

A bouquet of colorful lilies, including white, yellow, and pink varieties, is shown within a large purple circle. The background of the entire page is a light blue color with a repeating pattern of small, light brown circles.

LOGIC

A single lily stem wrapped in clear plastic is shown within a large purple circle. The background of the entire page is a light blue color with a repeating pattern of small, light brown circles.

CARE

ISSUE  
11

# LOGIC

magazine

11

---

care

summer 2020

# LOGIC

ISSUE 11: Care

## EDITORIAL

Managing Editor: Alex Blasdel

Interviews Editor: Jen Kagan

Editors: Ben Tarnoff, Moira Weigel

## CREATIVE

Xiaowei R. Wang, Celine Nguyen

## PRODUCTION

Jim Fingal, Christa Hartsock

/\*

Cover & interstitials: Celine Nguyen

Illustrations on p. 184, 204, 226 by Julie Sutherland

Text contents licensed under: CC BY-NC-ND 4.0

**ISBN (print)** 978-1-952550-04-1

**ISBN (digital)** 978-1-952550-05-8

**ISSN** 2573-4504

## ELECTRONIC TELEGRAM

[editors@logicmag.io](mailto:editors@logicmag.io)

## INTERNET

<https://logicmag.io>

Logic Magazine is published by the Logic Foundation, a California nonprofit public benefit corporation organized for the specific purpose of promoting education about technology.

\*/

**care**

## Editorial

**011** Where Deer Stroll Peacefully Past Computers

Moments of social mobilization enliven and expand our political imagination. Among the things that sorely need reimagining is our technology.

---

## Features

**053** Informatics of the Oppressed

Rodrigo Ochigame / An inquiry into the rich history of radical experiments to reorganize information.

**163** When You Say One Thing but Mean Your  
Motherboard

Matthew Seiji Burns / If your shrink were a computer, would you even know?

## Chatlogs

- 019** Community Defense: Sarah T. Hamid on Abolishing Carceral Technologies  
A conversation about how to break cages.
- 091** Logistics Workers of the World: A Conversation with Agnieszka Mróz of Amazon Workers International  
Building a global movement starts on the shop floor.
- 141** Call On Me: Interview with an Anonymous On-Call Engineer  
When the computers fail, they bring in the humans.

## Patches

**039** Inside the Paradise Bubble

**Ann Chen** / The story of Taiwan's surprisingly successful fight against COVID-19.

**077** Do No Harm

**Niloufar Salehi** / Social media is broken. Restorative justice offers a way to repair it.

**129** Bodies into Bits

**Joy Lisi Rankin** / An inquiry into how computerized healthcare education has reinforced racism and sexism.

**153** The Tower and the Tent

**Nitin Ahuja** / A doctor observes the pandemic from a tent in the shadows of a megahospital.

---

## Assets

**104** Open Casket Series

**Indira Allegra** / A series of woven memorials, reflections on mourning in "viral time."

# Maintenance Window

## 182 Introduction

Three conversations with government workers about lives spent maintaining critical computing systems.

## 185 Mike Schwab

On Mainframe Programming for the State of Illinois.

## 205 Adam Grandt-Nesher

On Modernizing Infrastructure.

## 227 Jed Wagner

On Single-Handedly Building and Maintaining the Veterans Appeals System.

## 245 Built to Last

Mar Hicks / Don't blame COBOL. Blame austerity.

## Contributors

**Nitin K. Ahuja** is an Assistant Professor of Clinical Medicine in the Division of Gastroenterology and Hepatology at the University of Pennsylvania.

**Indira Allegra** is reimagining what a memorial can feel like, the scale on which it can exist, and how it can function through performance, sculpture, and installation.

**Matthew Seiji Burns** is a writer and video game designer in Seattle, Washington.

**Ann Chen** is an artist, researcher, and filmmaker based in Brooklyn and Shanghai. She is currently an Assistant Arts Professor of Interactive Media Arts at NYU Shanghai.

**Sarah T. Hamid** is the policing tech campaign lead at the Carceral Tech Resistance Network.

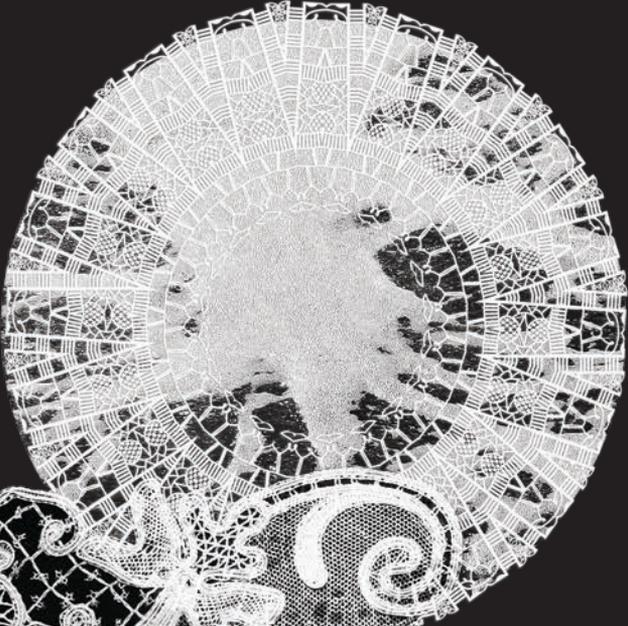
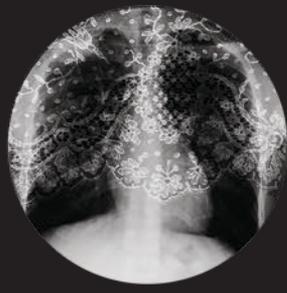
**Mar Hicks** is a historian of technology, professor, former sysadmin, and the author of the book *Programmed Inequality*.

**Agnieszka Mróz** is a Workers' Initiative steward at Amazon's Poznań, Poland warehouse and a member of the Amazon Workers International coalition.

**Rodrigo Ochigame** is a PhD candidate in science, technology, and society at the Massachusetts Institute of Technology.

**Joy Lisi Rankin** leads the research program in Gender, Race, and Power in Artificial Intelligence at the AI Now Institute at New York University.

**Niloufar Salehi** is an Assistant Professor in the School of Information at the University of California, Berkeley, where she directs the TO3 research group.



From the Editors

---

# Where Deer Stroll Peacefully Past Computers

1/

When the pandemic first struck the United States, it was not uncommon to hear people compare its virality to the other, online, kind. The numbers climbed impossibly — then inevitably — high. In a nightmare inversion of network effects, the more people who got it, the more deadly it became. In New York City ICUs, patients were dying in the hallways.

Then, as the weeks passed, a different technological metaphor started to seem more apt: the X-ray. The novel coronavirus itself remained, in many ways, inscrutable. But it revealed the brokenness of our systems for providing care.

Nursing homes and prisons became deadly zones of infection. Food production plants followed. Doctors and nurses were going to work in garbage bags, while governors, who had been told to compete with one another, bid up the price of PPE that never materialized.

Cities could not dispose of their dead fast enough. They closed schools and daycares indefinitely, without explaining what exactly the working parents who were now also full-time caregivers were supposed to do — and the working parents were the lucky ones. Every Thursday, the Labor Department reported that millions of Americans had applied for unemployment the previous week. That did not count all of the people who could not apply for unemployment because the state websites were crashing and the hold times on their hotlines lasted all day.

Did it even make sense to call what you were feeling “anxiety” or “depression,” when there were so many real reasons to worry? Amid so much loss and suffering, what feeling person would not grieve? Nonetheless, social media revealed a steady stream of people who could not be persuaded to care enough for their fellow citizens, or themselves, to wear masks on the bus or to the grocery store. Armed protesters showed up at government buildings to demand a return to business as usual.

## 2/

The pandemic put technologists, and technology firms, in an ambiguous place. On the one hand, their products have become more essential than ever. Internet traffic is up; cloud services are in high demand. Countless Americans have turned

to Amazon and Instacart to shop (or work), and Zoom to take classes or attend religious services (or work). Tech stocks are soaring, even as the real economy falls, and the fights that at least some prominent figures in the industry have picked with journalists and lawmakers reflect a new sense of invulnerability.

At the same time, COVID-19 has exposed points of fragility in the system. If the biggest tech companies are essentially vast engines for making predictions, the pandemic was unforeseen: you could see the breakdown of the machine learning machine. Moreover, this was a crisis of care, and care is precisely what software cannot provide: it is designed to coordinate, and sometimes eliminate, human work. The tech firms could offer gig and warehouse jobs, but they could hardly make up for plummeting employment. They could apply their expertise in digital surveillance to contact tracing, but the populations most at risk — Black and Latinx communities — had the most reason to fear being tracked.

Then the Minneapolis police murdered George Floyd, just two months after the Louisville police murdered Breonna Taylor. Cities across the country erupted. A government that couldn't be bothered to do the bare minimum to contain the pandemic quickly moved to mobilize battalions of militarized cops. There was no money for PPE, but no shortage of resources available for repression.

Again, the role of technology was ambivalent. On the one hand, smartphones produced the videos of police violence that sparked and spread the mobilizations. Technologists soon found other strategic uses for digital tools — for instance, mapping police movements by listening to scanners. On the other

hand, tech posed a constant danger. Drones developed for wars in the Middle East hovered above protests. Your own social media could be used against you — and so could other people’s, as a woman in Philadelphia learned the hard way, arrested after a stranger posted a photo of her with her face covered on Instagram. The networks used to organize and amplify collective action were all too easily weaponized by the state. If you go to a protest, leave your phone at home.

### 3/

We made this issue in the midst of these intertwined and rapidly unfolding crises. The pieces were written in the midst of the grief and rage of the past months, but also the moments of possibility and hope, which have so often taken the form of people taking care of one another, from street medics washing out pepper-sprayed eyes to militant nurses and teachers organizing for better conditions for patients and students. There was a reason so-called “Momtifa” captured the public eye.

Some of the pieces in these pages deal directly with the most current and urgent aspects of the crises. What they find is that technology is often part of the solution — but only a part. Smartphones can help, but only in tandem with functioning healthcare and state institutions. The clean lines of the tech-laden megahospital suggest a future of frictionless care. But in their shadow, essential workers are saving lives in tents.

Technology alone can’t save us. Often, in fact, it can hurt us. The harms it inflicts aren’t new. The software might have been made recently, but the social relations that software embodies

and enacts were made a long time ago. Contemporary digital surveillance emerged from older practices, and obeys old carceral logics. As Sarah T. Hamid explains in these pages, today's facial-recognition and predictive-policing algorithms belong to a centuries-long lineage of tools for "the control, coercion, capture, and exile of entire categories of people."

Taking care of one another will require dismantling these tools, whether in the form of a laser pointer that scrambles a facial-recognition camera or a legislative ban that outlaws facial-recognition technology. It will also mean constructing alternatives. Moments of social mobilization enliven and expand our political imagination. Among the things that sorely need reimagining is our technology.

#### 4 /

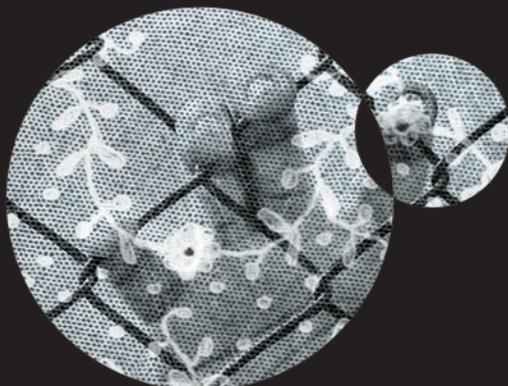
The poet Richard Brautigan once imagined "a cybernetic forest,"

*filled with pines and electronics  
where deer stroll peacefully  
past computers  
as if they were flowers  
with spinning blossoms.*

This issue explores what a more habitable digital world might look like. There are recovered histories and preliminary experiments, sketches of past and possible schemes for organizing networks differently, and for redressing networked harms.

The issue also asks, *Who cares?* In one sense, this means: Whose lives are touched by particular technologies, and who participates in their development and design and deployment? But it also means: Who performs the work of looking after and tending to people, and the machines that are integral to the systems that people need? The unglamorous work of maintenance and custodianship, of remembering abandoned knowledge — and programming languages — is what makes it possible for the millions of Americans who have lost their jobs in recent months to receive an unemployment check.

Computers cannot care for us as completely as venture capitalists might like. But we hold out the hope that, with some deep social and technical reconstruction, they can be put into service of creating a more caring world. ~~~~



# Community Defense

## Sarah T. Hamid on Abolishing Carceral Technologies

After the rebellions sparked by George Floyd's murder at the hands of Minneapolis police, abolitionist ideas and abolitionist demands are finding wider circulation. Campaigns to defund the police are gathering momentum in several cities. More people than ever before are asking questions about what actually keeps communities safe, and how to reimagine the project of public safety.

Given the large role that digital technology plays in modern policing, any attempt to dismantle the carceral apparatus must confront technologies of various kinds. To learn more about this terrain, we spoke to Sarah T. Hamid. Sarah is the policing tech campaign lead at the Carceral Tech Resistance Network and one of the creators of the #8toAbolition campaign. She talked to us about police software, militant research, and what it means to apply an abolitionist lens to technology.

## **What is the Carceral Tech Resistance Network?**

The Carceral Tech Resistance Network (CTRN) is a coalition of organizers who are campaigning against the design and experimentation of technologies by police, prisons, border enforcement, and their commercial partners. We work to abolish the carceral state and its attendant technologies by building community knowledge and community defense. Our group is made up primarily of femme, Black, immigrant, POC organizers. My own work is embedded in Los Angeles, the Bay Area, and Portland, Oregon, but CTRN has organizers in most West Coast US states.

