How did this icon turn into a design patent?

(... and how it evolved my design process)

Problem: How do we show where schedule changes come from?

Value: When managing environments that number in the tens of thousands, things can get complicated very quickly. Our icons create digital paper trails for the schedule changes and gives much needed visibility to all the administrators across the team.

Please note that I am respecting my NDA with IBM and do not go into all details intentionally.
I will be more than happy to answer any and all questions.

It’s not the icon

It’s how you use it

The icon is a combination of fixed resources from the vSphere library. So none of the icons used are unique, however how we ended up using them are.

But how?

Starting with such a technical topic, I knew I needed to have a conversation with a few of the developers to try to get some details of what this concept of **'schedule inheritance'** means to them.

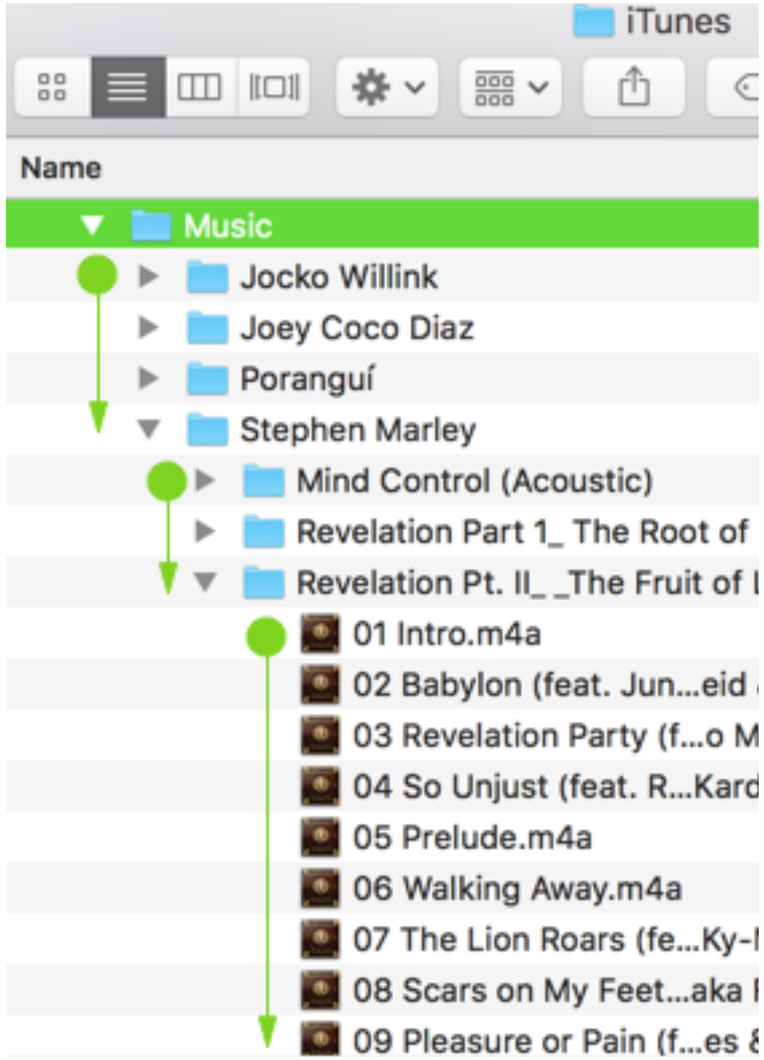
What I got however was a solution that was ready to be mocked up.

