TechStuff Podcast – March 2020

Smart Talks with IBM: Project OWL

JONATHAN STRICKLAND: Welcome to TechStuff, a production from iHeartRadio. Hey there, and welcome to TechStuff. I'm your host, Jonathan Strickland. I'm an Executive Producer at iHeartRadio, and I love all things tech. And today, we're kicking off a series of special episodes, called Smart Talks, some of which we'll publish here on TechStuff, and some that we'll publish over at Stuff to Blow your Mind.

Now in this series, we're sitting down with people who are leveraging powerful technologies, developed by IBM. And they're doing so in ways that are making a real difference out in the world. So, these are people who are implementing technological solutions to real world problems.

JONATHAN STRICKLAND: Today's episode is about Project OWL, an idea that won the very first Call for Code Competition in 2018. Project OWL aims to restore communications and logistics capabilities to areas that have been affected by natural disasters, specifically hurricanes. It's a combination of hardware and software that leverages mesh network technologies and IBM platforms such as cloud computing, the Watson Platform, and more.

We'll hear from Bryan Knouse, one of the co-founders of Project OWL, as well as Alisa Maclin of IBM to get a deeper understanding about the project, its origins, and the long-term goals, both for the call for code initiative in general and Project OWL, particularly.

Now I wanted to start the sentence that I'm speaking right now with the phrase, the remarkable thing about this story is, but it turns out I can't because there are too many remarkable things to pick just one. Here's a quick rundown. First, IBM has built a suite of technologies that are incredibly powerful. They range from cloud computing Services, which give developers the opportunity to take advantage of enormous processing power.

Cloud computing is where you've got servers that are able to do processing on the backend, and through an internet connection you can access that. So, you don't have to have a supercomputer at your own disposal. You've got a virtual supercomputer in the form of these cloud computing Networks.

But they also have the famous Watson Platform, which allows for an incredible range of AI applications that can tap into all sorts of different processes. So, you can think of this as a suite of refined extreme computing power. It kind of makes me tingle just to think about it.

Now second, this story is about taking those tools and actually applying them to solve real world hard problems. It's not just that the tech enables cool applications, it's that like-minded people who want to make a positive impact are finding one another, and they're coming up
with novel approaches to tackle these issues that affect millions of lives every day focuses on really big challenges. The theme for the first Call for Code in 2018 was natural disasters. The theme for this year is climate change. So yeah, they go big.

The third thing I find really remarkable about this story is that collaboration keeps popping up as an important component in the projects. Not only are individuals collaborating with their teammates, but also with subject matter experts and IBM professionals with deep knowledge and experience in the technologies and the company's products.

So, while we're looking at a competition framework, I mean it is a competition, the spirit of working together permeates the entire process. I sat down with Bryan from Project OWL, and I asked him to walk me through his own background and how he found himself participating in the 2018 Call for Code.

**BRYAN KNOUSE:** A lot of times I think about how lucky I was to grow up at this particular time, in this particular place, to have the resources I have and the tools at my disposal. And what I mean by that is growing up, a few things really inspired me to play around with digital technology.

One was the accessibility of personal computers and the growing accessibility of the internet. One of my proudest moments is I think the first real sophisticated thing I did with a computer when I was about eight years old is illegally downloaded music with Napster. And despite the concerns my mother had about that, I think it kind of opens your mind to this understanding that there's so many things you can do with digital technology, that previously just weren't possible in other industries or professions, careers, particularly things that are limited by the physical manifestations of the world around you.

Of course, software is kind of in a way I like to say it's like, not real, right?

**JONATHAN STRICKLAND:** Right.

**BRYAN KNOUSE:** It's just code. And similar experiences too, I'll never forget what pushed me to learn how to code originally was the first time I played, admitting my favorite video game of all time was the First Halo. I mean, did you ever play that?

**JONATHAN STRICKLAND:** Oh yeah, no, I'm a big Halo marathon fan, so.

**BRYAN KNOUSE:** First of all, amazing game. Right? But I'll never forget when I played that I thought to myself, "This is the coolest thing I've ever seen. I need to learn how to make things like that." So, I bought this book called, 3D Game Programming All in One. And it was like a thousand pages long, and it's all about C++ video game development. So, I thought to myself, "Yeah, how hard could that be?" Little, did I know.
But that was really an introduction. And those kinds of experiences that I mentioned from Napster, playing video games, thinking about how to build them, that was just kind of what I was doing through my teenage years, and then even through college. And while I studied mechanical engineering, I struggled to find an outlet for the work I wanted to do.