*“Carceral technologies are those that are bound up in the control, coercion, capture, and exile of entire categories of people.”*

The network was created out of two primary needs: first, we started to realize that these technologies, often rolled out at a local scale, have afterlives — they travel to other contexts, where communities may have less familiarity with them, or no organized base prepared to confront and dismantle them. So there was a need to knowledge-share and foster mentorship between community organizations. And second, we felt an urgent need to build a different relationship to the cataloging, databasing, and archiving practices that are widely deployed in movement spaces — but which also share a troubled history with the exact same surveillance technologies we are working to dismantle.

## **How did you first come to work on these issues?**

I started thinking about these policing techniques during the Ferguson uprisings. I became fascinated by predictive policing, an object that has captured popular and scholarly attention since its inception. Originally, I had aspirations to be an academic; I took the project of techno-criticism seriously. I described this recently as an impulse to “speak these technologies into illegitimacy.”

Things changed once I started to realize that academic research has a long history of being co-opted—even used against itself—by the particular systems that I was studying. Similar to prison industrial complex abolitionists in the 1980s and 1990s, I started to recognize that criticism was not going to be an effective tactic to enact change. So I started to look for other pathways. A couple of years ago, I came out to Los Angeles and began organizing here. And I realized that once you position yourself as an organizer, change becomes possible in a very different way.

**As an organizer, you’re focused on resisting and dismantling “carceral technologies.” What is a carceral technology? The rebellions in the wake of George Floyd’s murder have largely focused on the physical aspects of police repression, like killings by police officers and brutality towards protesters. But there are also various technologies of police repression that are less physical, and indeed sometimes invisible. Could you give us a sense of what some of those technologies are?**

Carceral technologies are those that are bound up in the control, coercion, capture, and exile of entire categories of people. CTRN organizers campaign against CCTV, face printing, DNA and biometric databases, acoustic gunshot detection, drones, electronic monitoring, AI and risk profiling algorithms—all of which function as weapons in the hands of law enforcement or prison administration.

When we talk about carceral technology, it's important to note that we are not just talking about *digital* technology. We are working with an inheritance that *predates* digital technology. We are talking about the long history of carceral technologies — lanterns (which Black, mixed-race, and Indigenous folks in the eighteenth century were required to carry if not in the company of a white person), rowdy sheets (colonial crime intelligence and profiling ledgers), sentry boxes (telegraph boxes that gave white, “reputable” citizens a direct line to police power in the early twentieth century), rogue galleries (image galleries of individuals criminalized by police bureaus), calipers (to catalogue biometric data from those in police custody), pin maps (analog “hot spot” mapping techniques used to criminalize entire neighborhoods and communities). And we are talking about a long history of carceral practices, like forced sterilization, medical experimentation in prisons, work homes, and security landscaping (architectural techniques popularized in the service of police surveillance, such as stripping entire neighborhoods of greenery). As abolitionists, we want to dismantle the system that makes those practices possible. And we are organizing in communities that have a long history of fighting those practices, communities that have acquired knowledge about *how* to fight and build safety against the rollout of experimental carceral programming — whether analog or digital.

**Sometimes, the argument against carceral technologies like predictive policing or facial recognition is framed as a privacy issue. I get the sense that you and your fellow organizers in CTRN don't share that analysis.**

When these technologies first captured popular attention, the anxiety over surveillance started to take up a lot of space in the room. There was an intentional move by white scholars to push back on these technologies by presenting surveillance as a

generalized harm; that is, as something that affected everybody. Because surveillance violated people’s privacy, everybody should care about it — not just racialized populations or communities targeted by the state.

*“The focus on ‘surveillance’ has a depoliticizing effect on the work we do.”*

This was a well-intentioned move. But it muted much of what directly impacted communities needed to talk about, what they wanted to build awareness about, and what they wanted to fight against. The privacy framing made it so that the harm enacted by carceral surveillance systems were fixed to a spectrum of *intrusion*, the one end of which is Target being creepy by spying on its customers, the other end of which is facial recognition-enabled immigration detention.

CTRN is very intentional in how we position our work. We organize against carceral institutions, actors, and systems — not surveillance. The focus on “surveillance” has a depoliticizing effect on the work we do. Organizers campaigning against carceral technologies are not organizing against “intense creepiness.” They are organizing against a category of violence — legally sanctioned violence by the carceral state — that has a long history of racialized surveillance, and a short history of digital surveillance. These technologies aren’t just creepy. These technologies don’t just make the subject feel watched, or like they can’t express themselves. These are violent technologies — *carceral* technologies. So the goal can’t just be to make them a little less intrusive.

**Who profits from these technologies? How do you see the role of industry in structuring or shaping these kinds of instruments and the social relations they embody? Does it make sense to speak of a “carceral technology industrial complex”?**

Speaking in terms of industrial complexes is very helpful. After the Ferguson uprisings, there was this particular way in which reform and technology acted together to incentivize certain modes of innovation, like body-worn cameras, that were linked to measures that were supposed to help improve police accountability. Not only did these technologies expand police investigatory and surveillance power, they fundamentally failed to improve the rate of violent encounters between over-policed communities and law enforcement. But this problem space of police reform was incredibly profitable — it was profitable then, and it’s profitable now.

**I’m reminded of a line from Foucault that Angela Davis uses in her book *Are Prisons Obsolete?*: “Prison ‘reform’ is virtually contemporary with the prison itself: it constitutes, as it were, its program.” The history of the prison, in other words, is the history of reform.**

We have to recognize that technological innovation, and the reformism that animates it, is a carceral tactic. It’s a means by which these systems have expanded over time. Police have been experimenting with different kinds of technology for hundreds of years. It has offered them a strategy to divert focus from the extreme conditions of violence that they enact on communities, all while amassing vast amounts of resources and connections. Technology is one way that police have historically mobilized academia to work in their favor. It has also helped police forge links with industry.

For instance, one early innovation in police technology was adding radios to squad cars. Who built those radios? Motorola. From the 1930s onwards, Motorola radios were installed in police vehicles. That turned out to be a lucrative line of business for Motorola, and it's no surprise that the company continues to maintain a large communications infrastructure for law enforcement. They have made a lot of money from decades of these public-private partnerships, and so many of the technologies that we enjoy as private consumer goods now were seeded through public funds intended to "fix" policing.

*“We have to recognize that technological innovation, and the reformism that animates it, is a carceral tactic.”*

It isn't just that these investments are system-sustaining—it's the very fact that these innovation ventures have never achieved the accountability or reconciliation they promised. They've just made policing deadlier and less accountable.

**Presumably companies can then market the products they develop to police overseas as well. Can you speak to the international dimension here?**

Yeah, absolutely. Because American policing and prisons have these entanglements with industry, companies have been able to set up different parts of the United States as test sites for new technologies. Certain cities have become spaces of experimentation. It's no accident that ShotSpotter, a gunshot detection system, exists in Chicago but is also marketed to

Johannesburg — two cities that also share a common history of racial segregation by city planning. Companies start to see and profile these places, these cities, as similar. These markets begin to resemble one another. So a product that's beta-tested in the United States gets sold elsewhere.

In fact, many of the technologies that are developed here are being developed with an eye to a global market. I'd go as far as to say predictive policing wasn't even really for the United States, which has a high threshold for things like accountability and transparency. When predictive policing first came to American police departments, the marketing line from industry was that the departments were resource-scarce. Predictive policing, the story went, could help law enforcement agencies save money. That argument is absurd. American police departments are far from resource-scarce. But that argument wasn't for us. That argument was for police departments that really *are* resource-scarce. It was a sales pitch for police departments in Karachi.

*“They test policing tech domestically, in places where the police have a hostile relationship with racialized communities, in order to design war tools for similar communities overseas.”*

But it's not just about global markets. It's also about global contexts. American policing functions as a research site for military innovation — the “green to blue” pipeline is bidirectional.

For instance, the National Institute of Justice’s 2009 predictive policing innovation grants (which funded Chicago’s now-deprecated Strategic Subjects List, or “heat list”) seeded the development of risk assessment technologies that served as templates for military detention algorithms in Iraq and Afghanistan, and that helped support counterinsurgency operations. Similarly, social media flagging systems designed for gang policing in urban contexts were studied by DARPA for monitoring ISIS recruitment. The racially hostile relationship that American police have with vulnerablized communities — what are commonly referred to as “low-information environments” — means that those communities can function as isomorphic innovation domains for US imperial contexts. So they test policing tech domestically, in places where the police have a hostile relationship with racialized communities, in order to design war tools for similar communities overseas.

This is why building transnational coalitions is so important, especially in this moment in American political history when we’re seeing so much momentum behind diminishing police power. Cities like Portland, Oregon, are enacting prohibitions on the kind of crowd-control armaments their law enforcement are able to use. But the Portland Police Bureau (PPB) has adopted Dhaka, Bangladesh, as a city that it wants to teach how to police effectively. That’s where my family’s from. Is that where PPB’s tear gas canisters are going to be shipped?

And how are other cities around the world going to get American policing out of their cities? I want to figure out how we can start collaborating with people in Dhaka to organize against the same systems. Confronting a transnational empire will require transnational networks. The US carceral state, through war, development initiatives, and arms and technologies exports, is a transnational phenomenon.

## Beyond Bias

---

**To return to the question of reform, I wonder if you could speak to the importance of taking an abolitionist framework when organizing against carceral technologies. For example, there are some people who argue that you can reform systems like predictive policing by “debiasing” them, so that they produce fewer racially biased results.**

Carceral technologies are racist because the institutions that develop and use them are intended to manage populations in a country that has a white supremacist inheritance. These technologies are not *incidentally* racist. They are racist because they’re doing the work of policing— which, in this country, is a racist job. There has been a lot of work devoted to proving that particular algorithms are racially biased. And that’s well and good. But there was no question that these algorithms were ever *not* going to be racist.

What would a not-racist predictive policing program look like? You would have to reimagine prediction. You would have to reimagine policing. You would have to reimagine the history of computation. You would have to reimagine the racial configuration of neighborhoods. You would have to reimagine a lot of things in order to arrive at even the slightest possibility of a not-racist predictive policing system, or a not-racist facial recognition system. So yes, they’re racist. There’s no question that they’re racist. But the reason that they’re racist is because they’re used to enact modes of racialized violence.

**In recent years, scholarly communities have focused more attention on issues of fairness, accountability, and transparency in machine learning. We’ve also seen a broader conversation emerge around “AI ethics.” What’s your view of these discourses?**

A lot of these research communities begin with methodologies from STS (Science and Technology Studies) and adjacent fields, where the emphasis is on trying to understand sociotechnical systems. But they often have an inability to apply that analysis to themselves — to interrogate the role that academia and techno-criticism play in the vast sociotechnical assemblage that buttresses the conception, design, and implementation of carceral technologies.

*“These technologies are not incidentally racist. They are racist because they’re doing the work of policing — which, in this country, is a racist job.”*

It’s not due to a lack of imagination that these scholarly communities have continuously circled the drain on questions such as the presence of racial bias in particular systems — this is a political arrangement. It’s a structural condition of how the grants that fund their work are allocated, and the relationships they have to industry and to government institutions. For decades, research questions have been *staged* to these scholarly communities in very particular ways by carceral institutions. There is a given-ness to the problems that these researchers are failing to interrogate. For instance, it’s no accident that for years everyone was like, “We need explainable AI,” and then DARPA started handing out millions of dollars worth of grants to develop explainable AI.

Historically, certain academic disciplines have had moments when they decided to reexamine their relationship with the military and police industrial complex. Consider anthropologists refusing to participate in the US military's human terrain systems in Iraq and Afghanistan, for instance. But the ethics-in-technology communities haven't had that kind of reckoning yet, where they start to deeply interrogate why they're asking the questions that they're asking. Because these technologies are moving so quickly, I think people in these research communities haven't had a chance to reflect on why they keep asking the questions that they're asking. Where do the questions come from? And why is it that they're asking the exact same questions that DARPA is asking? And why isn't that entanglement ethically complicated for them?

**You're no longer an academic but you're still very much a researcher. You're constantly doing research into how particular police programs function, how they were funded, and so on. How does the kind of research that you do now differ from academic research?**

Recently, one of the actions that I helped coordinate involved standing in front of City Hall and giving seven hours of testimony on police violence. Scholars might dismiss this sort of thing as being spectacle-driven. And that's fine. But it's also rooted in a desire to create alternative epistemologies. It's rooted in the recognition that you can't just offer another data visualization, because doing so reinforces a particular way of knowing and a particular entanglement with knowing institutions. So we ask: how do you diversify your ways of knowing? That's the question. How do you make interruptions into what is broadly accepted as valid knowledge—and make something new? How do you make interventions in the breadth *and* depth conditions of knowledge?

How we know, the *way we know*, our epistemic practices, are a political decision. They enroll us in technological and research traditions and transform our relationship to both the object of inquiry and the intention behind it. I remember this one moment when CTRN was archiving different policing program grants. We were working in a spreadsheet. There were blank cells in the spreadsheet, and we became obsessed with filling them in. And then after a week we were like, “Why are we doing this? Why are we so obsessed with having a complete spreadsheet?” We started to realize that our way of knowing and our mode of inquiry were being influenced by the nature of the spreadsheet. It wasn’t curiosity, or any real need to find the information. It was the structure of the technology.

Knowledge takes a particular shape when you start to use particular mediums. So it’s important to continuously reassess how your knowledge is being shaped because, at the end of the day, if you give into what the technology wants, then your work just becomes police work. Your organizing work just turns into a project to surveil the police, you cultivate a need to satisfy each blank cell, you strive for total information. You start to take on the state’s paranoid affect. You can lose yourself in that.

Beyond trying to build a different relationship to knowledge, another thing that we do that is fundamentally different from academic communities is that we always start with first commitments. That’s always the longest conversation of any new organizing formation. What are you committed to? What are you refusing to compromise on? What are you building towards? These are long conversations. When we first created CTRN, it took us more than six months to resolve and find agreement on our commitments.

