BoTSL#11 2016 JUL 01 2:17 PM

This bulletin was first published at the foot of a flag flying at 019, an artist-run space housed in a former welding factory building in the Port of Ghent, Belgium. The 7 × 10 m flag, designed by Dexter Sinister, was the second in a series of ongoing commissions. The text was released via a USB stick, installed as a "dead drop" at the base of the flagpole.

Bruno Latour's "The Berlin key or how to do words with things" was published (with the first half of its dual title clipped off) in *Bulletins of The Serving Library* #3 and naturally remains shelved online at www.servinglibrary.org.

Cover: A Universal Serial Bus memory stick, plugged into the wall at Kunstverein München sometime around November 2015.

Data is shuttled back and forth and in-between computers on a "bus." Take the Universal Serial Bus, for example. USB is made up of a physical specification plus a virtual protocol for communication between digital devices. It was cooperatively developed beginning in 1994 by seven leading computer companies: Compaq, DEC, IBM, Intel, Microsoft, NEC, and Nortel. Prior to then, connecting a desktop personal computer to an external peripheral might require installing custom software drivers, flipping dip switches, even cutting and re-soldering wires. USB was designed to make the process simpler for the user and more compatible for manufacturers. It succeeded. This essay, conceived 22 years after USB, is about the intercourse between machines and why *how* they do it matters not only for them but also to us.

Before USB, new bus standards had been introduced to improve specific cases, but solutions were too often bespoke to their function. The result was a pile of competing standards for plugging in different devices and different computers, including SCSI, Parallel, Serial, inPort, Apple Desktop Bus, IEEE 1394 (FireWire), GeoPort, PCI, PnP ISA, and so on. By 1994, it was clear that no computer was an island, and a common connection standard was urgently required.

From the outset this new model was designed to be truly *universal.* It would allow all types of new electronic products to easily plug in and communicate with each other. There would be no limit to the number of devices that might be chained together with hubs and cables, so a single computer port could be extended by any number of connections. And the specification was designed openly enough to work not only with devices then currently on the market, but with future ones too. Time's arrow dictates it's no simple engineering task to design a current product in light of products yet to come; but when faced with a choice, USB's engineers routinely chose lower-tech solutions that would allow the standard to easily evolve, as the devices it connected inevitably would.

By January 1996, Universal Serial Bus Specification Revision 1.0 was released. It lists three principal considerations in designing the protocol: ease of use for both users and manufacturers; "infinite" expansion via hubs and cables; and PC-to-telephone connection (which was never substantially implemented). As digital life at the time centered around personal desktop computers rather than today's laptops and phones, USB was initially intended to simply connect that bulky PC to the outside world via any number of devices. A partial list from the time would include: hard drive, three-button mouse with scroll wheel, keyboard, numeric keypad, digital camera, MP3 player, zip drive, scanner, and laser printer.

In the years since that initial release, this inventory has grown extensively (to say the least) to include: Wi-Fi router, iPad, flatscreen monitor, barcode reader, thumb drive, desktop fan, single-can refrigerator, personal vacuum, LED light, and of course, telephone. In many ways the "peripherals" USB was designed to connect have since moved to the center. If the standard was developed primarily to plug phones into computers, now that the phone *is* the computer, USB has been duly adapted for connecting smartphones to other devices such as laptops, speakers, lights, portable chargers, and selfie sticks.

. . .

It's important to understand that the new standard was more than just a physical form, or connector design. It was also a data transfer protocol. A crisp distinction was drawn between hardware and software, and this has proven crucial to the format's longevity. By outlining a set of principles that keeps hardware and software at arm's length, then defining loose terms for them to work together, USB was able to adapt to fundamentally different KINDS of upcoming improvements, which inevitably develop at different rates.

USB is asymmetrical, which means that one host is connected to an extendable number of ports. Devices are connected to the host at the ports, and one computer can be attached to up to 127 USB devices with a single physical connector. These devices might include additional USB hubs, which serve as further hosts, allowing even more connections. All of this is organized in what is called a "tiered-star topology." You might think of it like a tree and its branches.

