Twitter Thread by foone

foone
@Foone

I like how mediawiki exports pages because there's nothing I enjoy more than parsing a 1 gigabyte XML file

and I'm doing simple-english-wiki! it's way smaller than english wikipedia
I can only imagine how impossibly unparsable that nightmare is

fun fact: when programmers need to write code to extract data from an XML file, the process usually goes like this:
1. find XML file
2. open it in a text editor, see what the structure is
3. write some code that parses that

do you really want to open a 1 gigabyte XML file in your text editor?

do you want to fucking die?

I used to run into this problem back when I worked for the government, because sometimes I'd click a database query log (don't ask why our logs were in XML, it's... complicated) and it would turn out to be 100mb

and our IDE of choice was Eclipse

Do you know what happens if you open a 100mb XML file in eclipse?

it crashes, that's what.
And better yet, it crashes in a "safe" way.
It pops up a dialog saying "THE IDE HAS CRASHED! [Restart] [Quit]"

and this is what's known in computer circles as a massive trap

it's also what gamers call a thinking-outside-the-box puzzle.

because here's the thing: those two options?
they are both wrong

except maybe "quit"
you could argue that quitting eclipse and then uninstalling it and using a different IDE was the right option all along

but I was in an environment where that was not an option, so, eclipse it had to be.

Anyway, so the obvious option is "restart".
That sounds right, doesn't it? your IDE is crashing, you just need to restart it, and you'll be right back where you were!
Eclipse even saves all the tabs you have open!

wait.

Eclipse saves all the tabs you have open?

does that include the one that just crashed Eclipse?

YOU BET YOUR ASS IT DOES

eclipse will shut down, then start loading again, and 45 seconds later you'll be staring right back at the same fucking crash window

that's why it's a thinking outside the box question.
it's like those classic trick questions: "have you stopped beating your wife yet?"

They can't be answered with yes or no. And Eclipse's question can't be answered with "restart" or "quit"

You have to close the crashed tab behind the modal "restart or quit" dialog (because it turns out you can do that), and THEN you click restart.

I got very used to having to do this.

It's the kind of thing that could have been handled by having some other text editors on hand, but I had like Notepad, Microsoft Word, and Eclipse.

it's also the thing that could have been solved by having the developers running machines with more RAM than the ones used by the data scientists who only used excel and word, but that would complicate IT purchasing requirements so NOPE ENJOY YOUR 512mb XP BOX

doing programming work for the government was entirely a familiar experience, but it's not the kind you'd think it was.

it didn't feel like a job, really, where you have something that needs to be done, some limited resources, and you work

together with people to do it

it felt more like challenges. not challenges in the sense of "every thing that hasn't been done yet is a challenge we must overcome", but things like programming challenges or chess challenges or such.

things like "write a program in C to print 'hello world' which has no strings or integers in it" or "how many knights can you place on a standard size chess board such that none of them can take any other?"

it was a lot of "you need to do job X. You'd normally expect you'd have Y and Z to do it, but no, fuck you.
You have A and B. Yes, those are the wrong tools. Hard mode: You can't use C or D, either!"

"You need to make a website to let users download these 15 million files. You have to write it in Java. You're not allowed to have a database here which contains a list of the files, or a server containing the files for more than a week"

"So you have to connect to a 3rd party and query the file metadata from them, then they'll provide a server to get the files from, which users can't download from.
So you have to individually download the files and then rehost them, for the users, as they're requested!"

me: ok, that shouldn't be too hard, there's some difficult bits but they should be manageable
them: "HARD MODE TIME! the files on their server? they're in the wrong image format"
me: oh come on

me: no problem, I can just re-encode the files in my server and host the converted files
them: "SURPRISE! The server is an underspeced old redhat box, it doesn't have the RAM to handle loading these large TIF files!"
me: WHAT

me: ok so it turns out TIFF uses a compression format (inherited from fax machines, in fact) which PDF also supports, so it turns out you can turn (some) TIFF files into a PDF by just rewriting the header. That'll work, even without memory.

them: "SURPRISE! their backend server is very slow. downloading files can take up to 30 seconds"
me: that's okay, users can wait...
them: "GUESS WHAT? users will hit stop, and try again. so now there's two connections, making it SLOWER!"
me: FUCK

them: "And to make sure your site doesn't overload any of the other sites we're hosting on the same box..."
me: I'M SHARING THIS SERVER THAT CAN BARELY RUN MY SITE WITH OTHER SITES?!