I distinctly remember applying in six months prior to my graduation and the six months after applying to almost a hundred jobs. And these weren't just random things. They were within my area that I studied. I graduated with a degree in Mechanical Engineering in four years from the University of Rochester. And of all those applications, you know how many jobs I got?

JONATHAN STRICKLAND: How many?

BRYAN KNOUSE: Zero.

JONATHAN STRICKLAND: Yikes.

BRYAN KNOUSE: So, it was kind of at that moment that I was like, "Well, you know what, if this isn't going to work doing a traditional route, I might as well just do something I'm really passionate about." And that's when I kind of dug back into coding a little bit and really peeled back the curtain on the hackathon environment.

And now I'm sure you're familiar with the hackathon or the mechanics of it and probably most of your listeners, but for those who aren't, a hackathon's kind of just like a coding competition, right? And companies from all over the world will put on these competitions from IBM, certainly to Porsche or Mercedes-Benz to the U.S. Government, the Red Cross, all sorts of organizations.

And typically, what they'll do is they'll say something like, "Hey, if we're a car company, we've got this new API for you to build on our vehicles. And what we want you developers to do is come up with amazing stuff, be creative, build out something incredible, pitch it to us and the best ones you're going to get paid."

And for me, this was the coolest thing in the world, because it put together two things that I absolutely loved that I really couldn't find anywhere else. And that's I'm a huge sports fan, Philadelphia Eagles for life. I also played soccer my whole life growing up, and I'm a big fan of the sport. And I love that like innate competitive spirit. It's just, "Get on a field, and go and compete, and try to win."

But I also love going back to the Halo, Napster thing, just that interest to just build stuff, come up with ideas. And a hackathon, I think, is so unique because it's able to put those two perspectives together, the interest to just go compete in an environment, at the same time, do that while building incredible creative technology.
And so, for a few years I was hopping around going to hackathons, and I kind of got a groove in it, started winning a few, making a career out of it. And one thing led to the next, and then of course competed in the IBM Call for Code, and here we are today.

**JONATHAN STRICKLAND:** I asked Alisa Maclin of IBM to give an elevator pitch to kind of explain what the Call for Code is all about,

**ALISA MACLIN:** A little bit about the background for the Call for Code Challenge, and one of the things that, of course, is really important is understanding what matters to developers. And an interesting insight on developers is that 80% of them code as a hobby. They work on coding projects at night, in the morning, over the weekends in their spare time.

And they are inherently problem solvers. And for most software developers, coding isn't just a profession, it's really a passion. And one of the things that we really see also with developers is a tremendous use of open source technology. And I think an interesting insight there is that developers enjoy working together, collaborating on projects which open source helps them to do.

So, in thinking about, how can we help developers with their love of side projects, of learning and developing new skills, combined with the desire, which is a long-held focus of IBM's is using technology for good, creating innovation that matters. Taking the technology that we have with artificial intelligence and blockchain and others and using those to really help make the world a better place. And we know that the majority of developers are interested in the same thing. Most of them, as they're working on side projects, are doing a work that has societal benefit. So that's really what created the spark behind the Call for Code Challenge. We wanted to see what 24 million developers around the world would do if they were given access to the technology and also an understanding of some of these major world problems that they could help tackle. So, with Call for Code, where IBM is the founding member of Call for Code. we launched it in 2018 with the David Clark Cause and the Linux Foundation and working with the United Nations, and we wanted to focus on something that was really, really important each year. The Call for Code is a five-year program, and we started with a focus on natural disasters. We knew that this is an area that was an increasing problem around the world, and one that technology could help. So, in launching Call for Code in 2018, we asked developers to create solutions for these problems, and we were really blown away by the amount of engagement that we got. We had over 100,000 participants in that first year creating some 3000 software applications.

Over the last two years, it's grown tremendously. We had nearly 200,000 participants in the Call for Code 2019. What I think is really amazing too is we had participation across 165 countries. So, we saw a tremendous amount of interest and engagement in creating these sustainable solutions, leveraging open source and working not just developers, but typically teams that involved other experts. We had teams that were comprised of developers and first
responders, medical professionals, students, groups coming together to work on an amazing array of problems.

