Cloud Politics

Google has YouTube, AWS has Prime and Netflix. what do others Disney, etc use?

- **Disney** uses **Disney+** (along with **Hulu** and **ESPN+**).
- **Warner Bros. Discovery** uses **Max** (formerly HBO Max).
- **Apple** uses **Apple TV+**.
- **Paramount** uses **Paramount+**.
- **NBCUniversal** uses **Peacock**.

Were these all very different models, given the hugeness of AWS and Google

No, they are actually running on the exact same cloud infrastructure. Because **AWS** and **Google Cloud** own the underlying server and network technology, almost every major streaming competitor rents capacity from them rather than building their own global data centres.

The primary difference is not the hardware they run on, but **how they make money** and **where they source their content**.

1. The Underlying Tech: Almost Everyone Uses AWS or Google

Even though they compete for viewers, traditional media giants rely heavily on Amazon and Google's cloud backbones to actually deliver video to your screen:

- **Disney+ & Hulu.** Both rely extensively on AWS for video processing, storage, and global content delivery.
- **Max (Warner Bros. Discovery).** Uses AWS as its primary cloud provider for content storage, streaming observability, and live ad insertion.
- **Peacock (NBCUniversal).** Built its streaming system and live sports delivery on AWS.
- **Paramount+.** Built its global streaming platform on Google Cloud to handle live events like the Super Bowl.
- **Apple TV+.** Despite Apple's massive size, it does not operate a public cloud to rival Amazon or Google. It rents massive amounts of storage and server capacity from both **AWS and Google Cloud** to power iCloud and its media services.

(Note: Just like Netflix uses its own "Open Connect" boxes to cache video closer to local internet providers, most of these services use a mix of AWS/Google Cloud and third-party Content Delivery Networks like Akamai or Fastly to prevent buffering.)

2. The Business Models: Where They Actually Differ

Where these companies break away from Google (YouTube), Amazon (Prime), and Netflix is in their **business model**. Because they don't have massive tech or retail empires subsidizing them, their strategies look quite different:

Company / Service	Primary Business Model	How It Differs from Tech Giants
YouTube (Google)	**User-Generated & Ad-Supported.**	Doesn't pay to produce most of its content; relies on creator revenue-sharing and massive ad volume.
Prime Video (Amazon)	**Ecosystem Loss-Leader.**	Primarily exists to keep you paying for an Amazon Prime shipping and shopping membership.
Netflix	**Pure-Play Subscription.**	Relies almost entirely on monthly subscriber fees and ad tiers to fund a massive $18B+ content budget.
Disney+, Max, Peacock, Paramount+	**Legacy Studio & Ad-Tier Hybrids.**	Built to replace declining cable TV revenue. They rely heavily on existing movie/TV vaults, live sports, and cheaper ad-supported tiers to stay profitable.
Apple TV+	**Hardware & Ecosystem Lock-in.**	Like Amazon, Apple uses high-budget original shows to sell iPhones, Apple TV boxes, and **Apple One** / **iCloud+** bundles rather than needing the service to turn a standalone profit.

Not a lot is published. Google and AWS are very different, and Azure is uniquely MS.

You're spot on. Under the hood, **AWS**, **Google Cloud**, and **Microsoft Azure** are built on fundamentally different engineering philosophies, and the media giants keep their exact multi-cloud setups tightly under wraps, so they don't give rivals a competitive edge or lose pricing leverage.

While Disney, Paramount, and others rent space from these providers, the three tech giants themselves run very different architectures to power their own platforms and customers.

1. How the Big Three Clouds Actually Differ

- **AWS (Amazon):** Built originally to scale Amazon's massive retail site, AWS focuses on **modular, building-block services** (like S3 for storage and EC2 for computing). For video, it

relies heavily on its own global Content Delivery Network (**CloudFront**) and specialized media tools (**AWS Elemental**) to process and stream live video at massive scale.

- **Google Cloud (GCP):** Built on the exact same internal infrastructure that powers **YouTube** and **Google Search**. Google owns one of the largest private fiber-optic networks in the world, meaning video traffic stays on Google's private high-speed cables until the very last mile, rather than bouncing across the public internet.
- **Microsoft Azure:** Uniquely tied to **Microsoft's enterprise ecosystem** (Windows, Active Directory, Xbox). While Microsoft doesn't have a consumer Netflix rival anymore (after shutting down Mixer and Groove), Azure powers massive live-streaming events (like the NBA and NFL) using **Azure Media Services** and custom edge-computing hardware built for gaming and low-latency broadcast.