them: "yes. Don't interrupt. To make sure you don't break THEIR sites, we'll set a max page query time! So if it doesn't answer within 10 seconds, it gets canceled."
me: ... but how
them: "FIGURE IT OUT, YOU'VE GOT THE COMPSCI DEGREE!"

me: ok so the site can start a thread in the background (yes, webpages usually don't have threads that survive between requests but THIS IS WAR, PEACOCK!) and then the page can keep refreshing waiting for it to finish. that works. Anything else?
them: "NOW THAT YOU MENTION IT..."

them: "So the subcontractor who is hosting all this stuff uses MSSQL!"
me: so? I can just install the mssql connection libraries and talk to it natively...
them: "no you can't."
me: what? why not?
them: "Because we don't use those. We're an oracle shop. We hired oracle DBAs!"

me: but we need mssql connectivity to make this work?
them: "you do."
me: so I need mssql connectivity libraries to talk to them
them: "yep."
me: so I should just use those libraries...
them: "nope, you can't. we're oracle."

me: ok so I got on the line with the subcontractor and convinced them to install some software which provides a mssql-to-oracle gateway, which means they need more hardware (which we pay for) and we have another point of failure.

them: "and isn't that better?"
me: better than what, just installing mssql connectivity libraries?
them: "better than violating the ideological purity of our oracle-only software environment!"

me: yes, that's definitely worth spending another, I don't know, 10,000$ of taxpayer money a year?
them: "of course! BTW, we need you to manage this other program too. It's an access program"

me: WAIT, so I can't have MSSQL because "we're an oracle shop", but we have databases in MICROSOFT ACCESS that a production critical?
them: "no."
me: but you just said?

them: "it's a Microsoft Access PROGRAM. It's a bunch of Visual Basic and such which manages a database, modifying it and querying it and generating PDFs and printing things."
me: so it's not a database?

them: "well it's kinda a database. it accesses a database. it manages a database. but the database isn't access."
me: oh god, let me guess, it's oracle?
them: "OH YES"

me: I didn't even know you could connect a Microsoft Access database...
them: "PROGRAM!"
me: ... a Microsoft Access Program to an Oracle database.
them: "well guess what: you can!"

me: is that supported out of the box?
them: "Oh no! You have to install ODBC drivers."
me: wouldn't those have to be installed on every computer using the database?
them: "PROGRAM!"
me: right. Wouldn't they have to be installed everywhere?

them: "Yep. You have to install the oracle ODBC drivers on every computer that uses the program."
me: ok, at least that's easy. You just run the installer off the shared drive
them: "NOPE! users don't have access. you need admins!"
me: ... ok so I ask IT to remote install them...

them: "NOPE! Here's the fun part: the ODBC connections are also admin-only."
me: So the user can't just configure the database connection after IT installs the ODBC driver?
them: "NOPE! IT has to do that."

me: ok ok ok, so I just tell IT to install the oracle ODBC driver so that we can talk to it with Microsoft Access, and to pre-configure a connection to this server at this host, username, password
them: "NO. No credentials in IT tickets. it's insecure."

me: sigh. Can I tell IT in an email?
them: "NO CREDENTIALS IN EMAIL"
me: So can I call them on the phone and have them put it in their secrets store, so I just tell them "configure with the standard credentials for the FOOBAR project"?
them: "they don't have a secrets store"

me: can I walk over to IT with a fucking paper note with the passwords on it?
them: "no. IT hides behind a locked door and there's a whole call-in process to get there. You're not on the list."

me: so how in the fuck am I supposed to configure these machines?
them: "so you tell IT to install it and then when they are configuring it, you walk over to the cubicle of the user and wait through the whole process and type in the credentials yourself!"

me: ... for every user?
them: "YEP!"
me: but we have like 30 people in our department. I'd have to do this every time someone's computer is reinstalled or upgraded. How long does this take, anyway?
them: "oh, under an hour!"

