

Mike Pepi

Asynchronous! On the Sublime Administration of the Everyday

01/08

e-flux journal #74 — June 2016 [Mike Pepi](#)
[Asynchronous! On the Sublime Administration of the Everyday](#)

A cold pail of water passes through a line of workers, sloshing from hand to hand. Another follows behind it. And another. To coordinate this bucket brigade, the line of busy hands moves according to a fixed rhythm, each movement synchronized like a metronome. The analogy illustrates the primary principle of synchronous processing: no matter the speed of a single movement, the pace of the chain may not exceed the time it takes the slowest transfer to complete. This familiar scene is the basic unit of Fordism – an assembly line of exchanges locked in linear progression. One thing at a time. One thing after another. All you can really do is speed it up.

This dictatorship of synchrony – from clocked computer chips to supply chains and back again – hamstring productivity and constrains the marketplace. For the designers of scalable systems, it represents the ultimate barrier to progress. To break through this barrier, engineers dream of the *asynchronous*: a vision of the world where the bucket brigade stops following the tick of the metronome. In the event that one worker finishes passing their bucket early, they can accept the next from anywhere along the line. Instead of waiting for the second worker to pass their bucket, the third takes it directly from the first, or from a different line entirely. Work flows to available resources, regardless of where these resources are located in the traditional sequence. At first, the line becomes chaotic. But suddenly, the light accelerates past the heavy. Soon we have an asynchronous system and a new transaction can begin without waiting in line.

Inside every computer is a microprocessor ticking back and forth about a billion times a second. This tick organizes each transmission, signaling to the operating system when one process has completed and when the next can begin. Just as the bucket brigade's linear rhythm constrains the movement of the water, so too do synchronous computer chips limit the performance of our fastest information transmissions. At Sun Microsystems in the 1990s, Ivan Sutherland and Jo Ebergen used the bucket brigade metaphor to explain the advantages of their experimental research into asynchronous chip design. When computer chips become asynchronous, "actions can start as soon as the prerequisite actions are done, without waiting for the next tick of the clock."¹ But in the early days of computing, the market pressure for a straightforward, reliable solution meant that synchronous chip design, which was simpler, won out over the grander, theoretical plans for asynchronous computing. The processor that runs your MacBook is synchronous and clocked, running at about 2.7

GHz. Despite intense research, truly asynchronous chips took years to get out of the lab – and even then their commercial use was limited.

But something funny happened on the way out of R&D. Asynchronous processing hasn't simply left the lab and entered our devices and networks. Instead, the asynchronous principle – that complex systems should be designed to allow tasks to run independently as resources become dynamically available – has moved outwards from the chip to the server, from the server to the data center, from the data center to the workplace, and from the workplace to the city. Asynchronous processing has emerged as a new ideal, and it is increasingly being applied in fields as diverse as software design, biomedical engineering, and labor-force management.

No discussion of the contemporary can ignore the present drive to process more and more of society's moving parts in the fashion of an asynchronous bucket brigade. If today's lifeworld distinguishes itself by the ubiquity of computing in all its various forms, then the expansion of the asynchronous principle represents a fundamental shift. This expansion requires not just the datafication of everyday life, but a significant reformation of the social relations that grew around the modes of exchange proper to the pre-asynchronous era – what we might call linear information capitalism. With the introduction of asynchrony, these relations appear as so many bonds to be burst when the buckets begin arriving from everywhere, heralding the addition of a spatial dimension to what had, until now, been simply temporal sequences. As with all such arrivals, the asynchronous is initially apprehended in terms of the previous era, and so its borders remain frustratingly concealed behind inherited ideas about the individual's relationship to their labor, the market, and the state. How can we begin to uncover the contours of the new asynchronous present?

Beyond the Chip

It all started with hardware. Asynchronous systems were initially designed to transcend the material constraints of computer processors. Without an asynchronous architecture, clock-speed optimization would always be fundamentally capped by the physical limits of computing. Every speed increase of synchronous, clocked chips only produced diminishing returns. To go any faster, the governing clock would have to be replaced.