**It also sounds like you're trying to develop a model of *militant* research; that is, research that is rooted in a set of political commitments and organizing practices. It's a model of research where the ideas emerge through struggle, through practice, through social movement. That's a different approach than the standard scholarly one.**

The scholarly model also operates at a different scale. It's more interested in creating concepts that *govern* because they speak to multiple communities at once. But our work has to happen at a different scale because our inquiry is accountable to specific people. It's inquiry that's conducted through caring about particular individuals. Someone comes to you and says, "I don't understand what's going on. I don't understand why I got fired, why my husband didn't get the job, why my brother's parole was denied." And you start to begin to answer those questions, which are very personal.

Our work happens at this scale. The scale of friends, family, and loved ones. And yes, the answers often point to the role of giant sociotechnical systems. But we're answering individual questions. And we're doing it because we care about the people we are in community with, not because we're trying to develop the best idea to sell a book. Our intervention is effective when we're able to find the knowledge that allows people to enact meaningful change in their lives.

## **Abolitionist Futures**

---

**You mentioned earlier that your goal is to abolish these systems, not reform them. What does an abolitionist campaign against a carceral technology look like?**

I'm working on a campaign right now in Portland to ban both the private and public use of facial recognition technology. A

handful of cities have banned the use of facial recognition by local government entities like police departments, but private businesses have been unaffected. The Portland ban would extend to the private sector too.

It's been controversial because a lot of people who are civil rights oriented have been worried that you're infringing on an individual's ability to use this technology if they want to. But if you're organizing from an abolitionist perspective, you recognize that the private rollout of this technology is still a carceral technology. These technologies never exist without their carceral counterpart. Take the introduction of face-scanning software to unlock people's phones. Industry rolls out these artifacts of private consumption that normalize the existence of these technologies — technologies that have always been entangled in carceral systems.

We recognize facial recognition technology, a weapon used by law enforcement to identify, profile, and enact violence against categories of people. So individuals opting in to unlock their phones with facial recognition serves to improve a technology that has necessarily violent implications — individuals opting in, in other words, are participating in the creation and refinement of this weapon. So when we organize to abolish these technologies, we organize against their conditions of possibility as much as their immediate manifestation. We organize against the logics, relationships, and systems that scaffold their existence.

**You're also one of the creators of the #8toAbolition campaign, which was launched by a handful of prominent police and prison abolitionists during the George Floyd protests. Among the demands listed on the campaign website are "Invest in Community Self-Governance" and "Invest in Care, Not Cops." What might these demands look like within the context of technology?**

There are groups, like May First or Color Coded LA, that are working to create movement technology, technology with a different kind of political configuration. Their experiments don't always scale easily, because they too are working from first commitments. But investing in care and community self-governance when it comes to technology would mean supporting these kinds of experiments, helping them grow, and making them replicable all throughout the world.

We need technological alternatives, particularly now. In a world where people have to talk more over video chat, for instance, it's hard for organizations like us that are very privacy and safety focused. We don't want people on Zoom. We need to make sure that the tools we are using are safe for our communities. So we can't move too fast. We have to be slow, and difficult, and deliberately endure the drag because there are things that we're not going to compromise on.

Honestly, so much of our work is just mailing each other thumb drives. That's how we do our knowledge sharing. It's not high-tech and it's not glamorous. But that's the work that's effective in building these campaigns. It's easy to want to innovate our way to abolition. But you can't do that. You have to live in the friction. You have to be slow. You have to be methodical. You have to prioritize safety. You have to make sure folks aren't left behind because of your sense of urgency. That's just how it has to be done.

**During the protests, a few big tech companies also announced that they would halt or pause their work on facial recognition. IBM said that it would shelve its general purpose facial recognition product, Microsoft reported that it would stop selling facial recognition to law enforcement until there is a federal law that regulates its use, and Amazon declared it**

**would implement a one-year moratorium on the sale of facial recognition technology to the police. How should we view such moves?**

I'm suspicious. These companies profit from expropriative relationships with communities that are hyper-surveilled by the state. They're not just going to give up their bread and butter.

On the other hand, in recent years, we've seen that there is clearly something happening within these companies. Workers are taking action. And I think these moves on facial recognition partly reflect the pressure of tech worker organizing.

*“I will always say: don't join a big tech company.”*

To be honest, my work often has to take an adversarial posture towards the tech industry, including this kind of organizing, which is often informed by a drive to representation more than a drive to abolition. In many ways, saying that you need a more diverse, minority-sensitive tech company is like saying you need more diverse prison guards.

But as we saw with the successful campaign against the Project Maven contract at Google, this kind of organizing can make real gains. So I think their work is necessary. We need tech workers to organize so that contracts like Project Maven get cancelled. But I also think it's necessary for organizers like myself to remain antagonistic to the very existence of companies like Google and Amazon. In my political imagination, there's room for both.

I will always say: don't join a big tech company. I will always say, you're making war machines, don't get your paycheck from them.

But I also know that our work needs to be coalitional. We need solidarity between different groups that are working at different chokepoints. For instance, there are academics whose scholarship and concepts may be woefully inadequate to the work of abolition—but they’re also the ones who are teaching students before they enter the tech industry pipeline. Building a regulatory culture among technologists relies on their efforts.

It’s my hope that these different communities, that are sometimes ideologically at odds with one another, can all contribute to the project of defanging, dismantling, and interrupting these systems. How do we continue to create spaces of relief and spaces of emancipation? Because at the end of the day, that may be the best we’re ever going to do.

### **But isn’t the goal to abolish these systems?**

That’s the aspiration. That’s what we work towards. But what we celebrate as wins are the pauses and the breaks. We celebrate those moments where power recedes and people are able to live and to thrive. And we fight towards abolition because it’s an effective strategy to achieve those pauses and breaks.

The systems we’re fighting have been around for a long time. A very long time. But if you can introduce a bit of friction, you can open up some breathing room. 



# Inside the Paradise Bubble

by Ann Chen

Taiwan's response to COVID-19 has been incredibly effective. How much of its success comes down to technology?

---

“Taiwan is a paradise bubble,” my dad told me in March, during my first few days back at my parents’ home. “This is probably one of the safest places in the world right now,” he said. Seeing the rush hour crush on the Taipei metro and children in school uniforms clustering at bus stops after school, all without exhibiting signs of fear or anxiety, I couldn’t agree more.

When the pandemic began, I had been in New York. Throughout February and early March, I checked in frequently with my parents in Taiwan. Things are fine here, they said. Meanwhile, the situation in New York was worsening. Cases were beginning to appear, but the government response was hesitant and nebulous. The virus is coming, warned the media. It may already be here.

If it’s already here, I wondered, why aren’t we doing anything about it? Why is everything continuing as normal? The situation

felt out of control from the start. My parents urged me to come to Taiwan. On March 14th, 2020, I flew to Taipei on a direct flight.

My parents were right. From what I can tell, apart from masks on every face, life in Taiwan is uninterrupted by the pandemic. Schools, pharmacies, post offices, convenience stores, and parks are all open. Coffee shops, in abundance in Taipei, are full of people. Even the shopping malls are operating at regular capacity. Aside from a handful of attendants stationed at the front entrances, armed with temperature readers and hand sanitizer spray, every store remains open.

Early on, experts predicted that Taiwan, due to its close proximity to mainland China, would have the second-highest number of COVID-19 cases outside of the mainland. But the predicted wave of infections never materialized. As of late June 2020, Taiwan had reported only 447 known cases and 7 deaths. Compared with worldwide figures, these numbers are shockingly low; by comparison, more than 22,000 have died in New York City alone.

*“From what I can tell, apart from masks on every face, life in Taiwan is uninterrupted by the pandemic.”*

As countries around the world struggle to contain the virus, many observers are looking with great interest at Taiwan’s success. Rarely making international news and typically only in connection with mainland China, Taiwan has lately been held up by journalists and academics as a model for how to manage the pandemic. How did Taiwan, with a population of twenty-three

million, eighty-one miles away from mainland China, with over 800,000 citizens working there and frequently traveling back and forth, manage to avoid the public health crisis that is now destabilizing the rest of the world? One part of the answer, I discovered, is a unique mix of technological interventions—some led by the government, others coming from the grassroots—that have helped coordinate the massive mobilization of people and resources required to fight the virus.

## Fenced In

When I arrived in Taipei, I sailed through the airport. “Where are you coming from?” a health official asked. “New York,” I said. She took a health self-assessment form that I had completed, handed me a slip of paper with instructions for how to monitor my wellbeing, and waved me on. I had expected a more rigorous interrogation.

It turns out I had returned to Taiwan just in time. Had I landed two or three days later, when the authorities raised the United States’s travel advisory from Level 1 to Level 3, my experience would have been radically different. My friend Ting wasn’t so lucky. She arrived later than I did, also traveling from New York. And it was through her experiences that I first began to learn about the role of technology in Taiwan’s pandemic response.

By the time Ting left New York, the city had become a viral hot zone. Before she departed, Taiwanese authorities required that she fill out an online health screening form, providing her medical information and, most importantly, an address where she could quarantine in Taiwan. Upon landing, all passengers underwent testing, the results of which were given two days later. There was also a changing area where passengers could change out of their plane clothes. Alcohol was provided for

full-body sterilization. Ting passed through immigration and was escorted to a special quarantine taxi, to take her and other travelers to their quarantine locations. She was required to spray herself down with alcohol again before entering the cab.

When she arrived at her apartment, she was called by a local healthcare official (衛生所) assigned to her. They exchanged contacts on LINE, a popular messaging app in Taiwan similar to WhatsApp, to stay in touch. She also had to join a special LINE group, where she was expected to report her temperature and wellbeing twice daily, at 9 a.m. and again at 3 p.m. Soon after, on that same day, a government worker arrived at her door with a bag of supplies. This included garbage bags, extra masks, and some food. She signed some paperwork that stated she would commit to the entire fourteen-day quarantine process without leaving her apartment. If she had not owned a cell phone, the government would have provided one for her.

How would the government know whether Ting kept her quarantine? The authorities use mobile phone location data and cell tower triangulation to draw a “digital fence” boundary around an individual’s home. If you step outside of this zone, or if you turn off your phone, an alert is sent to the police and local health officials. Ting tells me a story of a friend whose phone shut down suddenly while in quarantine. Within a minute, the police knocked on his door.

When I asked Audrey Tang, a former computer programmer and tech entrepreneur who now serves as Taiwan’s first Digital Minister, about the privacy concerns raised by the digital fence, she was quick to point out the ways in which the program differed from surveillance tools used in other countries, such as smartphone apps or physical bracelets. Tang believes the Taiwanese approach is far less harmful. “First, it’s not GPS,”

she says. “We are not asking you to install an app that reports GPS.” The level of location specificity provided by such information would be unnecessary for enforcing home quarantines, she explained.

*“The authorities use mobile phone location data and cell tower triangulation to draw a ‘digital fence’ boundary around an individual’s home.”*

Moreover, telecommunication companies are already collecting the data being used to construct the digital fence. “This is not new information being collected,” Tang emphasized. The emergency warning broadcasting system, which sends texts about flash floods or earthquakes, relies on the same data. “Instead of collecting new data or requiring you to install a new app,” Tang said, “we repurpose existing data and existing notification mechanisms.” Most importantly, the program is sharply circumscribed: the data is only used to implement the home quarantines, Tang insisted, and nothing else.

## Fork the Government

Technological solutions have not only come from the top down, however. They have also come from the bottom up. This is due in large part to Taiwan’s uniquely robust civic technology community, known locally as g0v (pronounced “gov zero”). In its current form, g0v grew out of the Sunflower Movement in 2014, a student-led protest sparked by concerns that a trade pact with

mainland China would give Beijing more influence over Taiwan. But protesters also demanded more transparency and accountability from the Taiwanese government — demands that were taken up by the g0v movement.

Using the language of open-source programming communities, g0v claims that its goal is to “fork the government.” Digital Minister Audrey Tang, who is herself a veteran of the Sunflower Movement and a longtime contributor to g0v, explained the concept to me. Essentially, it means that g0v hackers produce alternative versions of government websites. “For each government website — which always ends in ‘gov.tw’ — that they don’t like, they just change the ‘O’ to a ‘0’” in the domain name and create their own, Tang said. In the process, the g0v community has implemented a wide range of digital tools designed to increase popular participation in policymaking, from online platforms for circulating petitions to data visualization dashboards that help citizens understand how budgets are allocated.

When the pandemic began, g0v responded creatively, using its “fork the government” model to help the authorities contain the virus. Perhaps the best known example is the collection of digital tools — apps, maps, chatbots — that g0v hackers created to make it easier for the public to buy masks through the government’s mask-distribution system.

Due to its previous experience with SARS, also a coronavirus, the Taiwanese government recognized early on the importance of mask-wearing to reduce the spread of COVID-19. The speed with which the government reacted most likely prevented a more serious outbreak in Taiwan. Taiwan reported its first case on January 21st: a woman traveling back to Taiwan from Wuhan. The next day, the government banned all Wuhan residents from entering Taiwan, the day before Wuhan went into lockdown.

Although the government made assurances to the public later that week about a sufficient supply of surgical masks, fear of future shortages led the government to halt exports of masks on January 24th. To control the mask supply, the government also decided to implement a mask distribution system, while working with local mask manufacturers to ramp up production.

Masks were initially sold in convenience stores. With over 10,000 convenience stores on the island, Taiwan has one of the world's highest ratio of convenience stores to population. Convenience stores in Taiwan provide services beyond what American convenience stores offer. You can pay your phone, electric, and utility bills; your taxes; and even your parking tickets there. You can buy everything from disposable underwear to towels to books to train and concert tickets. It even functions as a post office. Similar to Amazon Lockers, online packages can be delivered to any 7-Eleven or FamilyMart for pickup.