**JONATHAN STRICKLAND:** As Bryan would explain to me, Call for Code stands apart from other hackathon type events.

**BRYAN KNOUSE:** The IBM Call for Code focused on originally and still to this day, natural disasters, although they've segmented it and focused even a little more under that umbrella. I believe this year's competition is focusing on climate change, certainly has implications to natural disasters. One of the things that I think it's really interesting about the IBM Call for Code, apart from all the other hackathons I've been to, and one of the things that left me frustrated with the hackathon community was, you go to these events, sometimes they're smaller, sometimes they're larger ... The Call for Code is certainly a huge hackathon. But even, there are other in-person events. Call for Code's three month long, or at least it wasn't 2018. It might be longer or shorter now. Three month long virtual thing, so you can compete from anywhere.

But some of the hackathons I would go to, we would put 300 developers in the basement of the Venetian Hotel in Las Vegas for a weekend, and you'd compete on different enterprise, hackathons like PayPal would be there, Visa, some other folks. But what's so unique about Call for Code and what I think really sets this apart is their commitment to actually see the solutions through. I think that's really important, particularly within the context of natural disasters because so often, at these hackathons, you'll see great ideas or at least that first nugget of an idea. If great ideas are 1% inspiration and 99% perspiration, a hackathon is like that 1%. The part of Call for Code that really impresses me and the reason I think it's the most important technology competition in the world today is the commitment to see the work through in the end to actually make an impact in the world, and I think that has fostered an exceptional community of developers.

**JONATHAN STRICKLAND:** Like I said at the top of the episode, one of the interesting things about Call for Code is that it brings together two seemingly opposing philosophies, cooperation and competition. Well, Call for Code is structured as a competition. There's a deep culture of collaboration throughout the program. Alisa of IBM explains.

**ALISA MACLIN:** One of the things that really struck me early on is that the open source movement is about collaboration and about achieving more faster by working together, and that also means that projects live on in open source. So, one of the things that really inspired me hearing from I participated in a number of hackathons around the world. We do about 600 of these every year, bringing developers in different parts of the world, working with them together on their solutions. What I found incredibly motivating hearing from the developers is that they were excited that this wasn't a one and done. It wasn't a 24-hour hack where they come in, they create things, great ideas and then they move on. These are projects that can live on and be built out by developers around the world.
That was also the thing in working with the United Nations that they really keyed in on, because there are so many software-driven solutions that don't require a lot of infrastructure. Even things like early warning systems for tsunamis, as an example, if you have access to the weather data, if you have a smart phone, you can get those alerts which save lives. This is the type of thing that someone could create working on open source in one part of the world and then it could be adapted for conditions in different parts of the world by software developers, virtually any place anywhere somebody has a laptop and a smartphone, and the UN was particularly interested in this because they saw the long-term sustainability.

In terms of how the developers are working together, I think this is one of the things that really inspired me, as well over the last two years, is seeing how these groups come together, and I think Bryan may have told you that the five members of the team that he's on with Project OWL from different parts of the United States in different parts of the world and they met on a Slack channel. We saw a lot of that. The winner of Call for Code 2019 is a team that are Barcelona based and there's a firefighter, a nurse, and three developers. So, it's different groups collaborating together, meeting each other, either in person or hackathons, or meeting virtually working on these open source projects and they continue to collaborate over time.

**JONATHAN STRICKLAND:** Bryan confirmed what Alisa was saying, describing how his team came together during the first Call for Code.

**BRYAN KNOUSE:** I had known a few of the co-founders previously. We'd competed against each other at hackathons. But one of the founders of Project OWL, Magus Pereira, him and I had just met in a Slack. I think digital technology and meeting someone on a message board in Slack might seem like a little weird and different, right? You didn't run into this person in a room, you didn't meet at a conference, but it's really effective at putting people together who have like minds, like skill sets and like ambitions. I remember Magus had just posted some message about what he was interested in and we connected and had a call, and I still to this day distinctly remember his creativity, passion and interest in building unique solutions. So even after one call, we just agreed like, "Man, we got to work together. Well, how can we facilitate this? Because I think there's a lot you have to offer that you're interested in and I feel the same way about myself and the team we've already put together."