2. Why Media Companies Keep Their Tech Stacks Secret

You mentioned that not a lot is published—and that is completely intentional. Streaming companies treat their exact architecture as a trade secret for a few reasons:

- **Vendor Leverage:** Disney, Warner Bros., and Apple purposefully split their workloads across AWS, Google, and Azure (a **multi-cloud strategy**). If they keep the exact split quiet, they can pit Amazon and Google against each other during contract renewals to get massive discounts.
- **Custom "Last-Mile" Delivery:** Just like Netflix builds its own custom **Open Connect** servers and places them directly inside local internet providers (like Telstra or Comcast), Disney and Apple quietly build proprietary caching systems. They rarely publish the blueprints so competitors can't copy how they prevent buffering during massive events like a *Star Wars* premiere or live NFL game.
- **Acquisition Spaghetti:** Most of these apps are Frankenstein monsters of acquired tech. For example, Disney+ was built on **BAMTech** (originally created by Major League Baseball for MLB.tv), which had deep roots in AWS, while Hulu and ESPN+ ran on completely different legacy systems that are still being merged behind the scenes.

3. Who Uses Azure for Media?

Because Azure is so heavily focused on corporate IT and gaming rather than consumer entertainment apps, it flies under the radar in the "streaming wars," though it still powers major media backbones:

- **NBA & NFL:** Both leagues use Azure to power their direct-to-consumer streaming apps, live game stats, and broadcast workflows.
- **Xbox Cloud Gaming:** Microsoft uses custom Xbox server blades inside Azure data centres to stream live, interactive video games—which is actually harder to engineer than Netflix because it requires near-zero lag.
- **Sony (PlayStation):** Despite being Microsoft's biggest console rival, Sony famously partnered with Azure to power its cloud gaming and media streaming infrastructure because Sony didn't want to build its own global data centres.

Would you like to look closer at how **private fibre networks** (like Google's) compare to **local ISP caching** (like Netflix's Open Connect), or explore how **live sports** pushed these cloud providers to their limits?

Behind the Big Three (AWS, Microsoft, Google), Alibaba is huge in China. Oracle and IBM build their own infrastructure. Do Salesforce or SAP have preferences, or I imagine they keep close confidences, same as Apple, Disney and Max.

Unlike media companies that keep their infrastructure secretive to protect consumer streaming edges, **Salesforce and SAP are actually very open about their cloud setups**—because their enterprise clients demand to know where their sensitive business data lives.

Instead of building massive global server farms to rival AWS or Azure, both companies largely rent space from the Big Three and pass the choice on to their corporate customers.

Salesforce: From Own Servers to "Hyperforce" on AWS and Google

For its first two decades, Salesforce built and ran its own private data centres. However, to scale globally and meet strict local data-privacy laws (like keeping Australian data inside Australia), Salesforce retired most of its own hardware and rebuilt its software on an architecture called **Hyperforce**.

- **Strong AWS Preference:** Salesforce Hyperforce relies heavily on **AWS** as its primary global backbone. Because Salesforce competes directly with Microsoft Dynamics in the CRM space, it has historically avoided relying heavily on Microsoft Azure.
- **Google Cloud Expansion:** To avoid single-vendor lock-in, Salesforce expanded Hyperforce onto **Google Cloud Platform (GCP)**.
- **Alibaba for China:** Because Western clouds face strict regulatory hurdles in mainland China, Salesforce explicitly partners with **Alibaba Cloud** to host Salesforce locally for Chinese operations.

