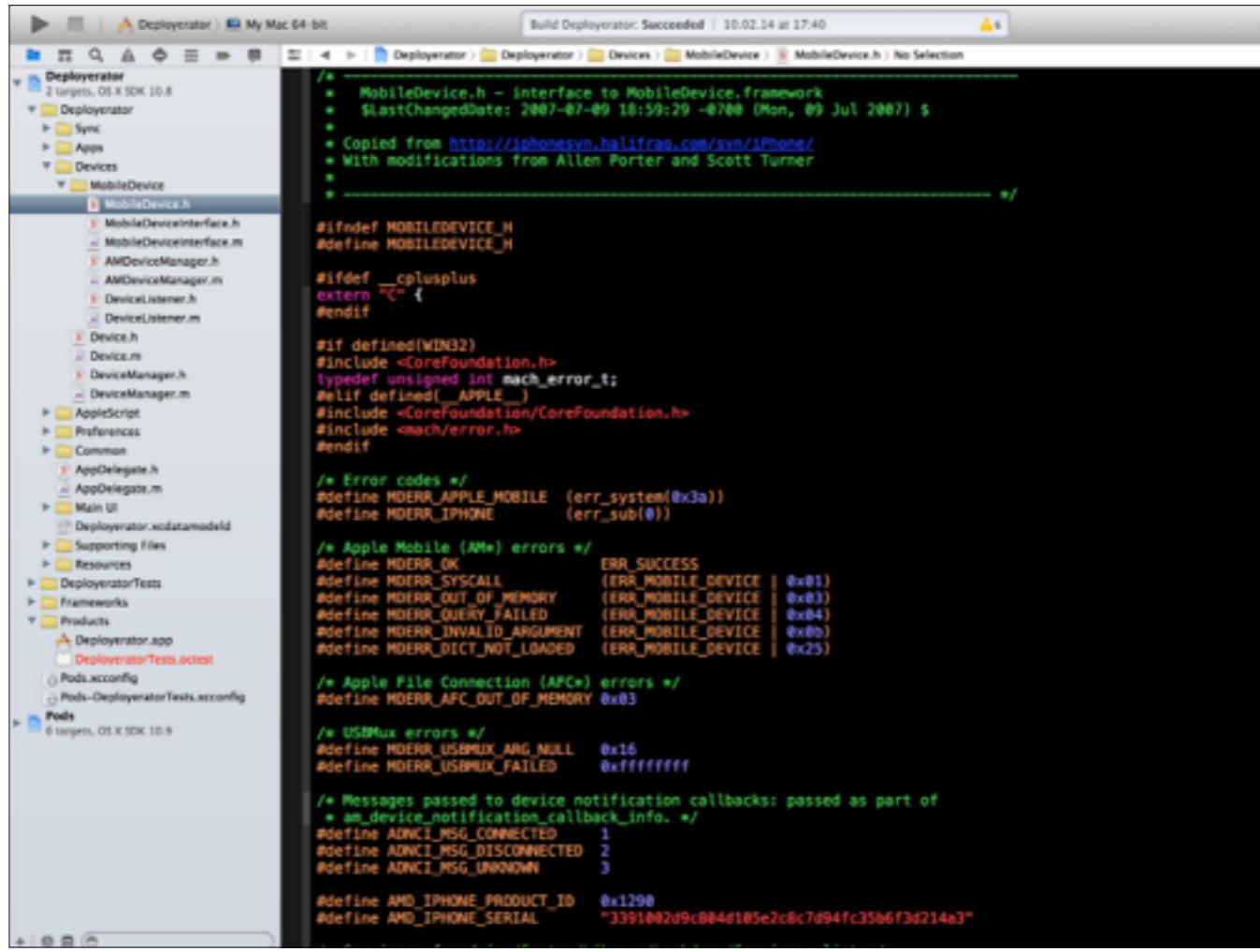


UI scripting: about what you'd expect



© eSpark Learning

http://en.wikipedia.org/wiki/File:Broken_heart.svg



The screenshot shows the Xcode interface with the Deployer project open. The left sidebar displays the project structure, including targets for OS X SDK 10.8, the Deployer target, and the MobileDevice target. The Deployer target includes subfolders like Sync, Apps, Devices, and MobileDevice, which contains the MobileDevice.h file. The right pane shows the code for MobileDevice.h. The code is heavily commented, indicating it was last changed on July 8, 2007. It includes imports for CoreFoundation and mach/error.h, defines error codes for Apple Mobile (AM) and Apple File Connection (AFC), and defines product IDs and serial numbers for the iPhone.

```
/*
 * MobileDevice.h - Interface to MobileDevice.framework
 * $LastChangedDate: 2007-07-09 18:59:29 -0700 (Mon, 09 Jul 2007) $
 *
 * Copied from https://iphonesyn.halifran.com/svn/iPhone/
 * With modifications from Allen Porter and Scott Turner
 */
#ifndef MOBILEDEVICE_H
#define MOBILEDEVICE_H

#ifdef __cplusplus
extern "C" {
#endif

#if defined(WIN32)
#include <CoreFoundation.h>
typedef unsigned int mach_error_t;
#elif defined(APPLE__)
#include <CoreFoundation/CoreFoundation.h>
#include <mach/error.h>
#endif

/* Error codes */
#define MDERR_APPLE_MOBILE (err_system(0x3a))
#define MDERR_IPHONE (err_sub(0))

/* Apple Mobile (AM) errors */
#define MDERR_OK ERR_SUCCESS
#define MDERR_SYSCALL (ERR_MOBILE_DEVICE | 0x01)
#define MDERR_OUT_OF_MEMORY (ERR_MOBILE_DEVICE | 0x03)
#define MDERR_QUERY_FAILED (ERR_MOBILE_DEVICE | 0x04)
#define MDERR_INVALID_ARGUMENT (ERR_MOBILE_DEVICE | 0x05)
#define MDERR_DICT_NOT_LOADED (ERR_MOBILE_DEVICE | 0x25)

/* Apple File Connection (AFC*) errors */
#define MDERR_AFC_OUT_OF_MEMORY 0x03

/* USBMux errors */
#define MDERR_USBMUX_ARG_NULL 0x16
#define MDERR_USBMUX_FAILED 0xffffffff

/* Messages passed to device notification callbacks: passed as part of
 * am_device_notification_callback_info. */
#define ADNCI_MSG_CONNECTED 1
#define ADNCI_MSG_DISCONNECTED 2
#define ADNCI_MSG_UNKNOWN 3

#define AMD_IPHONE_PRODUCT_ID 0x1290
#define AMD_IPHONE_SERIAL "3391002d9c004d105e2c8c7d94fc3506f3d214a3"
```

What's interesting is that the version we're using was last updated in 2007: this is a _stable_ API.

http://theiphonewiki.com/wiki/MobileDevice_Library

Deployerator

Sync Apps Log

12 connected devices

eSpark ID	Name	Installed	Assigned	Current App	All Apps	Current App
8429990A878F	K-westuti...	94	200	Lumi's Eyes		
8429990121F2	K-westuti...	95	200	A day with...		
842999023F11	K-05-7200	79	194	Math Add		
8429990A77FE	K-westuti...	61	200	MoreFun		
84299908EB0E	K-05-7194	82	194	Find Vowel		
8429990B705D	K-westuti...	90	200	MonsterTo...		
8429990B9A02	K-westuti...	96	200	Opposites		
84299904CD3D	K-westuti...	48	200	Letter Mun...		
842999080267	K-05-7188	72	194	1+1		
842999109008	K-05-7191	82	194	Find Vowel		
8429990F7862	K-05-7183	81	194	Mathtopia		
8429990F4868	K-05-7190	72	194	1+1		

App data loading finished.

Sync Apps

Automated Deployment



© eSpark Learning

Automated Deployment



Automated Deployment

```
JSON API  
GET /api/devices  
{"device_ids": ...}
```



Automated Deployment



© eSpark Learning

Automated Deployment



© eSpark Learning





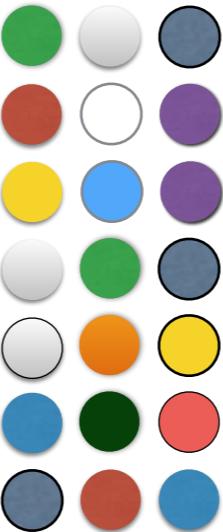


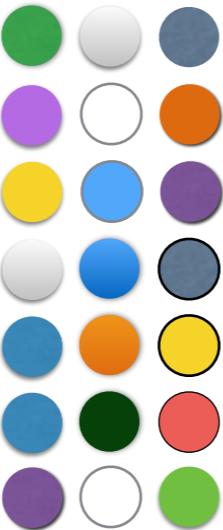
Installs the right apps every time
Each computer can sync 20 iPads at once
No setup time beyond plugging the iPad in



Not adaptive







Fall 2014: 60,000 iPads?



© eSpark Learning

© zir.com, <http://en.wikipedia.org/wiki/File:TARDIS1.jpg>



Apple's new tools

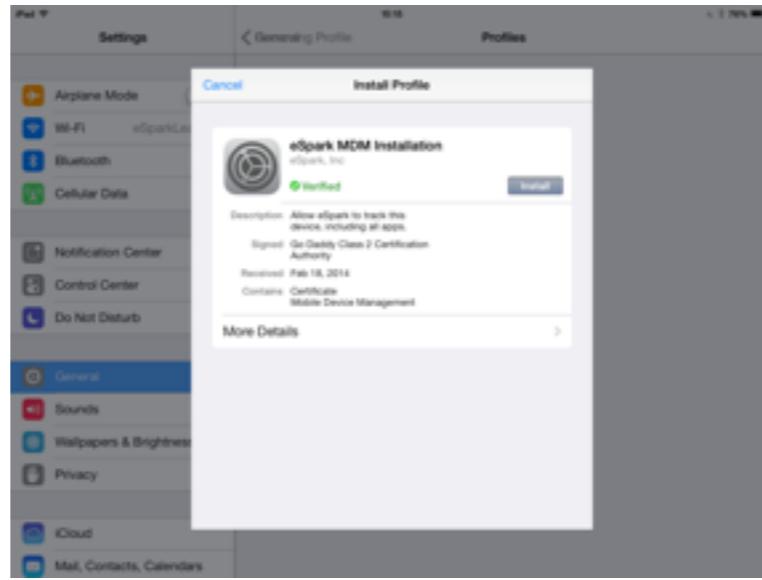
Volume Purchase Program for Education

The Volume Purchase Program (VPP) makes it simple for institutions of any size to find, buy, and distribute apps and books to enable endless learning possibilities.

[Enroll now >](#) | [Registered users login >](#) | [Download the VPP guide >](#)



Demo!



