EPISODE 1131

[INTRODUCTION]

[00:00:00] JM: Archive collects historical records of the internet. The Wayback Machine is tool from the Internet Archive which you may be familiar with. One project you may be unfamiliar with is book scanning. Internet Archive scans high-volumes of books in order to digitize them. In today's episode, Davide Semenzin joins the show to talk through the history of the internet archive and the engineering behind book digitization. We talk through OCR, storage, architecture and scalability.

I want to mention that we're looking for writers and podcasters for Software Engineering Daily. If you're interested in being a podcaster or a writer, email erica@softwareengineeringdaily.com.

[INTERVIEW]

[00:00:47] JM: Davide, welcome to the show.

[00:00:49] D: Thank you.

[00:00:51] JM: You work on the Internet Archive. What does the Internet Archive do?

[00:00:54] D: That's a great question. The Internet Archive is the world's largest digital library, and whereas most people may know of us because of the Wayback Machine, which is this really rather neat tool that allows you to go back in time and kind of see what pages used to look like. We really are a fully-fledged online digital library. And that's that. We have different types of media types. We hold texts and television, and audio, images, movies, all sorts of things. Yeah, the Internet Archive you can think of as this huge repository attached to the internet.

[00:01:31] JM: When did you start working these?

[00:01:32] D: I started here in 2016. It's been, yeah, 4 years.

[00:01:37] JM: And what do you work on there today?

[00:01:39] D: Well, I work on the books. That's mostly what I've always been on. Spending the bits inside of this. So, usually when we think about our media types, we think of them in terms of bits-in and bits-out. How we procure them and how we distribute them. My specialist is working on the book bits-in. So, in order to build up our collection of almost four million books, we have to scan them. My job is to sort of keep running the whole pipeline that allows us to do that. Over the last four years with my team, I built it. And now we achieved our objective of being able to digitize a million books per year, which we're doing. And it's been a pretty interesting challenge so far.

[00:02:23] JM: So, you work on book digitization. And I want to talk about that. But first let's talk more about the Internet Achieve at a high-level. Can you tell me about what is being stored across the Internet Archive and who pays for it? And how do people use it? Just share a little bit more about the Internet Archive.

[00:02:41] D: Yeah, that's a great question. So, I'm going to start from who pays for it, because I think there is a lot of depth in that question. The Internet Archive, if you think about it as a repository, it's just essentially a bunch of hard drive spinning connected to the internet. Somebody is got to pay for both the internet connection and the hard drives and the electricity and all of that.

Largely, you can think of our revenues in three different ways. So, we're a nonprofit and we don't really run for-profit businesses. We don't benefit in any way of the data that comes on our servers. We do benefit from your donations. And so, by and large, we are a community-funded effort. And so if you go to achive.org/donate, we actually just added the integration with Apple Pay. So if you want to help us test that, that'd be great. We receive a fair amount of the money that we need to run from patrons and just like people who support us.

On the side, we do have some businesses. We have our Archive It. Digital arm where essentially we contract out our Wayback Machine capabilities, and we are maintaining a very large amount of curated website collections. In fact, I think we have about 700 organizations

that are partnering with us to create these collections. A few tens of billions URLs that have been collected for our partners. And so they pay us to do this service and we do it for them.

The same is true for books digitization. So, as we have built up the large infrastructure that is required to do this these kind of tasks, we have to an extent the ability to contract that out to third-parties. So, we do get some revenue streams that way. Not anything particularly substantial in terms of like our ability to sustain ourselves, but every little bit helps. And then, obviously, throughout almost 25 years of our existence, our founder, Brewster Kahle, has chipped in here and there and like a significant amount, I want to guess, over the years to keep us running. So we have donations. We have a little bit of our non-for-profit business, and then we have Brewster, who is there.