The next obstacle was energy consumption. Because the “clock” is always running, synchronous systems do not adequately distribute energy according to demand. In

principle, an asynchronous architecture lets the system rest when no jobs require processing. This is illustrated in the example of the asynchronous bucket brigade: if there's no bucket coming down the line, the workers need not move at all. Breaking the clock means transcending a system's built-in ceiling while reassigning fixed resources more efficiently – a goal shared by engineers and capital alike.

Affordable, just-in-time computing is a commercial example of a large-scale asynchronous process. Cloud-computing storage services like Amazon Web Services apportion their server space among clients who pay through an on-demand model. The basic principle of Infrastructure as a Service (IaaS) is that you only pay for what you use. When your allocated space adjusts in real-time with your demands, you eliminate the pitfalls of predicting how much storage a project might demand, accelerating growth and reducing risk.

The asynchronous principle operates in software, too. A new set of asynchronous programming languages use what is sometimes called a “non-blocking schema,” where a task starts firing even if others tasks that are lined up before it haven't completed. Instead of going line by line, the component jobs run “all at once.” Consumer products have followed suit. The most popular products use the Software-as-a-Service (SaaS) model to make asynchronous production possible. Google Docs has quickly surpassed the local storage of Microsoft Word because many parties can edit simultaneously. Like the workers on the asynchronous bucket brigade, a line of code or a collaborating editor can start doing work as soon as it is ready.

The “sharing economy” – in which underused resources are rented via peer-to-peer transactions – is a means by which asynchronous processes have been introduced into the consumer marketplace. Asynchronous capitals do not require that resources be committed to a fixed sequence. The “hardware” of any given business process has come to be viewed like the physical limitations of computing; thus hotels, which are time-consuming and expensive to build, are now a drag on hospitality companies trying to compete with Airbnb, just as the requirement to have an official medallion is a drag on taxi drivers fighting for their livelihood against Uber drivers. No matter how efficient processes become, if they contain synchronous components in a blocking schema, they eventually create friction and are unable to compete, at the level of accumulation, with the asynchronous organization of information, labor, and capital.

While asynchronous processing is the latest in a long line of techno-determinist fetishes, the

02/08

e-flux journal #74 — June 2016 Mike Pepi
Asynchronous! On the Sublime Administration of the Everyday

asynchronous *principle* remains agnostic to any one mediating technology. From software startups to shoe companies, asynchronous processes are introduced when the immediate payoff of piecemeal execution appears to outweigh the advantages of performing tasks in a specified order. Despite these universalist ambitions, the promised increase in efficiency does not always materialize. Though the evangelicals might imply otherwise, only under certain circumstances are asynchronous methods more efficient than linear ones: namely, when the cost of each individual action has been driven down exponentially, making it feasible to spend exactly zero resources prioritizing the order of their execution.

Software can aid in bringing about asynchrony, but human capital frequently stands in the way. Paying the absolute minimum for labor – long the goal of supply-chain optimization and just-in-time manufacturing – can now be achieved through the asynchronous assembly of social interactions or physical labor. The impulse driving the multitasking web surfer, who spends their day in front of so many open and idle tabs, can be harnessed to the real labor behind dinner reservations, transportation, or apartment maintenance. A user who makes no upfront investment, who is free to leave at any moment, has little incentive to order their actions. This everyday arbitrage of simultaneity is already embedded in our cultural logic and

encouraged by the design of our interfaces. But the labor that would meet this demand must follow suit. In order to participate in this frenetic and ever-present auction, the laborer must remove themselves from the linear chain that once defined their market position.

We have dreamed about the revolutionary potential of self-organization for generations, but the apparent harmony between asynchrony and anarcho-syndicalism, libertarianism, or horizontalism obscures the extent to which an engineer's fantasy has become management's best friend. The decentralization achieved by asynchrony is different from the political ideal of decentralization. From the perspective of the individual worker, asynchrony doesn't remove authority as much as displace it. A non-blocking schema allows orders to pour in from everywhere, but they're still orders. The absence of a linear sequence means paying labor for only the time it works, and not a second longer; work need not be synchronized with the arbitrary designations of work days, licenses, or any other ordinal mechanism that produces artificial scarcity. You can work anytime you want, but there's no wage if you're at rest. And when you're at rest, demand will still be processed, perhaps by another worker who is faster and less expensive. The result: lower labor costs and higher profit. Nor is asynchrony simply flat. It is very interested in hierarchy – let the fast move faster and the slow drag down only themselves.

