

# You Are Already Living Inside a Computer

 [theatlantic.com/technology/archive/2017/09/you-are-already-living-inside-a-computer/539193/](https://theatlantic.com/technology/archive/2017/09/you-are-already-living-inside-a-computer/539193/)

Ian Bogost Sep 14, 2017



Like *The Atlantic*? Subscribe to [The Atlantic Daily](#), our free weekday email newsletter.

Suddenly, everything is a computer. Phones, of course, and televisions. Also toasters and door locks, baby monitors and juicers, doorbells and gas grills. Even [faucets](#). Even [garden hoses](#). Even [fidget spinners](#). Supposedly “smart” gadgets are everywhere, spreading the gospel of computation to everyday objects.

It’s enough to make the mundane seem new—for a time anyway. But quickly, doubts arise. Nobody really needs smartphone-operated bike locks or propane tanks. And they certainly don’t need gadgets that are less trustworthy than the “dumb” ones they replace, a sin many smart devices commit. But people do seem to *want* them—and in increasing numbers. There are now [billions](#) of connected devices, representing a market that [might reach](#) \$250 billion in value by 2020.

Why? One answer is that consumers buy what is on offer, and manufacturers are eager to turn their dumb devices smart. Doing so allows them more revenue, more control, and more opportunity for planned obsolescence. It also creates a secondary market for data collected by means of these devices. Roomba, for example, [hopes](#) to deduce floor plans from the movement of its robotic home vacuums so that it can sell them as business intelligence.

But market coercion isn’t a sufficient explanation. More so, the computational aspects of ordinary things have become goals unto themselves, rather than just a means to an end. As it spreads from desktops and back-offices to pockets, cameras, cars, and door locks, the affection people have with computers transfers onto other, even more ordinary objects. And the more people love using computers for everything, the more life feels incomplete unless it takes place inside them.

\* \* \*

A while back, I [wrote about](#) a device called GasWatch, a propane-tank scale that connects to a smartphone app. It promises to avert the threat of cookouts ruined by depleted gas tanks.

When seeing devices like this one, I used to be struck by how ridiculous they seemed, and how little their creators and customers appeared to notice, or care. Why use a computer to keep tabs on propane levels when a [cheap gauge](#) would suffice?

But now that internet-connected devices and services are increasingly the norm, ridicule seems toothless. Connected toasters promise to help people “toast smarter.” Smartphone-connected bike locks [vow](#) to “Eliminate the hassle and frustration of lost keys and forgotten combinations,” at the low price of just \$149.99. There’s Nest, the smart thermostat made by the former designer of the iPod and later bought by Google for \$3.2 billion. The company also makes home security cameras, which connect to the network to transmit video to their owners’ smartphones. Once self-contained, gizmos like baby monitors now boast internet access as an essential benefit.

The trend has spread faster than I expected. Several years ago, a stylish hotel I stayed at boasted that its keycards would soon be made obsolete by smartphones. Today, even the most humdrum Hampton Inn room can be opened with Hilton’s app. Home versions are available, too. One even [keeps](#) analytics on how long doors have been locked—data I didn’t realize I might ever need.

Computers already are predominant, human life already takes place mostly within them, and people are satisfied with the results.

These devices pose numerous problems. Cost is one. Like a cheap propane gauge, a traditional bike lock is a commodity. It can be had for \$10 to \$15, a tenth of the price of Nokē’s connected version. Security and privacy are others. The CIA [was rumored](#) to have a back door into Samsung TVs for spying. Disturbed people have [been caught](#) speaking to children over hacked baby monitors. A botnet [commandeered](#) thousands of poorly secured internet-of-things devices to launch a massive distributed denial-of-service attack against the domain-name system.

Reliability plagues internet-connected gadgets, too. When the network is down, or the app’s service isn’t reachable, or some other software behavior gets in the way, the products often cease to function properly—or at all.