I don't know if they still use Slack or, how they do the collaborative piece, but the environment of all these developers showing up to just want to build something great while still inspiring people to have that competitive nature, I think is a really fascinating and IBM puts on Call for Code satellite events, so mini hackathons, all over the world during the main overarching event. This is another way to plug in, try something quickly, see if you got an idea for the bigger competition and meet other technologists in the environment to really make this happen.

**JONATHAN STRICKLAND:** This brings us up with what Project OWL would specifically focus on, which all revolved around hurricanes.

**BRYAN KNOUSE:** Specifically, with regards to what we did, as we were going into Call for Code, we uniquely are ... Our original team of five was spread out across the United States. Still
is today, but we were at the time Charlie was in Houston, Magus in North Carolina, myself, Taraqur, and Nick in New York City, and very recently at the time, Charlie in Houston had gone through Hurricane Harvey, a massive hurricane, caused a lot of economic damage in Houston, and Magus, during the competition went through Hurricane Florence. At the outset, we felt, well, it seems like hurricanes are probably a pretty good thing to try to approach here. These have been quite a problem, and not only are they a problem that hits close to home for us because we've all been through them. Even still to this day, I'm here in New York City. I live on the L train. Hurricane Sandy ripped through New York City in 2012, and they're still shutting down the L train to repair it. We're like eight years later.

JONATHAN STRICKLAND: So, the team knew what type of natural disaster they wanted to focus on. Next they thought about how technology could help people affected by a hurricane.

BRYAN KNOUSE: So, in the absence of being able to stop these natural disasters, we really couldn't do that yet and we can't do that now. It may be in the long run. There might be something there. But in the absence of being able to stop them, we felt, "Okay, well, what we can do is enable people to prepare for and deal with them as effectively as possible," and the first obvious problem is that when a hurricane rips through, it destroys everything, most notably the infrastructure to provide organization, whereabouts and logistics in a community. That of course is partly where the name OWL came from. Our focus was really that if we could find a way to quickly, easily, and cheaply bring back communications in a place that either didn't have it or lost it, that could be a really advantageous solution to these communities.

JONATHAN STRICKLAND: While the immediate devastation left behind after a hurricane is plain to see, Bryan's team knew that the challenge lasts longer than a day or a week following a hurricane, and that communications plays a vital role in that timeframe.

BRYAN KNOUSE: I think it's important to note that particularly for hurricanes, though I suspect this is true for most other types of disasters too, the majority of the death toll and the economic devastation occurs not during the immediacy of the wind and the rain and the thunder. It's the long tail after, when you don't have roads and physical infrastructure. You can't get to places, you can't get to love the ones, you can't get to medical facilities. When you don't have communications, you can't effectively coordinate to put the right resources in the places they need to be, nor can you contact loved ones to see if they need help. Elderly can't get medications they need. Food is not adequately distributed to the places that need it most based on who needs what. When these infrastructures go down, particularly the communications piece, you lose the ability for society to function on the plane that we are currently accustomed to.

JONATHAN STRICKLAND: Their solution was twofold. Create small, durable and inexpensive network hardware that responders could rapidly deploy in a region, and a software platform that enabled the communications and other operations across that network. The idea seems
simple and elegant, but as Bryan and his team discovered, achieving that goal in the real world is a bit more complicated.

**BRYAN KNOUSE:**

To be clear, there were a whole host of challenges at the outset and there are many, many more today that we still need to solve. A phrase I like to use that never seems to connect with people, but I feel is appropriate, is maybe you've ever heard the saying, it's turtles all the way down. I think it came from a joke in physics that some person said to a physics professor, "You know, the earth isn't in space. It's just sitting on the back of a turtle." And the professor says, "Oh, okay, sure. So, what's that turtle sitting on?" And the person responds, "Another turtle." The professor goes, "Okay, and what's that turtle sitting on?" And the person goes, "Oh, it's turtles all the way down."

My point and the reason I think this is appropriate ... Maybe I'm crazy, maybe this has no relationship. But the point is as we are peeling back the covers, I'm ... By the way, I'm not an IOT guy. Sure, I'm a technologist. I can code, I know how to use Google, right? Which interestingly is like one of the core skills a programmer needs to be able to have, how to learn new technologies and figure things out.

