MY WEIRD PROMPTS

Podcast Transcript

EPISODE #163

Big Iron: Why Mainframes Still Run the Global Economy

Published January 04, 2026 • Runtime: 21:26

https://myweirdprompts.com/episode/mainframe-vs-cloud-infrastructure/

EPISODE SYNOPSIS

In this episode of My Weird Prompts, Corn and Herman dive into the world of mainframes, often called "Big Iron." They explore why, in 2026, the world's largest banks and institutions still rely on these massive machines instead of moving entirely to the cloud. From "seven nines" of availability to real-time AI fraud detection, discover how these systems handle billions of transactions with zero downtime.

DANIEL'S PROMPT

Daniel

"We've talked about workstations and desktops, but I'd like to discuss another type of hardware that is still very much alive: the mainframe. IBM still dominates this market, and they are used in the financial industry for things like credit card processing where high-volume parallel workloads are required. My questions for you are: why not just use the cloud? What is the decision-making rationale for investing in a mainframe today? What operating systems do they run—are they Linux-based like supercomputers? And finally, who manages these systems?"

TRANSCRIPT

Corn

Hey everyone, welcome back to My Weird Prompts. I am Corn, and I am sitting here in our living room in Jerusalem with my brother.

Herman

Herman Poppleberry, at your service. It is a beautiful day to dive into some deep tech, and our housemate Daniel has really delivered this week.

Corn

He really did. Daniel sent us this audio prompt about something that most people think belongs in a museum, or maybe in a grainy black and white photo from the nineteen sixties. We are talking about the mainframe.

Herman

The big iron. It is one of those topics where people either know exactly what you are talking about because they work in a bank, or they think you are talking about some ancient relic that was replaced by the cloud ten years ago.

Corn

Right, and that is exactly what Daniel was asking. If we have this massive, ubiquitous cloud infrastructure now, why are these giant boxes from IBM still running the world's economy? I mean, he mentioned credit card processing, but the scale of it is actually hard to wrap your head around.

Herman

It really is. We are talking about systems that handle billions of transactions every single day with almost zero downtime. When we talk about the cloud, we talk about three nines or four nines of availability. Mainframes are playing a completely different game. They are looking for seven nines. That is less than three seconds of downtime per year.

Corn

That is incredible. I want to get into the why of that, because for most of us, the idea of a single machine being that reliable feels almost impossible. But before we get into the technical weeds, let's set the stage. When we say mainframe today, in January of twenty twenty-six, what are we actually looking at? It is not just a room full of spinning tape reels anymore, right?

Herman

Not at all. The modern mainframe, like the IBM z sixteen which was a huge milestone, looks more like a very sleek, very heavy refrigerator. Or a series of them. Inside, you have processors that are designed for a completely different kind of work than what you find in a standard server rack. These are not just fast chips. They are chips with massive I O subsystems. I O stands for input output, for the folks listening who might not be in the hardware world every day.

Corn

So it is not just about raw calculation speed, it is about how much data you can move in and out of the processor at once?

Herman

Exactly. Think of a standard high performance computer, like a supercomputer, as a Formula One car. It goes incredibly fast in one direction. It solves complex math problems very quickly. A mainframe is more like a massive, high speed subway system. It is designed to move millions of people, or in this case, transactions, through the system simultaneously without any of them bumping into each other.

Corn

That is a great analogy. So, if I am a bank, and I am looking at my infrastructure in twenty twenty-six, why am I still buying these? Daniel asked about the decision making rationale. Is it just because they have been using them since the seventies and they are afraid to move?

Herman

That is a part of it, the legacy aspect, but it is actually much more strategic than that. The first big reason is data gravity. If you have forty years of transaction data and the core logic of your business sitting on a mainframe, moving that to the cloud is not just a technical challenge, it is a massive risk. But more importantly, the cloud is actually not very good at the specific thing mainframes excel at, which is extreme consistency in high volume parallel workloads.