SAP: The "4+1" Open-Choice Strategy

Because SAP runs the core financial and supply-chain systems (ERP) for the world's largest corporations, its customers refuse to be forced onto a single cloud. Instead of hiding its infrastructure, SAP actively markets a transparent **"4+1" strategy** through its flagship RISE with SAP program:

- **The "4" Hyperscalers:** Corporate customers explicitly choose whether they want SAP to host their system on **AWS**, **Microsoft Azure**, **Google Cloud**, or **Alibaba Cloud**.
- **The "+1" (SAP's Own Iron):** SAP still maintains its own private data centres (SAP Converged Cloud) for legacy systems, its public cloud edition, and ultra-sensitive clients who do not want their data on public hyperscalers.
- **Customer Preference Dictates the Cloud:** Unlike Disney+ (where you don't care which server streams your movie), an SAP client who already uses Microsoft Office and Teams will deliberately pick **Azure** for seamless integration. A company heavy on data analytics will pick **Google Cloud** to connect SAP directly to BigQuery.

Why They Don't Keep Secrets Like Apple or Disney

The reason Salesforce and SAP are transparent while Disney, Apple, and Max are secretive comes down to **who their customer is**:

Aspect	Consumer Media (Disney+, Max, Apple)	Enterprise Software (Salesforce, SAP)
Who pays the bill?	Everyday consumers buying a $15/month subscription.	Fortune 500 CIOs signing multi-million-dollar contracts.
Does the buyer care where the server is?	No, as long as the video doesn't buffer.	**Yes.** Laws and corporate boards require exact audits of where data sits.
Reason for secrecy vs. transparency	Secrecy protects pricing leverage with AWS/Google and hides custom anti-buffering tech.	Transparency is legally required for compliance, security audits, and data-sovereignty laws.

(Note: While Salesforce and SAP publicly tell clients which cloud provider hosts their data, they still keep their ***wholesale pricing discounts*** *with AWS, Google, and Azure strictly confidential.)*

Would you like to explore how **data sovereignty laws** (like GDPR or Australian privacy rules) force these tech giants to build local regions, or look at how **Oracle** broke away from the pack to become a major AI infrastructure provider?

Oracle is somewhat unusual, having SUN workstations, developing Java and Javascript at their earliest moments. A bit of a mini-IBM I sometimes think.

Calling Oracle a **"mini-IBM"** is a remarkably accurate description. Like IBM, Oracle grew up selling mission-critical software and hardware to massive enterprises, and it owns the underlying technology from the silicon up to the application layer.

However, Oracle's path to the cloud—and its current role in the tech ecosystem—is uniquely shaped by its aggressive acquisition strategy and its late, unconventional entry into cloud computing.

1. The "Mini-IBM" DNA: Full-Stack Control

When Oracle bought **Sun Microsystems** in 2010, it inherited a massive treasure trove of foundational computing technology—making it one of the few companies outside of IBM that could build an entire data centre from scratch:

- **The Hardware (SPARC & Exadata):** Through Sun, Oracle acquired SPARC processors and enterprise servers, allowing it to engineer custom hardware (like **Exadata**) specifically tuned to run Oracle databases faster than generic servers could.
- **The Software & Languages:** As you noted, Sun created **Java** (while **JavaScript** was actually created at Netscape by Brendan Eich, though Netscape and Sun partnered closely on branding and browser integration in 1995). Owning Java gave Oracle deep roots in the enterprise software world, alongside **Solaris** (Unix) and **MySQL**.

- **The IBM Parallel:** Just as IBM combined mainframes, DB2 databases, and consulting services into one monolithic stack, Oracle combined Sun hardware, the Oracle Database, and enterprise applications (PeopleSoft, Siebel, NetSuite) under one roof.

2. Why Oracle Cloud (OCI) Is Built Differently

Larry Ellison famously mocked cloud computing in 2008 as "complete gibberish," which meant Oracle was years late to the party behind AWS, Azure, and Google. Paradoxically, being late turned into an engineering advantage when Oracle built **Oracle Cloud Infrastructure (OCI)**:

- **Zero-Oversubscription & Bare Metal:** Early clouds (like first-gen AWS) relied heavily on virtualization—slicing one physical server into many virtual machines shared by different customers, which created "noisy neighbour" slowdowns. Because Oracle needed to run massive, unforgiving enterprise databases, it built OCI around **bare-metal servers** (dedicated physical machines with no hypervisor overhead) and ultra-fast, flat internal networking.
- **The Surprise AI Boom:** That exact architecture—raw, unshared hardware connected by ultra-low-latency networks—turned out to be ideal for training massive **AI models**. Today, companies like **xAI (Elon Musk)**, **Cohere**, and even **OpenAI** (via Microsoft partnerships) rent huge clusters of NVIDIA GPUs on Oracle Cloud because OCI's network architecture avoids bottlenecks better than older cloud designs.
- **Media Customers:** Even in streaming, Oracle carved out a niche—most notably, **Zoom** famously picked Oracle Cloud over AWS during the 2020 pandemic boom, and **TikTok** uses OCI to host its US user data ("Project Texas") due to Oracle's willingness to build isolated, custom-governed environments.

