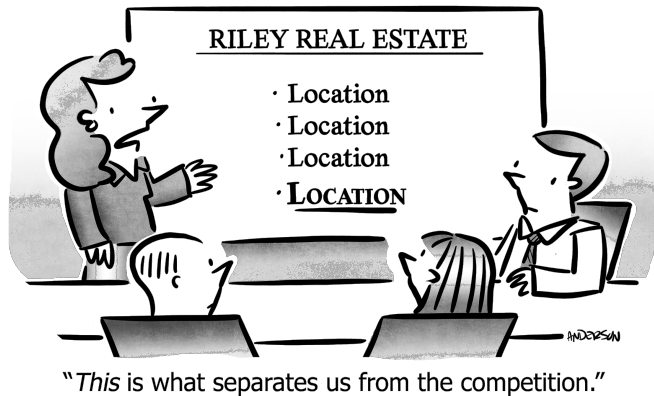# Moats: Durable competitive advantage

by Jason Cohen on May 7, 2022

Industries commoditize over time, delivering similar products at similar prices resulting in low profit. Moats are the antedote; your strategy must create some.

"*This* is what separates us from the competition."

Economics is useless for predicting what the market will do tomorrow, but it is prescient at predicting long-term forces that shape entire industries. Business strategists ignore this at their peril.

One of these forces is that industries commoditize: Companies copy the best ideas from each other, whether in features to attract customers or cost-savings to build profits, so they converge to similar products with similar cost structures. Undifferentiated in most dimensions, the flexible dimension is price, and in a real-life Prisoner's Dilemma, price reduces until profits are driven to zero. It's great for the consumer—the best products at the lowest prices—but bad for the companies.

To avoid homogenization, a company needs differentiation that others cannot copy. Best is differentiation that customers care about, because that should lead not only to higher prices (because customers cannot get the same value elsewhere) but also greater market share. Second-best is differentiation in cost structure, because this allows the company to convert those savings either into

profit (keep in the bank, or distribute by dividend or stock buy-back) or growth (spend in sales, marketing, or new product development).

Durable differentiation is rare, especially in the software industry where almost anything can be copied.

It is not even enough to have a competitive advantage; the advantage must also be *durable*. Snapchat invented the idea of "Stories," but Facebook and Instagram copied it, so the advantage didn't convert into market share:
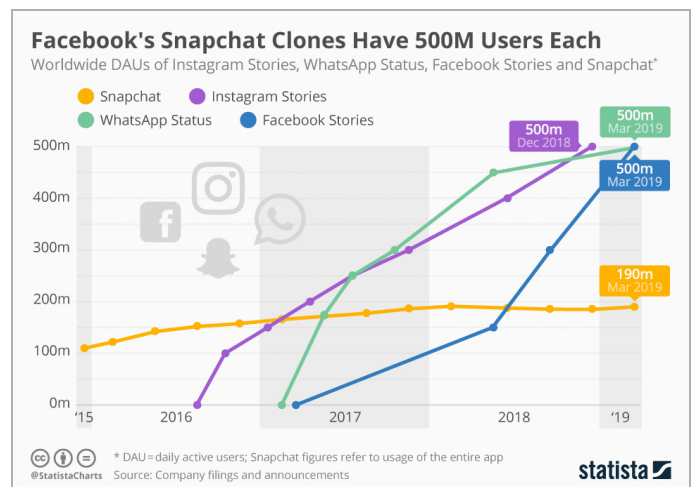
Facebook doesn't invent anything; they copy. But there is no "first mover advantage" unless that advantage is also durable, i.e. not easily copy-able by rivals.

A taxonomy of moats

> " *The most important thing to me is figuring out how big a moat there is around the business. What I love, of course, is a big castle and a big moat with piranhas and crocodiles.*"
>
> —*Warren Buffett*

Durable advantages are often called "moats," after Warren. The concept appears everywhere—a characteristic of fundamental truth—for example more recently and famously in Hamilton Helmer's Seven Powers.

Jerry Neuman collected, organized, and detailed a terrific taxonomy of moats. Such a reference is useful when analyzing other companies (which moats are they building?) or figuring out which moats are best for your business to construct.
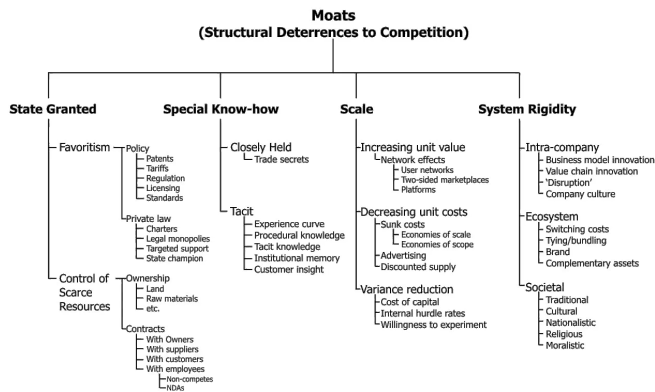
A real-world example will breath life into these trite bullet points.