*“Using the language of open-source programming communities, g0v claims that its goal is to ‘fork the government.’”*

Convenience stores, then, were an obvious choice to serve as mask distribution hubs. However, it soon became apparent that the stores did not have a means of verifying people's identity via their National Health Insurance (NHI) card to approve mask purchases, without violating personal privacy by collecting this data themselves. Eventually, a card scanning system was introduced to bypass data collection. But these challenges pushed the

government to switch to selling masks from pharmacies, which are already connected to the NHI database. This allowed people to maintain their privacy and gave the government the ability to monitor the rationing of masks.

Mask rationing had been put into place on February 6th, within a week of the government launching the mask distribution system. To prevent hoarding, every Taiwanese adult with a NHI card was limited to purchasing two masks per week. To reduce congestion and long lines, the ability to purchase masks was also systematized. The last digit of your NHI card ID number indicated on which days you could purchase your masks. ID numbers ending with an odd number could purchase them on Mondays, Wednesdays, and Fridays; even numbers on Tuesdays and Saturdays. Sundays were open to everyone.

The government also worked to keep prices low, threatening to fine sellers that were inflating prices up to \$50 million TWD (a little over \$1.5 million USD). These moves were designed to ensure that every citizen would have the opportunity to purchase a sufficient number of masks for daily use at an affordable rate.

Despite these measures, however, the early days of the mask distribution system were plagued with problems. In February, when the system was still in its infancy, masks were selling out fast and the public was having a hard time finding them. There were frequent stories of people waiting in long lines to purchase masks, only to have the pharmacy run out when they got to the front. There was no immediate way of verifying whether the pharmacy you went to had enough masks until you physically arrived on location.

In response, g0v hackers came up with a solution. The idea first originated in the g0v Slack channel: a digital map that would

visualize the quantities of masks available in different pharmacies. Howard Wu, a programmer and member of g0v, noticed that many of his family and friends were sharing information in LINE groups about which convenience stores still had masks in stock, back when convenience stores were the primary places to buy masks. He built a real-time “Mask Map” which relied on crowdsourced data to display mask stock levels in different stores. Users’ geolocation data would help them find nearby stores. Since there weren’t any existing comprehensive GIS datasets of convenience stores in Taiwan, Wu used Google Maps to obtain this data. Wu’s site had roughly 550,000 visits within the first six hours.

But relying on crowdsourced data wasn’t accurate enough. Digital Minister Audrey Tang showed Wu’s work to Taiwan’s Prime Minister, who immediately understood its usefulness. The government recognized that it could improve the accuracy of such civic digital tools by providing more up-to-date data. On February 4th, two days after Wu released his digital map, the government announced the switch to selling masks from pharmacies. In a coordinated effort with Tang, the Ministry of Health and Welfare released mask inventory data at pharmacies nationwide that was updated every thirty seconds.

Wu created another version of his site with the new data—and received 830,000 hits on the first day. Soon after, using the API that Wu had built for his map, g0v hackers created dozens more digital tools to help track mask availability, from more maps to smartphone apps to LINE chatbots. A government website now lists over 130 digital products for tracking mask inventories in Taiwan, all built by civic technologists.

The maps and apps have not only served as useful tools for people trying to purchase masks, however. The government has

also relied on these tools to improve its own distribution supply chain. Officials have been able to track the fluctuating numbers in different cities and provinces, which they can use to adjust mask shipments in real time. This reciprocity between the government and the grassroots technologist community has greatly benefited both parties, and Taiwan as a whole. It also stands in sharp contrast to the top-down approach of mainland China, where technological interventions to contain the virus have taken a far more authoritarian form.

## Care Works

---

There is no doubt that initiatives like the digital fence program and the g0v mask maps have contributed to Taiwan's effective management of the pandemic. As mechanisms to help coordinate the allocation of people and resources, these digital tools have proven invaluable. But the more I read, observed, and talked to the people around me, the more persuaded I became that technology had more of a supporting role in Taiwan's success.

When I ask my friends and family what Taiwan did right, they rarely mention technology. Instead, they talk about the soothing ritual of tuning in every afternoon to the Taiwanese CDC's daily press conferences, which are led by doctors, epidemiologists, academics, and public health experts rather than politicians. They talk about the government's decision to give journalists unlimited time to ask questions, and how this has resulted in more accurate media coverage, less disinformation, and greater public trust in the information being conveyed by the authorities. They also talk about the CDC's toll-free hotline, where you can call and talk with a person about anything related to the virus. One friend of mine, worried she had contracted COVID-19, called the hotline multiple

times. She described to me how comforted she felt being able to talk to someone who could ease her fears.

In part, the Taiwanese government's multi-faceted communications strategy reflects an attempt to make up for past mistakes. The government's mishandling of the SARS epidemic in 2003, which had a lower case count but a higher death rate than COVID-19, severely undermined public trust at the time. Unaware of the highly infectious nature of SARS, one woman's visit to an emergency room set off a chain of transmission that spiraled out of control. In a desperate attempt to contain the virus, the government sealed off Heping Hospital, with more than 1,000 people, infected and uninfected, locked inside. The inhumaneness of the approach shocked Taiwanese citizens. Twu Shiing-jer, Taiwan's Minister of the Department of Health, resigned in the aftermath.

*“The more I read, observed, and talked to the people around me, the more persuaded I became that technology had more of a supporting role in Taiwan’s success.”*

Post-SARS, Taiwan immediately began planning for the next health crisis. It could not afford to be caught off-guard again — especially since it had been clear during the SARS epidemic that Taiwan would have little to no direct communication with the World Health Organization, because it is not a member. Taiwan is isolated, and on its own. This realization may have

proven decisive in its pandemic response, as Taiwan was one of the earliest countries to sound the alarm on COVID-19 and begin monitoring the virus.

Above all, when I talk to Taiwanese people about what Taiwan did right, they talk about healthcare. In particular, they praise Taiwan's single-payer healthcare system. Almost 99 percent of Taiwanese citizens and residents are covered by Taiwan's national health insurance program. (The 1 percent, the government believes, consists of Taiwanese citizens residing outside the country.) When coverage hovered around 96 percent, the government made a concerted effort to track down the remaining 4 percent — composed primarily of Indigenous Taiwanese, the unemployed, the homeless, and orphaned children — to get them enrolled. Households below the poverty line receive free coverage. Essentially, no one is denied healthcare in Taiwan.

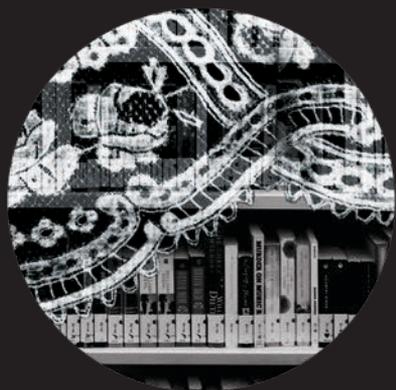
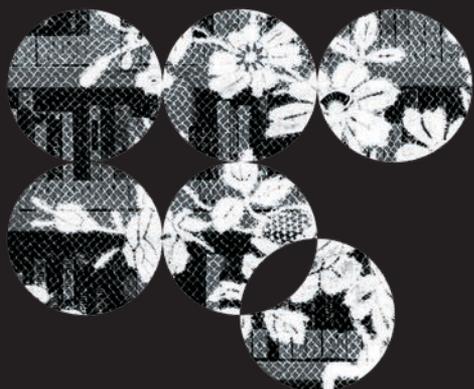
*“Ultimately, Taiwan’s success in containing COVID-19 has less to do with technology than with well-functioning state institutions that acted quickly and collectively.”*

With 99 percent of the population insured under one system, a centralized medical database made it possible for the government to rapidly implement its mask-rationing system. It also made performing contact tracing easier, as well as tracking community-based transmission. More importantly, universal health coverage means people aren't afraid of being denied medical treatment or going bankrupt from medical bills. The

government encourages citizens to report even mild symptoms, which enables the authorities to detect infection earlier.

Ultimately, Taiwan's success in containing COVID-19 has less to do with technology than with well-functioning state institutions that acted quickly and collectively. As a Taiwanese friend described it, the government's approach has resembled crossing a river. You inch forward step by step, feeling your way across and making decisions as you go. Along with this experimental, adaptable spirit, the government's focus on transparency and building public trust, paired with an excellent universal healthcare system, are the real strengths of the Taiwan model. Technology, while useful, cannot make up for the absence of strong public structures of care. In the United States, where the fight for universal healthcare is still an uphill battle, care is a luxury good with multiple prerequisites — employment, wealth, geography. In Taiwan, care is a basic human right that everyone receives equally. ~~~

Ann Chen is an artist, researcher, and filmmaker based in Brooklyn and Shanghai. She is currently an Assistant Arts Professor of Interactive Media Arts at NYU Shanghai.



# Informatics of the Oppressed

by Rodrigo Ochigame

Algorithms of oppression have been around for a long time. So have radical projects to dismantle them and build emancipatory alternatives.

---

How did you end up reading this text? If you're reading it online, you may have clicked a link in an algorithmically generated list of recommendations or search results. Or maybe a friend sent you the link — after finding it in an algorithmically generated list. Whatever the chain of events that brought you here, it likely involved a system of information retrieval.

Such systems select a handful of choices from billions of possibilities. Their existence is inevitable: at any given time, we can only comprehend a small portion of an immense world. The problem is that the systems that filter the world are not designed for your benefit but for corporate profit. No word captures the dominant form of information consumption on the internet more aptly than “feed” — a ubiquitous term derived from an agrarian metaphor. As in animal husbandry, your information diet is engineered to maximize the yield of a business operation.

If Silicon Valley’s PR departments claim that their products simply “find the most relevant, useful results in a fraction of a second, and present them in a way that helps you find what you’re looking for” — this is how Google describes its search algorithms on its website — executives and shareholders know better. They know that the whole point of the business is to get paid to show you things that you’re *not* looking for: ads.

The conflict of interest between advertisers and users has always been evident to the designers of commercial search engines. In 1998, a few months before the incorporation of Google, graduate students Sergey Brin and Lawrence Page presented their prototype of a web search engine at an academic conference. In an appendix to their paper, they commented, “We expect that advertising funded search engines will be inherently biased towards the advertisers and away from the needs of the consumers.” Indeed. More than two decades after this prophecy, all major search engines, Google first among them, now operate precisely on the business model of surveillance-fueled targeted advertising.

*“From the very beginnings of informatics as an institutionalized field in the 1960s, anti-capitalists have tried to imagine less oppressive, perhaps even liberatory, ways of indexing and searching information.”*

These search engines' algorithms are optimized for profit. The advertising industry governs the bulk of research and development in the field of information retrieval. Computer scientists and engineers often measure the "relevance" of potential results and test the "performance" of candidate algorithms according to evaluation benchmarks and validation data sets dictated by industry priorities. The predominant systems are designed to maximize ad revenues and "engagement" metrics such as "click-through rates." Consequently, these systems tend to promote content that is already popular or similar to what users have seen or liked before. Whether the predictions of popularity and similarity are based on simple correlation and regression analysis or on complex machine learning models, the results tend to be predictable and like-minded.

No wonder the public sphere seems so impoverished in the digital age. The systems that manage the circulation of political speech were often originally designed to sell consumer products. This fact has momentous consequences. Recent scholarship has documented the disastrous effects of "surveillance capitalism," and in particular how commercial search engines deploy "algorithms of oppression" that reinforce racist and sexist patterns of exposure, invisibility, and marginalization. These patterns of silencing the oppressed are so pervasive in the world that it may seem impossible to design a system that would not reproduce them.

But alternatives are possible. In fact, from the very beginnings of informatics — the science of information — as an institutionalized field in the 1960s, anti-capitalists have tried to imagine less oppressive, perhaps even liberatory, ways of indexing and searching information. Two Latin American social movements in particular — Cuban socialism and liberation theology — inspired experiments with different approaches to informatics from

the 1960s to the 1980s. Taken together, these two historical moments can help us imagine new ways to organize information that threaten the capitalist status quo—above all, by facilitating the wide circulation of the ideas of the oppressed.

## **Struggle on the Library Front**

What happens the day after the revolution? One answer is the reorganization of the library. In 1919, Lenin signed a resolution demanding that the People's Commissariat of Enlightenment "immediately undertake the most energetic measures, firstly to centralize the library affairs of Russia, secondly to introduce the Swiss-American system." Lenin presumably referred to the organization of the European libraries he had observed during his exile from Russia in the early 1900s. By imitating the "Swiss-American system," the Bolshevik leader hoped to create a single state system of centralized control over the distribution of books and the development of collections.

*“How could librarians take an active part in the revolution?”*

Four decades later, Cuban revolutionaries also recognized the importance of what Soviet leaders like Nadezhda Krupskaya had once called the struggle “on the library front.” In the aftermath of the Cuban Revolution in 1959, Fidel Castro appointed librarian María Teresa Freyre de Andrade as the new director of the Jose Martí National Library in Havana. A lesbian and long-time dissident who had been exiled and jailed by the previous

regimes, she had long been concerned with the politics of librarianship. In the 1940s, she had articulated her vision of a *biblioteca popular*, a “popular library,” distinct from a merely “public” one. Whereas the public library may be a “rather passive” one where “the book stands still on its shelf waiting for the reader to come searching for it,” the popular library is “eminently active” as it “makes extensive use of propaganda and uses different procedures to mobilize the book and make it go in search of the reader.”