me: You know I'm a programmer, right? I'm here to write and maintain programs, not do IT.
them: "TOO BAD"

them: "BTW, you're really gonna have fun with the printing part."
me: the printing?
them: "OH YES. See, this isn't just regular printing, all networked HP laser on letter paper"
me: oh no

them: "It's a thermal printer!"
me: ok, that's not too weird. I've used those, simple desk units, they plug in with USB...
them: "IT'S AN INDUSTRIAL ONE FROM 1994"
me: why

them: "it still works. replacing it isn't in the budget unless it breaks. here's the manual."
me: the only mention of windows in this manual is about Windows 3.1. The manual explains how to configure it using BASIC on the original IBM PC.

me: there's a spot in the back of the manual for the driver disks and they're 5.25" floppy disks! AND THEY'RE MISSING

them: "YEP!"
me: are we going to plug this into some retro computer that supports it?
them: "NO. old computers aren't allowed. they're insecure."
me: what about that one?
them: "oh, that signature machine only supports windows 2000. don't touch it."

me: ok so I talked to the other people on my team and the employee I'm replacing managed to find some XP drivers which still work with it. can IT install those?
them: "sure!"

me: I don't mean to tell you how to do your job, IT-man, but I filed a ticket saying "I have a special printer and some drivers for it, please install those"
them: "YEP"
me: and what is it, exactly, that you're doing now?
them: "googling for drivers"
me: I see...

them: "BAD NEWS, FOONE! these drivers don't work"
me: and by 'these drivers', you mean...
them: "the ones I found on google!"
me: yes... could you maybe install the ones I have on the network share, under "eltron printer, subfolder drivers, bracket 'working' close bracket?"

them: "oh I don't know, I'd have to talk to my manager. Just file another ticket and maybe we'll get to it sometime this week."
me: ok.

them: "OK drivers are finally installed. it works. Is that all?"
me: hang on I'm having a problem here. this printer takes a long time to print, like a print job can take like 35 minutes
them: "yes?""
me: and the screensaver timeout is 5 minutes
them: "so?"

me: well this printer has some weird CPU-timing issue. if the PC is doing too much stuff, it'll corrupt the printout.
them: "don't do stuff while it's printing, then!"

me: but... even if I leave it alone, the screensaver will come on

them: "and?"
me: AND THE SCREENSAVER USES TOO MUCH CPU AND THE PRINTOUT CORRUPTS
them: "well you should make sure the PC is active then, so the screensaver doesn't activate!"
me: but if I do anything on the computer, it corrupts.

them: "yeah just wiggle the mouse. Figure it out, smarty pants"
me: can I disable the screensaver temporarily or increase the timeout?
them: "absolutely not. that's not secure."
me: ok...

me: wait! I'm having another problem with the printer
them: "GOD WHY DON'T YOU JUST USE THE STANDARD NETWORK PRINTERS?"
me: can they print on thermal paper labels?
them: "no"
me: but I need thermal paper labels
them: "that sucks for you!"

me: so my problem is that my computer has an antivirus, which does full-drive scans twice a day
them: "yep. for security!"
me: but that uses a lot of CPU
them: "it does"
me: so if that happens while I'm trying to print, the print is corrupted

them: "well, those are scheduled for 9am and 2:30 pm"
me: yes? can I change that schedule for my machine somehow?
them: "no. just don't be printing at those times."

me: ok... why can't you do the full drive scan at 5am? no one is in the building at 5am!
them: "what if you turn off your computer overnight? then it wouldn't run!"
me: I don't turn off my computer overnight
them: "BUT YOU MIGHT!"

me: ok. So, that access database I manage...
them: "PROGRAM!"
me: right. it manages a bunch of boxes in the basement, which have barcodes on them. We haven't updated that in years, and we need to do an inventory. I hate to ask this, but... do we have any barcode scanners?

them: "YES! we do. Hang on"
me: this... what is this?
them: "It's a PSION Organizer!"
me: this is a proto-PDA which you can program in BASIC, with a barcode scanner attached to it. it's older than me.

them: "Yep! it may be a little outdated, but it works!"
me: You know it's 2010, right?
them: "what's that got to do with anything?"
me: hang on this building was built in 1995
them: "yes, so?"
me: so this thing was over a decade old when we started!

them: "IT WORKS, DOESN'T IT?"
me: it has 8K of RAM
them: "so?"
me: we have too many boxes in the basement to fit in 8K
them: "do it in batches! just periodically upload the data to your PC!"