So this is in terms of who pays for it. But the other question would be I guess who benefits from it, right? And that's a very, very large segment of the internet. We're not the biggest websites on the Internet. I think we're ranking about 200 and something on the Alexa rank. But since we've been around for a fairly long time, the users that love us, they love us. Every day, I am in contact with people who tell me their story about how they use the Internet Archive for their specific needs. And I'm always amazed by the depth and the breadth of the use cases that our users bring to us.

So, it expands from teachers, to researchers, to journalists, to lawyers. There is a very, very large diversity also in terms of like the countries and backgrounds from where our users come from. So, it's kind of hard to paint them with the same brush. But in general, I want to say they are people who have some degree of love for knowledge. And you may know, our motto, our slogan, our mission is Universal Access to All Knowledge. And so I guess people who have an interest in that eventually land on our website.

[00:06:02] JM: Okay. Well, let's talk about book digitization as a particular project that is under the auspices of the Internet Archive. What is book digitization?

[00:06:11] D: So, book digitization is the effort of transforming physical books into digital artifacts. So, that's the definition. It can take different forms. If you have a scanner in your home and you're scanning a document, in a way, obviously, that's digitization. If you should take

pictures of a book, that's book digitization. So, the definition that needs to be applied to the use case at-hand.

There have been other efforts at large-scale of books digitization. Famously, Google had one. By their way of doing it was slightly different from ours. For instance, where they did destructive digitization. So they would pull the spines from books and turn that process into a sort of essentially sheet feed kind of prop problem. We do nondestructive book digitization. And I think like the nondestructive bit, it's just every little bit as important in the definition as the fact that we're digitizing books, because we're digitizing them so that we can keep them, so that we don't destroy them. It's the process by which we turn books into bits, and then we return books to wherever they came from, or where ever they need to go.

[00:07:23] JM: So, why would I want to digitize a book? And how many books get digitized each day? Just tell me more about the volume that's going through this.

[00:07:33] D: Oh, I'm very happy to answer this. So, the reason why you would want to digitize a book, there are multiple. So think about, for instance, the first thing that comes to mind is obviously preservation. A famous [inaudible 00:07:44] is that accessibility drives preservation. So, if you don't have something, it's almost like it doesn't exist. Especially in this age of information, we do have immediate access to all of these resources. If you actually think about this, if you have to go to the library to procure a certain book, chances are you won't. And if the record of the book actually doesn't exist, you may never get to it.

Where this is a problem is for all of this huge amount of books that were printed in the 20th century, for which there is really no digital equivalent. Books nowadays that are published like currently, obviously, they have e-book artifacts. So that stuff is not going to get lost. And that stuff is searchable and it's reachable. But we have tens of millions of books that are just unaccounted for. And as time progresses, they're getting lost. And if somebody doesn't save them, they will be lost forever, and that would be a pity, and a huge loss of human effort.

Well, first of all, I think it's important to scope the problem. And I think in the estimates, that there is about 100 million books out there, give or take, unique books. And scanning them – We're probably not going to scan all hundred of them, first of all, you would be able to source it, and

that's by far the hardest thing. So, we try to scope down the problem and try to figure out, "Okay, how can we do this in the way that is useful for people?"

So, first of all, I think we had to come up with a list of books that we wanted to scan, that we knew these are books that are important and we need to scan these first so that we will get them to people, and this would be evidently immediately useful. And a good place for us to start was Wikipedia, which has collected a long list of ISBN's the were commonly cited in Wikipedia. Compiled the list and came out to a few hundred thousand books.

And so whenever we come up one of those in our sourcing process, we make sure that – We can talk about some resourcing process and a little bit later. But in general, we do have a little bit of a concept of priority, or at least we did. This was for the first million, million and a half. And then the problem was that we started running out of books. You would be surprised how hard it is to source books by the half a million. And if you do it by a smaller scale, it doesn't really makes sense to us in terms of maintaining our economics of scale. So, the whole system works only if you scan a huge volume and all the time.