After the revolution, Freyre de Andrade and her staff began to enact this vision. They brought books to the people by sending *bibliobúses*, buses that served as moving libraries, to rural areas where no libraries existed. They also began to develop a novel practice of revolutionary librarianship. Unlike with Lenin, the goal was not to imitate the organization of European libraries. In a 1964 speech, Freyre de Andrade argued that Cubans could not simply “copy what the English do in their libraries.” By doing so, “we would have a magnificent library, we would have it very well classified, we would provide a good service to many people, but we would not be taking an active part in what is the Revolution.”

How could librarians take an active part in the revolution? One answer was to gather and index materials that had been excluded or suppressed from library collections in the pre-revolutionary period, such as the publications of the clandestine revolutionary press of the 1950s. But librarians also became involved in a broader revolutionary project: Cuba’s effort to build its own computing industry and information infrastructure. This project ultimately led to a distinctive new field of information science, which inherited the revolutionary ideals of Cuban librarianship.

## The Redistributing of Informational Wealth

---

Both the revolutionaries and their enemies recognized that information technology would be a strategic priority for the new Cuba. A former IBM executive recalls that “all of the foreign enterprises had been nationalized except for IBM Cuba,” since the “Castro government and most of the nationalized companies were users of IBM equipment and services.” But from 1961–62, IBM closed its Cuban branch, and the US government imposed a trade embargo that prevented Cuba from acquiring computer equipment. This meant that Cuba would be forced to develop its own computing industry, with help from other socialist countries in the Soviet-led Council for Mutual Economic Assistance (Comecon).

Between 1969 and 1970, a team at the University of Havana created a prototype of a digital computer, the CID-201, as well as an assembly language named LEAL, short for “Lenguaje Algoritmico” (Algorithmic Language), an acronym that also means “loyal.” The design of the CID-201 was based on the schematics found in the manual of the PDP-1, a computer manufactured by the US-based Digital Equipment Corporation. Because of the US-imposed trade embargo, the team could not buy the necessary electronic components in Europe, but eventually succeeded — with the help of a Cuban man of Japanese descent who worked as a merchant in Tokyo — in bringing the components from Japan inside more than ten briefcases.

Cuban mathematicians also wrote a computer program in LEAL for playing chess; one of the CID-201’s engineers recounts that the computer even played — and lost — a game against Fidel Castro. Starting in the 1970s, Cuba manufactured thousands of digital computers, and even exported some computer parts to other Comecon countries.

The rise of digital computing transformed Cuban librarianship. Freyre de Andrade welcomed the digital age, paraphrasing Marx and Engels to analogize computing to communism: “a specter is haunting the informational world, the specter of the computer; and let’s be pleased that this circumstance has come to move our field [of librarianship], giving us a challenge that makes [the field] even more interesting than it already was by itself.” Cubans studied the techniques of informatics mostly with Soviet textbooks translated into Spanish. They combined the computational methods they learned from these books with the revolutionary ideals of Cuban librarianship. This synthesis produced distinctive theories and practices that diverged substantially from those of both Western and Soviet informatics.

*“Both the revolutionaries and their enemies recognized that information technology would be a strategic priority for the new Cuba.”*

Consider the concept of “information laws,” a staple of informatics textbooks. A classic example is “Lotka’s law,” formulated in 1926 by Alfred J. Lotka, a statistician at Metropolitan Life Insurance Company in New York, who sought to compute the “frequency distribution of scientific productivity” by plotting publication counts of authors included in an index of abstracts of chemistry publications. He claimed that the distribution followed an “inverse square law,” i.e., “the number of persons making 2 contributions is about one-fourth of those making one; the number making 3 contributions is about one-ninth, etc.; the number making  $n$  contributions is about  $1/n^2$  of those making one.”

Like Western textbooks, the Soviet textbooks of informatics adopted in Cuba covered such “information laws” in depth. Their main authors, Russian information scientists and engineers A. I. Mikhailov and R. S. Gilyarevskii, quoted a peculiar passage by US information scientist and historian of science Derek de Solla Price on the distribution of publication counts: “They follow the same type of distribution as that of millionaires and peasants in a highly capitalistic society. A large share of wealth is in the hands of a very small number of extremely wealthy individuals, and a small residual share in the hands of the large number of minimal producers.”

For Cuban information scientists, who had experienced a socialist revolution and an abrupt redistribution of material wealth, this unequal distribution of *informational* wealth also had to be radically transformed. Among these information scientists was Emilio Setién Quesada, who had studied and worked with Freyre de Andrade since the beginning of the post-revolutionary period. Setién Quesada contested the very idea of an “information law.” In an article co-authored with a Mexican colleague, he objected to the term “law,” which seemed to imply “the identification of a causal, constant, and objective relation in nature, society, or thought.” The mathematical equations represented mere “regularities,” without expressing “the causes of qualitative character of the behaviors they describe.” Those causes were historical, not natural.

Therefore, Setién Quesada and his colleague argued, publication counts did not conclusively determine the “productivity” of authors, any more than declining citation counts indicated the “obsolescence” of publications. Cuban libraries shouldn’t rely on these metrics to make such consequential decisions as choosing which materials to discard. Traditional informatics was incompatible with revolutionary librarianship because, by treating

historically contingent regularities as immutable laws, it tended to perpetuate existing social inequalities.

Cuban information scientists didn't just critique the limitations of traditional informatics, however. They also advanced a more critical approach to mathematical modeling, one that emphasized the social complexity and the historical contingency of informational regularities. In the 1980s, when Cuban libraries were beginning to adopt digital computers, Setién Quesada was tasked with developing a mathematical model of library activity, based on statistical data, for the purpose of economic planning. But he was dissatisfied with existing models of the "intensity" and "effectiveness" of library activity, devised by Soviet and US information scientists. (In the discussion below, I include mathematical explanations inside parentheses for interested readers, following Setién Quesada's own terminology and notation.)

*“Traditional informatics was incompatible with revolutionary librarianship because, by treating historically contingent regularities as immutable laws, it tended to perpetuate existing social inequalities.”*

Soviet information scientists computed the “coefficient of intensity” of library activity by multiplying the “index of circulation” (the number of borrowings  $m$  divided by the number of potential readers  $N$ ) by the “index of rotation” (the number of borrowings

$m$  divided by the total volume of holdings  $f$ ). Meanwhile, US information scientists computed the “measure of effectiveness” of libraries, combining the index of circulation with an “index of capture” (the number of actual library readers  $n$  divided by the number of potential readers  $N$ ). In contrast to these two approaches, Setién Quesada proposed an alternative “Cuban model,” which evaluated what he called the “behavior of Cuban public libraries”:

<p>“Coefficient of intensity” (Soviet authors)</p>	$\frac{m}{N} \frac{m}{f}$
<p>“Measure of effectiveness” (US authors)</p>	$\frac{m}{N} \left( 1 + \frac{n}{N} \right)$
<p>“Cuban model”</p>	$\sqrt{\frac{\frac{\bar{m}}{f} \frac{\bar{m}}{l} \left( 1 + \frac{\bar{l}}{n} \right) \left( 1 + \frac{\bar{n}}{N} \right)}{\frac{\bar{m}_i}{f_i} \frac{\bar{m}_i}{l_i} \left( 1 + \frac{\bar{l}_i}{n_i} \right) \left( 1 + \frac{\bar{n}_i}{N_i} \right)}}$

Setién Quesada argued that “the Cuban model is more complete.” It included many more variables, all of which he considered important. For instance, the Cuban model included an “index of communication” (based on the number  $l$  of readers who use the archive), while the Soviet and US models “do not express the precise level of the author-reader social communication that happens in libraries.” Moreover, those other models “do not consider the role of the librarian in the development of the activity.” For Setién Quesada, the librarians, “together with the readers, constitute the main active agents involved in the development of this activity.” Hence in the Cuban model, every variable was adjusted relative to the number of librarians (incorporated into

the adjusted variables denoted by a vinculum). Finally, the other models “do not offer an index that synthesizes the comparative behavior of places and periods.” By contrast, the Cuban model sought to facilitate comparisons of different libraries and time periods (each represented by the subscript  $i$ ).

Whatever the merits and limitations of this particular mathematical model, the broader story of Cuban information science encourages us to be skeptical of the claims attached to models and algorithms of information retrieval in the present. If yesterday’s information scientists claimed that their models ranked authors by “productivity” and libraries by “effectiveness,” today’s “AI experts” claim that their algorithms rank “personalized” search results by “relevance.” These claims are never innocent descriptions of how things simply are. Rather, these are interpretive, normative, politically consequential prescriptions of what information should be considered relevant or irrelevant.

*“We can design alternative models and algorithms in order to disrupt, rather than perpetuate, patterns of inequality and oppression.”*

These prescriptions, disguised as descriptions, serve to reproduce an unjust status quo. Just as print publications should not be deemed obsolete and discarded from library collections on the basis of citation counts, online information should not be deemed irrelevant and ranked low in search results on the basis of “click-through rates” and ad revenues. The innovative experiments by Cuban information scientists remind us that we can

design alternative models and algorithms in order to disrupt, rather than perpetuate, patterns of inequality and oppression.

## A Network Theory of Liberation Theology

---

The Cuban experiments were supported by a socialist state. But experiments with anti-capitalist informatics are also possible in the absence of such a state. In fact, another major undertaking took place in countries that were controlled by US-backed right-wing military dictatorships.

In many Latin American countries, including Brazil after the 1964 military coup, authoritarian regimes took violent measures to silence dissidents, such as censorship, imprisonment, torture, and exile. Some of the most vocal critics of these measures were Catholic priests who sought to reorient the Church toward the organizing of the oppressed and the overcoming of domination. A key event in the formation of their movement, which would become known as “liberation theology,” was a 1968 conference of Latin American bishops held in Medellín, Colombia. At the landmark conference, the attendees learned of the dynamics of oppression in different countries, and collectively declared, “A deafening cry pours from the throats of millions of men, asking their pastors for a liberation that reaches them from nowhere else.”

How could this cry be heard? The Medellín experience inspired a group of liberation theologians, largely from Brazil, to try to envision new forms of communication among poor and oppressed peoples across the world. Their objective was *conscientização*, or “conscientization”: the development of a critical consciousness involving reflection and action to transform social structures—a term associated with their colleague Paulo Freire, who had developed a theory and practice of critical

pedagogy. Towards that end, the theologians planned to organize a set of meetings called the “International Journeys for a Society Overcoming Domination.”

*“Their objective was conscientização: the development of a critical consciousness involving reflection and action to transform social structures.”*

But international meetings were prohibitively expensive, which meant many people were excluded. One of the project organizers, the Brazilian Catholic activist Chico Whitaker, explained that “international meetings rarely escape the practice of domination: in general they are reduced to meetings of ‘specialists’ who have available the means to meet.” To address this problem, the liberation theologians and allied activists envisioned a system of information diffusion and circulation that they called an “intercommunication network.” This network would make available “information that was not manipulated and without intermediaries,” break down “sectoral, geographic, and hierarchical barriers,” and make possible “the discovery of situations deliberately not made public by controlled information systems.”

By “controlled information systems,” the organizers referred to the severe state censorship of print and broadcast media that had become prevalent across Latin America. Liberation theologians wanted the liberation of information, which would enable a new phase of Freirean pedagogy: from the era of “‘conscientization’ with the intermediaries” to that of

direct “‘inter-conscientization’ between the oppressed,” in Whitaker’s words.

Since the modern internet was not yet available in the 1970s, the operation of the “intercommunication network” relied on print media and the postal service. The organizers set up two offices, called “diffusion centers”: one in Rio de Janeiro, at the headquarters of the National Conference of Bishops of Brazil where Brazilian bishop Cândido Padin, an organizer of the Medellín conference, served as project coordinator; and another in Paris, where Whitaker lived in exile with his wife, Stella, another Brazilian activist, because of his role in land reform planning before the 1964 military coup.

The diffusion centers received and distributed, by mail, submissions of short texts (or five-page summaries of longer texts) analyzing situations of “domination” from a worldwide network of participant organizations, connected via regional episcopal conferences in Latin America, North America, Africa, Europe, Asia, and Oceania. Whitaker emphasized that the texts should ideally be written by “those who have the greatest interest in the overcoming of domination, namely, those who are subject to it,” and should include “analysis of their own situations and the struggles that they were developing to liberate themselves from domination.” The organizers published every text that matched the basic requirements, without any editorial modification; translated each text into four languages (Portuguese, Spanish, French, and English); and mailed all texts for free to participants in more than ninety countries.

For Whitaker, the concept of intercommunication was rooted not only in “freedom of expression” but also in “liberty of information”: the ability for all participants to have access “to everything that the others wish to communicate to them and which serves the realization of the objectives which they share.”

Intercommunication sought to produce radical equality: “All must be able to speak and be listened to regardless of the hierarchical position, level of education or experience, social function or position, moral, intellectual, or political authority of each.” The practice of intercommunication demanded the “acceptance to heterogeneity and of the ‘dynamic’ of conflicts that go with it,” Whitaker wrote.

Finally, intercommunication required an exercise of “mutual respect” and “openness towards the others” that reflected the Christian principle of fraternity: as Whitaker put it, “the respect for what the other thinks or does... the receptiveness to what is new and unexpected, to that which poses questions to us or challenges us, or to perspectives and preoccupations that we would have been able to leave aside because they are difficult to accept.” Despite the importance of Christian values, however, the intercommunication network was open to anyone. Some participants were non-Catholic, non-Christian, and even non-religious. Padin explained that as “children of God, we are in Christ all brothers, without any distinction.”