The hardware spec defines the PHYSICAL FORM of the cables, their connectors, and the ports they plug into. In order to implement its one-way architecture, USB cables have different connectors at either end, identified as Type-A and Type-B connectors. The larger Type-A plug carries data and provides power, while the smaller Type-B plug carries data only. This disparity is fundamental, as connecting two power-supplying USB devices to each other creates a voltage spike that can fry the electronics at either end.

Meanwhile, the software spec establishes a VIRTUAL PROTOCOL of communication. A computer bus is either parallel (multiple bits at once) or serial (one bit at a time). Although necessarily slower, a serial bus allows for more robust error checking and is more readily adaptable as the underlying technologies evolve. This serial approach — finishing one thing before starting the next — extends to all the USB's communication procedures. Doing one thing at a time is less efficient than multitasking, but the results are more reliably correct, and this matters a lot when transferring data. A typical exchange works something like this:

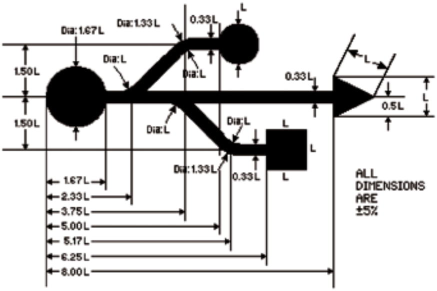
The host looks for all connected devices and assigns each a specific address. Once the device is located, the host sends a "handshake" sequence, which is received by the device and returned. The connection is then established and a transfer speed set. Next, the host sends a request to the device, which is answered in turn, and data sent as a response. This cycle of request and response continues until all the data has been transferred. Nothing can ever be sent by a USB device without first having received an explicit request from the host.

Think of the software protocol as similar to a busy intersection directed by a traffic cop. The cop is the host — she decides who goes where, and when; the cars are the data; and the merging roads are connected devices. No vehicle can proceed through the intersection without an explicit wave from the white gloves in charge.

Both hardware and software specs reinforce USB's lopsided design — the hardware via the cable's different connectors and corresponding peripheral ports, and the software via its "speak when spoken to" decorum that ensures all devices are subordinated to the master host controller. Because software develops far more quickly than the hardware it runs on, the split standard allowed for them to update at different rates. It allowed USB's physical form to evolve as technology marched forward and new products appeared on the market, and its virtual form to adapt in line with parallel advances such as the costs of chip sets, microcontrollers, and operating systems.

. . .

The industry acceptance of USB was considerably accelerated by the release of so-called "legacy-free" computers. Apple was first: their original iMac arrived in 1998 with *only* USB device connections, forsaking its own previous proprietary bus formats, and no floppy disk drive. This was the first new Apple computer produced after Steve Jobs returned to the company. Conceived by the firm's design lead Jonathan Ive, it looked absolutely nothing like other computers at the time, with a clear case to show off its innards, and in a range of colors with dubious names like "Bondi Blue." In a virtuous circle (or is it a vicious cycle?), such USB-only computers fueled a boom in the development and manufacture of USB peripheral devices, and the format was well on its way to fulfilling its claim to universality. An early adopter, Michael Slater of Microdesign Resources outlined the prospects of the new technology in a 1996 speech to an Intel user group: "It has been designed for today's PCs and with a view towards what PCs are going to be doing in the next 5, maybe 10 years … It's an essentially infinite connection."



The initial USB specification also included logo artwork. According to the spec's authors, the three-pronged graphic trident that mirrors the one-to-many logic of the USB interface was a collaborative effort. In a feat of symbolic geometry that would have been at home at the Bauhaus in 1932, a large circle identifies the host, forking lines indicate constituent ports, while a smaller circle, triangle, and square represent downstream devices.

The most common of these devices is a memory stick, a.k.a. flash drive, thumb drive, or USB key — a small, self-contained, self-powered solid state hard drive and microcontroller used both for simple storage, and the transfer of files from one machine to another. The first USB drive on the market was IBM's "DiskOnKey." These drives are often called "keys" in part because they are approximately the size and shape of a lock key, and often designed to be easily attached to a key ring. More significantly, they are named this because they are used to carry encryption certificates required by software installed on the host device. These certificates are paired to a specific copy of software, meaning the program will run only when the (digital) key fits.