And by huge volume, we're talking about a million books a year, which is about 3,000 books a day. Some days, we'll do 35. Some days we'll do 25. On the seven days week, it averages out to about between 20,000 to 25,000 books. Every book is about 300 pages. So that comes out pretty neat about a million pages per day, five to seven million pages per week. And that's not a huge amount of data in total. I think like last time I checked, it was about between 10 and 15 TB of data a week. So, we're not talking about huge amounts, but it's not a small amount either. And we can talk about the challenges of typing that data over the Internet in a reliable way later. But it's a significant volume, and this operation is running 24/7.

And so in terms of why even do this. So I covered the first part, which is obviously people want to get to the books. There is a second benefit in having digitized books, and that it's a wholly new format. It allows you to interact with the body of knowledge in a way that you never have before. If you have a physical book artifact, it is has some very desirable properties, for instance, very low random-access time and doesn't depend on a battery. It's very, very hard to censor. And these are not properties of a digital artifact, but digital artifacts are searchable.

In fact, we have this pretty amazing full-text search engine where you can instantly search all 40 million text items that we have. So, that's the 4 million books, plus all of the patents, papers, all sorts of stuff. And you can search that instantly. That was just not possible with the previous format. So, I don't think this is a dualism in any way. I think books in their digital format and books in their physical format will continue to coexist. They just help each other out. In fact, if we are able to digitize them in the first place it's because of the properties of physical artifacts that they don't just disappear. If we find one, we could scan it.

[00:12:42] JM: Well, that was a great summary of what you do. And I can tell how excited you are about it. Let's talk a little bit more about the high-level, and then we'll get into the engineering. So, can you describe the steps of digitization in more detail? If I have a book, how am I digitizing it?

[00:12:58] DS: Yeah. So, the books digitization pipeline is pretty simple, and it's like in a way, like if you're an engineer, and I think it's kind of what you expect. So first, there is a physical sorting step where your book is ingested into the system. It's given an ID, and it's placed in a container. So we know that it exists, so to speak. The second step is it gets to a scanner. The scanner picks it up, puts it in the machine, loads up the data as necessary. Whereby data, I mean the books metadata. We're going to have to talk about that, I guess. It's pretty interesting facet of it all.

And then they perceive to actually scan it, which means they turn the pages, page by page, and they take pictures of the pages. Once this process is done, they click upload and the book vanishes into the ether. And so at this point, we have a fork, the digital artifact goes into our servers. The physical artifact either goes back to the person who gave it to us in the first place, or it goes into our warehouse. And this largely depends on what kind of book it is.

Obviously, there is a larger conversation to be had about copyright and like what books. Is it okay to scan and under what guise it is? But suppose, we are just scanning your book, Jeff, and you just wrote your own book and you want to have it digitized. There is no claim on it. You just want it back at the end. So, after we're done scanning it, we're handing it back to you with a slip inside, which will tell you the Internet Archive identifier. And the identifier is just the name of the

item on the Internet Archive. Everything is an item. And you are just going to go to archive.org/details/ your identifier. And a few hours later, you will find your book.

While you wait, the second part of the pipeline is going to kick-off. So, that's the digital server-side stuff. And it's divided in essentially three phases. We have a first phase, which it's a preprocessing stage where we get all of these images that became raw from the camera. We'll look at them. We crop them. We dis-queue them, and we just make sure that everything is ready to go.

There's a second phase of manual review. So, currently, all books that we upload have to be checked by a human for correctness. And so, this is a step where a reviewer just goes through the images and ensures that everything is fine. And then when this is done, they kick-off the third stage of the pipeline, which is the real processing stage where we take all of this files and compile them in such a way that they are suitable for consumption by our web frontend, what we call our book reader. And from there, we derive other – We call them derivative formats, such as PDF, ABBYY, EPUB and other text files, OCR. It all happens at this stage. This is kind of like the bird eye view of the books digitization pipeline.

[00:15:58] JM: Let's start to get into the actual engineering of things. I think the place to start is the OCR. So, if a book is getting scanned, you're going to run OCR over that scanned book to determine what the characters are, right? So I'd love to know a little bit about the OCR process and if you use an off-the-shelf OCR system, or what role that plays.