me: wait how do I plug it into the PC?
them: "it has a serial port adapter, obvious. you just take off the barcode reader attachment, and swap in the serial port attachment!"
me: the computer you supplied me with doesn't have a serial port
them: "that sucks"

me: can I get a USB serial adapter?
them: "no. That's not on the approved hardware list. we can't buy you one"
me: can I bring in my own? they're like 10$ on amazon
them: "no unauthorized hardware may be connected to government machines!"

me: ok... I can talk to my contractor contacts and we can organize buying portable barcode scanner PDA thing of some kind
them: "NO! That's a computer. you can't use non-government computers in official government work"

me: isn't this PSION thing a computer?
them: "yes. but we've been using it for a long time, so it's been grandfathered in"
me: so I can use this, but I can't upgrade to the equivalent modern version?
them: "Yes."

me: what about a handheld barcode scanner? not a computer, just a scanner. it's connected by USB, to a laptop. We'll get a laptop and dedicate it to organizing the archive.
them: "eh... sure. but you can't have the laptop"
me: what?

them: "yeah, we don't want you contractors buying your own laptops and using them for government use. the USB barcode thing is fine, but you gotta use it with a government computer."

me: ok, can we get a laptop for the archives?
them: "no. computers have to be for a specific user. you can't have one unattached to a user, how would you even log into it? don't be silly. Also, they're desktops. we use desktops."

me: but surely sometimes people here have to go to conferences and things and present, don't they have laptops for that?
them: "yeah!"
me: where do they get those laptops?
them: "they borrow them for up to 7 days from the IT department!"

me: ok. so we just need to check out a laptop to do the inventory project, which we're estimating will take... two months. Can we get a laptop for longer?
them: "no."

me: ok, we can just keep checking it out every week, I guess. can you make sure we get the same one? it takes a while to set up all the software...
them: "no. we have to re-image the laptops after they're returned"
me: what? why?

them: "well you might have picked up a virus!"
me: IN THE BASEMENT?
them: "it's policy"
me: wait, don't the desktops have antiviruses? remember, they break my printer?
them: "yes of course"

me: so if I did get a virus, in the basement, wouldn't it be stopped by the antivirus on the desktop when I connected it up?
them: "yes! but we have to be sure."

me: ok... well our project is to compare the database to the actual boxes in the basement. We need to query the database while scanning boxes. Do these laptops have wifi?
them: "of course they do. it's 2010! who would buy a laptop without wifi?"

me: and is there wifi in the basement?
them: "no."
me: ...

me: ok, I guess I'll just save all the barcodes to a file, then take those back to my PC, and do the comparison , and THEN go back down and look at the shelf for any problems. sure would be nice to have a database on my laptop.
them: "you can't fit oracle on a laptop!"

me: ok. whatever. we got it done. it was a two-person job, one person holding the laptop and the other one doing the scanning.
them: "yeah, sounds hard!"
me: we needed a ladder. these shelves are pretty high. sure would been nice to have a handheld scanner-PDA

them: "OH YEAH THAT REMINDS ME... you know that site you run where the database and files are hosted by a third party"
me: ... yes?
them: "they're rebuilding it! no more MSSQL, no more FTP server full of files. it's all SHAREPOINT now!"

me: can I query sharepoint from java?
them: "nope!"
me: can I download files from sharepoint with java?
them: "Nope!"
me: who are they building this for, exactly?
them: "Just you!"

me: but it won't work for me... why are they using sharepoint?
them: "It's off the shelf!"
me: what?
them: "We want all our system to use off-the-shelf software! Then we don't need to do any development."

me: but they already did the development, for the current one. it works. they did a lot of development for this... how long have they been building this?
them: "oh, about a decade."
me: And they're throwing it out for a sharepoint solution?
them: "It's off the shelf!"

me: wait isn't sharepoint a sort of document database for microsoft office documents?
them: "Yes! it has all sorts of full-text search, it can find references inside documents and everything. it's great!"

me: does it support TIFF?
them: "what?"
me: all my documents are TIFF images. Can it search for text INSIDE TIFF IMAGES!?
them: "no, that's impossible"

me: that sounds like putting TIFF images inside sharepoint doesn't really gain us anything
them: "not for the documents themselves, no"
me: can we use it like a database, though? we have documents indexed by a bunch of columns in a database

them: "well, we can define custom attributes on each document, and then users can adjust the metadata, and then we can query on those!"
me: that sounds like a database with extra steps

them: "and when the next crawl happens, they'll be added to the index!"
me: hang on, crawl?
them: "Yeah! there's a program which goes through every document in the database and tries to parse out the text and indexes the metadata, and then you can search on it!"