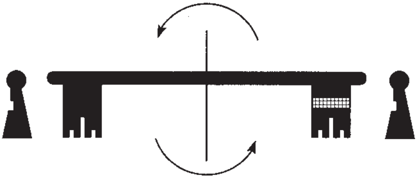
BERLIN KEY

In his 1991 detective essay "The Berlin key or how to do words with things," French sociologist Bruno Latour writes about a very particular key found in Berlin and its suburbs:

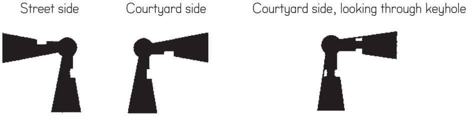
> **"What is this thing? What's it used for? Why a key with two bits? And two symmetrical bits? Who are they trying to kid?" ... The archaeologist turns the Berlin key over and over in her hands.**

The key opens the outside front doors of certain apartment buildings in the city — an idea that seems almost quaint today, when electronic keypads (or doormen) are more often tasked with granting or denying such access. Still, it's a markedly curious object, this key, which at first glance appears to have two perfectly symmetrical heads. This is not how we expect a key should look.

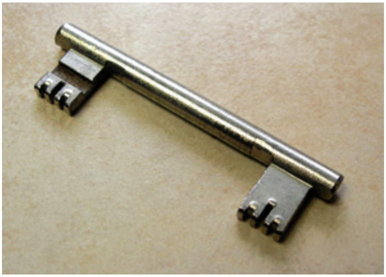
On closer inspection, however, the archaeologist protagonist notices something — the key is not quite so symmetrical as it first seems. On opposing sides of each head is a small crevice, a lip of sorts, which, as it turns out is carefully mated to a corresponding bump in the keyhole.



Stick this key in the lock it is meant to open, turn it 270° counterclockwise and it unlocks the door as expected. But because of the lips on either side of the lock, once the door is opened, it is no longer possible to withdraw the key as in any standard lock.



Instead, the Berliner who has mastered this entry technology first turns the key to unlock the door, then PUSHES it through the lock from the street side to the courtyard side, turns the key another 270° from the courtyard side, and withdraws the key. This additional counter-clockwise rotation locks the door once again when the apartment dweller is safely on the inside. It's a complicated transaction brokered by an austere and strangely-shaped piece of steel.



There's more. A second key, not issued to building residents, is used by the building's concierge. This key is not exactly the same as the first: it has a single head and a slightly thinner body. So, where the two-headed Berlin key requires its user to perform a symmetric action (if locked, it must be re-locked, if unlocked, it must be left unlocked), the concierge's key can *change the status of the overall mechanism.* It is thinner, and not obstructed by the lip in the keyhole, which means it can be inserted, the door locked and then withdrawn all on one side of the door—just like a "normal" key. Turns out, the building concierge typically flips the lock setting twice a day at 10 am, leaving it unlocked, and at 10 pm, locking it again for the night. Any tenant using their two-headed key is therefore obliged (i.e. mechanically coerced) to leave the lock on the front door in the condition that they found it in order to either enter or exit. So in the morning the door always remains open, and at night it is always locked.

According to Latour, this is the *script* of a device, or its *program of action.* The Berlin key's script says, "Please bolt the door behind you during the night and never during the day." Although it can be expressed in language, in this case the script is written not in words but in the idiosyncratic form of the metal key. And it seems that the physical, mechanical, technological form of the Berlin key turns out to be considerably more efficient than words. A sign that says "LOCK THE DOOR BEHIND YOU!" can be easily ignored, removed, or misunderstood. It also works better than a person: a rogue doorman might be fooled, bribed, fall asleep, or slip out for a quick drink.

But the Janus-faced key reliably regulates relations between building residents and managers, between delivery people and inhabitants, between tenants and owners, between dwellers and burglars. Says Latour: "It is because the social cannot be constructed WITH the social that it needs keys and locks." He adds, however, that the key is not simply a "medium" for this social exchange:

> **If the key is an intermediary, it does nothing in itself except carry, transport, shift, incarnate, express, reify, objectify, reflect, the meaning of the phrase: "Lock the door behind you during the night, and never during the day." …**

rather, it's a mediator:

> **Then the meaning is no longer simply transported by the medium but in part constituted, moved, recreated, modified, in short expressed and betrayed. No, the asymmetrical slot of the keyhole and the key with two bits do not "express," "symbolize," "reflect," "reify," "objectify," "incarnate" disciplinary relations, they *make* them, they *form* them … the mediator becomes at once means *and* end.**

The Berlin key enforces a certain action in its various users, and in doing so establishes relations *between* them. But it encodes something considerably more complicated and persistent than the cool term "social behavior" suggests—namely, etiquette. Etiquette is not a set of governing rules, but a set of expectations (social norms, conventions) that suggest how to act. It's a collective constitution rather than an individual legislation. It's the principle, or set of principles, upon which individual decisions are based.

