Chapter 8 | Okta

PodcastFlow, while it started as a single application striving to help podcasters plan, produce, promote, and profit from podcasting, the scope of the project grew during the early months. I already admitted we launched a quick venture with an incomplete plan resulting in gradually increasing scope.

Our team rapidly recognized a need for a vendor to manage financial transactions and help us stay on the good side of good rules.

Expanding our software to incorporate an external vendor or partner during the purchase process proved new to us but has become commonplace in the internet marketspace. We were slow to adopt this because our firm's traditional customers are governmental agencies. This technique of selling subscriptions in a retail-like fashion is a new business model for us. The technical challenges for us involved nearly instant communication of a purchase so that we could setup their account within our system.

In early winter of 2019, Kelly (Doctor Doctor Dodge) our collaborating partner on this business venture, stated what we should sell a course to better support podcasters. In response, I wrote a learning management system in three days using Oracle APEX – I will dedicate a future chapter to this. The learning management system, or LMS, hosts lessons. Lessons include a video, some narrative text, some actions, handouts, etc.

The LMS is a separate application from PodcastFlow. They do different things. Furthermore, not all people who buy PodcastFlow will likely buy a course. We just added complexity to the tools we now offer.

A customer can by X or Y or both. We need a means of allowing access to X or Y or both that is easy and seamless. Our Oracle tools as well as our historic process would work tolerably well but not perfectly. Additionally, we recognized that we also want to permit registered users access to our helpdesk articles.

why should we ask a user to log in and out of applications? Why can't we make all of these tools appear and behave as one suite? Why should we ask the user for all of those clicks and navigation efforts? Why ask for username and password as a user transits from a course to PodcastFlow and back. That seems dumb.

On the internet you can find anything, of course (even somethings you may not want to find). We looked for and found a service named Okta (Oh-Kay-Tee-Ay). These guys provide a service that validates the identity of users and gives them the appropriate access to web-based software.

One advantage of a third-party service, like Okta, is that we can use them for tools like our helpdesk. It is like giving a user one key to open all of the rooms that they need. It improves simplicity.

On the 8th of January 2020, we spoke with someone who may well have been our salesperson at Okta. Frustratingly, it took weeks to get a phone call scheduled. A call that ended in less than fifteen minutes. We want support on the services we buy from them. In the internet machine, the supply line is digital. And so is the delivery.

Back in June, looking for an answer to a different problem for a different application for a different client, we found Okta. We wrote a bit of code to explore their capabilities, bought a subscription at $50 per month for their “Developers” product. We walked away after three weeks frustrated with the firm. I’ll come back to that in a bit.

Let’s look at the internet machine from the user’s perspective.

PodcastFlow philosophy included the mantra: *mouse clicks are bad*. Therefore, logging into the application should be effortless and secure. Ease is accomplished when a single sign on is shared between tools or machines. When having multiple tools, then shared authentication of users between them reduces keyclicks without risking security.

Authentication asks: Are you, you? Oddly the best ways to answer that may not be with an email address and a messy jumble of letters typed on a keyboard.

Asking for a username and a password doesn’t inform the machine about the entity speaking to it. If a browser has remembered my credentials and automatically logs me in, then anyone at my browser will be treated as me. On a TV show about law or cops, this assumption would be: *circumstantial*.

“Detective, that evidence is circumstantial at best. I need something more to get a guilty verdict.”

We know the quote.

Our team password vault has 135 passwords that I have access too. Some are just mine; some are critical passwords for infrastructure. For websites, tools, and applications that I just don’t care about, I have a page in Notepad++ stored privately on my computer. This page has another 100 entries of username and passwords. I have one password that is 20-characters long. It is used for sharing programming code with the team via GIT. What does it take to use it?

Step one, open the VPN on my desktop.

Step two, open the right software tool.

Step three, do a GIT pull. Now you have access to some text files that is also our raw code.

If you are at my desktop, I have a significantly larger problem, don’t I?