Nobody on our team was a professional IOT developer. Even when we started developing, I would be writing code for the firmware, but I didn't actually know what language I was writing in. I just knew that it worked. None of us had a professional background to do any of this. So if anybody's thinking like, "Oh, you need some college degree or a career", no, none of that and how this relates to my point of it's turtles all the way down, we started peeling back the covers on, "Okay, well, what if we could get this little IOT device to connect to another device? That'd be pretty cool." We'd peel back the cover, play around with the radio technologies. We used LoRa 915 MHz here in the United States.

When we'd solve one problem, we make a little progress. We'd say, "Oh, that's really cool. All right, what if we could do the next thing?" And that's when we'd realize, "Wait, we need to write a codec for this radio." "Oh geez. Okay." All right. Well, once we figured that out, all right, these LoRa radios say they can communicate over two kilometers. Great. Well, wait a minute. When we went down to Puerto Rico, they're only working at 200 meters. What the hell is going on? Oh, heat and humidity are a big problem and foliage and metals, just like another huge problem for radio. Every problem we would find and then address, there would just be five others underneath it that we added to solve. I feel another way to think about this technology challenge is it's a game of Whack-a-Mole of infinite size. You keep knocking down a couple moles and a few more keep popping up. What's impressive and I think shows your ability to execute, is how quickly you can move through the field.

On the IOT side, it's hard. People say hardware is hard. It's true. Also, you have to consider in the long run the business case for hardware you're making. For Project OWL, this was a
discussion and a conversation we thought about for a long time because we understood and still do to this day that ultimately if we were ever to make a dent in the market, I think we've got a great brand. I think we have incredible nomenclature and design. But at the end of the day, if Intel looked at any of our stuff, they could say, "All right, let's put 10 engineers on this and give them five million dollars and they'll have a better, faster, cheaper product than you and we will put you out of business tomorrow. Oh, and we also have economies of scale for manufacturing." So, you need to consider how can this integrate into a business model because that's what it enables you to work on this for a very long period of time.

JO NATHAN STRICKLAND: Project OWL faces many challenges both from technical and market standpoints. Not only must the team build a working system in which custom built hardware and software work together, they must also find a way to make that a sustainable business. The hardware side of the project presented many challenges as the team worked to create a working mesh network infrastructure that was durable, deployable and cost effective. But I wanted to know more about the software side and how Bryan's team tapped into IBM's Watson platform and cloud computing capabilities.

BRYAN KNOUSE: Yeah. There were two halves to our solution and they nicely fit under the hardware and the software. Both of them enabled a unique capability that I think together made one complete solution. The way we looked at it was the software provided an incident management of sorts, a way to leverage data, perform analytics, do some intelligent things. And then the ducks, of course, provided the way to acquire that data on the ground, particularly in locations that don't have any infrastructure. Electricity's gone; communications are gone. The software has changed a bit from the competition. We pitched a lot of stuff that we found people just didn't care about it, our customers, clients, partners didn't care about, but we thought it was really cool.

But when we originally pitched the solution, I think at the time there were 12 IBM Watson APIs and we had incorporated every single one of them into the solution. These things were like text to speech, speech to text. There are some others like knowledge catalog, pattern recognitions. I don't remember all of them by name at this point in time, but they could enable you to do certain things. Whether it was speech to text being kind of giving it an intelligent feel. And then some other things like Watson Discovery, giving it more of an intelligent brain. These APIs enabled you to act on data, to do certain things that would be very hard to write code for yourself, you know, natural language understanding.

We had a conversational assistant you could just talk to for the OWL. At the time it was the IMS. It's now the data management system, the DMS. When someone would write in a message like, "Hey, can I create an incident Hurricane Florence and it's in North Carolina?" We could run natural language understanding and that would pull out things like the name, locations and tell us other interesting things this person said versus of course, if you're just writing raw code to do that, you're never going to be able to match the sophistication of a cloud software tool like this. IBM Watson was great in that it provided all these cool APIs you could play with.
I think one element that IBM did exceptionally well was they have these code patterns that you can just go online, pull down, play with. If anyone listening has ever written code before, you know that one of the best ways to learn is to just find someone else's code and screw around with it or adapt it to another use case. IBM is a great ecosystem of not only the services to build with, but the tutorials and the information to say, "Okay, this sounds cool. Can you just show me how to make something with it?"