[00:16:24] DS: Sure. So, OCR, as I mentioned happens towards the end of the pipeline. So, at that point we have all of this images that have already kind of like being scaled down, because the originals are very hyper solution. So you wouldn't want to feed that to an OCR. And they are kind of correct and ready to go. So, we use an off-the-shelf OCR solution. I believe it's called the LuraTech, and we OCR between 1 and 2 million pages every day. This includes both the books that I mentioned, plus like all of the other stuff that gets uploaded to the archive.

And what that is really useful for is creating this ABBYY file format that essentially it's an XML file that contains the position of every letter in the page. And it comes out to be like a huge file. And what we do is we use that to feed it to our full-text search cluster so that what we can do is

whenever you search for some term inside a certain book, we can highlight the exact portion where that occurs. So that's one place where OCR is really useful. Obviously, it's also useful to create sort of text version PDF files of books as well as EPUB's and such. But in terms of the engineering of it, it's kind of a black box for us. We have a very established interface, by which we know what the LuraTech software expects. And we get results on the other end.

[00:17:54] JM: So, the texts gets indexed only within the context of a particular book, or do you have a gigantic index where I can search over all of the books in the Internet Archive?

[00:18:05] DS: Ha! The second. We actually have a gigantic index, and we just recently rebuilt our cluster, and now it's blazingly fast. It's pretty awesome. It just uses like Elasticsearch in the background, and we just maintain this very large index. I think it's about 42, I think, million items. And so you can search everything at once. And then when you are like within the context of the book, obviously, you can scope down the search to that specific book. But I think it's really cool, because it only makes sense when you can search for everything. And that opens up like a huge amount of use cases, because you can almost do kind of free-form search. It's pretty remarkable use case of something that you couldn't do in a normal library. Just, "Hey, give me all of the books that have my name in it." That can be interesting.

[00:18:58] JM: Can you describe the storage architecture in more detail? So, you're stringing all these books. Are they stored on disk? Do you try to keep them in memory? How do you think about storage for all of these books?

[00:19:12] DS: Oh, absolutely. So, storage architecture is kind of paramount to everything we do, right? Because I kind of describe the archive as a mass, and in a way, that's what it is. So, I guess engineering-wise, the overarching principle is to just – It's essentially Unix philosophy of using off-the-shelf tools wherever it's possible. So, all our data is stored as files in directories. And these directories are stored on standard HTTP for file systems.

And we subscribe to a very simple file system philosophy, where other than some of the system stuff, everything is stored in /12345, however number of the disk is on your server. And so we don't do any sort of like striping. We don't do any sort of splicing. Every item just lives as a directory on some disk, and that's where it is. We do have duplication, and I'm going to talk

about that in a second. But the reason why we structure things this way is twofold. One, it's very simple. It's very simple to administer. It's predictable, and it makes platform migration overtime a lot more predictable. We've been keeping this data for 20, 25 year. So, the architecture needs to be simple enough that it can traverse at different stages of technology cycles.

The other reason for it is the disaster scenario. So, it pains me to say this. But, we are based in San Francisco, and the San Andreas Fault is pretty close. If something happens, God forbid, the principle is that you should be able to pull out whatever disc from the server, plug it into another computer and just read the data off of it. Obviously, if you do some degree of striping, that wouldn't be possible or easy. This way, the things are just as straightforward as they can be from a storage perspective.

This does add some complexity in our application layer where we have to be able to allocate, for instance, items basing on their size to different disks. So all of that logic that move stuff around to make sure that there is enough space and that when items grow in size, they are reallocated. All of that kind of is home built. And that's the other engineering principle. Simple most of the times. Crazy when we need it.

So, the general architecture again – This is the architecture within a single server. The larger cluster is just – And a lot of these servers all connected on the same network essentially and extremely distributed. So, whenever you hit your – Whatever webpage on the archive, looking for an item. What is going to happen is that you're going to hit the web head. We have just a pool of like a few VM's that will do like just fronting NGINX requests. And they're going to locate your item using our locator D protocol, which is kind of a cool little DNS for items as we call it. It sends a UDP broadcast to all of the cluster, and the machines that have the correct items will just respond with a UDP unicast. And so this is how we are able to locate items within the cluster.