Take doorbells. An ordinary doorbell closes a circuit that activates an electromagnet, which moves a piston to sound a bell. A smart doorbell called Ring replaces the button with a box containing a motion sensor and camera. Nice idea. But according to some users, Ring sometimes fails to sound the bell, or does so after a substantial delay, or even absent any visitor, like a poltergeist. This sort of thing is so common that there’s a popular Twitter account, [Internet of Shit](#), which catalogs connected gadgets’ shortcomings.

As the technology critic Nicholas Carr recently wisecracked, [these are not the robots we were promised](#). Flying cars, robotic homes, and faster-than-light travel still haven’t arrived. Meanwhile, newer dreams of what’s to come predict that humans and machines might meld, either through biohacking or simulated consciousness. That future also feels very far away—and perhaps impossible. Its remoteness might lessen the fear of an AI apocalypse, but it also obscures a certain truth about machines’ role in humankind’s destiny: Computers already are predominant, human life already plays out mostly within them, and people are satisfied with the results.

\* \* \*

The chasm between the ordinary and extraordinary uses of computers started almost 70 years ago, when Alan Turing proposed a gimmick that accidentally helped found the field of artificial intelligence. Turing guessed that machines would become most compelling when they became convincing companions, which is essentially what today’s smartphones (and smart toasters) do. But computer scientists missed the point by contorting Turing’s thought experiment into a challenge to simulate or replace the human mind.

In his 1950 [paper](#), Turing described a party game, which he called the imitation game. Two people, a man and a woman, would go behind closed doors, and another person outside would ask questions in an attempt to guess

which one was which. Turing then imagined a version in which one of the players behind the door is a human and the other a machine, like a computer. The computer passes the test if the human interlocutor can't tell which is which. As it institutionalized, the Turing test, as it is known, has come to focus on computer characters—the precursors of the chatbots now popular on Twitter and Facebook Messenger. There's even an [annual competition](#) for them. Some still cite the test as a legitimate way to validate machine intelligence.

But Turing never claimed that machines could think, let alone that they might equal the human mind. Rather, he surmised that machines might be able to exhibit convincing behavior. For Turing, that involves a machine's ability [to pass as something else](#). As computer science progressed, "passing" the Turing test came to imply success as if on a licensure exam, rather than accurately portraying a role.

That misinterpretation might have marked the end of Turing's vision of computers as convincing machines. But he also baked persuasion into the design of computer hardware itself. In 1936, Turing proposed a conceptual machine that manipulates symbols on a strip of tape according to a finite series of rules. The machine positions a head that can read and write symbols on discrete cells of the tape. Each symbol corresponds with an instruction, like writing or erasing, which the machine executes before moving to another cell on the tape.

People don't seek out computers in order to get things done; they do the things that let them use computers.

The design, known as the [universal Turing machine](#), became an influential model for computer processing. After a series of revisions by John von Neumann and others, it evolved into the stored-programming technique—a computer that keeps its program instructions as well as its data in memory.

In the history of computing, the Turing machine is usually considered an innovation independent from the Turing test. But they're connected. General computation entails a machine's ability to simulate any Turing machine (computer scientists call this feat [Turing completeness](#)). A Turing machine, and therefore a computer, is a machine that pretends to be another machine.

Think about the computing systems you use every day. All of them represent attempts to simulate something else. Like how Turing's original thinking machine strived to pass as a man or woman, a computer tries to pass, in a way, as another thing. As a calculator, for example, or a ledger, or a typewriter, or a telephone, or a camera, or a storefront, or a café.

After a while, successful simulated machines displace and overtake the machines they originally imitated. The word processor is no longer just a simulated typewriter or secretary, but a first-order tool for producing written materials of all kinds. Eventually, if they thrive, simulated machines become just machines.

Today, computation *overall* is doing this. There's not much work and play left that computers don't handle. And so, the computer is splitting from its origins as a means of symbol manipulation for productive and creative ends, and becoming an activity in its own right. Today, people don't seek out computers in order to get things done; they do the things that let them use computers.