“Failures do, of course, occur; I really hope none will happen today, but you have to prepare for them anyway.” - Franz Kafka on live demos



MDM menage-a-trois

MDM menage-a-trois



© eSpark Learning

<https://xkcd.com/869/>

MDM menage-a-trois



Hi!
I'M A SERVER!
WHO ARE YOU?



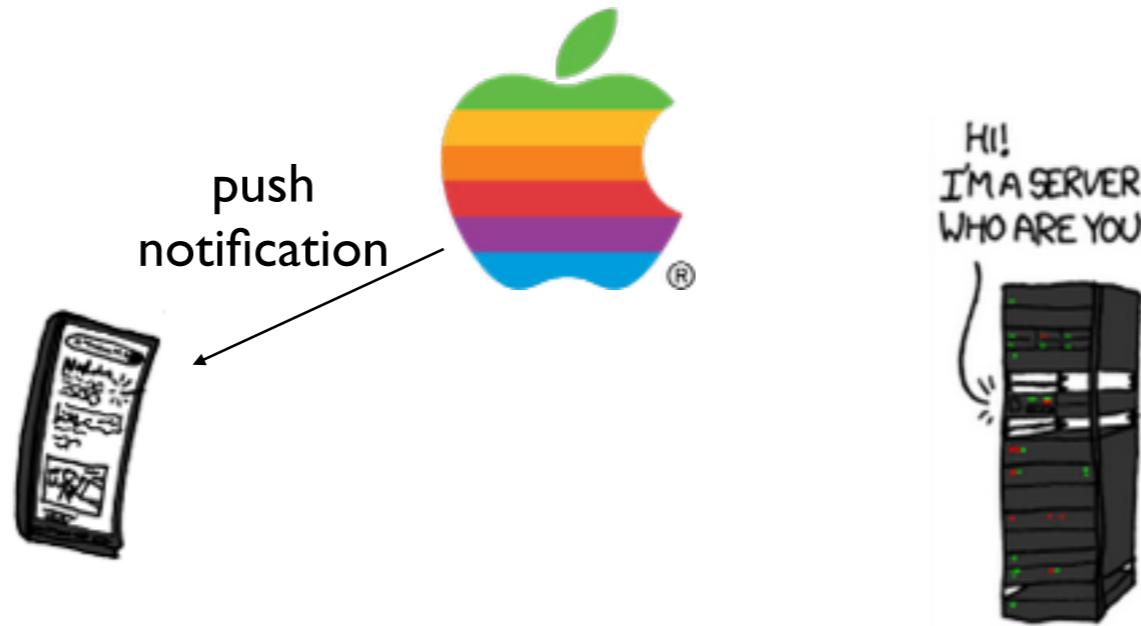
MDM menage-a-trois



MDM menage-a-trois



MDM menage-a-trois



© eSpark Learning

<https://xkcd.com/869/>

MDM menage-a-trois



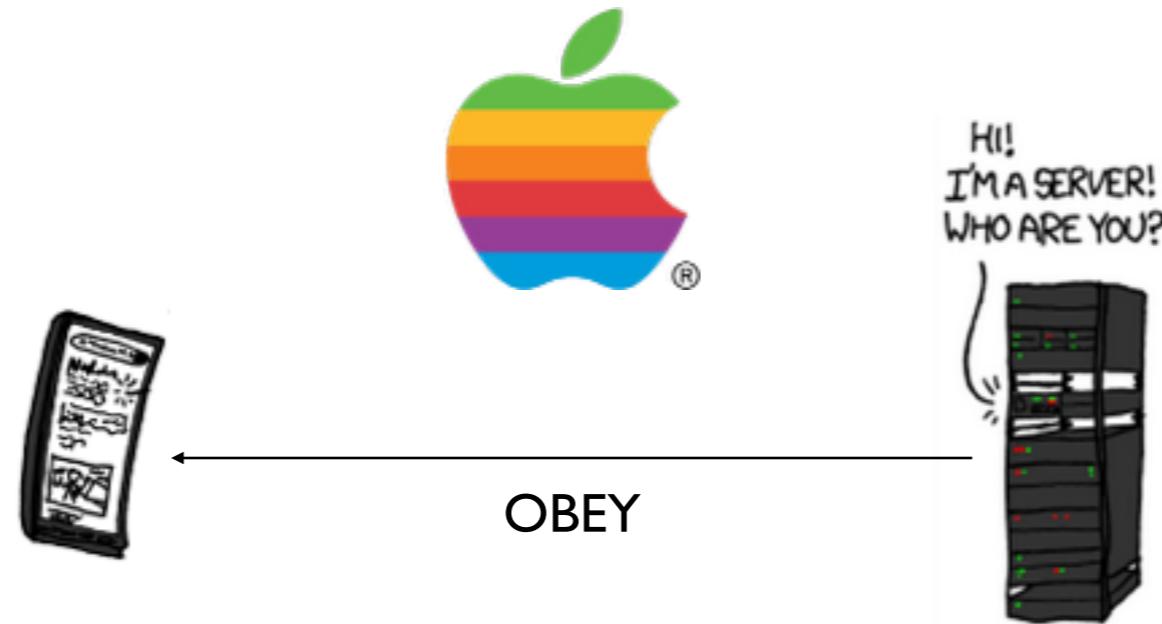