Am I more secure with 200 passwords?

Or less secure?

In 2019, the industry knows that username plus password employs circumstantial evidence to identify an entity. Let’s further admit that username/password do not discriminate between human and non-human. It is a suite of technology that is as old as the computer industry.

The old computers – the really-big ones, the “mini-computers” and the “micro-computers” of the 20th century enjoyed one added level of security that newer machines do not. The older, legacy, computers required a physical connection. It was a physical pair of copper wires that ran from one point to another point. I include a telephone line in this analysis. Before my high school days in the late 1970s, I used a teletype to connect to a mainframe computer. Uncle Lou dialed a phone, a touch tone phone, then put the handset into an acoustic coupler modem – the earlier form of a modem. This clunky device would modulate digital data into an audible stream. Audible data is analog. It will chart out on a graph as a waveform. On the other side of that call, a device answered. The analog stream returned to digital (demodulate). ***Mo***dulating then ***dem***odulating a data stream between digital and analog gives us the word modem. That teletype in Uncle Lou’s office was the terminal. Words spat out from the remote side on fan-folded paper that had green and white horizontal bars. You’d type words in response.

My first program resembled this:

```
10 BEGIN

20 PRINT CHR(7)

30 END

> run
```

It rang the bell on the teletype. 40 years later, ASCII character 7 is still “BEL” (written Bee-Ee-Ell). Security on that system involve significantly more barriers than today’s system.

Every modern system bares itself to the internet. It listens for a call constantly. Call it with HTTP (I mean your browser - the common internet ***h***yper***t***ext ***t***ransfer ***p***rotocol). It will answer. Call it with HTTPS? The “s” stands for secure, meaning it is encrypted. It will answer you. Call it on port 22, the SSH port? It will answer. Your doorbell, or thermostat, would answer if I called on the telnet port, port 23.

Telnet is a protocol that was established in 1969 as a means for a terminal to talk to a mainframe. 50 years later, that port would give me control of the camera in your electronic doorbell. Your doorbell will answer anyone who comes knocking via the internet. I would so like to be wrong about that.

Gee, if I am wrong about telnet on port 23, I guarantee I am NOT wrong about devices answering when called by browsers and probably not wrong about SSH. Your doorbell, firmly attached to your house, on your street is open to the entire world via the internet. Your doorbell monitors your front door. Meanwhile, your doorbell is listening for instructions from the internet on ports 22, 23, 80, and 443. It is harder for me to get to your door than for me to get to your doorbell with its camera, microphone, and speaker. Your internet-based doorbell is open to the entire world – everywhere!

To revise an annoying child’s joke.

Me (via internet): Knock Knock

Doorbell (via internet): Who’s there?

You know the rest, don’t you.

Me (via internet): Admin

Doorbell (via internet): Password

Me: Admin

Doorbell: Welcome Admin.

And the doorbell gives me a menu of actions including turn off camera, erase data. Go ahead substitute doorbell for thermostat or refrigerator or television. Or Wi-Fi and internet router?

Our firm managed $5 billion dollars on behalf of the Government of Puerto Rico following the hurricanes of 2018. 2,000 users uploaded 400,000 documents to our system as proof of legitimate financial activity. Usernames, passwords, automatic session timeouts burdened the users, so they told us. “It’s too much, we just need to get this job done.” Our grants management system lifted the spirit and construction of a 40-year-old mainframe database application to the internet. We painted it blue and white. It used internet technology in as much as it had to.

A fuller adoption of internet technology while adhering to the *clicks are bad* mantra, has us exploring a better way of identifying the entity (or human) asking for access. In 1995, when I first got to Alaska my boss’ email password was “read”. Why? She knew she needed to “read” email. Done.