\* \* \*

When the use of computers decouples from its ends and becomes a way of life, goals and problems only seem valid when they can be addressed and solved by computational systems. Internet-of-things gadgets offer one example of that new ideal. Another can be found in how Silicon Valley technology companies conceive of their products and services in the first place.

Take abusive behavior on a social networks as an example. Earlier this year, Chris Moody, Twitter's vice president of data strategy, [admitted](#), "We have had some abuse on the platform." Moody cited stopping abuse as the company's first priority, and then added, "But it's a very, very hard challenge." To address it, Twitter resolved to deploy IBM's Watson AI to scan for hate speech. Google has a similar effort. One of its labs has developed [Perspective](#), an "API that uses machine learning to spot abuse and harassment online."

People choose computers as intermediaries for the sensual delight of using computers.

Sometimes tech firms will make efforts like these a matter of business viability—the search for “scalable” solutions to products and services. When I asked Twitter about Moody’s comments, a spokesperson told me that the company uses a combination of computational and human systems when reviewing safety content, but they couldn’t share many specifics.

It sounds promising, but the results are mixed. Twitter [claims](#) it’s getting better at fighting abuse, but it still [seems to ignore](#) even the worst cases. And Google’s Perspective [can be fooled](#) by simple typos and negations.

Even though it’s in these companies’ [immediate business interest](#) to solve abuse problems now, making online spaces safer is assumed to require a computational answer. Human content moderation is [notoriously hard](#), admittedly. And the volume of content is so high, a matter Twitter emphasizes, that computational systems are needed to manage and operate them.

## PUBLICITÉ

[inRead invented by Teads](#)

But perhaps managing abuse is “a very, very hard challenge” largely *on account of* that assumption. The very idea of a global network of people able to interact with one another anonymously precludes efficient means of human intervention. Twitter’s answer assumes that their service, which is almost entirely automated by apps and servers, is perfectly fine—they just need to find the right method of computational management to build atop it. If computer automation is assumed to be the best or only answer, then of course only engineering solutions seem viable.

The Turing test works best when everyone *knows* the interlocutor is a computer but delights in that fact anyway.

Ultimately, it’s the same reason the GasWatch user wouldn’t choose a cheap, analog gauge to manage cookout planning. Why would anyone ever choose a solution that doesn’t involve computers, when computers are available? Propane tanks and bike locks are still edge cases, but ordinary digital services work similarly: The services people seek out are the ones that allow them to use computers to do things—from finding information to hailing a cab to ordering takeout. This is a feat of aesthetics as much as it is one of business. People choose computers as intermediaries for the sensual delight of using computers, not just as practical, efficient means for solving problems.

That’s how to understand the purpose of all those seemingly purposeless or broken services, apps, and internet-of-things devices: They place a computer where one was previously missing. They transform worldly experiences into experiences of computing. Instead of machines trying to convince humans that they are people, machines now hope to convince humans that they are really computers. It’s the Turing test flipped on its head.

\* \* \*

There’s a name for that, as it happens: the “reverse Turing test.” [CAPTCHAs](#), those codes in online forms that filter out automated bots, are reverse Turing tests in which the computer judges whether a user is a human. There are also reverse Turing tests [in which](#) people try to guess which actor is a human among a group of computers.

These works use the Turing test as an experience in its own right, rather than a measure of intelligence. That precedent dates to the early days of computing. One of the most famous examples of imitation game-style chatterbots is Joseph Weizenbaum’s 1966 program ELIZA. The program acts like a Rogerian therapist—a kind of psychotherapy built on posing clients’ questions back to them. It’s an easy pattern to model, even in the mid-1960s (“What would it mean to you if you got a new line printer?” ELIZA responds heroically to a user [pretending to be an IBM 370 mainframe](#)), but it hardly counts as intelligence, artificial or otherwise. The Turing test works best when everyone *knows* the interlocutor is a computer but delights in that fact anyway.

Doorbells and cars and taxis hardly vanish in the process. Instead, they just get moved inside of computers.