If the physical design of the Berlin key dictates that anyone entering an apartment after 10 pm will lock the door behind them, it simultaneously implies that residents should not welcome visitors after that time in the evening. A similar soft coercion applies in the morning, when apartment residents must leave the front door open from 10 am onwards: sleeping late is not encouraged.

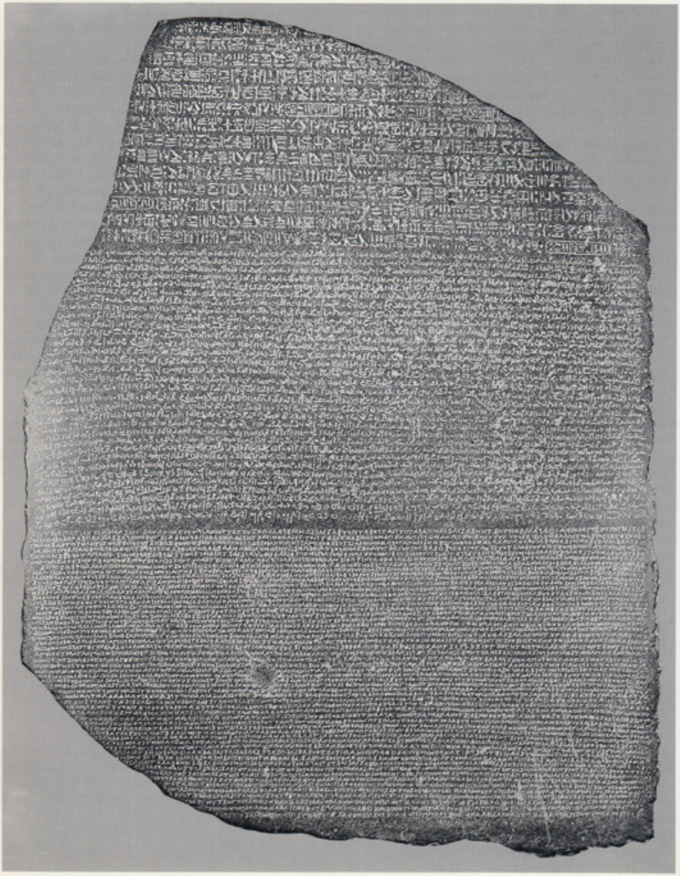
Of course, as a Berlin key-carrying resident, you CAN still welcome visitors (only you'll have to schlep out with your key to let them in), just as you CAN sleep in if you want (only you'll be more likely to be awakened by people coming and going—or burglars). The key doesn't strictly ENFORCE certain behaviors; it slyly PROMOTES them. It's the difference between what you *could* do and what you *should* do. And again, the same etiquette can be just as easily "promoted" by immaterial language than a material object. Indeed, Latour stops at the electronic keypad that lets him into his own apartment in Paris and wonders how its digital entry system relates to the Berlin key:

Which is more fragile, "45-68E" (my door code) or the handsome steel key? Which is more technological, the steel or the little counting-rhyme "end of the war, May '68, Europe" which I say over to myself at the end of the day in order to remember the thing that authorizes me to reenter my house? Which of the two, this solid key or that mnemotechnological counting-rhyme wired into my neurons, is more durable?

KEY/WORDS

In *Design Interface* (1987), Italian writer Gianni Barbacetto coined a well-punctuated neologism, "key/word," to name an emerging linguistic category. Yes, this new "word" includes a forward slash, "/," which already suggests it doesn't work exactly the same as other words. Indeed, key/words are magic words, brand new terms whose enthusiastic adoption and wide use dilutes their meanings. The jargon of new technology is filled with them. Sloppily defined and exaggeratedly profiled, these words are traps, yet the fact that they are so quickly assimilated and madly contested affords them a certain power. They have latent potential as "skeleton-keys to open doors of all sorts," as Barbacetto puts it—including the networks of references and conditions that produced the terms in the first place.