Applying moats to a commoditized market: Cloud Computing

AWS (Amazon Web Services) was the first and is still the largest "cloud computing" provider. Its industry is an archetypical example of a market that commoditizes. Indeed, all computing infrastructure markets that preceded the cloud also commoditized, from individual components (e.g. RAM and disks) to entire physical data centers (with power, internet connectivity, climate-control, and physical security).

Cloud computing—paying only for what you need, by the hour, by the megabyte, scaling up and down at will—is a discrete leap forward in customer value, but commoditization is inescapable, as every by-the-hour CPU, gigabyte of disk, and gigabyte of network transfer from one vendor is the same as from another. If renting infrastructure

is the only strategy, the only dimension remaining to compete on is price, so price falls, and the industry commoditizes.

Indeed, that's exactly what has happened, for example in storage:
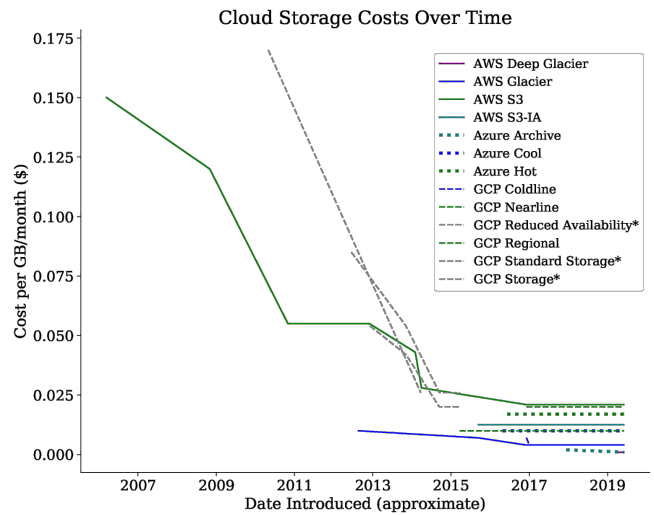
Different types of storage have different prices, but each type has identical price across vendors. All prices are low.

But AWS has largely avoided commoditization. The proof is in the market-share and in the profit. Every year since 2014, AWS's profit has been more than half of Amazon's entire profit; in 2022 it's generating $21B of profit at a 30% operating profit margin, and growing:


Amazon Web Services, 6-year financials

Source: Amazon Earnings Reports, In Millions Per Fiscal Quarter   GEEKWIRE

If it were possible for a competitor to steal business by undercutting prices, they would. Indeed, they *have*; competitors like Google, Microsoft, and Alibaba charge less for similar service. Google in particular spends billions more dollars per year than they make, plus they have all the requisite "smart people," but even at scale in 2022, on annual revenues of $20B for their cloud computing services, they're still *losing* nearly $4B annually. None of this is hurting AWS's profits, demonstrating that AWS has not commoditized.

Diverging Fortunes
Operating profit/loss for **Google Cloud** since the end of 2018 and for **AWS** from mid-2015, when both companies were generating nearly the same revenue*

*On a trailing 12-month basis
Source: Company filings
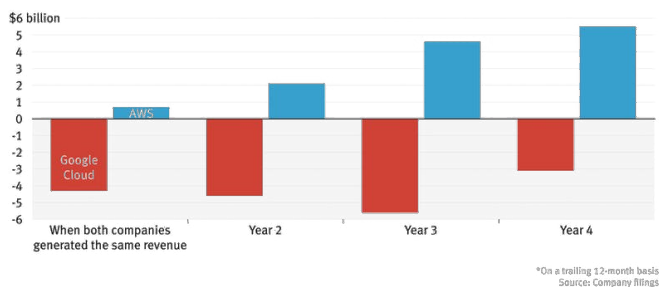
Why not? We might expect the answer is "moats," but what are they?

AWS Moats

In 2019, then-CEO of AWS (now-CEO of Amazon) Andy Jassy, on stage in a live interview with the great Kara Swisher, was asked directly: Why hasn't AWS commoditized? Jassy immediately fired off the moats from their anti-commoditization strategy, annotated here with the bullet points from the taxonomy diagram:

Innovates the fastest [Economies of Scale + Willingness to Experiment]
Jassy says they use their size to fuel more and faster innovation. Specifically, revenue today plus line-of-sight to revenue growth for tomorrow, means it is easy to justify a massive investment in innovation. The platform with the most innovation, will have the best tools for developers, which they believe is the key to winning.