MDM menage-a-trois



MDM menage-a-trois



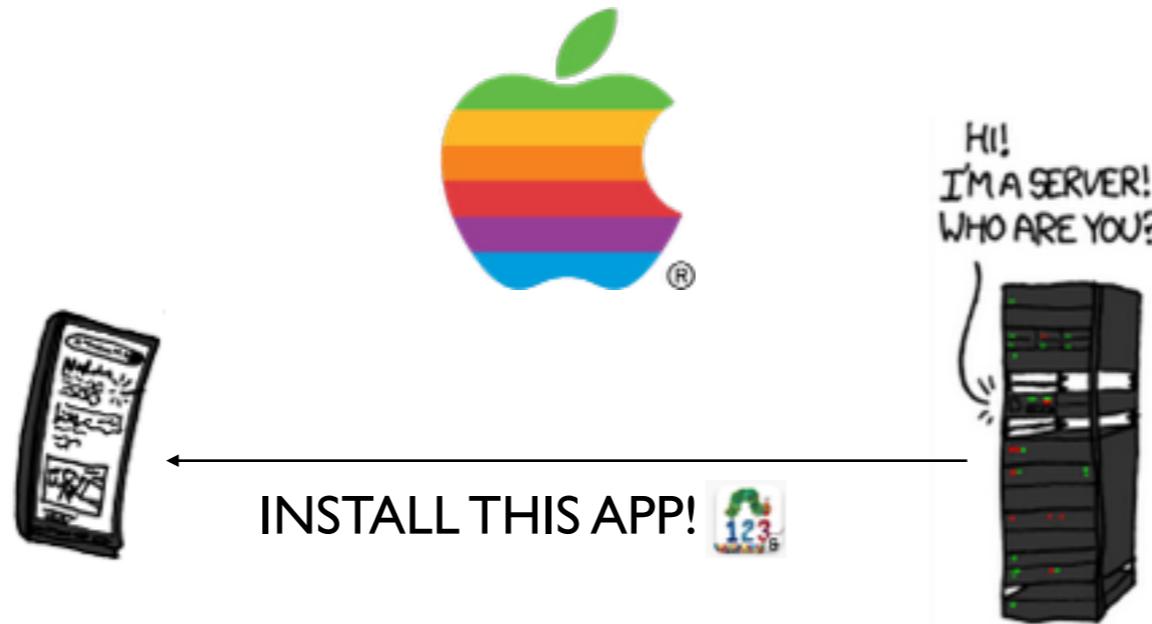
MDM menage-a-trois



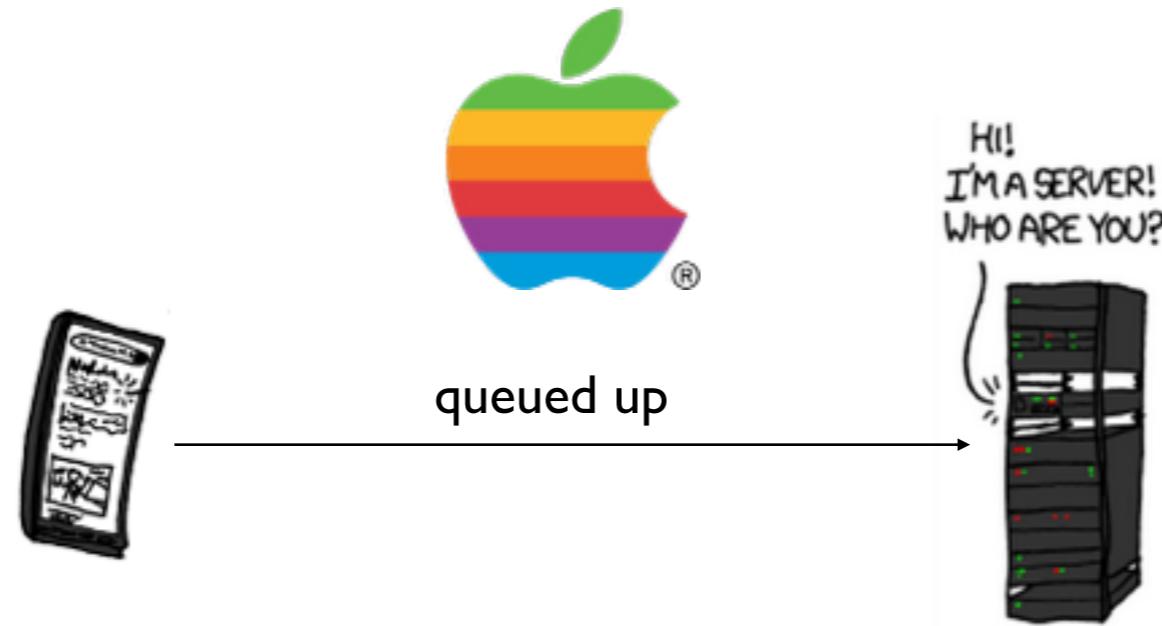
MDM menage-a-trois



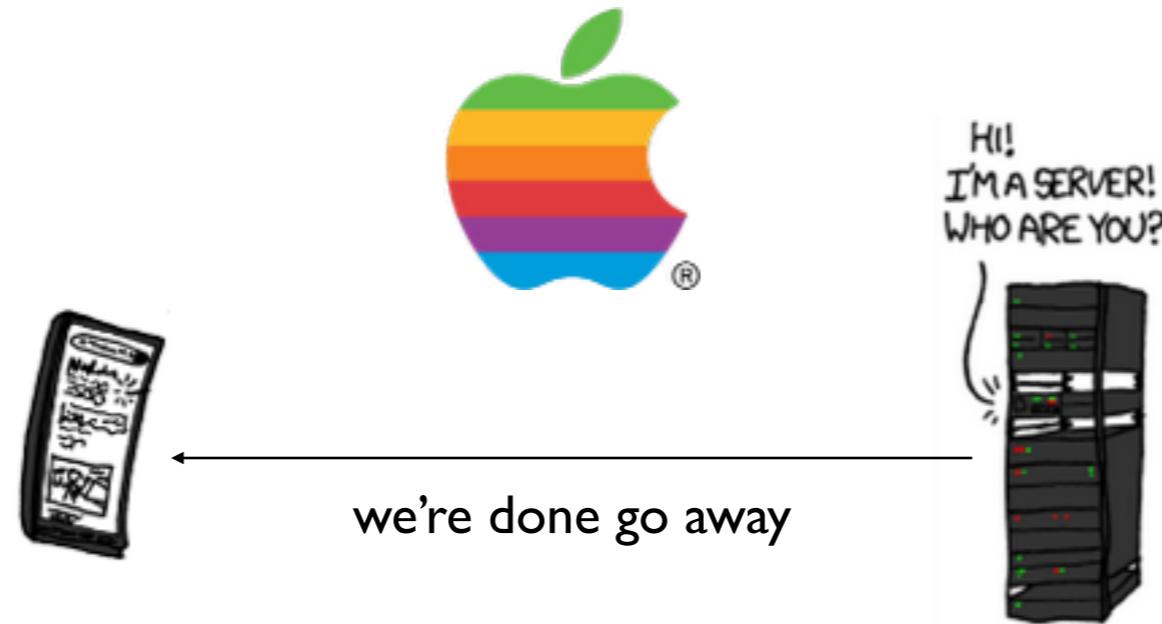
MDM menage-a-trois



MDM menage-a-trois



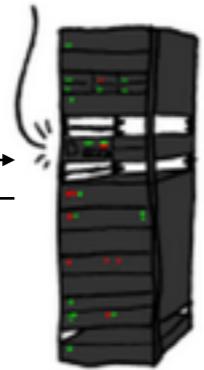
MDM menage-a-trois



MDM menage-a-trois



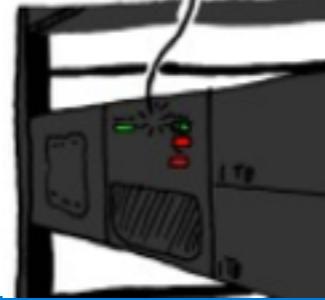
Hi!
I'M A SERVER!
WHO ARE YOU?

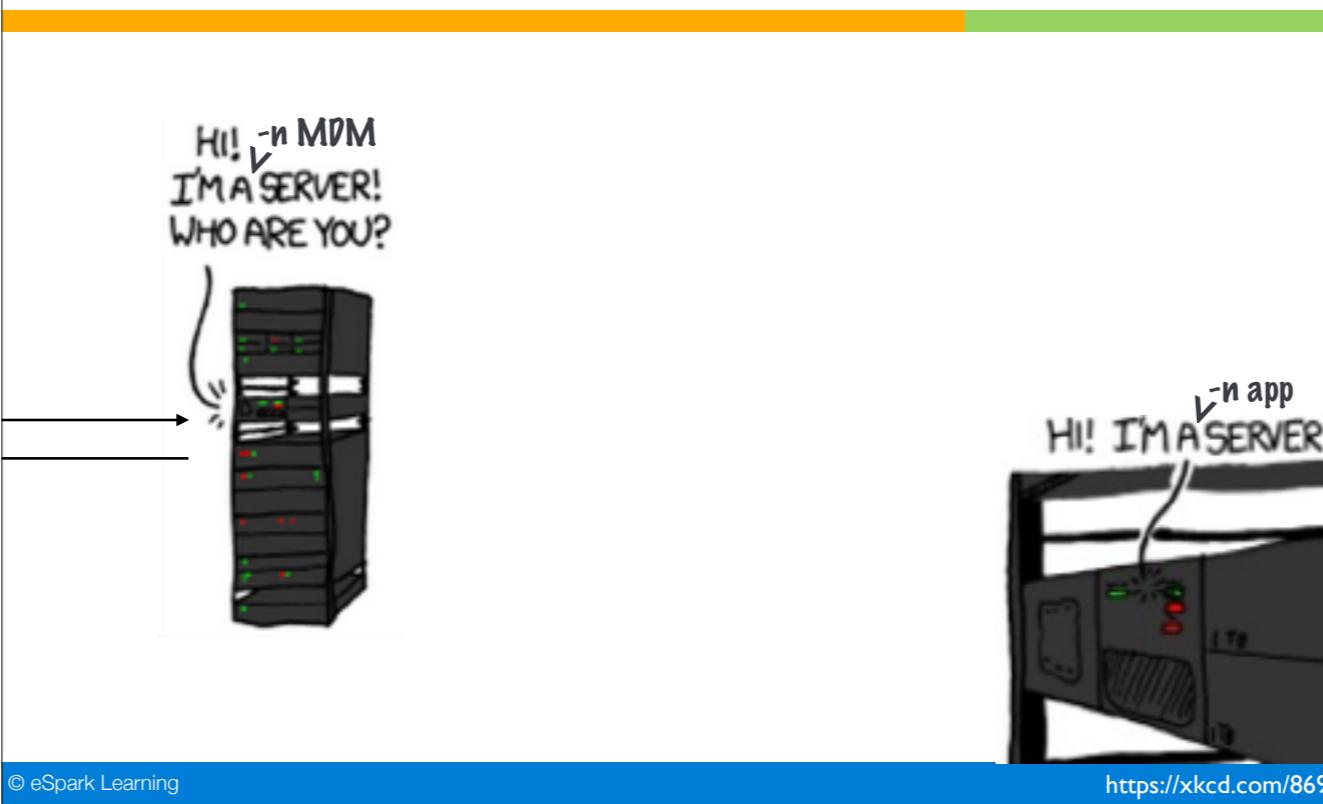


Hi!
I'M A SERVER!
WHO ARE YOU?



Hi! I'M A SERVER!