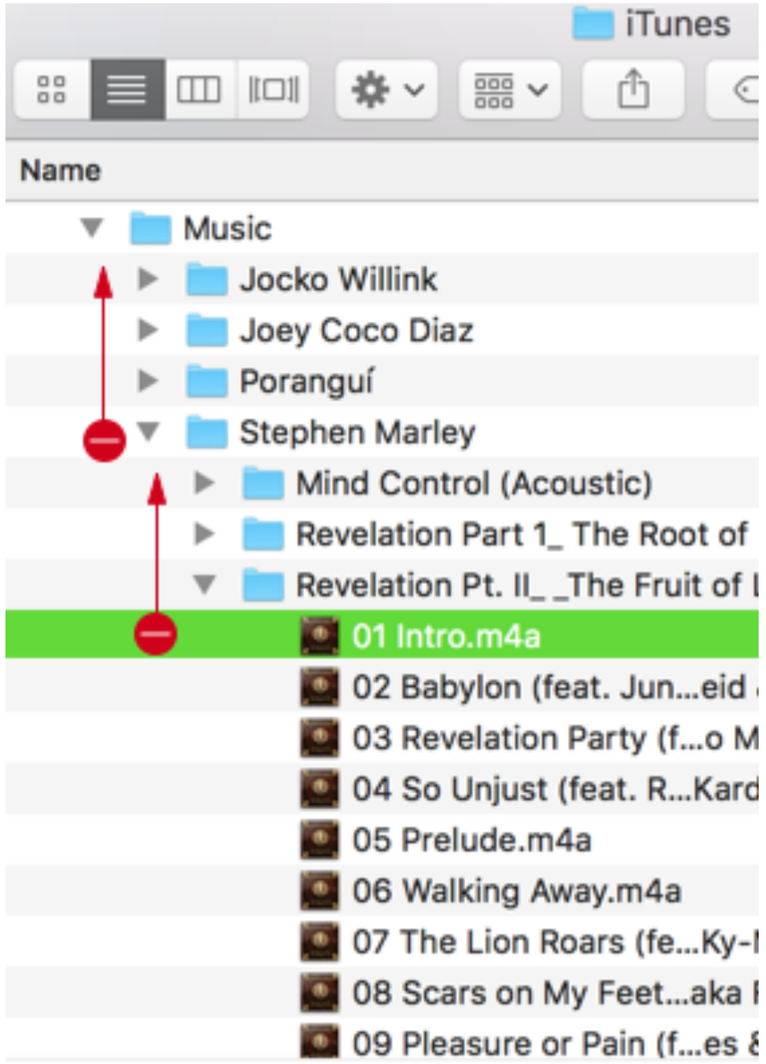
Translation

Now the conversation I had over the course of a few meetings was pretty dense.

The essence of what we did can be described with an analogy using an iTunes music library and Apple's Time Machine tool that is used to back up your computer.

Scenarios





Top-Down vs. Bottom-Up

Which one?

Let's say you want to use Time Machine to backup your Music folder. Once a month a schedule tells it to make a backup of everything in that Music folder. It copies the 4 artist folders, the albums that belong to the artists and of course the songs in. (This is a top-down model of schedule inheritance)

But! Storage space costs money, the less you use the less you spend. We all skip the intro tracks, so to shrink the size of the backup, I can create another schedule to exclude all tracks labeled Intro. I create this schedule when I'm already on a song so I could then send that schedule up and it would go through the folders and removes all Intros. (A bottom-up model)

Let’s do both!

Incorporating both a top-down and a bottom up approach eliminated the most amount of edge cases, development was certain this was the obvious path to take and we should just start building it.

The benefits of talking early

Getting this conversation out early was the key, because I was able to say, "this is a good idea and we can user test this, but we should also test the top-down and bottom-up models separately.” My initial fear was that using the ‘both’ model, we would create unnecessary complexity for our users.

Iteration instead of arguments

I was able to work with the team to create scenarios and as we worked out the details, we able to test all three models with our users. It is fair to point out that sometimes the timing of things just works out, we were able to finish this before our yearly big meet up with our VM Administrators, so we were able to show this to 60+ users. The top-down approach was the favorite by 95%.

Success!

By using the combination of icons and a clickable link to the source of the schedule, nothing like that existed in the vSphere so six months later we were approved for a design patent!

Talk Early and Talk Often

Now this is not meant to be a story of design was right and development was wrong.

It is meant to be a story of the team being right and on the same page, that delivered a solution that solved the customers needs.

1

Insight One

The more perspectives you can introduce to the problem, the more refined the solution.

2

Insight Two

Second (this might be an IBM thing), but some developers and architects are very solution oriented and it can lead to jumping right to 'THE' answer and if you are not communicating early, you can get feedback that guides you toward their original idea. I was able to pick up on this because we talked early and until we user tested, we kept having conversations that very strongly argued for ‘both’.

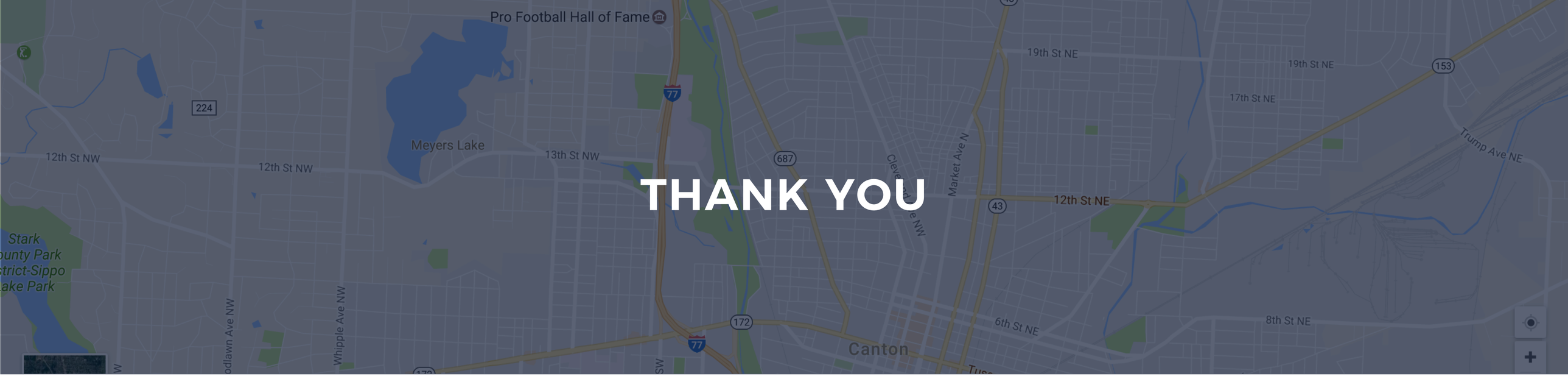
3

Insight Three

This solution would not be what it is without the help of the developers.

We did not receive a design patent because we just went through a good design process.

We received the patent because we went through a good process AND delivered a solution that worked.



THANK YOU

Steven Voyk

330.209.1485

stevenvoyk@gmail.com

www.stevenvoyk.com