## **The Freedom to be Heard**

---

Over the years, the intercommunication network circulated an extraordinary diversity of texts. Chadian participants examined the social consequences of cotton monoculture since its imposition under French colonial rule. Sri Lankan participants reviewed the labor conditions in the fishing industry, the profiteering tactics of seafood exporters, and the limitations of fishing cooperatives set up by the state. Panamanian participants narrated their struggle for housing and their formation of a neighborhood association. From Guinea-Bissau, a group of both local and foreign educators, including Paulo Freire, wrote about the challenges of organizing a literacy program and

changing the education system in the aftermath of the war of independence. Between 1977 and 1978 alone, nearly a hundred texts circulated in the network. These were later compiled into a monumental volume, published in four languages and discussed at regional meetings of network participants across the world.

*“By building a distributed worldwide network via regional conferences, the liberation theologians had bypassed the central authority of the Vatican.”*

This volume featured an unusually sophisticated system of indexing. Each text had a code composed of a letter and a number; for example, the aforementioned Chadian text had the code “e35.” The letters indicated the type of text — “e” for case studies, “d” for discussion texts, “r” for summaries — and the numbers were assigned chronologically. The volume was divided into sixteen numbered sections, each about a different theme of “domination.” Section III focused on “domination over rural workers,” section IV on “non-rural workers,” section VII on “domination in housing conditions,” section X on “health conditions.”

Each text was printed inside one of the thematic sections, but since the classifications were not mutually exclusive, the index of each section also listed texts that intersected with the theme despite being from different sections. For instance, the index of section IX, on education, listed some main texts — “e4” from Thailand, “e6” from Guinea-Bissau, “e38” from the Philippines — as well as other texts from different sections,

like “r3” from section X, which discussed the intersection of health and education in structures of domination. The end of the volume featured an additional index that classified texts according to “some particular categories of victims of domination”: “women,” “youth,” “children,” “elderly people,” and “ethnic groups.”

The astonishing diversity of texts circulated by the intercommunication network soon brought its organizers into conflict with conservative factions of the Catholic Church. In 1977, some readers were especially scandalized by text “e10,” submitted by a small, women-led, self-described “community of Christian love” in rural England. The text bothered conservatives not only for its explicit denunciation of “the Roman Catholic Church as an instrument of domination” engaged in “a kind of efficient and specialized ‘brain washing,’” but also for its feminist proposals, which included the refusal “to call anyone ‘father’ in a clerical context” and the commitment to “calling the Holy Spirit ‘She’ and not ‘He.’”

After a long deliberation at the Rio de Janeiro diffusion center, the project organizers decided to publish the text along with a note restating their commitment to free expression and reminding readers of the minimal requirements for publication. Still, conservative bishops complained to Vatican authorities, who were increasingly concerned by the rise of liberation theology in Latin America and beyond. Pope Paul VI, who did not sympathize with the project, sent emissaries to Brazil to intervene. The Vatican demanded that the bishops stop, claiming that the conference in Rio de Janeiro “could not take an initiative of such breadth, and had surpassed its competence by inviting other episcopal conferences to join the project.” By building a distributed worldwide network via regional conferences, the liberation theologians had bypassed the central authority of the

Vatican. Despite the Vatican's order to stop the project, a group of Brazilian organizers continued in disobedience until 1981.

*“The theologians understood that the problem is not just whether one is free to speak, but whose voices one can hear and which listeners one’s voice can reach.”*

Later on, former organizers reflected on the relationship between their intercommunication network and the modern internet. They did not know that in the original paper on the Transmission Control Protocol (TCP), which outlined the technology that serves as the basis of the internet, engineers Vinton G. Cerf and Robert E. Kahn had spoken of a protocol for packet “network intercommunication” — or simply an “internetwork” protocol, leading to the contraction “internet” a few months later. The paper had appeared in 1974, when the liberation theologians were planning their similarly named network.

In 1993, reflecting on the two internets, Chico Whitaker theorized that the “network” is an “alternative structure of organization,” much less common in “Western culture” than the “pyramidal structure”:

*Information is power. In pyramids, power is concentrated, so also information, which is hidden or kept to be used at the right time, with a view to accumulating and concentrating more power. In networks, power is deconcentrated, and so is information, which is distributed and disseminated so*

*that everyone has access to the power that their possession represents.*

There is no doubt that Whitaker and his colleagues were prone to techno-utopianism. Their hope that technological progress would finally enable a “free” circulation of information was a fantasy, since various sorts of machine decisions and human labor, structured by political-economic conditions, always filter what information circulates and to whom. Techno-utopian conceptions of “information freedom,” whether in the Californian libertarian-capitalist version or in the Brazilian liberation-theological one, are never quite right.

Yet there is a crucial difference between the two conceptions. The Californian version of information freedom is largely limited to a particular understanding of freedom of speech. The Silicon Valley firms that manage public discourse on the internet, such as Facebook, appeal insistently to “free speech” as an excuse for their business decisions to profit from posts and ads that spread right-wing misinformation.

The remarkable innovation of the Brazilian liberation theologians is that they moved beyond a narrow focus on free speech and toward a politics of *audibility*. The theologians understood that the problem is not just whether one is free to speak, but whose voices one can hear and which listeners one’s voice can reach. The intercommunication network was meant to produce more equitable conditions not just for speaking, but for listening and for being heard. Ultimately, the network’s purpose was to amplify the voices of the oppressed. Today, our task is to reformulate this more critical conception of information freedom for the digital age. Information will be “free” only when the oppressed can be heard as loudly as their oppressors.

## The Retrieval of History

---

The history of technology is too often told as a linear progression, as a series of tales of triumphant inventors, emanating mainly from North America and Western Europe. Such tales are pervasive in part because they are easy to tell. After a certain technology prevails, the storyteller can simply follow the records and narratives given by the handful of people who are already credited for its invention.

Such commonplace narratives serve important ideological functions. First, they legitimize capitalist accumulation by framing the inventor-entrepreneur's fortune as the merited payoff for an ingenious idea. This requires erasing all other contributors to the given technological artifact; in the case of search engines, it means forgetting the librarians (whose feminized labor is never valued as creative) and the information scientists whose cumulative work over the course of decades laid the foundation for Google.

More insidiously, such narratives also serve to sanction the dominant technologies by presenting them as the only ones ever conceivable. They overlook the many possible alternatives that did not prevail, thereby producing the impression that the existing technologies are just the inevitable outcome of technical ingenuity and good sense.

If peripheral innovations like the Latin American experiments with informatics did not become mainstream, this is not because they were necessarily inferior to corporate, military, and metropolitan competitors. The reasons why some technologies live and others die are not strictly technical, but political. The Cuban model was arguably more technically sophisticated than its US counterparts. Yet some technologies are sponsored by the

advertising industry, while others are constrained by a neo-colonial trade embargo. Some are backed by the Pentagon, others crushed by the Vatican.

It is crucial to recover those lost alternatives, for they show us how technologies could have been otherwise — and could still become so in the future. However, these histories are difficult to retrieve. Their protagonists may remain anonymous and their records unpreserved.

No search engine pointed me to the Latin American experiments. I could never have found them through traditional methods for searching the internet. Instead, I came across subtle clues through serendipitous conversations. I was chatting with Theresa Tobin, a retired librarian at the Massachusetts Institute of Technology who co-founded the Feminist Task Force at the American Library Association in 1970. She commented that after she fundraised to donate a digital computer to a Sandinista library in the 1980s, Nicaraguan librarians used it to implement a Cuban system for indexing materials.

I set out to learn more about the Cuban system, a task that proved laborious. Even the most important sources on Cuban information science are hard to find using conventional search engines and databases. For instance, despite the prominence of María Teresa Freyre de Andrade, Google Scholar does not index her main books, and Wikipedia lacks an entry on her in any language. On the other hand, the Cuban online encyclopedia, EcuRed, features an extensive article on her. I also managed to find a few initial references on Cuban informatics in SciELO, a Latin American bibliographic database. I then contacted Cuban scholars directly to ask for help.

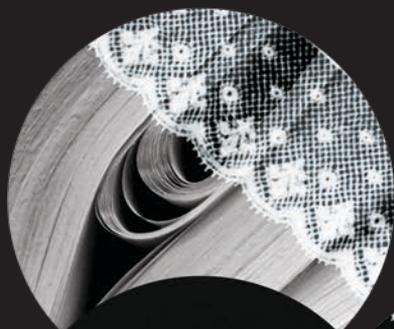
My discovery of the liberation theologians' intercommunication network took a similar path. When I first met Stella and

Chico Whitaker at the World Social Forum in Porto Alegre, which they co-founded in 2001, I had never heard of the inter-communication network. It was only years later, when I was helping the couple donate their personal papers to a public archive, that they mentioned in passing that one of the dusty boxes in their apartment contained documents from an old project involving informatics. They were surprised I showed interest. Sometimes the best method of information retrieval is talking to people.

Many more vital ideas for alternative futures, technological and otherwise, remain forgotten in dusty boxes across the world. The repressed dreams of past struggles will not easily appear on our corporate algorithmic feeds. To recover these lost ideas, we must develop more critical methods of information retrieval, continuing the work that the Latin American experiments left unfinished. In short, we need critical search.

The project of critical search would recognize that any quantification of “relevance” is an interpretive, normative, and politically consequential act. Critical search would actively strive to increase the visibility of counterhegemonic intellectual traditions and of historically marginalized perspectives. We must build systems of information diffusion and circulation that seek to amplify critical voices and to cut across linguistic, national, racial, gender, and class barriers. Let us draw inspiration from our predecessors, and try to follow in their footsteps. Let us experiment with algorithms, interfaces, and tactics for reindexing the world anew. ~~~~

Rodrigo Ochigame is a PhD candidate in science, technology, and society at the Massachusetts Institute of Technology.



# Do No Harm

by Niloufar Salehi

Social media companies are framing the problem of widespread online harassment in a way that maximizes profits without limiting harms. It's time for a new model.

---

In early 2017, an investigative journalist uncovered a private Facebook group called Marines United, where hundreds of veterans and active-duty marines were circulating nude or invasive photos of military servicewomen without their knowledge. “Dozens of now-deleted Google Drive folders linked from the Facebook page included dossiers of women containing their names, military branches, nude photographs, screenshots of their social media accounts and images of sexual acts,” the journalist, Thomas Brennan, later wrote for the news site *Reveal*.

The Marines United scandal stood out for the coordinated nature and scale of the abuse, but it was only one egregious example of how toxic social media platforms have become. According to a study by the Pew Research Center in 2017, 41 percent of Americans have personally been subject to abusive behavior online, and one in five have been the targets of particularly severe forms such as sexual harassment, stalking,

revenge porn, physical threats, and sustained harassment over time. Those who experience online harassment suffer from mental or emotional stress or even fear for their personal safety, and the stakes are particularly high for young internet users: other studies have found a significant association between cyberbullying and depression, self-harm, and suicidal ideation in people aged 12–18.

Following the Marines United revelations, Facebook quickly shut down the private group, but similar ones immediately began cropping up on the platform. Most major social media platforms ban sexually explicit photos, particularly those flagged as non-consensual, so if a victim sees the content they can report it. But this doesn't keep the images from being shared on private groups where they are less likely to be reported. As a result of mounting pressure from advocacy groups for victims of sexual assault, Facebook promised to take a more active role in addressing such harassment.

The result, which Facebook began rolling out in April 2017, and which now includes partnerships with nine organizations across eight countries, is one of the most proactive efforts by any of the social media companies to address online abuse. And yet, despite the amount of time and money that Facebook has spent on the program, the whole thing is ultimately doomed to fail. It's a revealing failure, though, because it points to fundamental limitations in the way that social media companies think about—and have encouraged the rest of us to think about—the problem of online harm.

## **The Narrowsight Board**

---

Facebook's process for addressing revenge porn and other non-consensual sexual images requires victims to upload

their images to the company, where a “specially trained” employee reviews the image and then creates a digital fingerprint of it. This allows image-matching software to detect if the same photo appears elsewhere on the site or is later uploaded again. This can potentially help some victims, but it has major shortcomings. It assumes that the victim has access to the non-consensual image, and it requires that the victim trust Facebook with extremely sensitive content. The image-matching software is also remarkably easy to fool; slight alterations to an image, such as changing the background, have been shown to elude the technology.

More importantly, the technology takes control away from the victim by assuming that deleting the images automatically is all that a victim wants. The victim never learns whether anyone else has tried to upload the images and has no proof for further action. Platforms even routinely ignore “preservation letters” from lawyers of victims of revenge porn, and delete crucial evidence.

At the most basic level, Facebook’s process, like other attempts to address online harm, suffers not from faulty algorithms, but from a crucial misrepresentation of the problem. Social media companies have construed a wide range of online harms as essentially problems of content (violating photos, violent or threatening posts, Nazi symbolism). As a result of this framing, the solution to online harm has largely been presented by these companies as “content moderation”: removing posts that a platform deems against the rules or toxic, and occasionally banning the user who posted the content.

Social media companies have a strong incentive to adopt the content moderation framework, which was originally developed to minimize spam, and all of the large social media companies

moderate content to some extent. That's because the quality of their platforms would spiral downward if they didn't. Imagine logging into Facebook and seeing an unabated stream of violent images and junk messages — you probably wouldn't want to log in again. Most advertisers don't want their ads to show up next to such content either. A decline in user engagement and ad sales is bad for a social media company's bottom line, and removing potentially offensive content is the cheapest way to ensure that doesn't happen.

But although it might maximize profits, this way of thinking about online harassment is almost entirely unable to address the harm that harassment causes. It assumes that the problem is individual pieces of harmful content that must be moderated — not people and their relationships. As a result, content moderation fails to serve the needs of those who are harmed online or to change the conditions that make such harm possible.

Once the problem of online harm is framed as content moderation, it is already a lost cause for victims. Inevitably, platforms claim that the sheer amount of content makes it impossible to monitor. This is true, but it conveniently leaves out the fact that every single decision made by platforms prioritizes scale, and platforms generally avoid taking actions that might reduce user engagement. At the same time, as the scholar Sarah T. Roberts has detailed, they strive to minimize costs, especially for things like human moderators.