me: so it's a database except you can't search on new changes until the crawl happens?
them: "yep!"
me: how often does the crawl happen?
them: "we were thinking every 3-7 days"

me: but our main publication gets daily uploads. thousands of them.
them: "yeah, that's why we have to batch them into a crawl! databases just can't handle that many documents being added so quickly"

me: it can't handle... THOUSANDS OF IMAGES A DAY?
them: "no, that'd slow down the system too much to do it in real time. we have to batch them and run it in the middle of the night."
me: you know I used to work for 4chan, right? They get that many images an hour.

them: "ok we'll add some more hardware and increase the crawl rate for the main busy library"
me: ok, great. what's the frequency now?
them: "every 24 hours!"

me: so let me get this straight. so I upload an image and assign some metadata, and that image becomes available to users who are searching for it... the next day?
them: "Yep! assuming the crawler doesn't timeout"

me: timeout?
them: "yep, sometimes it takes too long to crawl the database, and the crawl is still running when the next one would start, so we have to kill the old run"

me: so wait... it can take more than 24 hours to parse the couple thousand new images added a day?
them: "oh, no, that'd be silly. it's not that slow"
me: ok good
them: "it has to crawl all the images"

me: wait, ALL the images?
them: "yes! to get a full index, it checks every document"
me: but there are like 15 million of those
them: "yeah, that's why it takes 24 hours!"

me: why are we re-crawling the ones that haven't been changed since 2003?
them: "we need a full index! that's how sharepoint works."
me: ok... this feels like it's not a good fit for what we need.
them: "but it's off the shelf!"

me: ok, fine, whatever. so how do I query this database? I don't have sharepoint connectivity with my java shit. Are you gonna set up another mssql2oracle gateway?
them: "Nope! it doesn't use mssql. Well, it's built on it, but we don't query it at that layer. we use LINQ!"

me: that's a .dot technology for writing queries using native code, right?
them: "YEP!"
me: can I use that from java? across A FUCKING NETWORK?
them: "Nope!"

me: SO HOW IN THE FUCK AM I TO QUERY IT
them: We're gonna develop a SOAP endpoint!
me: SOAP?
them: "OH YES. You'd like it, the code is... very clean"
me: get out.

me: so how does this SOAP endpoint work?
them: "You write PSEUDO-SQL and encode it into an XML remote procedure call!"
me: pseudo-sql?
them: "Yes, we're writing a bunch of code to take pseudo-SQL and convert it into LINQ queries"

me: You're having to do a lot of development to make this work, eh?
them: "oh yes, we're working around the clock."
me: that doesn't sound very off-the-shelf
them: "SILENCE!"

me: ok so I do like SELECT * FROM TABLE_COOP where STATE='CA' and it'll give me the list of publications for california?
them: "NO! we don't support SELECT *. You have to explicitly name columns"

me: how do I get the list of column names?
them: "look at the web interface! they're very like the ones shown in the metadata sidebar"
me: "very like"?
them: "yeah, they're not exactly the same. Spaces are encoded weird, they're case sensitive and not capitalized the same"

me: ok... can you give me a list of the columns?
them: "Sure! here you go. This may change, btw."
me: ok, so will you update me when they change?
them: "What makes you think WE'LL know when they change?"

me: what
them: "Oh yeah! These column names are being autogenerated by some .net serialization bullshit that we don't really understand. They might change some weekend due to an upgrade"

me: but my service can't do "select *". if the column names change over the weekend because of an upgrade, it'd take my entire site down for days
them: "yeah. that sucks."

me: OK, so I just do like select Station__#0020__Name, Date from TABLES_COOP
Them: "Nope! we don't support table names"
me: table... names?

them: "Yeah, we just numbered all our databases! except this is sharepoint, so they're libraries. So for COOP you want Library 109."
me: so it's SELECT ... from 109?