**JONATHAN STRICKLAND:** As Bryan mentioned, Project OWL tapped into IBM's Watson platform in several significant ways. Watson is a suite of services from IBM and it leverages artificial intelligence to an incredible extent. It's designed in such a way that developers can tap into these powerful processes without having to build everything themselves. If a developer has an idea for a cool application that would lean heavily on something that's traditionally really hard to do, like natural language processing, having a computer understand what we mean when we communicate the way we typically communicate, not as a machine would, but as a human would.

Machines are not naturally good at that. You have to really work hard to make them understand. Well, most developers can't do that on their own, but they could lean on a Watson API, that's the application programming interface, to handle that part of their service. They can focus on whatever it is the app is supposed to do and the natural language part can be handled by the Watson platform. Now I think most people who have heard about IBM Watson, think back to the system's famous appearance on the television game show Jeopardy. Watson acquitted itself pretty well on that show. It won the competition.

But it turns out it's much more than a trivia answering machine. In a way you can think of Watson as access to an array of AI capabilities. The Project OWL team attempted to take advantage of every single one of those. Interestingly, as Bryan mentioned, they found out that in the real-world people didn't necessarily use all the features Project OWL had included in their service, and this helps illustrate another big challenge facing any developer or engineer. What seems like a brilliant idea in the conference room or as was the case with Project OWL on the Slack channel, may not translate in the real world.

It's not that the idea itself is bad necessarily, but rather that it's less applicable than the designers first thought. So, some ideas might turn out to be best suited for other applications in the future. The development process at Project OWL continues. The original design relied heavily on individual mesh network devices the team called ducks because like a rubber duck, they were meant to be small and capable of floating. The individual ducks linked together through a hub unit called a mama duck, which can then send information over to an infrastructure component called a papa duck. That links the mesh network to the internet at large. But even without papa, the mesh network itself can provide onsite communications and logistics support within the region. The team's pitch won the 2018 Call for Code, and I asked Bryan what that actually meant on a practical level.
BRYAN KNOUSE: Of course, there was a monetary prize and that kind of enabled us to not focus on other things. There was also a lot of support from IBM. So, a commitment from IBM-ers to help us see the work through. And one of the biggest manifestations of that was the corporate service corps deployment. So, in March of last year, five handpicked IBM-ers from around the world... If I’m recalling correctly, they were from the United States, Canada, Israel, the United Kingdom... Yeah. That's total, two of them were from the United States. And this group was exceptional with a diverse skill set of talents and being able to go to Puerto Rico with our very rough hackathon project and have five expert IBM-ers there with skills in... from branding and communications to design, to engineering, to backend to front end software development. This really kicked our solution into high gear, not just with the work they produced, but also, I have vivid memories leading up to our Puerto Rico one, as we called it, deployment, last March, our first official deployment to Puerto Rico.

I have memories of hopping on our screen shares in the mornings and hearing the corporate service corps members saying to us like, hey, Project OWL, none of your stuff works right now. We were like, yeah, you're right. Because again, a hackathon is about pitching an idea, and anyone who's been through a hackathon understands that you don't have a fleshed-out enterprise product. It's just not possible to do that given your time constraints and resource constraints. So, if you can cobble together the idea, you can kind of work out the kinks if people take an interest in it. And I think the commitment of IBM to provide that support, to provide experts, to provide the help in the field as you're deploying is a huge benefit to your fledgling organization's ability to scale and grow. And so that for me was the in a way was the most valuable part that we didn't really even consider at the time when we had won. But in hindsight it was really exceptional.

JONATHAN STRICKLAND: Project OWL continues to refine their technology and approach, taking the experiences they've encountered in the real world and using them to create a more focused approach to achieving their goal, creating a nimble, robust and effective communications platform using custom built hardware and software. Winning the competition wasn't the end of the line, but just the beginning. Bryan told me about the next steps for the project as it strives to achieve the goals of the co-founders.

BRYAN KNOUSE: So, what's next for Project OWL? We really, after the competition where we had spent a lot of time focusing on engineering and design, product development, coming up with the idea, executing on that. To enable your organization to work on a challenge over a long period of time, you have to build a sustainable business model. So, we've spent a lot of time and energy. If I was engineering software in the beginning, I'm spending a lot of time engineering the company now. So that's... How does Project OWL fit in the market and who do we service? I had mentioned earlier our incident management system, we had all these capabilities in it, a whole bunch of different things it could do, and we realized that the people we wanted to go, and support were only really asking for one or two of those things.