If platforms accept that something needs to be done about online harm, but frame the problem as needing simply to detect and remove content that breaks platform rules, the logical next step becomes automating this process. When Mark Zuckerberg was questioned by Congress in 2018, he mentioned “artificial

intelligence” more than thirty times, framing it as “the scalable way to identify and root out most of this harmful content.”

Many experts disagree that AI will ever be able to apply content moderation rules effectively by itself. Even if it could, abusers quickly adapt to automatically enforced content moderation rules, for instance by editing non-consensual sexual images to bypass AI detection, creating fake porn videos of victims using deepfake techniques, or by coordinating attacks and using memes to mask their intent. What the vague promise of a forthcoming technological solution *has* been effective at doing is dampening calls for greater regulation of social media platforms.

*“This way of thinking about online harassment assumes that the problem is individual pieces of harmful content that must be moderated – not people and their relationships.”*

The content moderation framework also severely limits our collective ability to push back against the way social media companies deal with harm on their platforms. Most platforms have no mechanism for contesting moderation decisions. In response to pressure from users and lawmakers, Facebook recently created an oversight board for content moderation, but because the problem itself is so narrowly defined, so is the solution. The oversight board can only review individual pieces of content, and only if the author of the content objects to

their removal from the platform. This means that the oversight board has no influence over what content is allowed, how private groups are governed, or what the platform's algorithms promote. Nor can it do anything to support people who have already been victims of harassment.

In order to do that, we need to think differently about the problem of online harm. Over the past two years I have been working with collaborators in communications, information studies, and computer science to imagine alternatives to the content moderation model. Our greatest resource has been abolitionist traditions that have challenged the criminal justice system for dealing with offline harm, and which have been extremely influential in shifting public perceptions of policing in the wake of the police murder of George Floyd.

*“Violations create obligations, and the central obligation of restorative justice is to right the wrong.”*

In particular, those rallying to abolish prisons have developed models of restorative and transformative justice that can expand our imaginations of what can be done to address harm more generally. In listening to members of online communities, my research group has found that what is needed is not more sophisticated ways to identify and remove offending content — just as we don't need better ways of policing and imprisoning people — but ways of supporting survivors and transforming the societies in which harm happens, including our online social worlds.

## Obligation and Restoration

---

Practices of restorative justice, which have their roots in Indigenous ways of repairing harm, are particularly useful to rethinking how we address online harm.

Restorative justice views harm not as a crime against the rules of the state, but as a violation of people and their interpersonal relationships. Violations create obligations, and the central obligation of restorative justice is to right the wrong. At minimum, restorative justice requires that we address the victim's needs related to the harm; hold offenders accountable to right those wrongs; and involve victims, offenders, and communities in this process.

The primary tool of restorative justice is communication, and practitioners have developed structured modes of communication to take people through a process of reckoning with and repairing harm as much as possible. Although restorative justice has to be adapted to different communities and circumstances, its principles have been successfully codified into processes and formal training and embedded in churches, workplaces, and neighborhoods. In Oakland schools that have implemented restorative justice, students feel more respected by and connected to their peers, and detention rates have fallen to zero.

What would an approach to online harm grounded in restorative justice look like? My research group has conducted extensive interviews with restorative justice practitioners, as well as with the moderators of online communities on platforms such as Reddit and Discord.

As part of this work, we practiced speculative design: giving participants real-world scenarios of online harm and asking them to

discuss the ways that those harms might be addressed, including assessing which scenarios that were most likely to happen, most desirable, and most unexpected. Here is a basic table contrasting the kinds of questions that would be asked when using a content moderation framework versus one based on restorative justice:

CONTENT MODERATION	RESTORATIVE JUSTICE
What content has been reported?	Who has been hurt?
Is the content against the rules?	What are their needs?
Should the content be removed, demoted, flagged, or ignored?	Whose obligation is it to meet those needs?

For example, in the case of non-consensual sexual images like the ones shared in the Marines United scandal, a restorative justice process might begin with the following questions:

- *Who has been hurt?* The person whose private, intimate photos were shared without their consent.
- *What are their needs?* Victims have a diverse range of needs, including: protection, if they are in immediate danger; validation; and contextual information (Are they being stalked? Have the images been shared? Where?). While “blocking” is often presented to victims as a way to deal with abusers, in practice many victims seek to actively monitor their abuser so that they are not caught

off-guard. Some victims want the harm to be acknowledged by the person who committed it, or by the community. Many victims want the harm not to happen again, to themselves or anyone else.

- *Whose obligation is it to meet those needs?* The obligations mentioned above fall to many people: the person who committed the harm, the wider community of people who know and care about these people, and the platform that provided the space for the harm to occur.

While the obligations of the person who committed the harm and of the wider community have been extensively discussed by restorative justice researchers and practitioners, the obligations of platforms are new and unique to online harm, and deserve special attention. Each case is different, and should be addressed individually and with care, but here are several possible approaches, which may need to be combined:

- *Assign a trained caseworker.* In the most extreme cases of harm, such as sharing non-consensual sexual images, trained caseworkers should be assigned to the case. Facebook currently assigns a “partner” to someone who seeks to report revenge porn, but their role is only to instruct the person on how to upload their images to Facebook. Instead, the case worker should support the victim, provide validation, share information on their options, and help them identify and activate a community of people that cares for them and can be involved in the restorative justice process.
- *Be trauma-aware.* Current methods of addressing online harm, including Facebook’s revenge porn process, are detached from the reality of being a victim of sexual harm. For instance, sharing private,

intimate photos with unknown Facebook employees can retraumatize the victim. An alternative could be giving victims the tools to create a digital fingerprint of the image themselves with the help of trained outside support groups, who could verify and share the symbolic identifier with Facebook without sharing the actual photos.

- *Support the harmer to take accountability.* With the victim's consent, their case worker may reach out to the offender, have a conversation about the harm, and gather and provide information. At present, Facebook deletes the photo or prevents it from getting uploaded, and in some cases they ban the offender with little explanation. This actively discourages responsibility on the part of the offender. But offenders need to be encouraged to understand the harm and to work towards repairing it. That might mean apologizing and working to not repeat the harm, becoming educated about the broader effects of the harm, or paying reparations for harms done in the past.
- *Stop the continuation of the harm.* This may mean placing limitations on posting, or even removing the person who has committed harm from the platform. This does not mean that anyone who commits harm should be immediately banned forever, but does mean that removal from the space is a valid and possible consequence of harm. Removal should be done with care so that it is not simply a punitive response but is actually necessary to stop the harm and is done with explanations and an opportunity to engage with the process.

- *Create structures for continued accountability.*  
Platforms should make addressing harm a priority before it occurs. This can mean setting expectations that sexual harm will not be tolerated when members join, and periodically reminding them as well as explaining what accountability protocols exist to address harm. It can also mean training and supporting users to intervene in sexual violence. Facebook Groups currently rely on admins from the group to moderate it; the platform could mandate that groups above a certain size conduct a yearly review of their protocols for dealing with sexual harm, and this could include designating at least two point people to whom such issues should be reported and a process for addressing incidents.

## **Communities of Care**

---

Restorative justice approaches to online harm require resources, time, training, and labor. Each harm case—from Donald Trump threatening violence against protesters to trolls sending journalists violent memes—will be unique and require a creative response. It’s a revolutionary idea that victims of harm, who are often people with the least power in society, should be cared for, and we cannot expect today’s large social media companies to commit the resources necessary to do this of their own accord. But we can pressure them to, and we can create alternative platforms that do.

The project of content moderation is costly, complicated, and controversial. To move beyond it, platforms need to spend considerably more resources to hire experts who are trained in restorative justice, trauma response, community accountability,

and anti-racism. They would also need to help fund outside organizations such as victim support groups. This would mean lower profit margins; it would also mean adapting protocols to different cultures and geographical regions.

All of this would be difficult and expensive to achieve at the scale of current platforms, which is an argument for breaking those platforms down into smaller online communities governed by the users themselves. If companies whose main activity is to host user-generated content can't effectively address harm on their platforms, maybe they shouldn't have such high profit margins — or maybe they shouldn't exist at all.

*“The future we should be working toward is one in which every single person has the skills to identify harm and work towards justice.”*

My research group has begun to practice some aspects of restorative justice in online communities in coordination with the moderators of those communities. Pre-conferencing, which involves one-on-one conversation between the mediator and different people involved in the harm, is often the first step of a restorative process. In order to get a deeper understanding of the types of harm that happen, the needs of those who are harmed, and what potential next steps could look like, we are currently conducting pre-conferencing interviews with people who have been harmed in online gaming communities, those who have been banned from certain games, and moderators.

In building a just future we cannot rely solely on the intervention of platforms, or on restoring justice one harm at a time. Even as we work towards restoring justice right now, our long-term aim must be to transform the societies in which harm occurs. This is the work of transformative justice, which was popularized by women and trans people of color as a way to address interpersonal violence and tie it to structural and systemic forms of violence. As the organizer and educator Mariame Kaba puts it: “I am actively working towards abolition, which means that I am trying to create the necessary conditions to ensure the possibility of a world without prisons.”

The future we should be working toward is one in which every single person has the skills to identify harm, hold themselves and others accountable, and work towards justice. At the same time, we must transform the social conditions, including patriarchy and racism, in which harm thrives. This kind of work leads us to fundamentally transform our relationships with one another, and it cannot be scaled or outsourced. When building a future that addresses online harm we should not seek mere alternatives to content moderation; we should work towards a world where no content moderation is needed. ~~~~

The work discussed here is in collaboration with Sijia Xiao, Anna Gibson, Amy Hasinoff, and Shagun Jhaver.

Niloufar Salehi is an Assistant Professor in the School of Information at University of California, Berkeley, where she directs the TO3 research group.



# Logistics Workers of the World

## A conversation with Agnieszka Mróz of Amazon Workers International

In order to manifest its customers' dreams of frictionless consumption, Amazon has built hundreds of warehouses around the world and filled them with hundreds of thousands of warehouse workers who ensure that the things we order online appear, as if by magic, on our doorsteps. But warehouse workers, not magic, are how freight trucks get unloaded, forklifts get driven, items get picked from endless shelves, and boxes get packed for distribution. COVID-19 has made their work essential.

Rather than quadrupling their pay, giving them more breaks and ample time off, and providing them with PPE, "the company has thanked us in announcements and on bathroom stalls for our great contribution to keeping society running," according to a letter to Amazon leadership from the coalition of warehouse workers Amazon Workers International. The working conditions have always been bad. By now, many of us have read accounts describing the nightmare of "making rate" – the algorithmically generated and biometrically tracked productivity quotas that workers must meet in the warehouses.

But warehouse workers ask that we keep the pity party to ourselves. They are too busy to comfort us between making plans on the company bus, stealing time from the boss, sharing notes about the choke points across the company's vast physical and digital infrastructure, and having each other's backs across international borders.

We talked to Agnieszka Mróz, a member of Amazon Workers International and a shop steward at Amazon's Poznań, Poland warehouse, about how to organize to win against the great monopolist.

---

### **What is Amazon Workers International and how did it come about?**

I'll start at the very beginning just to give you the big tour.

The Poznań warehouse was Amazon's first warehouse in Poland, and it opened in 2014. I've been working in that warehouse since then. When it opened, we thought that it was going to be part of a larger shift, where Amazon was going to move warehouses from Germany to Poland. There were a lot of strikes in Germany at that time, and we've also seen that shift before with other factories and warehouses because we are much cheaper labor than Germans. We earn about four dollars per hour, so three to four times less than they do.

At the same time, German workers were also afraid that the warehouses were going to close and that they would lose work. We connected with some of them through social movement organizing and met in person for the first time in 2015. We quickly realized that Amazon was building power by exploiting the differences between countries. Over time, it's become clear that the company is not closing German warehouses; they just

want to use us as a cheaper, more flexible workforce to limit the bargaining power of workers in Germany. We decided back then that we needed to stay in touch with each other. The Polish-German connection was very important to the beginning of Amazon Workers International.

We've met every year since then, twice a year, and we've also expanded to become a larger network that includes French and Spanish warehouse workers. Workers from a few US warehouses have also joined. The Amazon Workers International name and logo are new, but it's not a new organization. We met in Madrid in March this year and decided to go more public.

The network relies on the organic connection between the warehouse workers from different countries — we don't have union officials talking in the name of workers — and the fact that we share many of the same problems. Our warehouses look exactly the same inside, and Amazon uses the same disciplinary tools against all of us. We update each other about the struggles in our warehouses and think about how they're connected. We share an understanding that because Amazon is a global company, we need to have a global movement.

To give you an example, we have very different labor laws in Poland than they do in other parts of Europe. It's much easier to strike in France and Spain and Germany. So part of what we do together is look for ways of struggling beyond those legal differences. In Polish warehouses it is nearly impossible to organize a legal strike, but we are connected with other workers who can. That gives us power; we don't have to sit down and cry. We can support strikes in Germany with petitions, rallies, stickers, leaflets, press conferences, actions that share their slogans — and then the German workers strike for the common cause. That's how we fight together against something like rate increases.

## **What does it look like for you all to organize in an environment where you can't legally strike?**

There was an action in 2015, a year after the warehouse opened, that gave us momentum early on in our organizing.

I was there on the night shift that night. As I mentioned earlier, we'd connected with German warehouse workers at the beginning of 2015, and we knew when they were going on strike. We'd been distributing leaflets in the warehouse and on the company buses saying the Germans would be going on strike and what their demands were.

*“We also don't have to limit ourselves to the restrictive legal frames of labor and union law.”*

Around the same time, Amazon had announced that we'd be required to work obligatory overtime. Eleven-hour shifts instead of ten. People understood what was happening: that the Germans were going on strike and Amazon wanted us to work longer to make up for them not working. That made us really angry.