And then the connection is redirected and everything happens over HTTP with that specific data node. This is in read path. In the write path, everything – Like if you want to write to an item or create a new one, you are not just able obviously to talk to a server and say, "Hey, create a directory for me." You have to talk to our catalog system, which is our centralized task queue. And it will receive the data. Make sure that it ends up in an item. Perform all the checks that are

necessary in what I was talking about earlier. So, in terms of like space being available, etc., etc. And then it will write the item and like leave a trace. So we have like some degree of also forensics trace of whatever happens in the archive.

And then the other big piece of the architecture would be the metadata API, which is the directory service for all of these items. So, what this service does, it aggregate data from catalog locator data within the item and it collects that into JSON files that guarantee that you have up-to-date information about the metadata of all 40, 50 million items that we have right now.

From a high-level perspective, this is kind of what the storage infrastructure looks like and what the pieces are. You have the cluster. You have the cluster with the items. You have the catalog and you have the metadata API.

[00:24:09] JM: Can you tell me some general lessons that have emerged in how to do digitization at scale?

[00:24:15] DS: Oh, sure. Plenty. I think one that comes to mind immediately is the same between building up our cluster and building up our infrastructure, which is using as much as you can off-the-shelf cheap things. So, our scanners, which I guess I should talk a little bit about, because maybe like you're imagining a scanner like a flatbed scanner. Not quite.

We have three different families of devices for scanning books. The older one we call the full print scribe. It's kind of like a telephone box. There is a tabletop scriber, which is a scaled-down version of the same system that just like sits on a table. And then we have something called foldout scribe, which is kind of like a table with a camera on top and we just have to take pictures of large format, and maps, newspaper, and stuff like that.

So, as we built up the capacity to digitize very large amounts of very different kinds of formats of books. So one lesson was do this with as little specialized equipment as you can. And this is especially true for cameras. So, there is a ton of very sophisticated industrial sensors that you could use for this. But it turns out, what works for us is just using Sony cameras. We used to use Canons before this. We used to have 5Ds, but we phased those out because after a couple million shots, like the mechanical shutter activator would fail.

And, so Sony has this pretty new cameras, they're pretty cool. We started with the a6300 Alpha. They're pretty cool cameras, because they have this shutter-less mode. And so there is no mechanical activation involved in a shot. In fact, I think we rolled out 300 or 400 of those at this point. And the failure rate is negligible. I think it's like literally like less than 10 that have probably failed. Sometimes it's the lens that fails and stuff like that. But amazing reliability from consumer hardware there.

However, the other lesson that we learned is that books specifically are kind of weird beast, because in every engineering problem, I guess at some point you have to model it. You have to make some assumptions about, "Okay, this is the size of book that I'm expecting. This is the kind of content that I'm expecting." If I have to do like auto-cropping, I expect like there will be white area here, or whatever, or like I can inspect this kind of size. And it's amazing how quickly any assumption you can make about books will go out of the window, especially when you scan like thousands of them a day. It's really interesting how many different formats they will come in. And into how many edge cases you will run in. And this goes from books that have like puppets in them. The books that have CDs. To books that have electronics. To books that have like bizarre foldouts that are really difficult to capture.

So, the lesson here is don't be too sure about what you will encounter and be ready to accommodate it. So, for example, one of the things that we had to do is modify that pipeline that I talked about earlier. If you recall, I said you upload your book and then it goes into this pipeline. It comes out of the other end and some of the website, whereas the physical book goes away into a warehouse.