them: "Yep!"
me: that doesn't seem very pseudo-SQL

me: wait I'm testing with this program you gave me and I can't seem to get the data for california or alaska
them: "oh yes, the XML is corrupt for those"
me: WHY IS THE XML CORRUPT

them: "well sharepoint times out the request if they're taking too long, and you just get a truncated XML file"
me: so if I want to query CA or AK, I get... invalid XML back?
them: "YEP!"

me: our system can't handle invalid XML. is there any way you can fix that to return it all?
them: "no. But We'll can put in an xml element towards the end that says so you know your didn't get all the results"

me: ok so then I can just do a query that's like SELECT .... from 109 limit 0,50 and paginate through the results?
them: "guess what else we don't support?"
me: pagination?
them: "you're learning!"

me: so how do I get all the documents for the bigger states? they have more data than will fit into one query!
them: "well you'll have to query for less stuff"
me: BUT THE THING I NEED IS A LIST OF ALL STATIONS IN CALIFORNIA, BECAUSE THAT'S WHAT THE USER IS SELECTING FROM

them: "well you'll have to make the search more narrow"
me: OK... I can do some offline weirdness and do queries for state='CA' and station_name like 'A%' and iterate through all the letters and merge them on my end and "cache" that here so it works.

them: "yeah, it better be a cache. you're not allowed to store that database here!"
me: yeah, that's... kinda weird. Why is that again?
them: "congress!"
me: ... what

them: "oh yeah, the senator for west virginia is the oldest and so he has seniority over budgets and he made sure you're not allowed to have this database"
me: why... what? how is that a budget thing?

them: "well you're not allowed to have the database"
me: yes?
them: "and there is a stimulus for west virginian tech companies!"
me: let me guess: to pay for them starting up database serving companies for the US government?
them: "You're catching on now!"

me: ok, so as long as I call it a "cache" I can keep this list here and not have to do 26 separate queries to your database on every page load, fine. So, how do I download the TIFF images? is it still the FTP server I've been using?
them: "No, FTP servers are not off-the-shelf"

me: great. So how do I get the TIFF images?
them: "SOAP AGAIN!"
me: isn't that an XML-based web query thing?
them: "It's actually not web-only, and it's REMOTE PROCEEDURE CALL, not query!"

me: ok, so it's an XML... remote proceedure call thing. So, what, I call it and it tells me the HTTP address of the file to download?
them: "No, you call it to download the file."
me: wait... in SOAP itself?
them: "YES!"

me: so I have to download the files, embedded in XML?
them: "yep! it's very efficient if what you're downloading is XML itself"
me: but it's not. it's TIFF. that's binary data.
them: "Don't worry, we can base64 encode it."

me: won't that make all the downloads like 1/3rd larger?
them: "yep! but that's the only way to embed them in XML"
me: why do we have to embed them in xml in the first place... never mind. OK, so is there a timeout limit on this one? because some of these files are big

them: "no"
me: oh thank god
them: "There's a file size limit!"
me: what

them: "yeah, we can't send more than about 5 megabytes at once. But don't worry, we'll provide a pagination system for this. We're working around the clock building it!"
me: right, very off the shelf.
me: wait, 5mb? I have some documents that are like 61mb

me: do you seriously expect me to download a document in THIRTEEN SEPARATE REQUESTS for something the user is waiting for?
them: "well, you gotta. that's just how HTTP works"
me: ... what

them: "oh yeah, HTTP isn't very reliable. if we allowed you to download large files all at once, it'd break too often! you can't really expect to transfer like 20mb in an HTTP request without it breaking"

(I swear to god, this is basically verbatim what they said.
They tried to convince us that HTTP WAS NOT RELIABLE FOR FILES OVER ABOUT 10 MEGABYTES as a reason for

why their download service was being designed so incredibly badly)

anyway I gotta go before my throat falls out because I haven't had anything to drink yet. But to sum up what actually happened:
We convinced them to use MTOM, which a SOAP thing where you have a binary-attachment to the XML part, so we don't have to base64 encode our binaries

and we got them to not do the pagination thing on downloads. we had no issues.

so I have no idea what their problem was and why they said that.