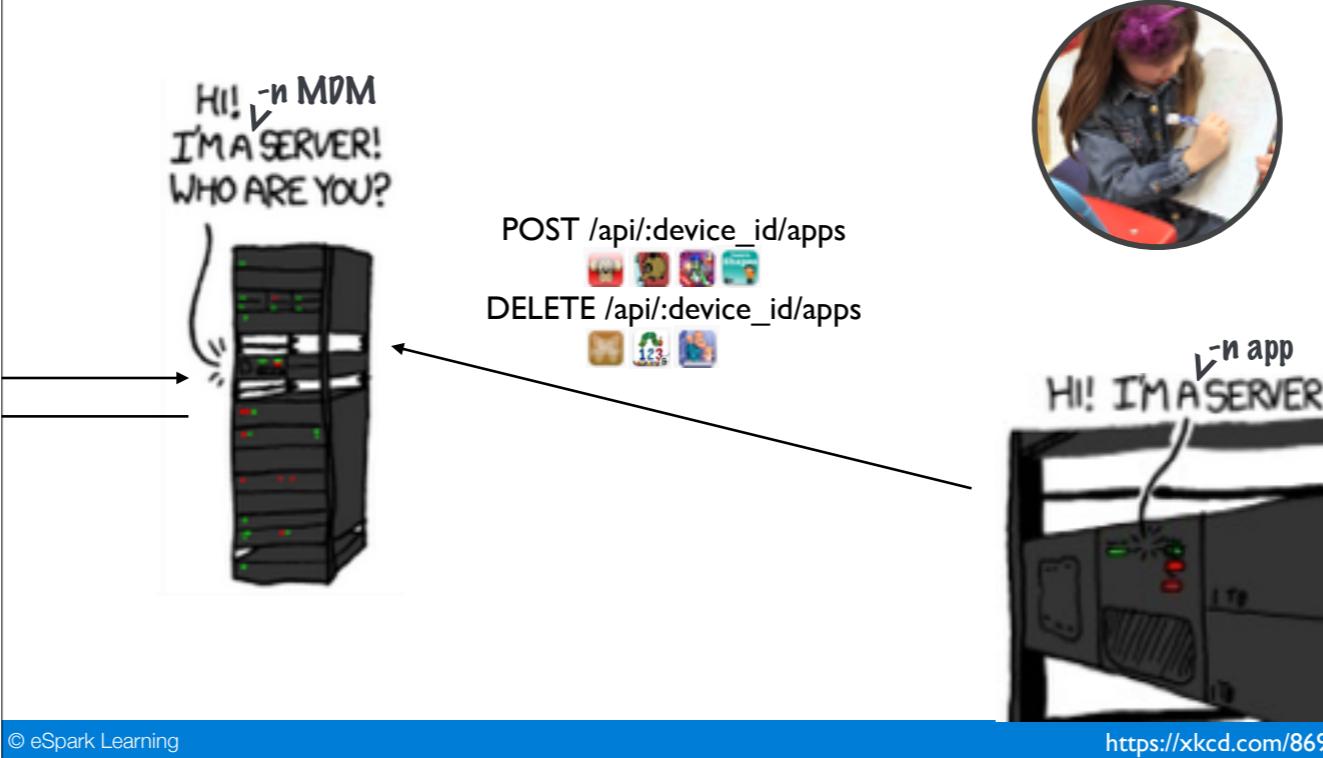


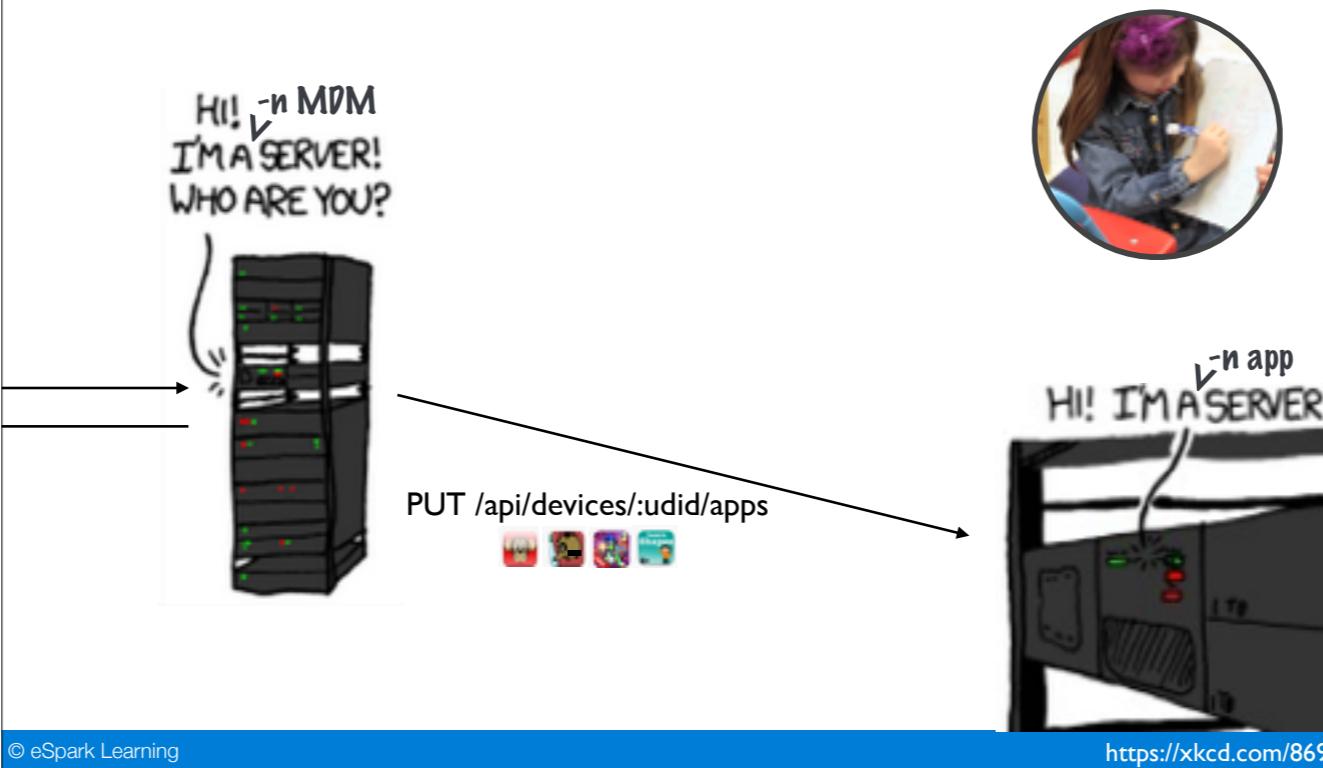


© eSpark Learning

<https://xkcd.com/869/>



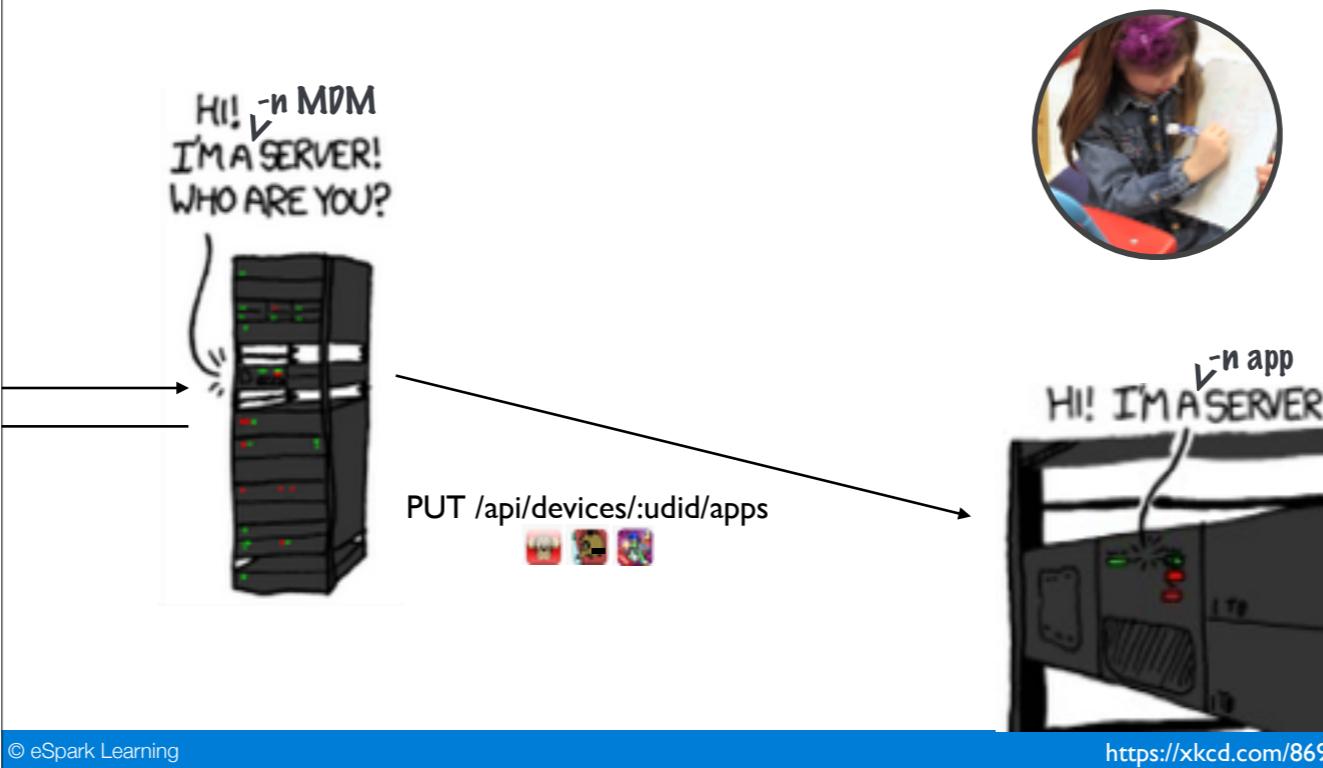


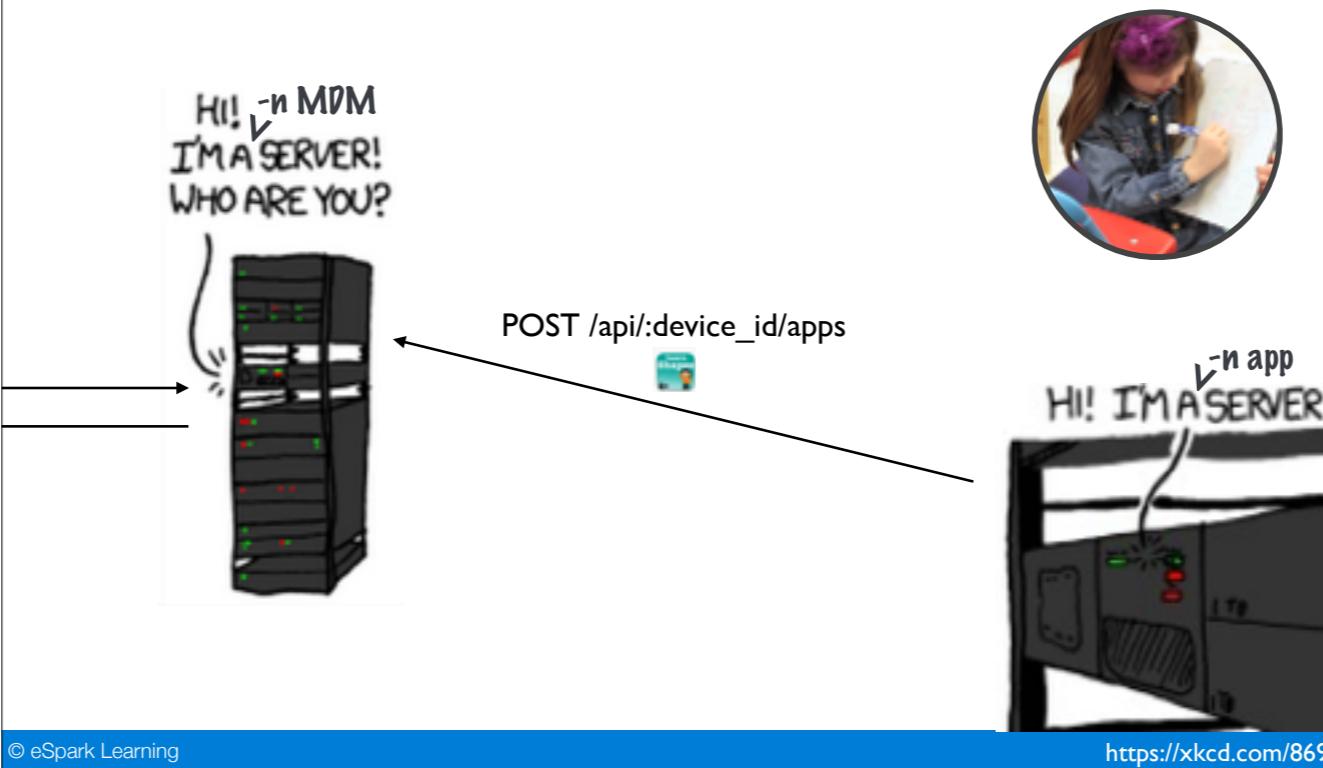




© eSpark Learning

<https://xkcd.com/869/>



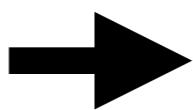


What have we learned?