Biggest Ecosystem [Complementary Assets]
The biggest vendor will attract the most 3rd-party support, which in turn is vital for developers who must integrate with other tools, especially in the modern world where everything is a SaaS.

Advanced operating maturity [Learning Curve]
Jassy says, "There is no compression algorithm for experience." Even Microsoft and Google, throwing billions of dollars and thousands of engineers at the challenges, are not able to overtake AWS.

There's another moat that Jassy didn't list, probably because he didn't want to talk about market-share[1], and possibly because it could be interpreted as gently insulting to their customers:

---

[1] Later in the interview, he refused to answer basic questions about how much market share AWS has, or the relative positions of Microsoft, Google, and Oracle, even under pressure from Kara Swisher, the excellent interviewer. Being the market-leader invites regulation and puts you on the defensive in anti-trust cases, so public companies shy away from such claims; for example Amazon owns 43% of all eCommerce at the time of this writing, but they publicly position themselves as a general "retail" company, so that their share is "only" 6%.

Brand of "The Leader" [Brand]
The old adage was "no one was ever fired for picking IBM." Now that's true of AWS. The idea is that if you pick the market leader, your career isn't in jeopardy if it doesn't go well, because you picked the least-risky choice. Often "the leader" is also "more expensive," so this is not only good for long-term differentiation, but for profitability.

There's another moat that is clearly part of every major cloud provider's strategy; perhaps Jassy didn't mention it because it's slightly nefarious, as it involves lock-in:

Has the biggest menu of technology to select from [Switching Costs]
If customers only buy commodity services like CPU, RAM, disk, and network, it's easy for them to leave for a cheaper cloud. So, AWS has created hundreds of "services" covering every imaginable corner in the world of infrastructure: big data storage and retrieval,

web analytics engines, AI algorithms, logging, alerting, change-management, source code repositories, network firewalls, and support for dozens of open source software packages, all for rent by the hour or gigabyte. Furthermore, they encourage use of these services by making each one free when each is only lightly used. A software developer who avails herself of this incredible array of services is more productive than a large team of developers would have been ten years ago, however her Faustian bargain is that she gives up her ability to switch to a competitor, which means as her project grows and the pricing tiers kick in, she's locked in.

Moats are integral to strategy

Moats don't appear quickly or by accident; they require consistent investment and prioritization over a course of years. This is exactly what a "strategy" is supposed to organize, and what "just be agile and react to whatever happens" will not. Therefore, a good strategy will have identified one or very few moats that the company will create, and the primary themes of action that will excavate them and fill them with Buffett's piranhas.

The example of AWS shows that the taxonomy is only a starting point. The details of exactly how one item applies to a specific company are essential. A strategy that states "We will create switching costs" is not a strategy; it is a statement of what you hope will become true, rather than an explanation of how you will cause that to become true. A strategy to intentionally create switching costs looks more like this:

Objective
Create permanent customers through high switching-costs, caused by extensive integration with their technology stacks, development workflows, and IT administration.

How
Maximize the number of cloud services each customer uses. This, in turn, requires:

1. Create the most number of services as quickly as possible, so there's more to integrate with. Must have more than the competition, and a reputation of "innovating faster than anyone else," to attract customers in the first place, and so that customers don't look elsewhere if we don't have their favorite service quite yet.
2. Create compelling "on-ramps" so that services are actually used (e.g. free tiers, great documentation, pre-integration with other APIs, "bundled offerings" that includes many APIs)

Consequences

1. If we build many services, quickly, some of those services will turn out to be duds, or might be strategic only as a part of the larger offering, not high-growth or even profitable on their own. This is a "cost of doing business" and will not constitute evidence that a team has failed or that the strategy is failing.
2. Many individual services will be commoditized, especially when we're just reselling existing open-source projects. We still need to create these, because integrating many of them into a single customer project is necessary to fulfill the objective, and customer discovery shows that they specifically want to buy those products from us. Creating genuinely unique services are more valuable and defensible, so we need to do that as well, but they also take 100x more effort.
3. Given that (1) and (2) demand a large quantity of software, built quickly, this is a costly bet; we require $X billion dollars over three years, plus a scaled-up hiring process.

The moat is named, but also how it will be built, along with second-order consequences (high-cost, and frequent failures of usage or of differentiation on a case-by-case basis) that we not only *accept*, but *expect*, in service of the objective of creating that moat. With these details and decisions in place, it is a strategy, not a hope.

Make sure your strategy is explicit about which moats are being constructed, and the major multi-year activities required to construct them.

"I told you to order the piranhas earlier."