I sometimes thing subcontractors lie as their primary goal, and all the working for the government stuff is just a minor side-effect of their main focus, which is fractal fraud

anyway the last bit of nightmare madness that happened during the setup phase is that they could only set up their SOAP service using NTLM2 authentication

there are many ways to authenticate SOAP services and they can be supported by lots of different programs.
NTLM2 is one that only really makes sense if you're an all-microsoft shop and you want to authenticate using your windows logins

they were an all microsoft shop, we were an all unix shop.
so this was not a good fit. But whatever, they can make a special "query the database" user login and let us use that

did they remember to disable password expiration on that account? no, they did not.
60 or 90 days into the service being active, the site goes down, because it's trying to pop up a "your NT password has expired! change it now." dialog on some screen that didn't exist

but the real fun is that we were talking SOAP from our servers using Apache Axis, which is an open source industry standard SOAP implementation.
It supports many authentication methods.
NTLM2 is not one of them

so after a lot of arguing with different security teams we were unable to get them to relent on the NTLM2 requirement.
we could confirm all our code worked just fine if we ran it with authentication disabled or set to other methods, but for production, it had to be NTLM2

but don't worry. we found a solution:
there was a closed source binary-only java library someone had written which could hook up to apache axis and make it support NTLM2.
so we just spent like 5000$ on a server copy and all and bam, it's done, it works

it made debugging a fucking NIGHTMARE because it was closed source and heavily obfuscated but hey, it worked. technically.

did I mention this all was happening over a inter-datacenter VLAN? it's not like this was the open internet. we could have saved a lot of money and development time if they'd just switched to another authentication method, but They Didn't Want To.

anyway, this story only got worse from here.
There was a whole saga where we had to stop working with them because it turned out they were massively defrauding the government by falsifying all their timesheets

and they tried to lawyer their way out of giving us back our data because we wrote a bad contract which said that if we ever canceled the contract, they had to give us back "our servers", which we ASSUMED BUT DID NOT REQUIRE would store our data

and they were actually storing only the indexes and crawlers on the 100,000$ worth of servers we loaned to them, and they had to host the data elsewhere, because it was far less efficient than they were claiming

I heard a rumor (but never was able to confirm this) that a large portion of the data was distributed on the workstations being used by the programmers at their HQ

anyway because they were about to be massively sued for fraud we had to take down the web service stuff with them, and the "getting the servers back" plan was a bust, so now we had the program of getting the data back

we solved it with hard drives. you'd think this would just mean "we went down to bestbuy and picked up Xty terabytes of drives, loaded up the drives from their servers, then drove them to our datacenter and loaded them up"

but because of budget reasons, they had to be the ones to buy the drives. and they refused to buy more than two drives. which were either 2tb or 3tb, and we had a bit over 15tb of data to copy

so they had to keep being mailed back and forth.
they'd load it up, mail it to us, I'd copy the data off, then mail it right back to repeat the cycle

yes, this ended up costing far more money in guaranteed insured overnight shipping than just buying 15tb of drives would have, but importantly it was US paying for the shipping, not them.

would have been nice if we could have just used that money to buy more hard drives, but no, that was in the mailing-shit budget, and you can't just move money around like that, this isn't the vietnam war, there are RULES

anyway this story had extra bits of madness. like IT not letting me scan the drives from my computer, because maybe viruses. so they had to scan them, which took several days, and then they'd be attached to the server.

it turned out the only way to do this successfully involved me getting temporary access to the IT section with a networkless laptop that was about to get erased, and doing a pre-scan, then they scanned it, then it would be attached to the servers

except it then turned out that our servers didn't support NTFS, because they were old, and couldn't be upgraded because of a hostage situation involving retirement pay.
So in the end, they had to get imported by connecting them right to... my government desktop computer

which was the whole thing we were avoiding in the first place.
did we abandon this convoluted process once we realized it was pointless? of course not.

there's more details and scary bits over in my other threads about my government job, which are linked from my wiki (as this thread soon will be):
https://t.co/vF4QY3Cf5q

anyway if you enjoyed reading that, feel free to send me a dollar or two on ko-fi. I have scars from that job, if that's not obvious, and I need to drown my tears in a frappuccino
https://t.co/fxSZjxyBm0

or set up a monthly donation on my patreon:
https://t.co/Sd9bKPHLf2

and yes, I need to write a book. it'll happen, eventually.

and sorry if you replied to this thread: I was in 100% WRITING MODE so I wasn't seeing all the replies as they came in, and twitter sometimes makes it hard to see them all after the fact.
I'll do my best to read through them and reply, though!