3. Oracle vs. IBM Today

While both share that "full-stack enterprise" heritage, their modern cloud strategies have diverged sharply:

Dimension	Oracle	IBM
Cloud Strategy	**Hyperscale Challenger.** Competes head-on with AWS/Azure on raw compute speed, databases, and massive AI GPU clusters.	**Hybrid & Open-Source.** Stepped back from competing directly with AWS on raw scale; bought **Red Hat** to manage software across *other* companies' clouds.
Hardware Legacy	Retired general-purpose SPARC development to focus on custom database appliances (Exadata) and commodity x86/ARM/GPU cloud servers.	Still dominates banking and airlines with **IBM Z Mainframes** and **Power** systems.
Database Lock-in	Uses its dominant **Oracle Database** to pull enterprise clients into OCI (and recently struck deals to embed Oracle hardware *inside* AWS, Azure, and Google data centres).	Relies on **DB2** and mainframe lock-in, paired with massive IT consulting (IBM Consulting).

Oracle's willingness to actually put its own custom Exadata hardware *inside* AWS, Google, and Microsoft data centres—so customers can run Oracle databases inside Azure or AWS without moving their apps—is a uniquely pragmatic move that neither IBM nor the Big Three would have imagined a decade ago.

Would you like to look closer at how **Oracle's multi-cloud deals** (putting Oracle servers inside AWS and Azure data centres) work, or explore how **IBM's Red Hat acquisition** changed its cloud strategy?

While Apple has always kept itself separate, and at a Unix level, still does. :-)

You hit the nail on the head—that Unix DNA is arguably the single best decision Steve Jobs made when he returned to Apple in 1997, and it explains why Apple remains such a fascinating contradiction today: **fiercely independent on the device, yet deeply pragmatic in the cloud.**

1. The Unix Foundation: NeXTSTEP to Apple Silicon

While Microsoft built Windows NT from scratch and Linux grew out of the open-source movement, Apple acquired **NeXT** and inherited a rock-solid Unix lineage (combining the **Mach** microkernel with **FreeBSD** userland into **Darwin**):

- **True POSIX Certification:** Unlike Linux (which is "Unix-like"), macOS is a fully certified **UNIX 03** operating system—putting it in the same historical bloodline as Sun's Solaris and IBM's AIX.
- **The Ultimate Portability Engine:** That clean Unix foundation is the secret weapon that allowed Apple to pull off three massive hardware transitions—PowerPC to Intel (2006), Intel to ARM on iPhone/iPad (2007), and Intel to Apple Silicon M-series chips (2020)—without breaking its developer ecosystem.
- **Walled Garden on Top of Open Roots:** Apple took open-source Unix plumbing, locked down the user interface and hardware integration (Cocoa, Metal, Secure Enclave), and created an ecosystem where it controls the entire stack from the **silicon transistor** up to the **App Store**.

2. Why Apple Keeps Devices Closed—and Clouds Outsourced

Apple's philosophy has always been to **own the primary technology** behind whatever differentiates the user experience:

- **Where Apple Builds (Silicon & OS):** Apple designs its own M-series and A-series chips, writes its own Unix-based OS, and even builds its own custom **Apple Silicon servers** inside its data centres specifically for **Private Cloud Compute** (so Apple Intelligence AI requests stay encrypted and stateless).
- **Where Apple Rents (Commodity Storage & Streaming):** For raw bulk storage (iCloud photos) and video delivery (Apple TV+), owning generic hard drives and global fiber cables doesn't differentiate an iPhone. So Apple happily rents massive capacity from **AWS and Google Cloud**—wrapping everything in its own encryption keys so neither Amazon nor Google can ever read the data.

It really is a unique hybrid: **1980s Unix purity** powering **2020s custom silicon**, backed by **rented hyperscale clouds** whenever raw bulk capacity is needed.