Corn

When you say consistency, are you talking about the ACID properties of transactions? Atomicity, Consistency, Isolation, and Durability?

Herman

Precisely. In the cloud, if you are using a distributed database, you often have to deal with what they call eventual consistency. That means if I send you money, there might be a few milliseconds or even seconds where the system isn't quite sure if the money is in your account or mine. For a social media post, that is fine. For global finance, that is a disaster. Mainframes are built from the ground up to ensure that every single transaction is perfectly recorded, instantly, across the entire system.

Corn

I remember we touched on some of these architectural trade offs back in episode two hundred twelve when we were talking about AI benchmarks. It seems like the mainframe is the ultimate expression of vertical scaling rather than the horizontal scaling we see in the cloud.

Herman

That is a perfect way to put it. Instead of adding a thousand small servers, you make one giant server incredibly robust. And the security is on a different level. Modern mainframes have hardware based encryption that can encrypt all of your data all the time with zero impact on performance. In a cloud environment, that kind of pervasive encryption often comes with a significant overhead.

Corn

So it is about security, reliability, and that massive throughput. But what about the cost? Because mainframes are notoriously expensive. Is the ROI actually there compared to a massive AWS or Azure bill?

Herman

It is funny you mention that, because as cloud costs have spiraled over the last few years, the mainframe is starting to look more economical for certain workloads. When you are doing billions of transactions, the egress fees and the sheer number of virtual machine instances you would need in the cloud can actually end up costing more than the lease on a mainframe. Plus, the longevity is insane. You buy a mainframe, and it will run for a decade or more with minimal intervention.

Corn

That leads into another part of Daniel's question. What are these things actually running? Is it some weird, proprietary language that only ten people in the world understand? Daniel asked if they run Linux like supercomputers do.

Herman

The answer is both yes and no. The primary operating system for IBM mainframes is called z O S. It is a direct descendant of the operating systems from the sixties and seventies, but it has been modernized every single year. It is not Linux. It uses a different file system, different memory management, and it is designed specifically to manage those massive I O workloads we talked about.

Corn

But wait, I have definitely heard about Linux on the mainframe. Is that a separate thing?

Herman

Yes! That is called Linux on Z, or sometimes IFL, which stands for Integrated Facility for Linux. IBM realized years ago that if they wanted to stay relevant, they had to support the modern open source ecosystem. So, you can actually run thousands of Linux virtual machines on a single mainframe. It is like having a private cloud inside one box.

Corn

So you get the reliability of the mainframe hardware but the flexibility of Linux? That seems like the best of both worlds.

Herman

It really is. You can have your core banking database running on z O S for that extreme consistency, and then you can have your web front ends and mobile app APIs running in Linux containers on the same physical hardware. The latency between those two environments is basically zero because they are sharing the same backplane.

Corn

That is a huge advantage. In a typical cloud setup, your web server might be in one data center and your database in another, and even a few milliseconds of latency can add up when you are doing millions of requests.

Herman

Exactly. And let's not forget about the latest developments. In the last year or so, IBM has been pushing AI on the mainframe. Not for training massive models like GPT four, but for real time inferencing. Imagine a credit card transaction happening. With a mainframe, you can run a deep learning model to check for fraud on that specific transaction in less than a millisecond, while the transaction is still happening. In the cloud, you usually have to do that after the fact because the latency is too high.

Corn

That is a massive use case. If you can stop the fraud before the transaction is even approved, you save billions.

Herman

Precisely. It is all about that tight integration. But I think we should take a quick break before we get into the "who" part of Daniel's question—who actually manages these beasts and what the future looks like for the people in those roles.