This brings us back to Okta. Tracy Kidder wrote *The Soul of a New Machine* in the late 1970s. Most of the companies he mentioned are gone. Commonplace names of my youth and neighborhoods have disappeared: DG, DEC, Wang. Curiously, for those who watch Boston-based broadcast television in 2019 (boy did I just narrow that audience), Digital

Credit Union also called DCU seeks your patronage. This was the DEC employee credit union. DEC is gone, their credit union remains.

I wonder what of Okta will exist 40 years hence, in 2059. I do have a guess based on the shear arrogance of their posture in the marketplace, their staff, and their sales position. Do they know history? They probably do.

Business Decision

Podcast Flow exists within a suite of related internet applications. Podcast Flow aids in the management of Podcasts. The learning management system, currently called Maestro, hosts the courses. We want to give users ready access to the help desk tools. By extension, we want to have a single login for all three tools and let people navigate between with ease. Three tools, one tool – an unnecessary distinction if access and navigation are shared.

Okta states: “Okta is the identity standard”. To quote further, “the most complete access management platform for your workplace and customers, securing all your critical resources from cloud to ground.”

In the architecture of the internet machine, Okta became part of our supply line. They plod through their mundane days unaware of that we are their customer. They do their thing. We do our thing. They deliver a digital service via the internet. We pay monthly via credit card.

In June, we wrote an ***a***pplication ***p***rogramming ***i***nterface or API, a connector, between them and us. This connector was written in our favorite programming language with Oracle, PL/SQL. We got frustrated and abandoned that effort. Okta has a confusing message for their peers in technology. For $50 per month, you can use our service if you are self-guided and have a few thousand logins per month. We wrote the connector, contacted the company and were told by them we bought the wrong product from them – whilst our connector served our needs just fine. Confused? So were we!

We walked away in June of 2019.

Implementation

The business decision and implementation exist as separate activities. To my left is the stated operational goals, to my right are the pathways forward. Few vendors stand alone in their marketspace. For each Hertz, there is an Avis. For Coke, Pepsi. And Okta is not alone.

Implementation includes our goals to use Google, Facebook, Microsoft, and Amazon credentials to identify the user. If you are already logged into Microsoft and proven yourself to Microsoft, then do your work with Podcast Flow – if you are a customer.

With research and testing, we returned to Okta. Our prior effort with Okta focus on our government-facing software. This new effort, Podcast Flow, seeks thousands of retail customers. We want to get the product to the market with haste. And we already written a significant portion of the code for an Okta connector or API. Within days, Okta can become integrated into our internet machine and its supplier network.

We write. We test. We are successful. It is not a lot of work. Here is an inventory of the elements in our connector: list groups, list users, add new user, assign user to a group, suspend a user, un-suspend a user, delete a user. Done. Each of these is button someone will need to push at some point.

The connector required 1,542 lines of code and comments in Oracle, just over 4000 words. We measured the investment in days of work. Testing the connector, again, required only days. A connector works or fails. If it fails, we must fix it.

That's the secret with connectors. Okta provides the service. We own the problems. We own the code. If there is a problem, we must investigate and fix it.

In October and November of 2019, we keenly remembered our frustrations of the recent June and our brief Okta engagement. Today, they are an industry recognized leader. Today, they are trusted. Today, they have respected customers. Today, we have a functional connector. Today, they cost us $50 per month.

November 12th or 13th of 2019 remains on my calendar as the first viable launch date for Podcast Flow. We missed the date. On December 20th, our connector failed. The failure demonstrated erratic behavior. Intermittent failure means there exists a gremlin in the process.

Let's examine the variables in the supply line… Our connector remained unchanged for weeks, and functional. We own our piece of The Cloud. We are firmly located in the mathematical center of the internet. As such, our internet connection is guaranteed by Amazon Web Services, approximating reliable. Our key troubleshooting question becomes: Our problem or theirs?

Intermittent failure on user authentication is the same as complete failure. Therefore, intermittent failure related to paying users gaining access to their service is a catastrophic system failure. In the lingo, we call this a P1 or Priority One issue. December 20th and the weekend that followed permitted us to step through a P1 issue while not in production.