Well, sometimes after you scan a book or while you scan it, you realize that it does a weird foldout, and it would be a pity not to capture that. So, I modified the pipeline so that after a book is uploaded, it can actually be downloaded in a different station. Rescanned with the missing pages and then re0uploaded, and this process can be repeated for like an arbitrary number of times. So, we have books that have traveled, like two stations, then they got back corrected. Pretty interesting process and kind of involved, but it was a classic example of we can fix this with software. But in that edge case that we would've never thought about. Yeah, those are two lessons right there.

[00:28:26] JM: Are using any cloud services or is this all on-prem stuff?

[00:28:30] DS: 100% on-prem. Zero cloud, nil.

[00:28:35] JM: Is that for cost reasons or consistency reasons? Or because you've been doing it for so long, why not reasons?

[00:28:43] DS: Great question. So, I think there is different orders of reasons. First of all, is cost, first and foremost. Again, this is not a storage project, like a high school storage project. This is a long-term preservation effort. And it turns out that like fads are fads. But what we're interested in are kind of long-term reliable technologies. And we found that if we rollout our own solution, we can be a lot cheaper than the cloud. I think right now we are more than 10 times cheaper than the cloud depending on how you define cloud, or whatever you compare it to.

Obviously, if we were to store 60 petabytes in Glacier, it would be very different than storing it in S3 or EC2. But what is certainly true as well is that we have been lucky enough that the cost of storage has decreased in a manner that is kind of similar to the manner in which users have increased. And that is not at all guaranteed. But we managed to keep that pretty on point.

And so, as our demand for space has ballooned, it turned out that we could kind of keep up just by relying on the fact that the hard drives were getting cheaper. And we've been expanding our capacity significantly. But all of that, still, in terms of cost, significantly cheaper than what it would have been if we had outsourced it. Also, in terms of engineering, because then you need to re-architecture your platform over a couple of years. If this platform fails, then you move it elsewhere, and whatnot.

So this is like the first reason. Then, there are some more philosophical reasons why we like to do it that way. First of all, we like to have control over our data. And that means that even if a national security letter arrives, at least we know that it arrived. And we successfully filed one a few years ago. But it also allows us to ensure that our patterns data is safe. And by that, I mean something very specific. I mean, that is a very long and sad story of libraries being asked about in list of things that people were reading. We want to make sure that such lists don't exist. And

to the extent to which the exists, we know where they are and nobody else can see them. By that I mean, obviously, we have some web server logs before we get rid of them. We anonymize them and everything. But even that is something that we want to have on our own premises. We wouldn't want to have them in somebody else's computer, so to speak.

And the way we are able to do this, because this may sound pretty interesting. How do you get to be 10 times cheaper than the cloud? Well, we have a very different uptime requirement than the cloud. The clouds needs to up 95.59% or whatever. We decide that it's better to take a hit on availability. And it's amazing, when you drop that requirements from 95s to just 99%, you are making an incredible amount of savings on the infrastructure that is required to keep you going then.

So, you will find that throughout the year, the archive ever so often may be down. It may be down because we get a fiber cut, like we did last year. Could we have another fiber? Yes, but it would be very, very expensive. In fact, we're getting another one. That's a separate story. Could we have 3, 4, 5 replicas of every piece of item that we have? Sure, we could. But would it be worth spending three, four times the hard drive budget and everything else is killing up things so that we can guarantee a .1% of availability? If you are amazon.com and you have a cart that needs to be dispatched, maybe. We calculated that we're not and that it makes more sense for us to save and sort of push the limits of frugality that way. And so, yeah, this is why.

[00:32:52] JM: It's very interesting about the sacrificing a reliability for cost management. Are there any other ways in which that the tradeoffs of the unique domain that the Internet Archive is solving for. Anything else, like any other interesting tradeoffs or anecdotes you can share about that?

[00:33:13] DS: Sure. Well, I mean, you were asking I guess in terms of storage. One that I mentioned this – And maybe we didn't really talk about this much, but one thing that I mentioned is that our data obviously doesn't exist in a single copy, right? As a policy, we keep paired copies of everything. So, the way it works is that every server has a mirror image of itself in a different data center.