Corn

Good idea. Let's take a quick break for our sponsors. Larry: Are you worried about the upcoming solar flare cycle? Do you feel like your digital life is just hanging by a thread in the vast, unprotected wilderness of the internet? Introducing the Invisible Data Shield. Our proprietary blend of rare earth ions and pressurized ozone creates a localized protective field around your laptop, smartphone, or even your smart toaster. It is completely odorless, invisible to the naked eye, and guaranteed to stop at least some forms of electromagnetic interference. Probably. Do not use near open flames or small pets. The Invisible Data Shield—because if you can't see the danger, the danger can't see your data. BUY NOW!

Corn

...Alright, thanks Larry. I am not sure I want pressurized ozone near my laptop, but okay. Back to the big iron.

Herman

Yeah, I think I will stick to standard firewalls for now. Anyway, Corn, you were asking about who manages these systems. This is actually one of the most fascinating and, frankly, stressful parts of the mainframe world right now.

Corn

I have heard the term "graying of the mainframe." Is it true that everyone who knows how to run these things is about to retire?

Herman

It is a real concern. For a long time, universities stopped teaching mainframe skills. Everyone wanted to learn web development, mobile apps, or cloud architecture. So you have this huge generation of systems programmers and operators who are in their sixties and seventies. They are the ones who know how to tune a C I C S region or manage a D B two database at massive scale.

Corn

C I C S? That is a new acronym for me.

Herman

Customer Information Control System. It is basically the application server for the mainframe. Most of the world's bank accounts are managed by C I C S. It is incredibly powerful, but it is not something you just pick up in a weekend bootcamp.

Corn

So what happens when those people retire? Does the world's financial system just... stop?

Herman

Well, that is the big push right now. IBM and the big banks are pouring money into training a new generation. And they are doing it by making the mainframe look more like the tools young developers already use. They have added support for Python, Java, and even Node dot J S. You can use Visual Studio Code to write mainframe code now. You don't have to use those old green screen terminals anymore.

Corn

That makes a lot of sense. If you can use the same tools you use for web dev, it lowers the barrier to entry. But is it working? Are people actually signing up to be mainframe engineers?

Herman

It is starting to happen. There is a certain kind of engineer who loves the idea of being at the center of everything. If you work on a mainframe, you are not just building a feature for an app that might disappear in two years. You are managing the infrastructure that keeps the global economy moving. There is a weight to it, a sense of importance that you don't always get in other areas of tech.

Corn

I can see the appeal. It is like being the lead engineer on a nuclear power plant or a massive bridge. The stakes are incredibly high. But Daniel also asked about the decision making rationale today. If I am a startup, I am obviously not buying a mainframe. So is this market just slowly shrinking as the older companies eventually modernize?

Herman

It is actually the opposite in some ways. While the number of mainframe customers might not be growing rapidly, the amount of work being done on mainframes is increasing. As more of the world goes digital, the number of transactions per second is exploding. And the mainframe is still the most efficient way to handle that growth. We are seeing a shift toward what is called hybrid cloud.

Corn

Hybrid cloud. We talked about that a bit in episode two hundred sixty four with the Model Context Protocol. The idea that you don't have to choose just one environment.

Herman

Exactly. You keep your core, high security, high volume data on the mainframe, and you use the cloud for your customer facing apps, your data analytics, and your experimental features. The mainframe becomes the "system of record," and the cloud becomes the "system of engagement." It is a very powerful combination.

Corn

So, it's not a "this or that" situation. It is about using the right tool for the job. But what about the "why not just use the cloud" part of the question? If AWS or Google Cloud really wanted to, couldn't they build something as reliable as a mainframe?

Herman

They are trying, certainly. But it is a fundamental architectural difference. The cloud is built on the idea of commodity hardware. If a server fails, you just spin up another one. It is designed for failure. The mainframe is designed to never fail. Every component in a mainframe is redundant. There are two of everything, and they are constantly checking each other. You can literally pull a processor card out of a running mainframe, and it will keep working without dropping a single transaction.

Corn

That is the hardware level reliability. But what about the software? Is z O S really that much better than Linux for this?