03/08



A photograph of Amazon's dash buttons promotes the automatic consumer goods ordering service.

The new asynchronous regime optimizes coordination at the expense of that which is coordinated. Any newfound autonomy applies only to the system itself. This is why, although asynchrony has established itself at the level of infrastructure, its most substantive expressions will be political. A critical history of the aspiration to asynchrony is necessary to separate utopian visions from a real politics that accounts for the new socio-technical capacities of the asynchronous.

Sublime Administration

Asynchronous capitalism is already a rallying cry for Silicon Valley. Venture capital firms are heeding the call, investing in a “platform economy” that promises to transform any job, project, or endeavor that can be represented as a “unit of work” in an asynchronous system. J. P. Morgan calls it “unbundling a job into discrete tasks” and has joined other investors in funding the platform economy to the tune of \$9.4 billion since 2010. Not only will platform companies reap the financial benefit of massive growth; they also stand to play an outsize role in reshaping the distribution of goods and services once provided by the state.

The extent to which the platform economy replaces this infrastructure will be a battle waged in public. But the internal governance of platform companies is a private affair, first and foremost a matter for management methodologies and open-floor plans. A new breed of such methodologies has emerged, viewing labor as little more than a problem of human-platform engineering. These management philosophies have been encapsulated in a kind of shorthand notation: “agile,” “lean,” “open source,” “holacracy.” These labels – which are half brand, half method – signify the various efforts to extend asynchronous systems to human resources, each time wrapped in the promise to distribute employee authority in the name of autonomy and productivity.

In “agile” project management, teams work on incremental iterations in highly visible and simultaneous cycles. In the scrum – agile’s signature form – team members communicate to rapidly remove blockers, organize sprints, and collaboratively squash known issues as they occur. This approach can also be found in open-source software development, which, following Eric Raymond’s famous text *The Cathedral and the Bazaar*, should be run more like a bazaar – a babbling, participatory community in which many hands make light work through concurrent collaboration – than a cathedral, where a closed team toils in isolation, adding one new section at a time.² After the rise of cloud storage and

instant communication protocols, the new networked age of software development doesn’t require the sort of restrictive physical rituals of the white-collar office. It follows that open-sourcing the code itself allows for asynchronous production anywhere on a network. This method has come to replace a local, sequestered practice of shipping software that, not unlike the sequential bucket brigade, was restrictive, blocking, and expensive.

“Lean” management methodology takes the “test and learn” ethos latent in the provisional nature of agile and open source to its logical conclusion. In recent years, lean management has stressed experimentation and rapid customer feedback to optimize the outcome of each new movement. A manager schooled in lean methodology ships a product to market prematurely, monitors results of split tests, and “pivots” accordingly. Lean’s extreme reliance on preemptive action and real-time feedback could not exist in a linear bucket brigade, since no lean manager would set up a structure that lacks a contingency plan for its abrupt dissolution.

“Holacracy,” which is perhaps the most extreme and putatively emancipatory of all the new methodologies, attempts a total rewiring of the manager-employee relationship. Its name derives from the Greek word *holon*, meaning a part which is simultaneously a whole. True to form, its foundational tenet is a relinquishing of authority, replacing managers with self-governing circles comprised of each department’s component tasks.³ As in lean, this design empowers dynamic “human sensors” to identify tensions and enact change from any position in the organization. Appeals to the ideal engineering environment of the human body are central. Holacracy founder Brian Robertson asks: “How can we reshape a company into an evolutionary organism – one that can sense and adapt and learn and integrate?”⁴

On their face, these management methods rid the workplace of blocking schemas, foster spontaneous, data-driven collaboration, and build organizations with a responsive and collapsible pseudo-structure that can be dissolved on demand. Each of these forms attempts what we might call sublime administration. Increasingly, they rule the shop floor, but their aspiration is the town square.