Would you like to look into how Apple uses its own **Apple Silicon servers for Private Cloud Compute**, or explore the history of how **BSD Unix** survived inside macOS and iOS?

It’s interesting, Microsoft is so huge, but the others gravitate to AWS and Google Cloud. Makes sense, Microsoft can change the interface without warning.

That is a sharp observation, and it actually points to a very real divide in how software engineers and infrastructure architects view Microsoft versus AWS and Google.

While Microsoft's enterprise licensing dominance is massive, companies building pure internet-scale services (like streaming apps or modern SaaS) often hesitate to build their core consumer infrastructure on Azure for several reasons—and that "Microsoft can change things on a whim" reputation plays a surprisingly big role.

1. The "Microsoft Ecosystem Gravity" Problem

AWS and Google Cloud were born on the open web (Linux, open-source databases, standard APIs), whereas Azure was born as an extension of **Windows Server, Active Directory, and .NET**:

- **Opinionated Tooling:** Microsoft historically favors tight integration within its own stack. If a company already uses Office 365, Teams, and Windows servers, Azure feels seamless. However, if a company's engineering team works on **macOS and Linux** (which describes almost every Silicon Valley startup, streaming service, and Apple/Disney engineering team), Azure's tooling, identity management (Entra ID / Active Directory), and portal workflows often feel clunky and overly complex.
- **Frequent UI & Naming Churn:** Your point about Microsoft changing interfaces without warning is a running joke among IT administrators. Microsoft frequently rebrands products, reorganizes menus, and changes licensing terms (for example, renaming *Azure Active Directory* to *Microsoft Entra ID*, or constantly shifting portals between Office, Defender, Intune, and Azure). AWS's web console may look utilitarian and dated, and Google Cloud's is starkly minimalist, but their underlying APIs and command-line tools rarely suffer from corporate rebranding chaos.

2. Competitive Paranoia ("Don't Feed Your Rival")

Even more than interface quirks, corporate rivalry dictates who uses which cloud:

- **Retailers Avoid AWS:** Walmart, Target, and Kroger refuse to use AWS because Amazon is their biggest retail threat—they overwhelmingly choose **Microsoft Azure** or **Google Cloud** instead.
- **Enterprise Software Rivals Avoid Azure:** Companies like **Salesforce**, **Workday**, and **Slack** compete directly with Microsoft (Dynamics, Viva, Teams). Giving Microsoft billions in cloud revenue helps subsidize the very competitor trying to put them out of business, so they gravitate toward **AWS and Google**.

- **Media & Consumer Tech:** Apple and Disney view Microsoft less as a direct streaming threat today, so their avoidance is mostly historical—AWS had a massive head start in video encoding (AWS Elemental) and content delivery (CloudFront) when Disney+ and Apple TV+ were being built.

3. Where Microsoft Actually Dominates

Even though consumer streaming and Silicon Valley startups lean toward AWS and Google, Azure quietly dominates the **traditional corporate world**:

- **Fortune 500 Lock-in:** Over 95% of Fortune 500 companies use Azure. Because CIOs already have multi-year Enterprise Agreements with Microsoft for Windows and Office, Microsoft bundles Azure credits into those renewals—making it financially effortless for banks, airlines, healthcare networks, and governments to adopt Azure.
- **The OpenAI / AI Catalyst:** Microsoft's early multibillion-dollar partnership with **OpenAI** gave Azure a massive advantage in enterprise AI, pulling huge numbers of corporate workloads onto Azure just to access GPT models inside a compliant corporate firewall.

In short: **AWS** owns the startups and the internet backbone, **Google** owns data-heavy and open-source native teams, and **Microsoft** owns the corporate boardroom—even if its engineers drive admins crazy by moving the buttons every six months.

Would you like to look at how **retail giants like Walmart** built their tech stacks specifically to avoid Amazon, or explore how **enterprise licensing bundles** give Microsoft its biggest edge?

We'll finish here. Thanks for the historic politics.

You're very welcome! It really is a fascinating blend of engineering philosophy, corporate history, and competitive turf wars once you pull back the curtain.

Enjoy the rest of your weekend, and feel free to drop back in anytime if another tech rabbit hole catches your eye!

If you ever want to revisit this topic, we could explore how **sovereign AI data centres** are reshaping global cloud politics.