≠



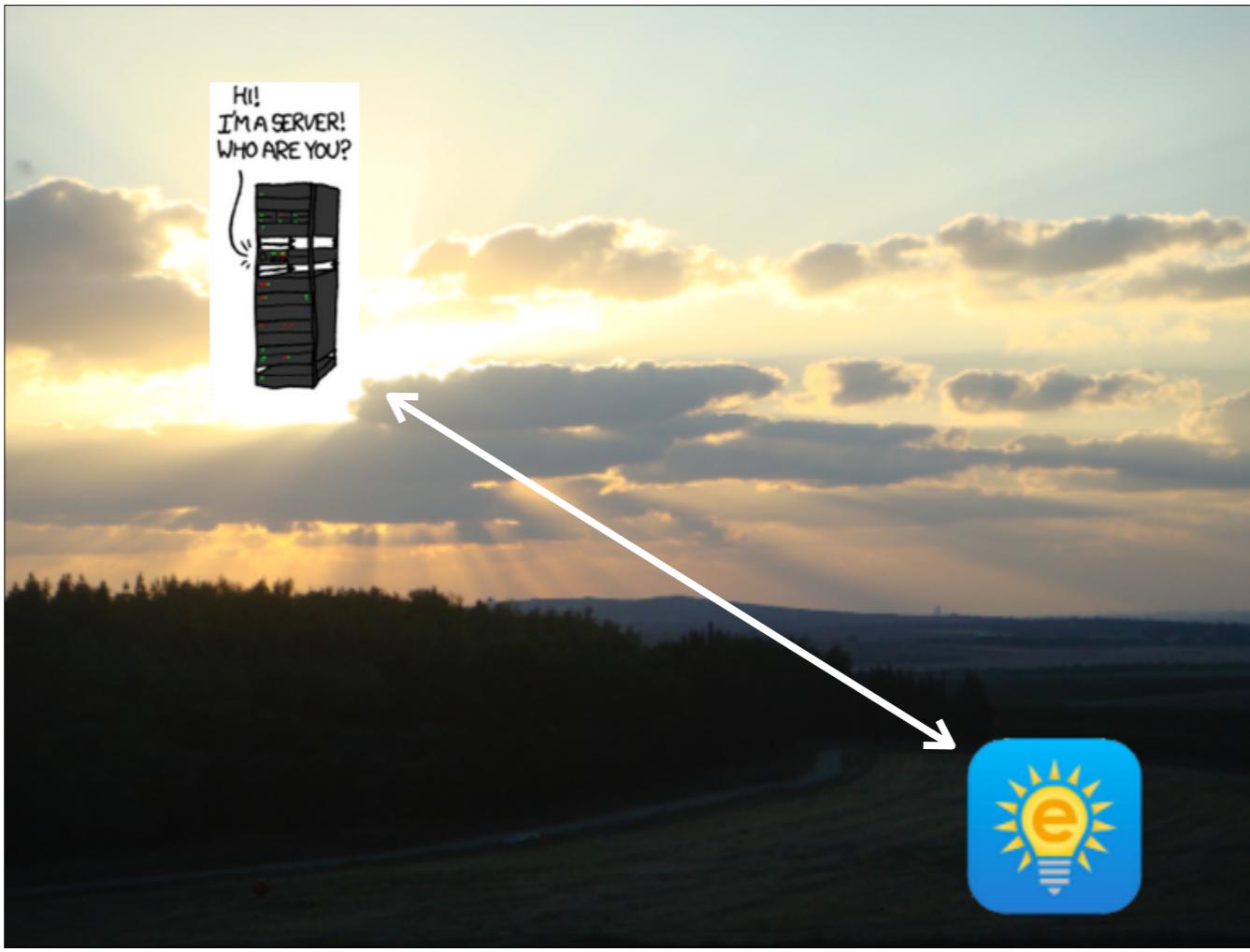




As an app developer, we're used to the server being the main source of truth. The app may have a different state from the server, but until it syncs it back, that's not actually recorded or "real". Devices could die at any time, only the server is forever.



As an app developer, we're used to the server being the main source of truth. The app may have a different state from the server, but until it syncs it back, that's not actually recorded or "real". Devices could die at any time, only the server is forever.



As an app developer, we're used to the server being the main source of truth. The app may have a different state from the server, but until it syncs it back, that's not actually recorded or "real". Devices could die at any time, only the server is forever.



Mention that the MDM does talk to Apple

When you're managing devices, though, the device is the ultimate source of truth. When a 3rd grader turns it on to read about dinosaurs, it's the device that determines what they see. It doesn't matter what your server thinks the state of the device is, whether it "ought to be" correct based on what you're seeing.

Of course, the picture is really complicated, because it's not just the MDM server in the picture. There's also:

- * our application, the one that knows about students. It talks to the device through the MDM...
- * ...but also interacts with devices via our app, which students open and run on a daily basis.
- * Apple's also in the mix — in order to install apps from the iTunes store we have to allocate iTunes licenses through Apple.



Mention that the MDM does talk to Apple

When you're managing devices, though, the device is the ultimate source of truth. When a 3rd grader turns it on to read about dinosaurs, it's the device that determines what they see. It doesn't matter what your server thinks the state of the device is, whether it "ought to be" correct based on what you're seeing.

Of course, the picture is really complicated, because it's not just the MDM server in the picture. There's also:

- * our application, the one that knows about students. It talks to the device through the MDM...
- * ...but also interacts with devices via our app, which students open and run on a daily basis.
- * Apple's also in the mix — in order to install apps from the iTunes store we have to allocate iTunes licenses through Apple.

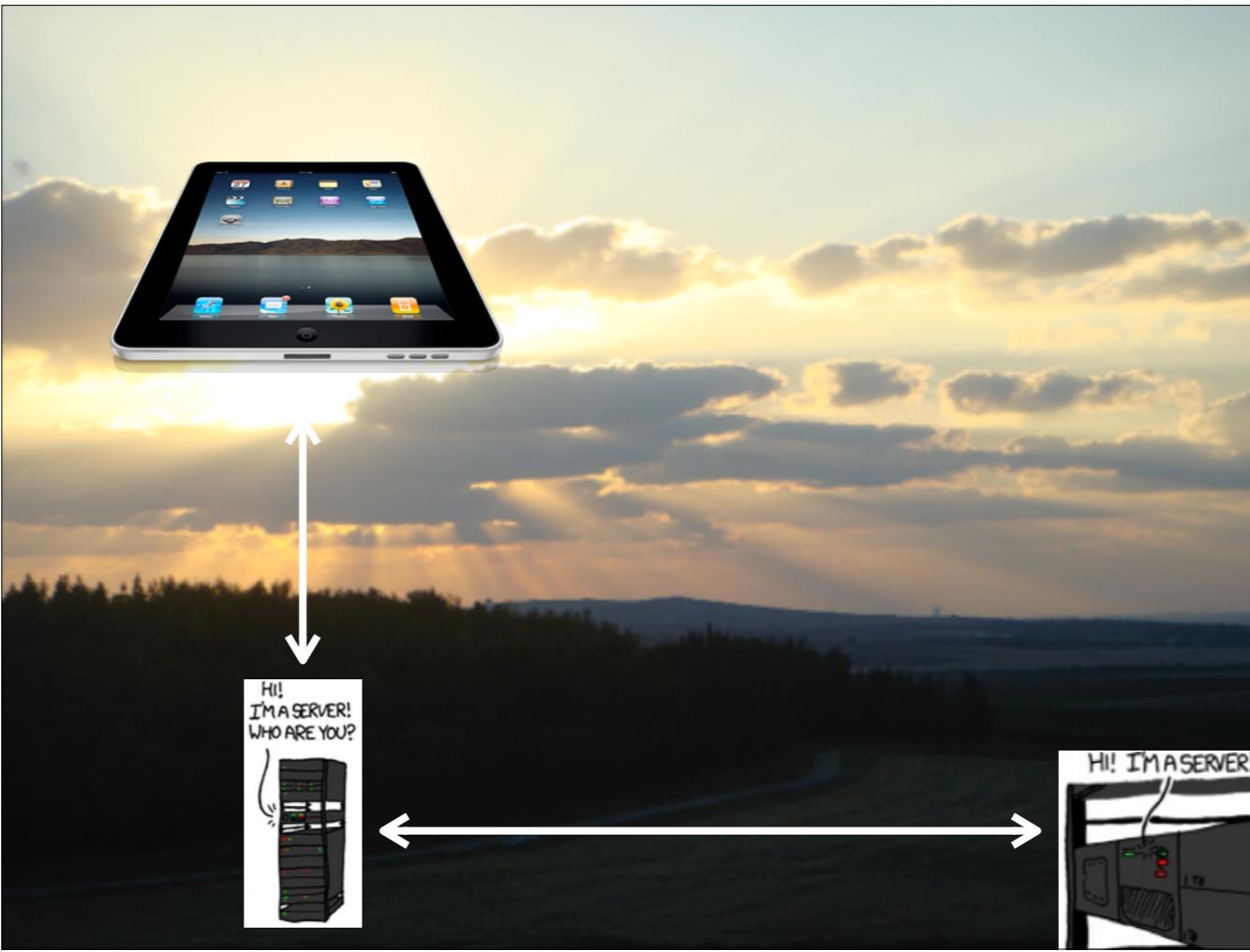


Mention that the MDM does talk to Apple

When you're managing devices, though, the device is the ultimate source of truth. When a 3rd grader turns it on to read about dinosaurs, it's the device that determines what they see. It doesn't matter what your server thinks the state of the device is, whether it "ought to be" correct based on what you're seeing.

Of course, the picture is really complicated, because it's not just the MDM server in the picture. There's also:

- * our application, the one that knows about students. It talks to the device through the MDM...
- * ...but also interacts with devices via our app, which students open and run on a daily basis.
- * Apple's also in the mix — in order to install apps from the iTunes store we have to allocate iTunes licenses through Apple.

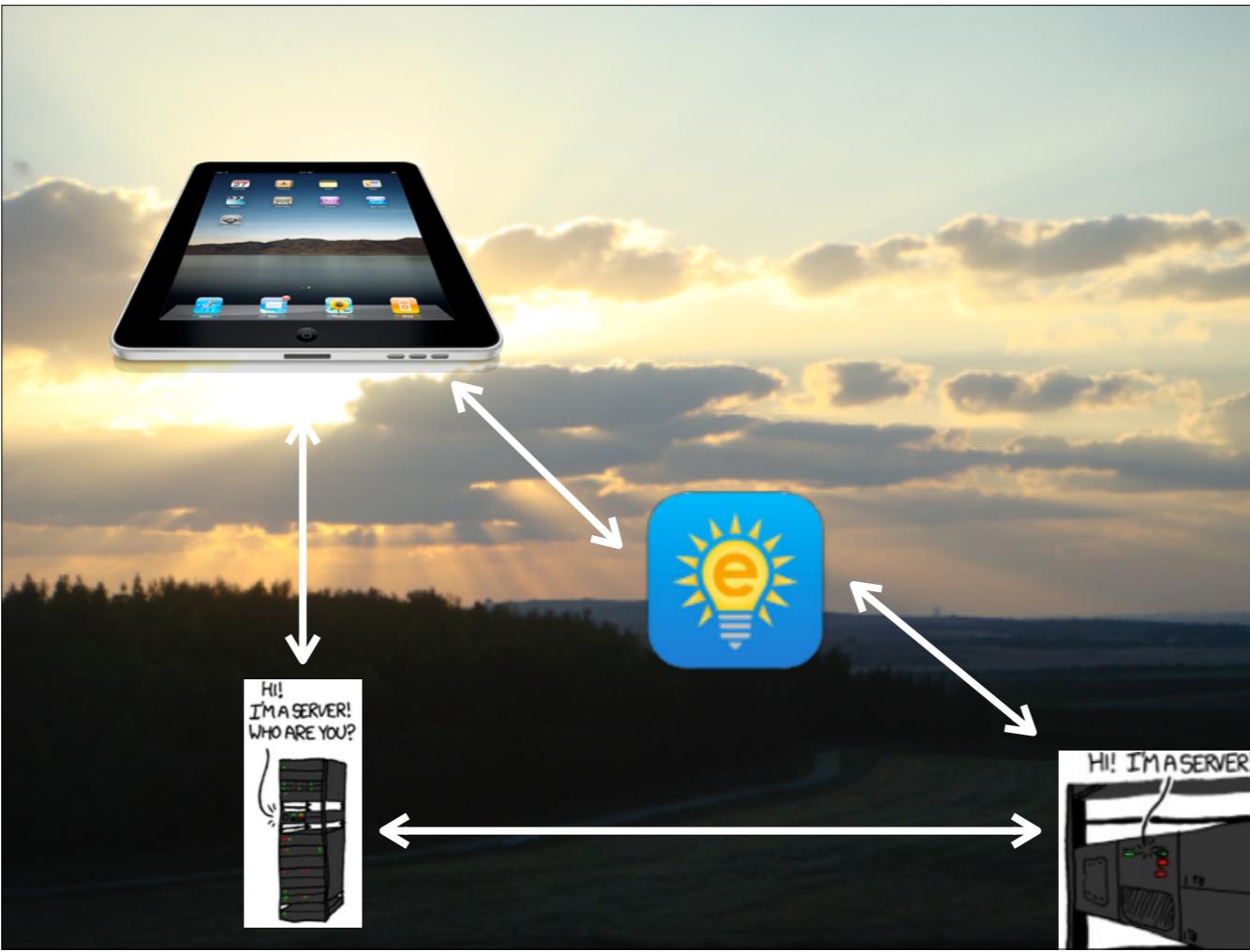


Mention that the MDM does talk to Apple

When you're managing devices, though, the device is the ultimate source of truth. When a 3rd grader turns it on to read about dinosaurs, it's the device that determines what they see. It doesn't matter what your server thinks the state of the device is, whether it "ought to be" correct based on what you're seeing.