The Rosetta Stone (British Museum, London)



"Interface" is one of these words, he says, and 30 years on it remains an extraordinarily elastic example. Definitions from chemistry, theater, fashion, and computer science variously describe it as "a shared boundary," "a contact surface," "a border condition," and "a process or active threshold." After offering up "interface" as a prime example of a key/word, Barbacetto devotes the rest of his book to unpacking it. He begins:

Whatever lies between is called interface, whatever allows us to link two different elements, to reconcile them, to put them into communication.

Facing this definition is a reproduction of the Rosetta Stone—an ur-interface that facilitated communication between otherwise irreconcilable languages and cultures. King Ptolemy V's decree is etched into the rock in three different languages, each with a unique script—Egyptian hieroglyphs, Demotic letters from the Nile Delta, and ancient Greek. The crystal clear interface offers a model for translation: three equivalent horizontal bands establish a non-hierarchical relationship between three places, languages, and writing systems, all on a single stone surface.

Another exemplary interface in Barbacetto's book is St. Mark's Clocktower in Venice. Built in St. Mark's Plaza in the late 15th century on the main square facing the lagoon, the tower's facade contains a number of distinct "clocks." At the top are two bronze figures, one old and one young (to mark the passing of time) who strike the hours on a central bell. The next level is home to a winged lion, and the level below that contains a "digital" clock constructed from bright blue ceramic tiles. This display presents the current hour in Roman numerals on the left, and the current quarter hour in Arabic numbers on the right. The tiles are changed manually behind the scenes every 15 minutes. Twice a year, on the Epiphany and the Ascension, three Magi and an angel emerge from the number tiles and parade around a semicircular path to announce these two key days in the Catholic year. Another level down is the great clock face containing one ring with 24 hours marked, another with the signs of the Zodiac showing the sun's position in the year, and an inner ring with the monthly phases of the moon.

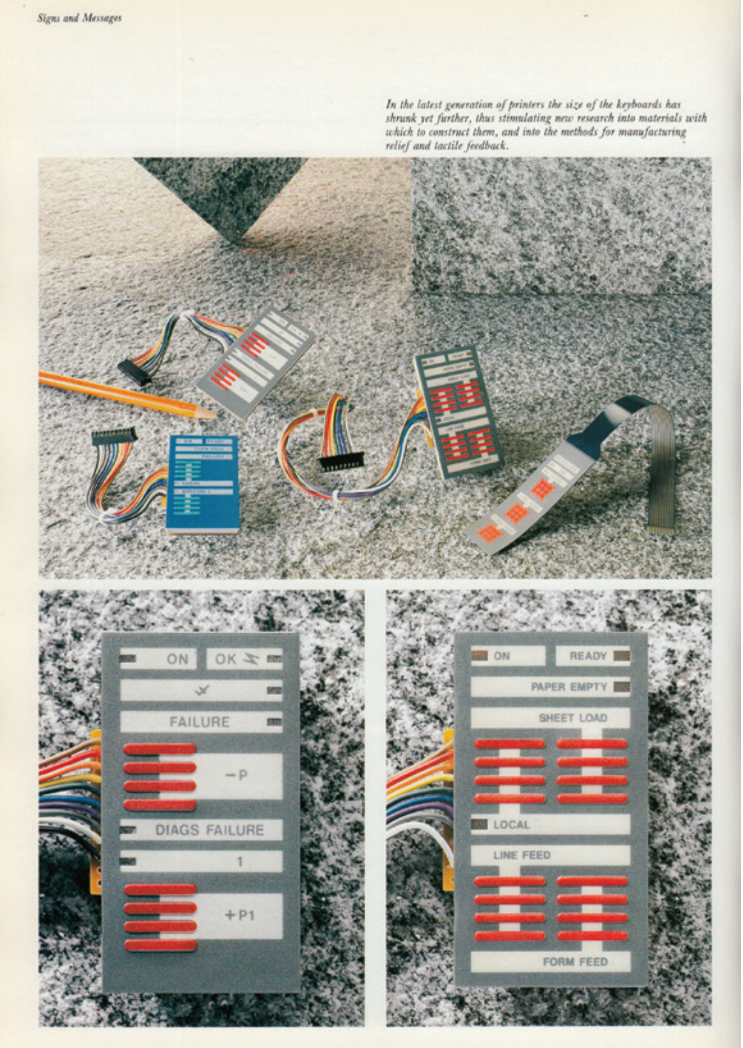
This overachieving clock has provided a regular, legible, public, and shared rhythm for Venetians for over five centuries, interfacing the daily activities of the merchant city with the procession of calendar, solar, and religious time. As such, the clock does not merely "represent" or "express" the time; rather, it *produces* it. Without a communal form of public display, the mechanical time of a calendar would remain a private and therefore relatively useless abstraction. The tower's interface allows the citizens of Venice to forge social relations, manufacturing meetings, deliveries, ceremonies, and other events on the social schedule.