So, like 90% of them, we can just get rid of them, and that's a realization that took a long time to come to, through a lot of conversations and a lot of trial and error. And so, I spend a lot of
time thinking about working with the team, certainly still on technology. We’re all nerds here and this is what we love to do. But a lot of times thinking about how we can put Project OWL in a position to succeed over the long term, because if we do that then that enables us to think about some of this other crazy technology.

**JONATHAN STRICKLAND:** When Bryan and I first talked way back at IBM Think 2019, the Project OWL team was mainly working with the basic duck units, and they were meant to be spread over a region on the ground, mostly. But things have evolved a bit since then and the team has come up with some more variations on this basic technology design that they hope to develop further.

**BRYAN KNOUSE:** We have a whole variety of ducks, many more so than when I last met you. We have detector ducks, we have disco ducks, we have cluster flocks, we have duck ducks, we have space ducks, and there are many others. You can actually go to our open source firmer, clusterduckprotocol.org, and there it lists a bunch of the different duck variants. You can check out some of the other ones we made. But space ducks was a project I did in collaboration with some engineers at Cal Poly, to put a duck on a very large helium balloon and send it up to a hundred thousand feet to acquire sensor readings and try to transmit and see if it would break.

And so, these are kind of... That early seedlings of ideas, not only... Sure, we got great photos from that, but I’d been thinking to myself too, and the rest of the team, I should say Project OWL had been thinking, putting this communication stuff on the ground is great, but what if a hurricane comes through? It kind of just destroys everything, so we’re going to lose a lot of stuff. Well, okay. Where could you put communications things where they would stay there and even if a hurricane rips through, they would still stay there? Well, interesting space might work. Oh, and by the way, you know what environment radio frequencies work really well in? A vacuum. So, the more we were thinking about it, the more we were considering, hey it might be worth our time to just start throwing some space ducks out into space and see what catches, because there might be a real long-term opportunity here.

**JONATHAN STRICKLAND:** Bryan and his team continued to develop Project OWL, to move beyond a testing phase and into fully fledged deployments and implementations. Meanwhile, the 2020 Call for Code is underway. I asked Bryan if he had any words of advice for competitors in this year’s Call for Code.

**BRYAN KNOUSE:** I would encourage any developer working on solutions in the upcoming competition to think about how you can distill a concept down to its fundamental atomic parts. Because if you can do that and you find the right atomic components, like a proton, neutron and electron, what you find is, oh wow, actually people can make a whole periodic table of elements with these things. So, I think from Project OWL’s perspective, we’re very fortunate. We certainly didn’t know this at the time, but the original duck link we’ve now developed into a whole host of variants for different use cases, but still leveraging that same fundamental core.
And I think, similarly goes for the software side in the cloud that our data management system is really slimmed down and simplified to the core components. Getting data in, seeing what it is, having dots appear on a map, and then API-ing that data out if you need to put it in external systems. So those core components are like that fundamental atomic nucleus that we're still just in the early days of this, but we think can inspire a lot of folks to solve problems around the world in unique ways.

**JONATHAN STRICKLAND:** I want to thank Bryan and Alisa for joining me on this episode of Smart Talks, and this is just the first in this series. You'll hear more conversations with people using technology to make a positive impact in the world very soon. Episodes we'll publish here on TechStuff, and also over on Stuff to Blow Your Mind. You'll learn about how some really smart people are changing things for the better in incredible ways. So, make sure you catch every episode. You know, I talk a lot about tech on this show and sometimes it's easy to get lost in how tech works, and you lose sight of why it's important. Competitions like Call for Code and companies like Project OWL remind us that these powerful tools can bring about incredible change and help those who need it most.

I have no doubt I'll be talking more about Project OWL in the future, describing how it helped communities see to vital functions despite natural disasters, and I can't wait to see what the participants in Call for Code 2020 come up with as they tackle climate change. Make sure you check out the other episodes in the Smart Talk series as they publish over the next few weeks, and if you have any suggestions for future episodes of TechStuff, feel free to reach out to me on Twitter or Facebook. The handle for both of those is TechStuffHSW, and I'll talk to you again really soon.

TechStuff is an iHeartRadio production. For more podcasts from iHeartRadio, visit the iHeartRadio app, Apple podcasts, or wherever you listen to your favorite shows.