Of course, the picture is really complicated, because it's not just the MDM server in the picture. There's also:

- * our application, the one that knows about students. It talks to the device through the MDM...
- * ...but also interacts with devices via our app, which students open and run on a daily basis.
- * Apple's also in the mix — in order to install apps from the iTunes store we have to allocate iTunes licenses through Apple.

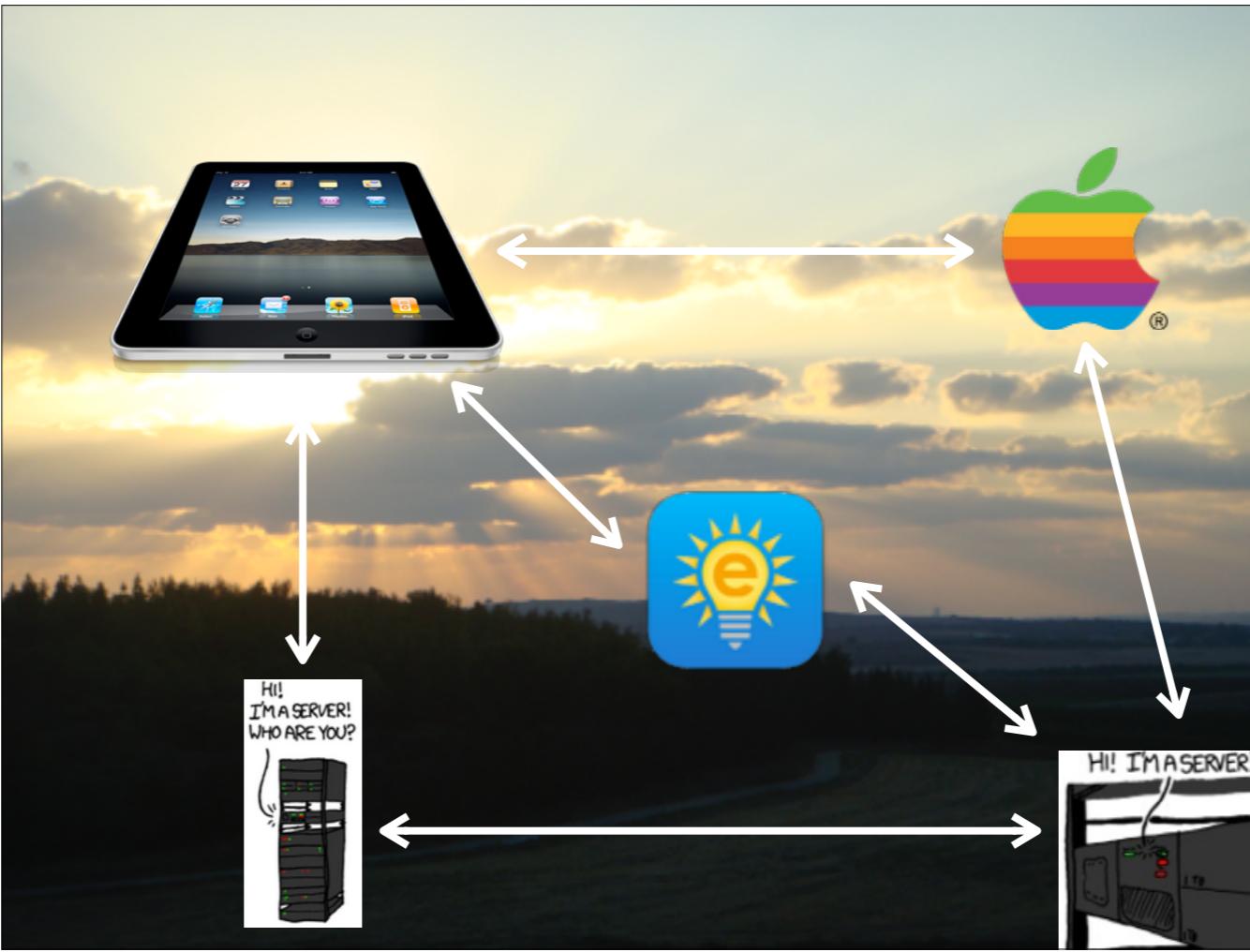


Mention that the MDM does talk to Apple

When you're managing devices, though, the device is the ultimate source of truth. When a 3rd grader turns it on to read about dinosaurs, it's the device that determines what they see. It doesn't matter what your server thinks the state of the device is, whether it "ought to be" correct based on what you're seeing.

Of course, the picture is really complicated, because it's not just the MDM server in the picture. There's also:

- * our application, the one that knows about students. It talks to the device through the MDM...
- * ...but also interacts with devices via our app, which students open and run on a daily basis.
- * Apple's also in the mix — in order to install apps from the iTunes store we have to allocate iTunes licenses through Apple.



Mention that the MDM does talk to Apple

When you're managing devices, though, the device is the ultimate source of truth. When a 3rd grader turns it on to read about dinosaurs, it's the device that determines what they see. It doesn't matter what your server thinks the state of the device is, whether it "ought to be" correct based on what you're seeing.

Of course, the picture is really complicated, because it's not just the MDM server in the picture. There's also:

- * our application, the one that knows about students. It talks to the device through the MDM...
- * ...but also interacts with devices via our app, which students open and run on a daily basis.
- * Apple's also in the mix — in order to install apps from the iTunes store we have to allocate iTunes licenses through Apple.



So there's a lot of state to keep in sync.

We've built systems that manage synchronizing data and put a lot of effort into moving data quickly between our various servers and pulling data quickly from Apple.



© eSpark Learning

There's another icon you're responsible for too: settings. (Or as it's known to those who are too young to remember the great skeuomorphic massacre of 2013...)

If anything settings harder than apps, since:

- * took over setting accidentally in 2012, since we were touching devices anyway
- * fewer good tools for changing core system settings (unsurprisingly)

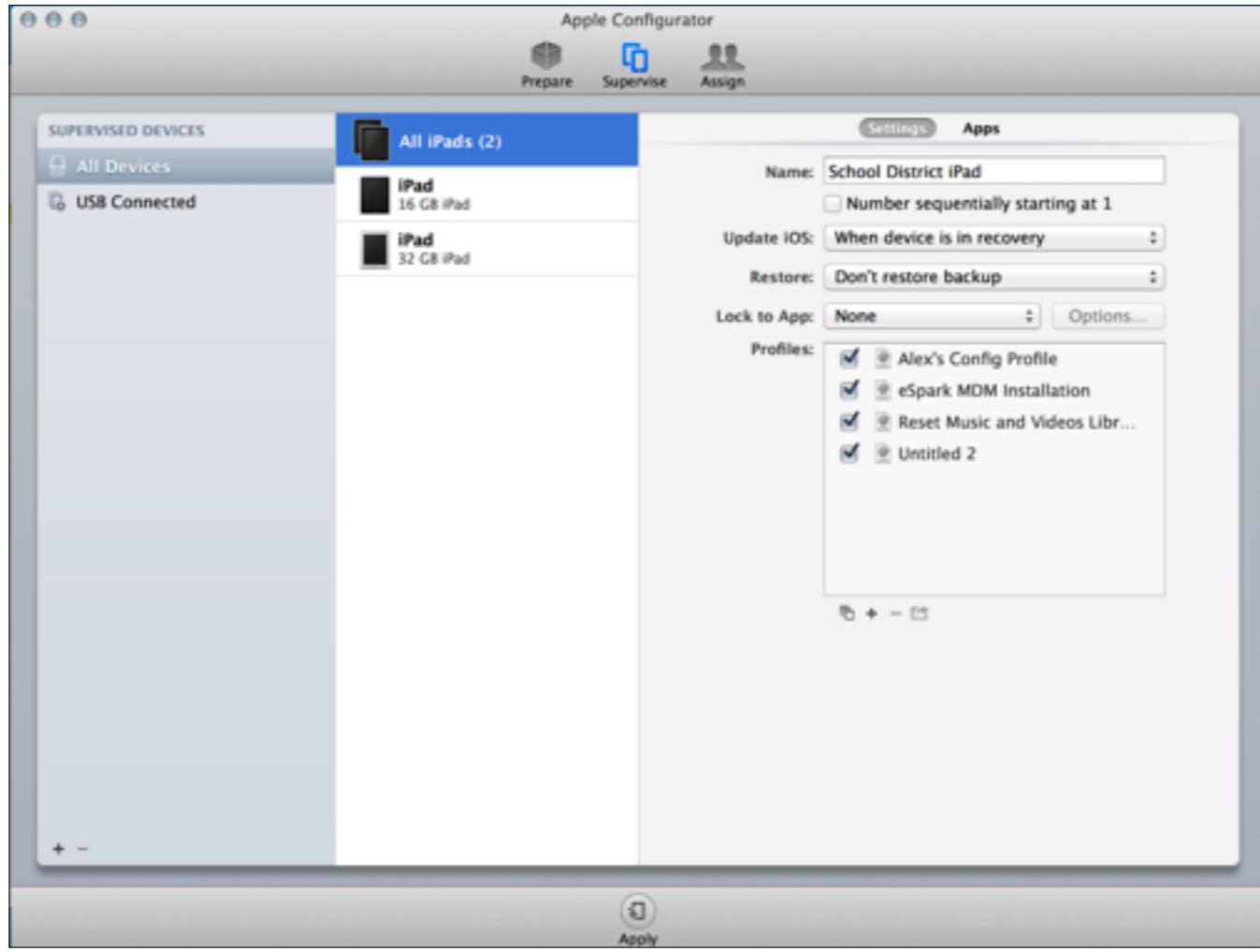


© eSpark Learning

There's another icon you're responsible for too: settings. (Or as it's known to those who are too young to remember the great skeuomorphic massacre of 2013...)