How did we come to the realization that two was enough is simply – Is a case of this. So, it would be fairly easy I think to re-architect things so that we could have triple copies or quadruple copies for like certain items, or whatever. But the reality is that that cost is not offset by the benefit that we would be giving to our patrons. I think, our patrons – A good way to think about this is like to think of, again, back to our users. It's hard to find them with bit of a broad stroke. But what brings them together is the fact that they are interested in our offer. Certainly, the content that we have, but are also not paying for it. And some of them are donating, but everybody understands that sometimes the library may be closed, right? You're not going in to the library with the same attitude as you would be going to a store.

In fact, it's very important that the library remains free. Andrew Carnegie, who is by – Anyway you can imagine an expression of capitalism, when he founded the Boston Public Library, had free to all written in big letters on top of it. So this is an interesting way in which the engineering is actually serving the mission. Allowing us to trade reliability for cost makes sure that we actually make the Internet Archive available in the long-term.

[00:35:09] JM: Tell me an anecdote about something that has been incredibly hard in the digitization process. What kinds of tasks have been really difficult?

[00:35:17] DS: Oh! There's been a few. I'll talk about my personal experience of rewriting the software stack. So, when I picked up this project back in 2015, that we had an existing digitization system. We call that Scribe 2. And it was kind of a relic of the past. It hadn't doesn't really be maintained. But it was kind of working and chugging along, and it's what most people did their scanning on. It's what they knew. And my job was to essentially rewrite this thing from scratch. And there were a few things that were necessary. One major requirement was that instead of working like the previous system did, which was kind of hooked up into our infrastructure almost at the network level. So you would have like every scanner was like kind of VPN-ed into employer main cluster. This could only interact through the Internet Archive over APIs.

And one huge challenge there was that we just didn't have all of these APIs, and we had to find a way to expose like part of the guts of the system to the outside world in a way that was kind of secure and acceptable to the existing developers of the internal software. So that was like a first

challenge in re-architecting the system. The second one was that I wasn't starting from scratch. In fact, I had inherited this little script. And the script was like what we call Scribe 3, was a new version of the Scribe scanning software. It was maybe like 1000 or 2000 lines of Python in this GUI system called Kivy. And what it did is it essentially mimicked some of the main features that like the previous system had while it did some crazy things trying to mimic the – Sorry. Trying to get to the backend APIs the we still didn't have.

And so, the problem with this is that it was already being used. By the time I came on board, like 10 or 15 people were already trained on this, and it wasn't great software by any – Like it was just very early. And so a huge challenge was developing this software backend in our cluster. Creating all of this new set of APIs while we were creating a new frontend for which we had existing legacy. So users already had expectations about how we would have to work. And all of this, we couldn't lose a day of production. It's not like I could say, "Hey, let's pause everything and get back to the drawing board."

So, paying back that technical debt took a couple of years. And I think we are now in a good place where the whole thing was re-architected from being just a scribe. Now it's like essentially a little operating system. So that was an interesting little challenge.

I'll tell you another one. So, I mentioned that in the previous incarnation of the system, every node, every scribe was hooked up to our main cluster at the network level. And the reason this was interesting and made the system unnecessary was so that these nodes were running the same PHP monolith as the rest of the cluster. And we're able to act as Internet Archive storage nodes. So whenever you scan the book, you scanned it into this thing. And then it would just get shuffled away to its final destination once it was done.

What this allowed you to do was also to have access to all of these wealth of internal APIs for reprocessing that I mentioned earlier. Once we decided to cutoff the cord, not only did we lose access to all of these APIs. But we also lost taxes to the ability to control the scanners remotely, right? Because suddenly you no longer have all of these nodes that have IP addresses. They are just clients that talk to our HTTP API. And the reason this was a problem is that a lot of management actually happens in a centralized fashion, where for instance you have a manager that wants to say, "Show me all of the books that are being processed in my station center." Or

something happens and you want to clean up all stations that have that belong to this other center where we know that there has been a problem.