To overcome the political nature of their autonomous subjects, sublime administration must paradoxically erect a baroque set of protocols that are hyperfocused on distributed autonomy and asynchronous assembly. This is all done under the guise of empowerment and individual choice. But the sum total of this framework creates value at the expense of the subjects it administers. In its most extreme

04/08

e-flux journal #74 — June 2016 Mike Pepi
Asynchronous! On the Sublime Administration of the Everyday

05/08



The Société booth at Frieze New York 2016 featured a collaboration between Sean Raspet and Soylent, the drink brand. Image courtesy of Société.

forms, sublime administration purports to administer a (human) resource that it fundamentally feels it would be better off without. The mechanics of sublime administration trade not in the employee's innate human capacities, but in the ability to confront and remove the bottlenecks created by such capacities.

It is no coincidence that the tactics of sublime administration are increasingly deployed in the fields of software automation and large-scale market disintermediation. Like asynchronous systems, sublime administration seeks to unlock the surplus profit yielded by a passive mediation of interactions, which proliferate without the constraint of the queue. Under sublime administration, parties to an exchange are removed from the jaws of time and liberated from locked resources. Its frequent appeals to the worker as a mini-entrepreneur, able to produce at a pace unrestricted by a sclerotic hierarchy tied to outdated modes of production, reduces the bucket brigade to ruins. If your guess is as good as mine, then a productive dissolution is always just around the corner.

In sublime administration, management acknowledges its own inability to define an organization's optimal route, which is why it distributes incremental authority across the organization. Management's ignorance about the most profitable direction for the company is evident in the way its decision-making apparatus privileges future information over the events of the present. Sublime management is speculative and deeply skeptical of all things recently accumulated. It is quick to discard the past, unless the past can be used to construct an anticipatory model.

In the social order that follows, everyone works on their own, self-directed and requiring little investment of resources by superiors. But this "free-for-all" is always facilitated by the platform that most successfully executes the processes themselves. Asynchronous processes achieve the appearance of autopoiesis for what is in fact a hardened marketplace. Running such a platform is the ultimate goal of sublime administration: to maintain power while not appearing to seek it.

No Shifts, No Boss, No Limits

To understand the totalizing vision of the asynchronous we need look no further than a video advertisement from Uber. Entitled *Bits and Atoms*, the video purports to reveal the company's *grand récit*. It begins by dividing the known world into the two eponymous building blocks, and goes on to assert that the bit has changed communication and business in less

than seventy years' time. The bit represents Uber's technology – it is "complex, precise, and advanced. But when it's expressed, it's effortless, and refined." The atom is far older, but much more impressive – it is "responsible for everything – from the BLT, to moms everywhere, to New York City." The rest of the video depicts the city of the future, a platform utopia of benevolent and frictionless people-first mobility.

While Uber's *Bits and Atoms* makes titular reference to the technology that is integral to the company's business model, Uber's true innovation has been a political reformation of the economy. While Uber's app relies on widely available protocols and devices, its competitive advantage derives from the company's innovative and asynchronous organization of its contingent labor force. A slogan Uber pitches to prospective drivers crystallizes this central fact: "No shifts, no boss, no limits." Uber's asynchrony removes the governing clock, facilitating an army of entrepreneurs who suddenly need not wait in line.

The bucket brigade, too, uses technology. And its metronome principle likewise enhances the performance of the total system. But the bucket brigade's synchronous structure, its technology (buckets), and the humans that mediate its transactions are all balanced in their contingency. The unbundling of the client and the (now precarious) service provider is only tenable if the mediating platform can continue to maintain an asynchronous state. Under this framework the human element quickly becomes an obstacle.

Fueled by speculative capital, Uber's asynchrony aims for growth rather than stability. It privileges the potential redistributions of the future over the social continuities of the present. In *Bits and Atoms*, Uber stakes a claim to asynchrony's expression of harmony, purpose, and spirit – the very elements that commonly figure into human judgment, both moral and aesthetic. The work of *Bits and Atoms*, then, is to redefine the structures of human activity according to the logic of the asynchronous process.