If anything settings harder than apps, since:

- * took over setting accidentally in 2012, since we were touching devices anyway
- * fewer good tools for changing core system settings (unsurprisingly)



Our main tool until this year: Configurator

Lets you update not just apps but also settings for many iPads

Used in conjunction with Deployerator: 2 Mac apps per group of iPads

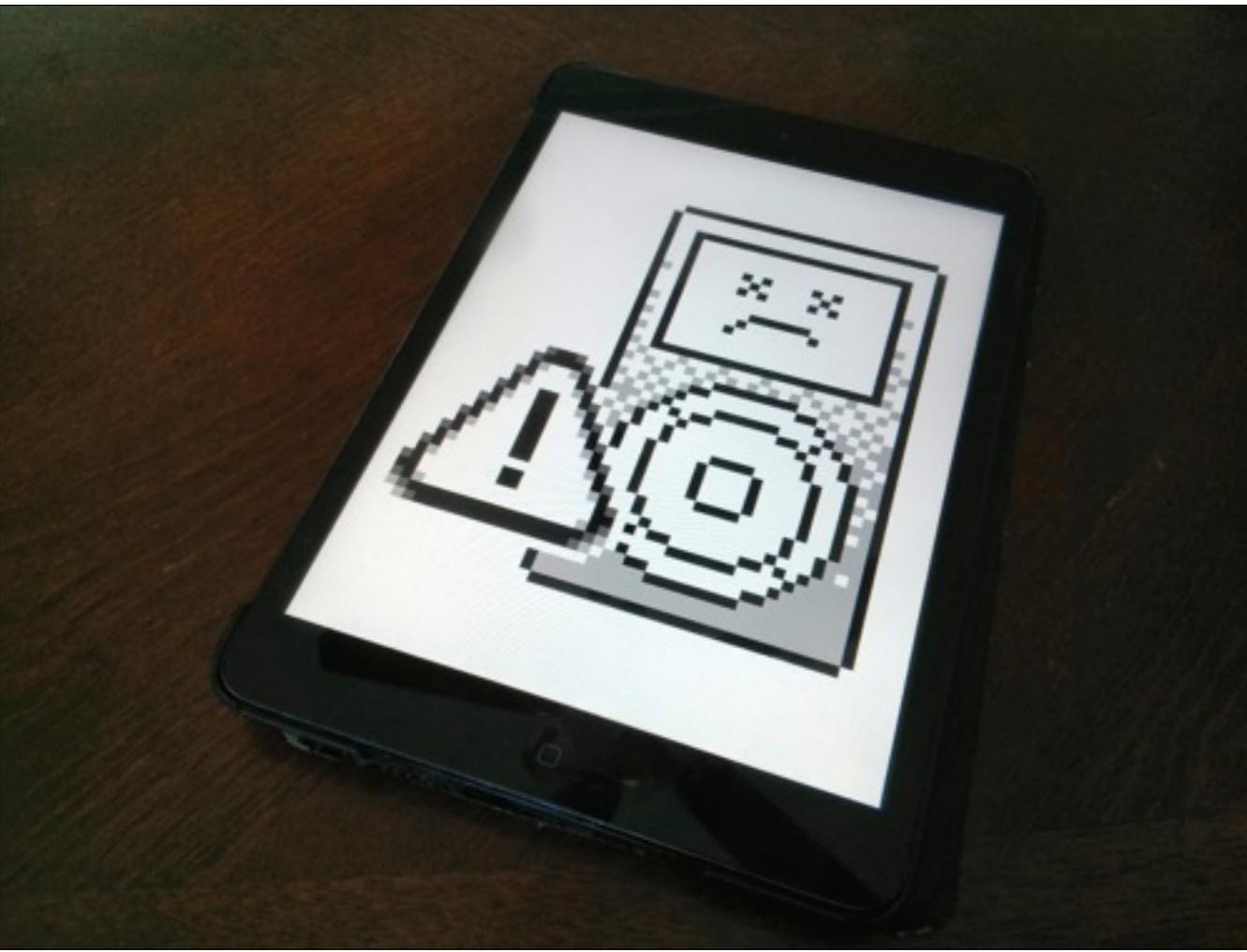


Same wired issue: can't update settings without collecting devices
PITA: McDonalds issue



http://commons.wikimedia.org/wiki/File:SRI_Dish_radio_antenna.jpg

Fortunately, the tools now exist to do all this wirelessly, so that settings can be downloaded as soon as device is turned on, and never needs wired connection.



Third learning: things go wrong, and when things go wrong, it's our responsibility to get it fixed, even if they're not our fault.

With apps, device state is a given, online or off. You have to support on and offline, but are not responsible for whether the user can get online.

As an iPad deployer, tho, we're responsible for the device state.

If connectivity issues (our apps, installation), that's seen as "eSpark is broken".



© eSpark Learning

What happens if we don't hear from a device?

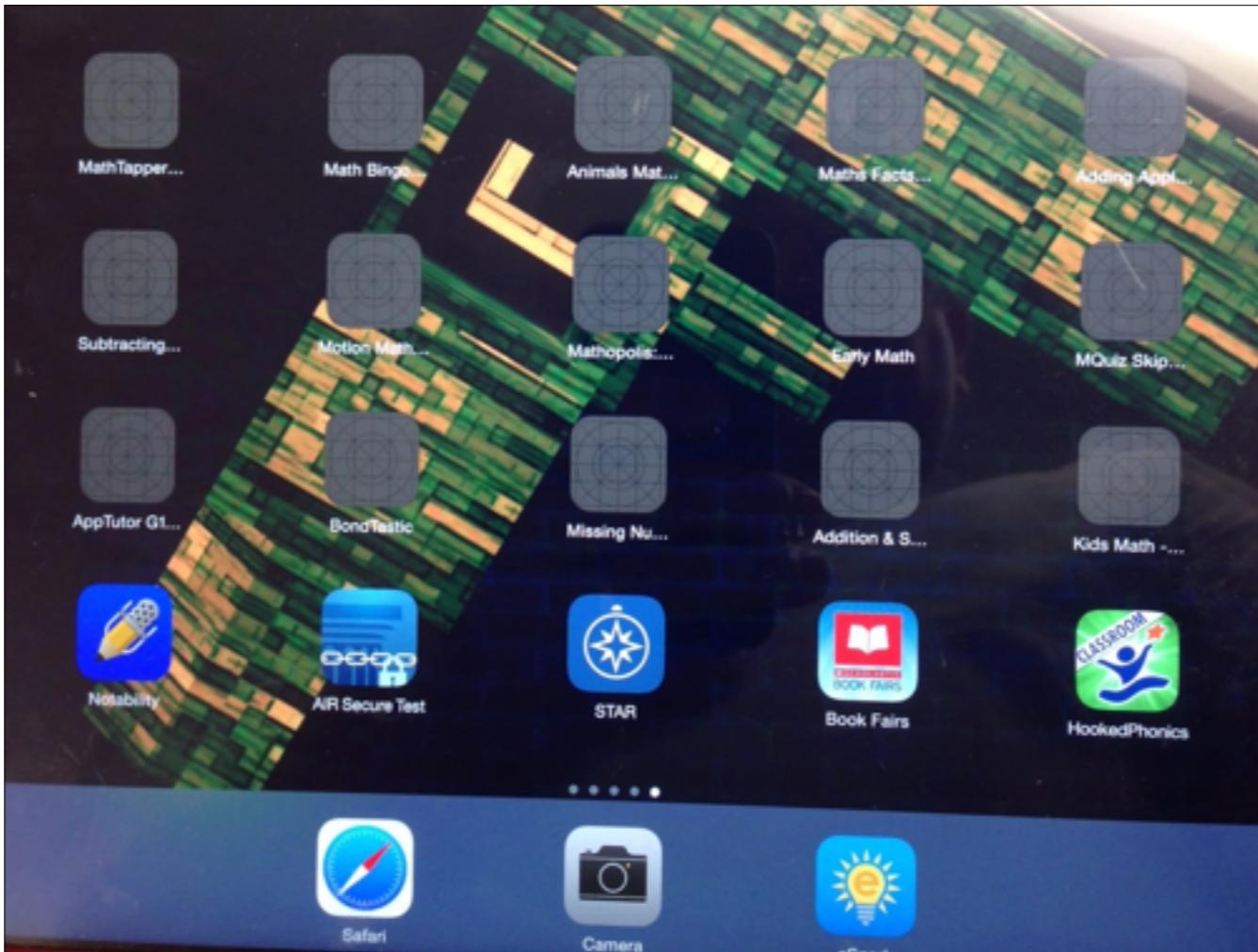
- * Devices reset/off MDM
- * No wifi/bad wifi
- * Date/time wrong (SSL)
- * Push gateway down
- * Device ignoring pings

COLLATE



© eSpark Learning

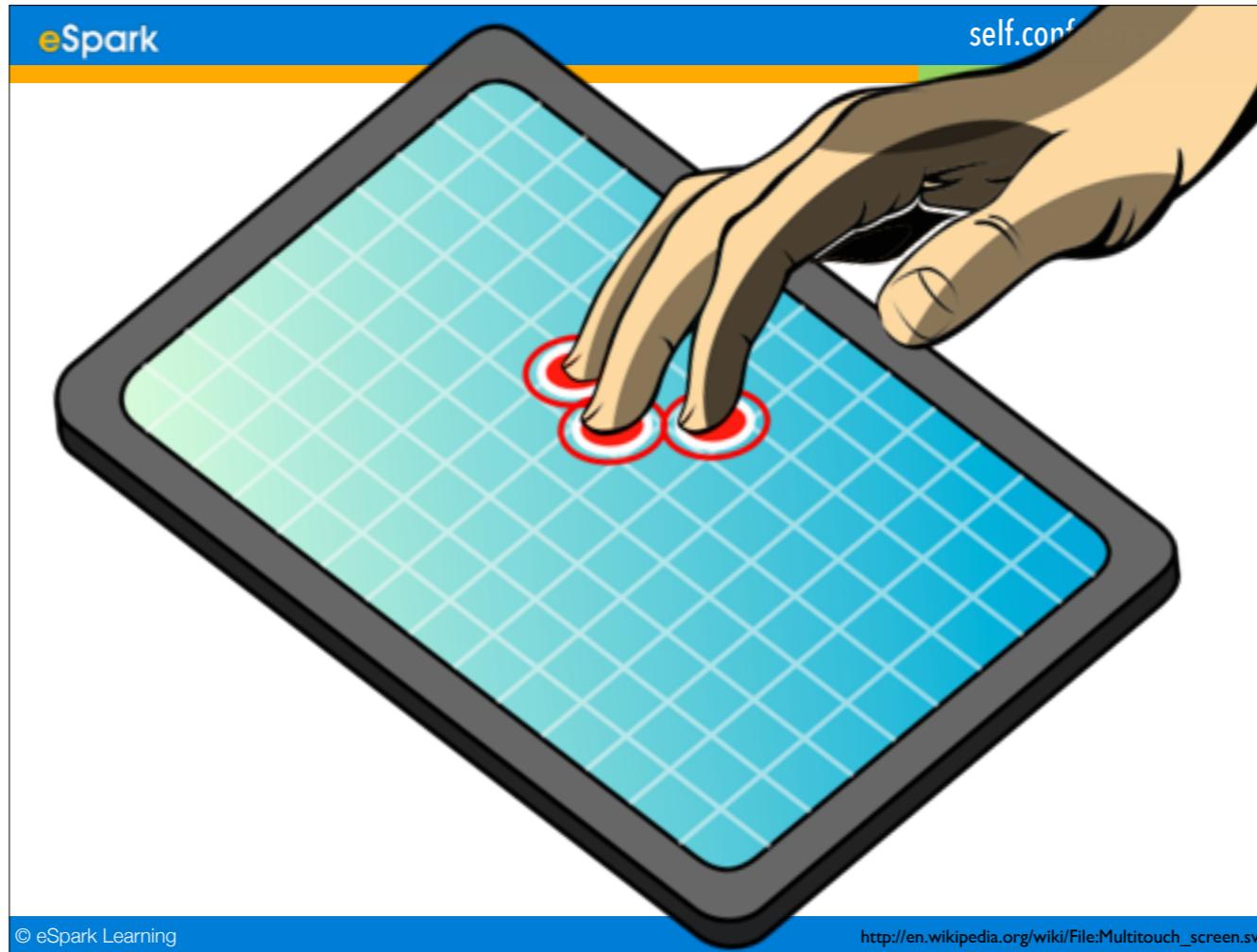
- * Has the student used the app recently?
- * What system time did the app pick up?
- * Are other devices in the school checking in?
- * Can we run diagnostics on the network? (build in diagnostics?)