“Being a computer” means something different today than in 1950, when Turing proposed the imitation game. Contra the technical prerequisites of artificial intelligence, acting like a computer often involves little more than

moving bits of data around, or acting as a controller or actuator. Grill as computer, bike lock as computer, television as computer. An intermediary.

Take Uber. The ride-hailing giant's main business success comes from end-running labor and livery policy. But its aesthetic success comes from allowing people to hail cars by means of smartphones. Not having to talk to anyone on the phone is a part of this appeal. But so is seeing the car approach on a handheld, digital map. Likewise, to those who embrace them, autonomous vehicles appeal not only because they might release people from the burden and danger of driving, but also because they make cars more like computers. Of course, computers have helped cars run for years. But self-driving cars turn vehicles into machines people *know* are run by computers.

Or consider doorbells once more. Forget Ring, the doorbell has already retired in favor of the computer. When my kids' friends visit, they just text a request to come open the door. The doorbell has become computerized without even being connected to an app or to the internet. Call it "disruption" if you must, but doorbells and cars and taxis hardly vanish in the process. Instead, they just get moved inside of computers, where they can produce new affections.

The present status of intelligent machines is more powerful than any future robot apocalypse.

One such affection is the pleasure of connectivity. You don't want to be offline. Why would you want your toaster or doorbell to suffer the same fate? Today, computational absorption is an ideal. The ultimate dream is to be online all the time, or at least connected to a computational machine of some kind.

This is not where anyone thought computing would end up. Early dystopic scenarios cautioned that the computer could become a bureaucrat or a fascist, reducing human behavior to the predetermined capacities of a dumb machine. Or else, that obsessive computer use would be deadening, sucking humans into narcotic detachment.

Those fears persist to some extent, partly because they have been somewhat realized. But they have also been inverted. Being away from them now feels deadening, rather than being attached to them without end. And thus, the actions computers take become self-referential: to turn more and more things into computers to prolong that connection.

\* \* \*

This new cyberpunk dystopia is more *Stepford Wives*, less William Gibson. Everything continues as it was before, but people treat reality as if it were in a computer.

When seen in this light, all the issues of contemporary technology culture—corporate data aggregation, privacy, what I've previously called [hyperemployment](#) (the invisible, free labor people donate to Facebook and Google and others)—these are not exploitations anymore, but just the outcomes people have chosen, whether through deliberation or accident.

Among futurists, the promise (or threat) of computer revolution has often been pegged to massive advances in artificial intelligence. The philosopher Nick Bostrom has a name for AI beyond human capacity: "superintelligence." Once superintelligence is achieved, humanity is either one step away from rescuing itself from the drudgery of work forever, or it's one step away from annihilation via robot apocalypse. Another take, advocated by the philosopher of mind David Chalmers and the computer scientist Ray Kurzweil, is the "singularity," the idea that with a sufficient processing power, computers will be able to simulate human minds. If this were the case, people could upload their consciousness into machines and, in theory, live forever—at least for a certain definition of living. Kurzweil now works at Google, which operates [a division devoted to human immortality](#).

Some even believe that superintelligence is a technology of the past rather than the future. Over millions of years, a computer simulation of sufficient size and complexity might have been developed, encompassing the entirety of what Earthly humans call the universe. The simulation hypothesis, as this theory is known, is of a piece with many ancient takes on the possibility that reality is an illusion.

But the real present status of intelligent machines is both humdrum and more powerful than any future robot apocalypse. Turing is often called the father of AI, but he only implied that machines might become compelling enough to inspire interaction. That hardly counts as intelligence, artificial or real. It's also far easier to achieve. Computers already have persuaded people to move their lives inside of them. The machines didn't need to make people immortal, or promise to serve their every whim, or to threaten to destroy them absent assent. They just needed to become a sufficient part of everything human beings do such that they can't—or won't—imagine doing those things without them.

There's some tragedy in this future. And it's not that people might fail to plan for the robot apocalypse, or that they might die instead of uploading. The real threat of computers isn't that they might overtake and destroy humanity with their future power and intelligence. It's that they might remain just as ordinary and impotent as they are today, and yet overtake us anyway.

---