. . .

The focus of Barbacetto's book, however, is a considerably more recent and straightforward instance. *Design Interface* chronicles an early attempt to design user controls for Italian firm Olivetti's upcoming product line. Industrial designers Perry King and Santiago Miranda were tasked with designing a set of user operator controls (button layouts) for a new range of Olivetti photocopiers. These new machines were electronically controlled, and so no longer required physical buttons to trigger the mechanical operation of the machine. Unlike the typewriters or adding machines that had made Olivetti's name, the controls on these new electronic products merely close a circuit that sends an instruction to produce an action. This was a new human-machine interface that required a new form.

King and Miranda were ambitious: they set out to completely reconsider the human-machine interface not as a site for human control of the machine, but rather for exchanging information in two directions *between* the two parties. They proposed a single continuous information surface, a zone of exchange articulated by schematic shapes and colors, and rendered as a single membrane in flexible Lexan plastic connected to the central processing unit. There was a clear separation of areas dedicated to user control (with buttons that triggered actions) and machine status (that displayed the current configuration).

The Clocktower in St. Mark's Square in Venice. With public display of clocks on city towers, the first "machines" made available to the populace at large, the relationship of men with time began to change. "Cyclical," lunar, and seasonal time was replaced by an image of linear, progressive, and cumulative time.



Signs and Messages

In the latest generation of printers the size of the keyboards has shrunk yet further, thus stimulating new research into materials with which to construct them, and into the methods for manufacturing relief and tactile feedback.

Equivalent “instructions” on Olivetti’s former mechanical products were typically printed directly onto the relevant buttons (whether A or STOP or RETURN), which meant that carrying out the function — pressing the button — involved covering up the information with a fingertip. For the new interface, however, King and Miranda insisted that information and action were kept separate to allow for the possibility of the machine communicating back to the human. For instance, a status indicator light on the “copy” button was kept apart from the active surface, so any error in the process could be immediately and clearly conveyed to the user. The same displays were designed to indicate the number of copies requested, how long the user would have to wait, ink shortages, mechanical errors, and so on. This was no longer a mechanical but an *informational* interface — a grey area where two discrete entities can communicate:

The user asks the machine to make a copy. The machine replies with a visual confirmation in the lighting of an LED (a sort of thumbs-up). The user indicates how many times to repeat, perhaps to combine multiple pages, and collate. The machine responds with a confirmation of each instruction, tells the user how long to wait, confirms how many copies have been made while running, and notifies when the job is finished. It might suddenly warn of a paper jam … at which point the human-machine interface becomes considerably more intimate.

King and Miranda’s human-machine interface consists not ONLY in molded plastic and applied graphics, any more than the human-human interface of a Berlin key consists only in steel, or the machine-machine interface of a Universal Serial Bus consists only in silicon. Each of these material designs also requires its virtual script, which is, in turn, betrayed by the form of the object. The back and forth between the Olivetti copier and its user is visible in the control panel’s segregated layout; the Berlin key’s script is written into the grooves of its symmetrical head; and the “program of action” that drives the Universal Serial Bus lies in its asymmetrical cable ends and unidirectional port design. It’s true enough that such implicit use is embedded into the physical form of all kinds of things. The handle of a mug tells a user how to lift it, or the two ends of a pencil suggest a more forgiving manner of writing. But things get considerably more complicated where software's concerned. From the binary informational atoms of 0 or 1, to machine language, assembly code, compilers, languages, and APIs, software progressively abstracts itself further and further from its hardware shells. Counterintuitively, perhaps, software necessarily becomes more technically baroque with each improvement that makes it simpler to use. That is, our machines are not getting any simpler, even as their user interfaces appear to be.