That's a luxury. A drill without the artificial aspects of a drill.

System Down

On the 20th of December 2019, the connector with Okta demonstrated intermittent failure. Each hour we got one inquiry answered, maybe. Our logs showed use rates of failure and success precisely. Had we paying customers, they could not log in. If customers cannot log in, your system is down. Whose system is down? Our system is down.

We have no paying customers, on that day. Thankfully.

The first actions involve scoping the problem then seeing if we control the technology that breaks the connector. Troubleshooting technology looks like magic, appears to be a flurry of activity. Yet it should be as simple as flipping a light switch.

Flip light switch: lights on. Flip it again: lights off.

If you flip the switch and the light does not come on, then there is a problem. A finite list of candidates exists. If I start at the street and work in, it sounds like this:

1. Is there power from utility company?
2. Is there a circuit breaker or fuse that should be checked?
3. Is the wiring intact?
4. Is the switch working?
5. Is the light doodah working?
6. Is the lightbulb good?

Switch on – light on. Switch off – light off. With 100 cycles, I expect that same behavior 100 times. Note that lack of magic. There is no magic.

At some point, our team said: It ain’t us, it is Okta.

Our ticket to Okta went unanswered.

On Saturday, our ticket remained unanswered. I called Okta’s service line. While on the phone, I was informed that they do not provide support to customers of our sort. With a description of both a critical failure and demonstration via logs and other proof that they owned the responsibility for the problem, I got put through.

They consented to opening a ticket. In the literal, I heard: “We’ll do you a favor this once.” I got connected to a technician. Within an hour, the problem was solved. He swore he did nothing. Problem disappeared.

On that day, I asked to be contacted by sales to buy a support contract.

Sales Meeting

The sales meeting took 12 minutes and executed on the 9th of January 2020. That was the tenth business day following my request.

Our account manager, Nolan Doyle, immediately corrected the introduction his colleague provided. He upgraded his title to *sales executive*. Based in the Washington DC area, he stated he aided emerging markets along the eastern coast of the United States. He went to describe Okta as a publicly traded company on the New York Stock Exchange, giving ***New*** and ***York*** special verbal emphasis. Okta has a duty to their shareholders and investors. He did not tell us his honored duty.

He asked: how many users are we expecting? We tossed a number – twenty thousand. A thousand. We were already shamed by him for being too small to earn a spot in his definition of emerging markets. His word selection and tone communicated, that he had no time for us or our questions.

Arrogance.

What a wonderful tool arrogance can be. Done right, it provides an individual the confidence to try new endeavors and expect success. I can often go toe-to-toe with any arrogant bastard: family legacy, family connections, personal accomplishments, professional accomplishments, publications, earnings – I can win on so many of those fronts.

Arrogance in sales can work too. It certainly did for Data General. Founded by four employees of Digital Electronics Corporation in 1968, they competed for customers against DEC and IBM for a decade-ish. Sadly, their internet domain name “DG” is now owned by Dollar General, a discount department store. DG emerged to fight for customers in marketspace then defined by their competitors. In the 1980s, these guys even printed t-shirts saying: “We did it on a desktop.”

I remember the fleet of DEC and IBM salesmen (yes, sales*men*). The stories ring with hyperbole. The strict dress codes; the codes of conduct; the uniformity in approach. I once had an office adjacent to a regional DEC sales office. These guys had nearly identical Ford Taurus sedans. Sales managers did a “bounce test” on their cars to confirm that they were locked with alarms set. In that same time frame, I got a holiday card from the local Xerox sales team. Fifty people standing in front of a suburban tech-based office park – complete with the pond. Fifty faces on fifty identical suits and bodies. Pre-Photoshop, that was a pretty good sight gag. Well done, Xerox. A very dated sight gag. Who is Xerox and why are identical people a funny thing? Hum.