In general, we lost entirely visibility and the ability to contact our stations from the central mothership. And so what I ended up doing to fix that was create – I just used IRC. It's still a great solution to this day. There is now an IRC server that acts as a command and control for all of these stations, all of the stations essentially talking their own channel. And there is an HTTP API that you can use to send messages to all of these scribes. And so that integrates with our management tools. Yeah, that was pretty cool. I didn't expect that IRC would be a solution in 2019 to a problem that I had.

[00:40:19] JM: Fascinating. Let's talk a little bit more about the Internet Archive. Now that we've given a good overview of one of the particular projects under the auspices, generally speaking, how does the Internet Archive scale? Are you just racking and stacking new servers and scaling that way?

[00:40:39] DS: Pretty much, yes. For the core data product, that is exactly the case. We will add bigger and bigger disks over time. I think our most recent addition, our 12 terabytes. So the density of a single rack will go up over time. But other than that, yeah, that's how we scale, by direct pair. It's never one rack. It's always two, because items are paired.

And I think at this point – So every rack is then data nodes. Every data node is 36 disks. I think right now we're about 4, 4.5 petabytes per rack pair. And yeah, we scaled that way. There are other components of the infrastructure that don't scale or don't scale as well, or scale differently. So, we use Redis a lot for caching all over the place. And so we scale that to a 10 nodes cluster. It was an interesting project. We scale Elasticsearch. Elasticsearch also scales pretty horizontally. And we do that within the context of our high-performance virtual machines pool. And then there are like some other components of the infrastructure that we'll scale less.

So I mentioned this catalog task queue. That's kind of a problem for us still. But we managed to make it go faster by scaling it late vertically and making better things happening and adding more layers of priority, making faster database access, all that stuff. But to any practical purposes, the core of the Internet Archive scale just horizontally. We just rack and stack.

[00:42:17] JM: What's the worst case catastrophe scenario that could happen? Are there any worst-case scenarios that can happen? Like an earthquake, or a hurricane that you're not resilient to because you're not on the cloud?

[00:42:30] DS: That's a great question. And I think that's above my pay grade, and that like it requires a certain amount of scenario planning that I wasn't necessarily involved in. But from my understanding, what keeps me up at night is the earthquake more than anything else. We do have off-site backups. We do have partial backups. We do have some ability to recover our data even we're not fettered to the cloud. But the reality is that the majority of our data is here. And if something happens to the Bay Area, I think that's the night mercenary for me. I am less worried about cyber threats. I am less worried about other kinds of man-made threats. What really worries me is the big one, and that is not so much because the data will be lost. It's because – Again, I said we've already done our best to make sure that data will be available even in a disaster recovery scenario. It's because the data will no longer be available. And so that goes back to accessibility [inaudible 00:43:40] preservation. The moment the archive goes offline in a disaster scenario, it starts losing value that very second, and that scares me. I think that's the disaster scenario for me.

[00:43:53] JM: What are the plans for the future of the Internet Archive. Are there any new projects that are going to be taking up a lot of the time?

[00:44:00] DS: There are, but I cannot talk about them.

[00:44:02] JM: Okay. Fascinating.

[00:44:06] DS: But stay tuned, because there are always things cooking for sure.

[00:44:11] JM: What keeps you working at the Internet Archive?

[00:44:13] DS: The people and the mission, I want to say. And I want to spend just a few words commending the people that I work with. And they are really some of the finest engineers I have had the pleasure to interact with. I learned a lot at the Archive, and I was given the opportunity

to build up a pretty large system with a pretty significant amount of backing from high-up. It is been incredibly empowering and not just because you feel like you're building something. You also feel like you're building something for people who care. So does the mission part of it all.

I believe in universal access to all knowledge. I believe in books being important and access to information being important. And I think, ultimately, this dataset will prove really, really useful. I don't know. It's pretty simple, really. It's the people and the mission.

[00:45:08] JM: Okay, Davide. Well, thanks for coming on the show. It's been great talking.

[00:45:11] DS: Awesome. Thank you, Jeff.

[END]