. . .

So let's return to the Universal Serial Bus now, 20 years after it first began connecting one machine to another. While those original seven computer companies that developed the standard are either finished or barely shadows of their 1996 selves, USB remains robust. Version 3.1 was released in 2013, and the league of peripherals it now connects — phones, computers, and myriad other electronic devices — has expanded perhaps a hundredfold and shows no sign of slowing down. USB has become the plain vanilla of computer connections, the simplest and most ubiquitous way for a couple of devices to get it on.

The 3.1 spec heralded a new cable and connector standard, USB-C, more glamorously known as "SuperSpeed+." It promises to be The One — a truly universal cable for connecting all kinds of electronics. The industry is enthusiastically embracing this new standard to the extent that Apple, again with characteristic bravado, has released its new MacBook ultra portable laptops with a single solitary USB-C port for power and peripherals, and no other connections whatsoever. In "Meet the World's First Truly Universal Cable," *Scientific American* columnist David Pogue pronounced: "The best new technology of the past year wasn't some phone or an app. Believe it or not, it was a new kind of USB cable." He continues to inventorize the cost savings and environmental dividend of an industry settling on one universally accepted solution.

However, blogger Andrew Cunningham disagrees. In "A brief history of USB, what it replaced, and what has failed to replace it," posted on tech blog *Ars Technica*, Cunningham recounts stories of USB's remarkable ability to adapt to changing conditions, and suggests that what may be its undoing is not a new standard, a new cable, an updated, faster, more robust connector, but rather the complete disappearance of any physical connector at all. Wi-Fi, AirDrop, AirPlay, and Bluetooth are increasingly essential and can take care of most of what the physical USB interface previously facilitated. Data is increasingly stored on remote servers (the "cloud") and transferred wirelessly over the internet to our devices as and when required.

It turns out USB was even ahead of this turn: wireless USB has been developed and promoted since 2005, though it never really caught on. But something called Media Agnostic USB (MA USB), released only three years ago in 2013 by the USB Implementers Forum, shows much more promise. As the name suggests, this standard retains USB's fundamental "speak when spoken to" protocol, but allows the interface to work over any existing medium without prejudice—including wireless connections like Wi-Fi and Bluetooth.

Recall that a deliberate communication scheme is implemented between USB devices with a clear division between the traffic-cop host and its dependent ports. No data is ever sent in advance of a request. There's a certain politeness and order to the whole exchange: it's clear, forgiving, resilient—and consequently mirrored in many other systems.
A similar client-server relation between a local machine and remote host is the root-level of the world wide web, for instance; while a mobile phone likewise identifies itself on a network by initiating a call and response loop with nearby cell towers. Even the communication between

a user account and its operating system relies on the same ingratiating manners.

An interface exists, only and always, *between.* It is a grey area, an adjustable relation, an agreement; it is a contract written in silicon, steel, plastic, or binary code that intimates how humans and machines should communicate. We know (approximately) how humans should speak to each other, and we are busy figuring out how humans should communicate with machines. The Universal Serial Bus suggests how machines should talk to other machines—a strange new decorum that becomes ever more important as the relative global population of humans and electronic devices continues to shift. If etiquette guides our responses in changing circumstances, it is particularly important in today's fast-moving world of electronic machine-to-machine communication.

Which is more durable?, mused Bruno Latour back in Berlin: material object or immaterial code? Clearly, it's the latter. "All that's solid melts into air," said Karl Marx, and it's clear, too, that the Universal Serial Bus never really stood a chance. What remains once the hardware disappears is only the soft etiquette of its interface.

This, finally, is the key legacy of the Universal Serial Bus: it offers a model for machine-to-machine communication that has been duplicated often enough to form a rough consensus—and one that's considerably more long-lasting than any hard physical equivalent. Two decades on, the communication protocol that USB put in place has become a code of conduct for machines, a set of manners that guide a conversation conducted in 1s and 0s:

"After you ..."

"No, no, after you."

*