The delight I have in writing about the arrogance of IBM, DEC, and Xerox sales force is their obsolescence. Gone. Bye-bye. Bye-bye Taurus. Bye-bye white starched shirts – or the light blue version. Bye-bye to that arrogant sales manager who said: I can tell everything I need to know by looking at a guy’s right heel, his watch, and the fit of his jacket. This man wanted to see a smoothly warn shoe-heel indicating hundreds of hours of driving. Shabby shoes bad, but worn in the correct way with polish, good.

In 1979, DG states: “We intend to make a lot of money”. They barely made it from 1979 to the internet before dying. They staked a claim that they will sell a lot of units at a deep discount.

My arrogant account executive from OKTA proved himself and his value with a link to the New York Stock Exchange following an IPO from less than 2 years ago. His message: Okta is a big deal. And we have big-deal customers. Are you a big-deal customer? Really, his questions should have been, will you be a big deal? He didn’t even get that right – asshole.

Nolan Doyle informed us that we were on the Developer’s Trial. To that I offered a correction. We are on a paid service, so it is not a trial. And it is the service we need. He told me, that they call it The Trial. I returned the comment. We did a 30-day trial for free, but then executed with my American Express, therefore not a trial – a paid service.

“Well, semantics aside, you are on the Developer’s product. To buy service you must be buy One App.”

“Nolan, I don’t want One App. We don’t need it. We have already integrated you into our products as is.”

“But to get support, you must buy One App.” He then explained, “You have support now. You have basic-success. This is gives you access to support twelve hours per day, five days per week.”

“Right, with a returned email at two to three business days. I cannot run a production system and have you as an ecosystem partner without support.”

“You’ll need One App”

“Ok, I’ll bite, how much is One App?”

“$17,000, but this is the end of our fiscal quarter, so I am in a position to get you an amazing discount.” In time, after some more conversation, the price came to $11,000. The support on One App costs $3,000. He has no ability to discount support costs because of the very real investment in human capital. Support, he explained to me is a capital-intensive venture.

That little snark-generator in the lower part of my brain tossed a few poorly chosen phrases in the conversation.

In short, if you commit to spending $20,000 per year with us, then we will answer the phone. Until then, fuck you.

Well, fuck you.

Click.

No, I didn’t say fuck you. And neither did Nolan Doyle, account executive. He is the modern-day equivalent to the starchy-shirted IBM and DEC guys. With their gently worn right heel from resting next behind their Ford Taurus accelerator pedal. He’s that same guy. Today, he likely wears some fitness-focused not-a-watch on his wrist.

So, at $50 per month, we have a supply-line partner for our machine that will not answer the phone. Our machine has multiple separate applications tied to one: project management, training, support.

Frankly, if our growth predictions are correct, we will exceed Okta’s definition of “One App”, so we’ll be required to move to their $42,000/year enterprise account.

What is the cost for me to replace Okta?

First option, find a competitor, write a connector – so another 2000 lines of code? That’s barely a week’s work. My goal would be to have a supply partner I can call with a critical failure for less than $20,000 per year – that’s a saving right there.

My second option is to write the tools we need internally. Can I write a single-sign-on solution for less than $50,000? These connectors and authentication processes predicate on public and shared standards. Yes, we can. That’s my arrogance.

What is the cost for me to operate with Okta and face a catastrophic failure on their part? Well, Okta and Podcast Flow both depend on Amazon Web Services. Study the Okta website, you’ll learn that if they have a failure, we all starve. None of us can fly. The court system fails. There are no airport car rentals. Okta will minimize this risk. And thus, they will minimize our risk.

[sfx oops]

Oh, that list is so Pre-COVID. Clearly when I wrote this in January of 2019, the worst I could think of was empty shelves at the grocery, and grounded airplanes. I think the point is made: Okta sold to a lot of large internet companies who rely on them. Their failure will impact many.

[sfx out]

We were told by Okta, we don’t need support because we are too small and bought the wrong product.

Ok Okta, we hear you.