Herman

For transaction processing, yes. z O S has something called the Workload Manager, or W L M. It is an incredibly sophisticated piece of software that can prioritize millions of different tasks based on business goals. If a high value trade comes in, the W L M ensures it gets the CPU cycles it needs, even if the system is at ninety-nine percent capacity. Linux is great, but its scheduler wasn't built for that specific kind of extreme multi-tenancy.

Corn

It sounds like the mainframe is the ultimate "set it and forget it" machine, but with the caveat that when you do need to change something, it requires a very specific set of skills.

Herman

Exactly. And that is why the "who manages them" question is so critical. It is not just about writing code; it is about understanding the whole ecosystem. A mainframe systems programmer needs to understand hardware, networking, storage, and security at a very deep level. They are the last of the true generalists in a world of extreme specialization.

Corn

That is an interesting perspective. Usually, we think of mainframe people as the ultimate specialists, but they actually have to know how the whole stack works because the stack is so integrated.

Herman

Precisely. In the cloud, you might only care about your specific container or your specific API. On the mainframe, the performance of your application is tied directly to how the whole machine is configured.

Corn

So, let's look at the practical takeaways here. For our listeners who are in the tech industry, maybe they are developers or IT managers. Should they be paying attention to the mainframe? Or is it just a niche for the big banks?

Herman

I think everyone should pay attention to the architectural principles of the mainframe. The focus on reliability, the focus on I O throughput, and the way they handle data gravity—those are lessons that apply to the cloud too. If you are building a high volume system in the cloud, you are essentially trying to recreate a lot of what the mainframe already does.

Corn

That is a great point. We often reinvent the wheel in the cloud because we don't look at the history of computing.

Herman

Exactly. And for younger developers, don't dismiss the mainframe as a career path. The salaries are often higher because the skills are rare, and you get to work on some of the most challenging problems in the world. Plus, with the integration of Linux and modern tools, it is not the "boring" job it used to be. You could be the person who figures out how to run a massive AI model on a mainframe to prevent global financial fraud. That is a pretty cool resume line.

Corn

It definitely is. And I think it is important to realize that "legacy" doesn't mean "obsolete." Sometimes, the old way of doing things is still the best way because it was designed for a level of scale and reliability that we are still struggling to match with newer technologies.

Herman

Well said, Corn. The mainframe is a survivor. It has survived the rise of the PC, the rise of the internet, and now the rise of the cloud. And every time, it has adapted and found a new reason to exist. I suspect we will still be talking about mainframes in twenty thirty-six.

Corn

Probably so. Maybe by then they will be running on quantum processors, or whatever the next big thing is.

Herman

Oh, don't get me started on quantum. IBM is already working on quantum-safe cryptography for the mainframe because they know that their customers need to be protected from threats that don't even exist yet. That is the level of forward thinking we are talking about.

Corn

That is actually a perfect place to wrap up. The mainframe is the past, the present, and apparently, the quantum-safe future.

Herman

It really is. Daniel, thank you for that prompt. It was a great excuse to nerd out on some big iron.

Corn

Absolutely. And to our listeners, if you enjoyed this deep dive into the world of mainframes, we would really appreciate it if you could leave us a review on your podcast app or on Spotify. It genuinely helps other people find the show and allows us to keep exploring these weird and wonderful topics.

Herman

Yeah, every review counts. And if you have a question or a topic you want us to tackle, you can head over to our website at my weird prompts dot com and use the contact form. We love hearing from you.

Corn

This has been My Weird Prompts. We are available on Spotify and all your favorite podcast platforms. I am Corn.

Herman

And I am Herman Poppleberry.

Corn

Thanks for listening, and we will see you next time.

Herman

Keep those prompts coming!

Corn

Until next week, everyone. Stay curious.

Herman

And stay off the pressurized ozone.

Corn

Good advice, Herman. Goodbye!

Herman

Bye!