App installation goes wrong: it's can go really wrong.

- * Slow wifi
- * Timeouts
- * Download errors that bork app state
- * License service down
- * iTunes store down

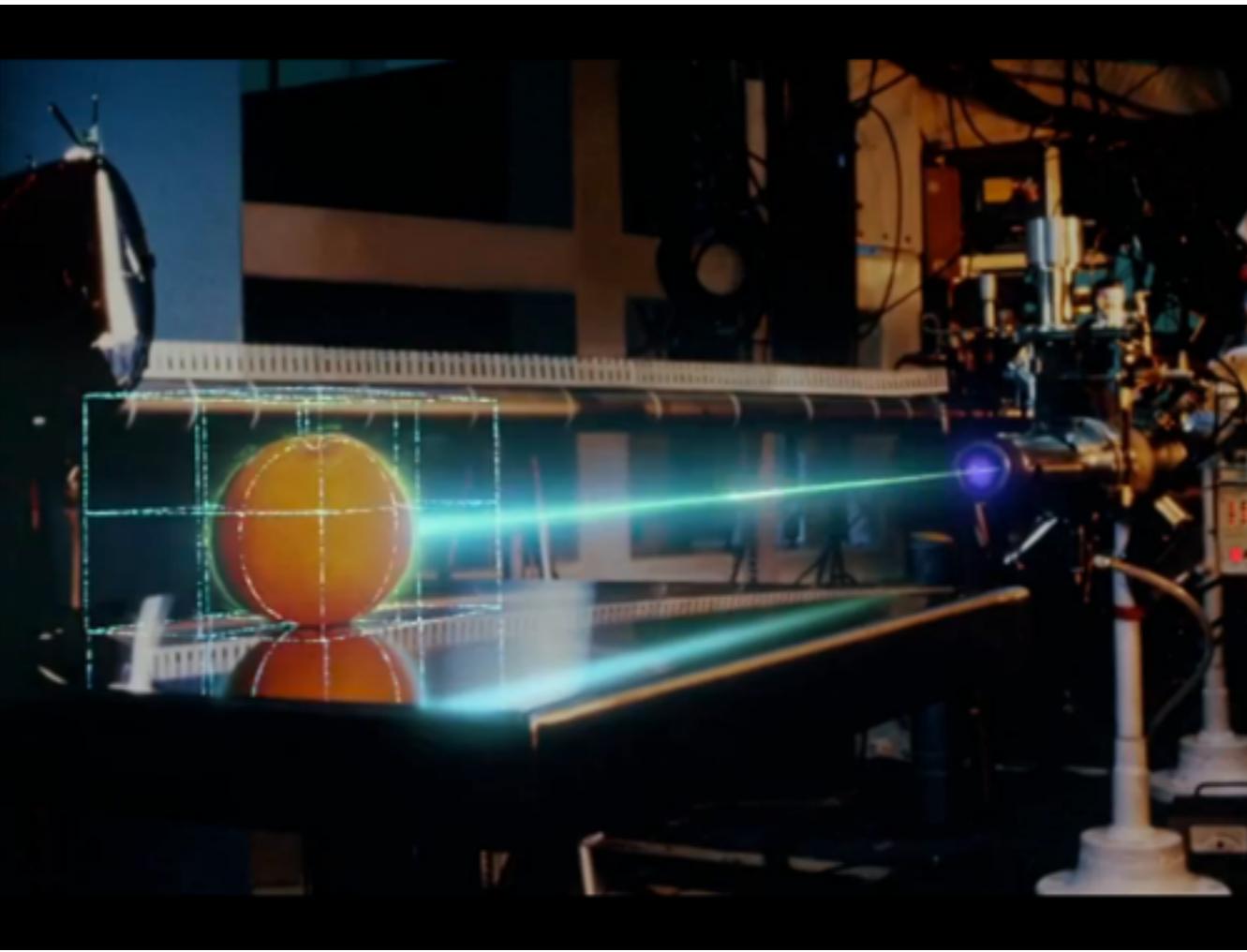
MDM can't be fire-and-forget
Track installation status
Install apps a few at a time
Install apps during off hours
Retry, retry, retry



© eSpark Learning

http://en.wikipedia.org/wiki/File:Multitouch_screen.svg

Sometimes you have to touch the device



Finally: this is real world stuff. Changing state of actual objects (albeit computers).

- * Read device state
- * Add/remove apps
- * Add/remove configurations/restrictions
- * Clear security settings
- * Wipe device

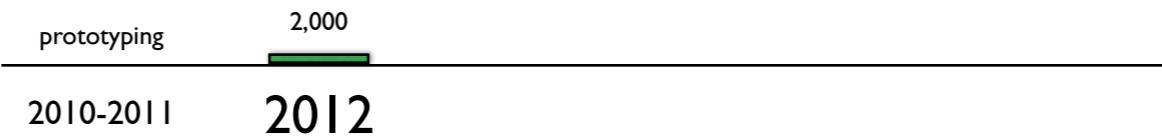
Recap

iPads eSparking

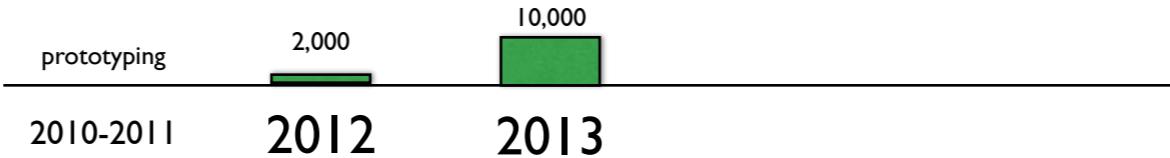
prototyping

2010-2011

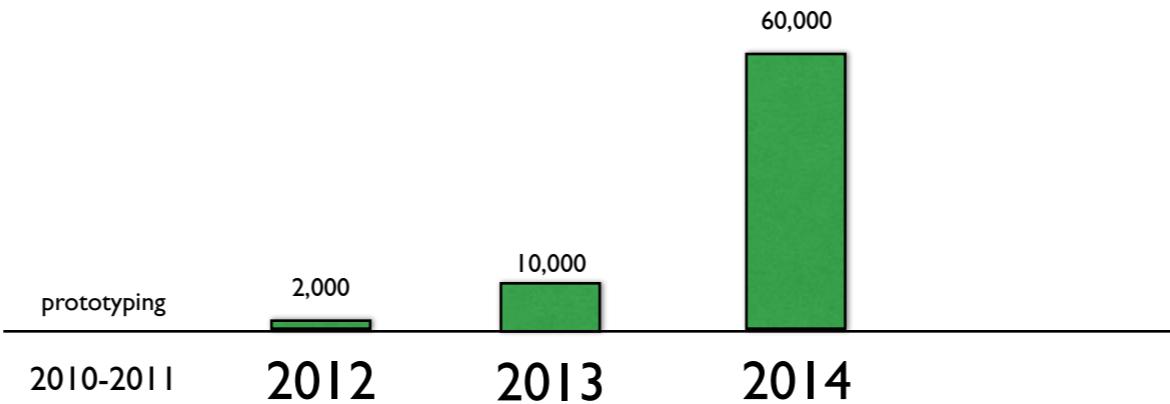
iPads eSparking



iPads eSparking

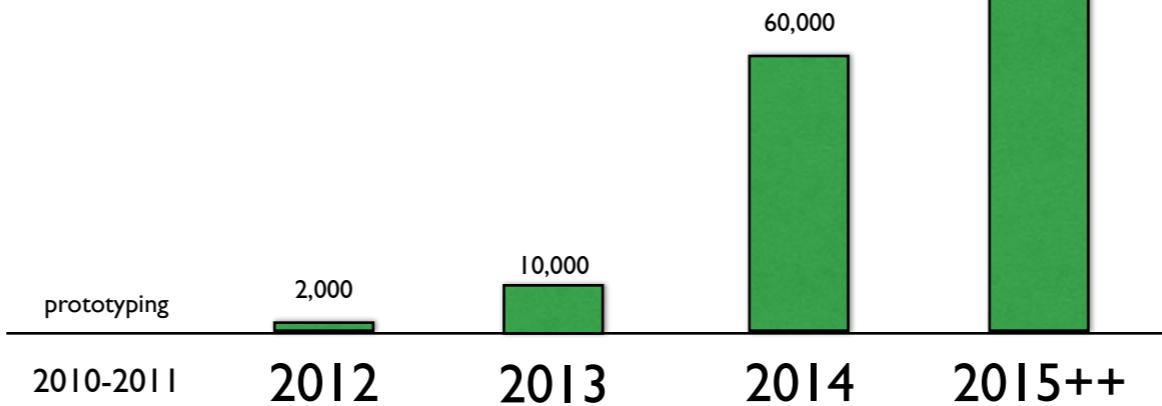


iPads eSparking





iPads eSparking





Thanks!
Questions?

Thanks!

Questions?

Engineers!



Want to help us help students learn?

<http://esparklearning.com/careers>