Uber's video manifesto ends with a rosy tautology: Uber creates "industries that serve people, and not the other way around." The asynchronist erects a socio-technical system that enjoins people into competitive transactions, and yet Uber contrasts its monopolistic platform with an imaginary inverse scenario where "people serve industries." In effect, Uber is arguing that it serves its users and employees alike. This turn of phrase really attempts to inaugurate a new political logic that privileges one type of circulation over another. Nestled within this logic can be found the impact

06/08

e-flux journal #74 — June 2016 Mike Pepi
Asynchronous! On the Sublime Administration of the Everyday

of the asynchronous on aesthetics, understood here in Rancière's terms, where aesthetics is the "distribution of the sensible" responsible for the "conceptual coordinates and modes of visibility operative in the political domain."⁵ This distribution determines the forms that are available to artistic practices. Each time a linear component of the bucket brigade breaks down, we are treated to new forms of production and therefore new ways to move about the stage of society – or in Rancière's terms, new "ways of doing and making." How could the production of an asynchronous subjectivity appear without some corollary reformation of the aesthetic subject, the position from which we sense, judge, and act? Art's political interventions – or its near constant mode of articulating our relationship to society – must now contend with the new relations of our asynchronous present.

Though Uber's aspirations seem to be of a piece with the overall economy's drive toward full automation, the company's articulation of the asynchronous principle has more expansive intentions. Automation derives in part from a Taylorist drive for efficiency; but asynchrony dissolves industrial ambition altogether. To the asynchronist, even the fully automated assembly line is a cost center where the firm must still perform the labor itself. But the asynchronous platform harbors none of this risk. The asynchronous achieves the most desired effects of automation even before the hardware or software is introduced. In the perfect asynchronous system, labor almost seems to disappear from the system itself. In *Bits and Atoms* we see users and goods transported, but barely any drivers. Uber has made no secret of its plans to eventually deploy self-driving vehicles, or, as it hints near the video's conclusion, the "safe, efficient movement of people and things at a giant scale."

In 2015, Ford Motor Company hired Pivotal Software, a management consulting firm specializing in the agile development method, to transform their IT and software engineering department. Ford CIO Marcy Klevorn explained: "We need to iterate, take more risks, learn. That requires a different culture. Our culture is very risk averse, and rightfully so. But we need a different way of thinking of IT and the way we do business."⁶

A report on the partnership provided one rationale for the move:

Ford is not so much an automobile manufacturing company as a mobility and transportation company. "They're thinking beyond just cars," says Ashok Sivanand, senior product manager at Pivotal. "They're thinking about mobility and realizing they

need to transform a lot more aggressively into being a software company."⁷

The symbolism behind Ford's reimagination as a "mobility" provider cannot be overstated: the very company that perfected and scaled the assembly line has imported the management style of software companies. "Mobility" usurps "automobiles" because asynchrony works best with platforms, not products; and sublime administration focuses on building infrastructures for abstract activities, not giving life to activities themselves.

x

07/08

e-flux journal #74 — June 2016 Mike Pepi
Asynchronous! On the Sublime Administration of the Everyday

Mike Pepi is a writer living in New York. He runs Cloud-Based Institutional Critique, a New York-based reading group dealing with digital technologies and their relationship to arts institutions, organized through thepublicschool.org.

08/08

1

Ivan Sutherland and Jo Ebergen, "Computers without Clocks," *Scientific American*, August 2002.

2

Eric Raymond, *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary* (Sebastopol, CA: O'Reilly, 1999).

3

Brian Robertson, *Holacracy: The New Management System for a Rapidly Changing World* (New York: Henry Holt and Co., 2015).

4

Ibid., 7.

5

Jacque Rancière, *The Politics of Aesthetics*, trans. Gabriel Rockhill (London: Continuum, 2004), 82.

6

"Ford draws on Pivotal to reshape developer culture," cio.com, December 16, 2015 <http://www.cio.com/article/3016017/developer/ford-draws-on-pivotal-to-reshape-developer-culture.html>

7

Ibid.

e-flux journal #74 — June 2016 Mike Pepi
Asynchronous! On the Sublime Administration of the Everyday