The night of the action, everyone on the shift was talking on the company buses on the way to work. The idea was to do something in the last hour, during our obligatory overtime, but it ended up starting much earlier. The slowdown took place mostly in the Pick department. The pickers picked one item from the shelves for each tote instead of the usual twelve or fifteen. Sending the boxes to the Pack department like that made a mess

of the conveyor belts; thousands of these mostly empty Amazon totes were falling from the belt, which then brought the Pack and the Ship departments to a standstill.

It didn't take hundreds of people. It was really clever to recognize that the Pick department is a choke point. Some people say that the Dock or Ship departments are the choke points in the warehouse since, when you do a labor action in the Ship department, you block trucks from leaving. But this was in Pick.

Pick is where they send people who join Amazon on short-term contracts from temp agencies because they can train a picker in a few hours. That's what was unique in this action, that these workers who don't have special training—they weren't, you know, forklift drivers—understood how to shut down a warehouse. So it was amazing, this popular wisdom. It showed us that we don't need a labor sociologist to tell us “do it this way” and that we don't have to limit ourselves to the restrictive legal frames of labor and union law.

### **What happened after that?**

Retaliation. Amazon interviewed about ten workers and some of them, under pressure, signed a statement saying they took part in the action and regretted it. Amazon only stopped when we made their interrogations public. We defended a woman in court who was fired afterwards—or, she was not technically fired, but her contract was not renewed. One permanent worker was also fired and we've been fighting that in court for the last four years.

After the action, our union entered into a formal “labor dispute” procedure where we brought our demands to negotiations with Amazon management. It was not very useful. Our union believes that actions, not negotiations, are the best way of talking to Amazon.

In the years of organizing since then, we've had ups and downs. There have been other important actions, and one of them happened last summer. Five thousand workers took part in a strike vote. That's still not enough to win the right to a legal strike because you need 50 percent of the whole company. To give you a sense, there are about 8,000 workers including temps in our warehouse, and there are nine warehouses in Poland. So while we didn't get enough votes this time, there's an army of people in Poland who *did* vote to strike. That's what we think about work conditions in the warehouses.

Still, it's difficult because of the permanent turnover — people joining the warehouse and then leaving, and more people on short-term contracts who don't have labor protections, so their contracts are not prolonged if they're going on sick leave, not meeting rate, or are open union members.

**That seems like difficult terrain on which to build long-term organizing relationships. How does your union adapt to Amazon strategies like short-term contracts or their “employee forum,” which sounds kind of like an internal company union?**

You have these employee forums all over Europe. Amazon uses theirs to advertise that they have “very good contact with the workforce” and “eight ways of communicating with employees.” One is the employee forum, another is a board at the company that any worker can send questions to and the board will answer. They have opinion polls every day! They're really proud of this.

We call it a yellow union, which is a union that was started by the boss to use against a proper worker's union at the negotiating table. They do this so they can always bring their own union and say, “We listen to the workers, but you all are just trouble-makers who promote your own interests.”

Amazon is, of course, quite clever, so they advertise this body as a form of worker representation, and there are some people who are interested in representing workers. But the critical point is that it's not covered by labor law in any way, while the union is. If you're not meeting rate, or if you get any form of punishment or write-up, it doesn't matter that you're part of that body; the employee forum doesn't protect you. It's fully controlled by Amazon, and they can kick you out and fire you. It's a clever management trick.

*“While we didn't get enough votes this time, there's an army of people in Poland who did vote to strike.”*

There is a general election for that body every four years. We vote on who is going to be a rep. Amazon organizes the whole process of elections, and they use the employee forum in certain formal situations. For example, by law, if there is a work accident involving a union member, the union member has the right to ask that a union representative be present. Amazon always tries to say that employee forum members should be there instead. This is a site of constant struggle for us.

**It sounds like the HR department within companies, where theoretically you can report harassment to HR and they'll help you. But that's often not what happens because part of the HR department's job is to protect the company from lawsuits.**

**Is there a general consensus among you and your coworkers that that's what the employee forum is for? Or are there some people who see it more positively?**

Some workers find that the employee forum can be useful for solving individual problems, like if someone has an issue with their manager. But this body is legally not allowed to participate in negotiations over our essential issues: wage and rate, how much we get paid and how fast we work. They mostly just talk with Amazon about, like, where to position the fan in the canteen.

## **Wildcat Since Way Back**

---

**You talked about the importance of the fact that your union is worker-to-worker. Can you talk more about that and the organizing tradition you are coming out of?**

In Europe, you have many different union traditions, from business unions to more grassroots unions. And there are a lot of different union networks and organizations working on Amazon labor issues. In Amazon Workers International, we are convinced that the way we build power is in the workplace, in our local warehouses, and with warehouse workers in other countries. Our only criteria for joining is that you have to be a worker who's organizing with others in your warehouse. That one rule reflects how we think labor movements should grow. We do not think that consumer boycotts or politicians making a spectacle of our situation will help us. That is not how we build power.

We are also different from many big unions in that we don't have professional organizers with full-time union jobs; we all work in the warehouse. We meet directly with other workers and don't have union bosses above us who tell us what to do. Despite language differences — because sometimes we speak seven languages in our meetings, it's really crazy — it has been easy to find a common language with Amazon workers from different countries. If you've ever worked a full shift scanning items or packing

boxes, you just understand how it is, how they exploit you. There is a desire to talk to each other and hear how others are fighting against things like quotas and disciplinary actions.

*“Our only criteria for joining is that you have to be a worker who’s organizing with others in your warehouse.”*

We are invited to demonstrations and debates with groups and networks that come from other traditions. And we do attend and cooperate. But in Amazon Workers International, we have a shared recognition of where our power comes from.

**On the organizing tradition question, I have to ask: have you read the Wikipedia article on Poznań?**

No.

**I was reading it ahead of our conversation and it links to an article about these protests in 1956.**

Oh, yeah.

**The government raised the work quota so that it wouldn’t have to pay workers at the Cegielski metal factory their full compensation. Workers responded by walking out one morning in what turned into a march of 100,000 people. Raising an arbitrary quota in order to lower pay sounds like something Amazon would do, so I was curious if that history impacts organizing in your city today.**

It's interesting that you mention this. Our union is just one section of a larger union called Workers' Initiative. Workers' Initiative was started in the early 2000s by workers at the Cegielski factory. What happened in that factory in 1956 was a massive moment in the history of organizing against the Communist state, and it was also connected to the Hungarian Uprising later that year. Those workers faced harsh retaliation and eighty protesters were killed. What we've done doesn't compare, but we are inspired by that history.

*“The challenge for our work together will be whether tech workers are able to see themselves as workers who are dependent on their wages.”*

To give you a bit more of the historical context, there was a transformation in Poland when the old regime collapsed. The new regime that came into power in the 1990s was basically shock therapy for working-class people, and all the unions supported it, including Solidarność leaders who used that period to get into politics. In the 2000s, young workers at the Cegielski factory had had enough and decided that they wanted a new form of labor organizing. Workers' Initiative came out of that. We are inspired by that tradition and the rejection of the big unions that supported company “restructuring,” which always meant dismissals.

So we are connected to that factory emotionally, but there's another connection as well. The factory had 20,000 workers

in 1956. Now it has something like 800. The old working class that made up the heavy industry sector — that factory makes engines — was destroyed in Poland in the 1990s and 2000s. Our union had a lot of discussions about what the new field of working-class formation would be. As Poland has become a big warehouse for Western Europe, we've come to think that logistics will be the crucial sector for the future of the labor movement.

So yes, we know the story of 1956 and we've tried to learn from it.

**Shifting gears, I want to ask you about the role of tech workers. Amazon recently fired two tech workers in Seattle for organizing. Shortly after that, a VP “quit in dismay” after watching the event that you spoke at, partly because he was so moved by hearing from warehouse workers. How do you see the role of tech workers in your struggle against Amazon?**

In Poland, we don't really have... Well, we do have a tech arm of Amazon in Poland. We know there are a few hundred tech workers in Gdańsk working on Alexa, but we've never been in touch with them. Amazon doesn't have many of these upper-level workers in our country. For the most part, Amazon in Poland is just warehousing for the West. So we don't have an organic connection.

We do appreciate our discussions over the last few months with tech workers from Seattle. I think what they did was brave, and we need their support. We don't have the symbolic position they have, so it's powerful when they can give us access to the space they get. But the challenge for our work together will be whether tech workers are able to see themselves as workers who are dependent on their wages. If they are able to organize on those grounds, then we'll have a foundation to build on together. What we would rather avoid is a situation where they only see us

as pitiful, helpless people. If the only thing they do is talk about how scandalous our conditions are, that's not useful. We need to recognize our power, and increase it together so we can make real change. The balance of power is so unequal now. We're past the point of calling on Amazon management to make a little change here and there.

Another challenge is that as warehouse workers we build our movement on our own anger; we know exactly why we're angry with Amazon. But if you are a tech worker and you design all these tools to discipline us, your experience is very different. You have to be aware of what you're doing. The tools they're creating are not neutral. They're designed to spy on us every second of our ten-hour shift, constantly increase our productivity, and literally work us to death. Last week, a worker in our warehouse died on the shop floor. The tech workers don't see that. 

# Open Casket Series

Indira Allegra  
2017

The Open Casket Series began in 2017 as a series of memorials to families who have lost loved ones to police violence. Here, I use the cloth structure of crepe, a material used to line caskets, to create digital “fabrics” from public interviews with grieving families. In the Open Casket Series, the structure of these fabrics are cut, flipped, juxtaposed, and layered one atop another, forming sometimes chaotic compositions which respond to the feelings of profound grief, distress, and anger expressed by these families.

Through this multipart series I am able to weave together tensions from multiple experiences, time periods, and locations. I had originally intended each of these images to be shown in distinct, three-dimensional installation contexts for public viewing – not unlike the way in which an open casket funeral is conducted. During the pandemic, however, the ability to collectively mourn deaths has changed dramatically and so these images live again in a two-dimensional space where we must now learn how to collectively mourn – either through the frame of the page or the screen. Now that we are on viral time, days feel collapsed. Past wounds feel like present pain. The uprisings of today mirror the resistance of yesterday. The families of Gerald Javon Hall, Philando Castile, Frank Clark, Norman Gary, Jamal Rollins, Carnell Snell, and Kajuan Raye wait alongside the families of Stephon Clark, Breonna Taylor, Tony McDade, George Floyd and so many more for justice.

## The mourners in this work include:

Darnelle Bell, the best friend of Gerald Javon Hall (1987–2016).  
Diamond Reynolds, the partner of Philando Castile (1984–2016).  
Jasmine Lloyd, the girlfriend of Frank Clark (1984–2016).  
Margaret Johnson, the mother of Norman Gary (1987–2016).  
Monique Douglas, the cousin of Jamal Rollins (1995–2016).  
Monique Morgan, the mother of Carnell Snell (1997–2016).  
Karonisha Ramsey, the mother of Kajuan Raye (1997–2016).  
Michelle Raye, the aunt of Kajuan Raye (1997–2016).



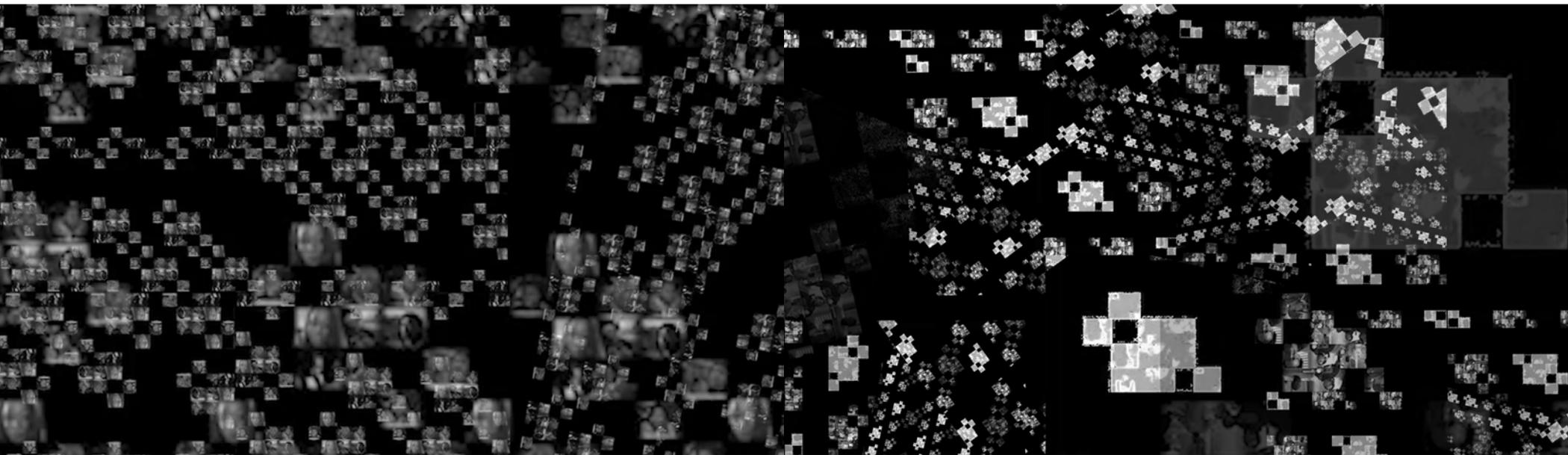
## *Open Casket IX*

Weaving drawdown for crepe (casket upholstery), video, 2500 pounds (1133 kg) hand-shoveled marble, dolomite, granite, concrete, three channel audio of families who have lost love ones to police violence.

42 x 42 x 22 FT | 12.8 x 12.8 x 6.7 M

2017

*Courtesy of the artist and Kadist Collection.*



***Open Casket I***

Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*

***Open Casket II***

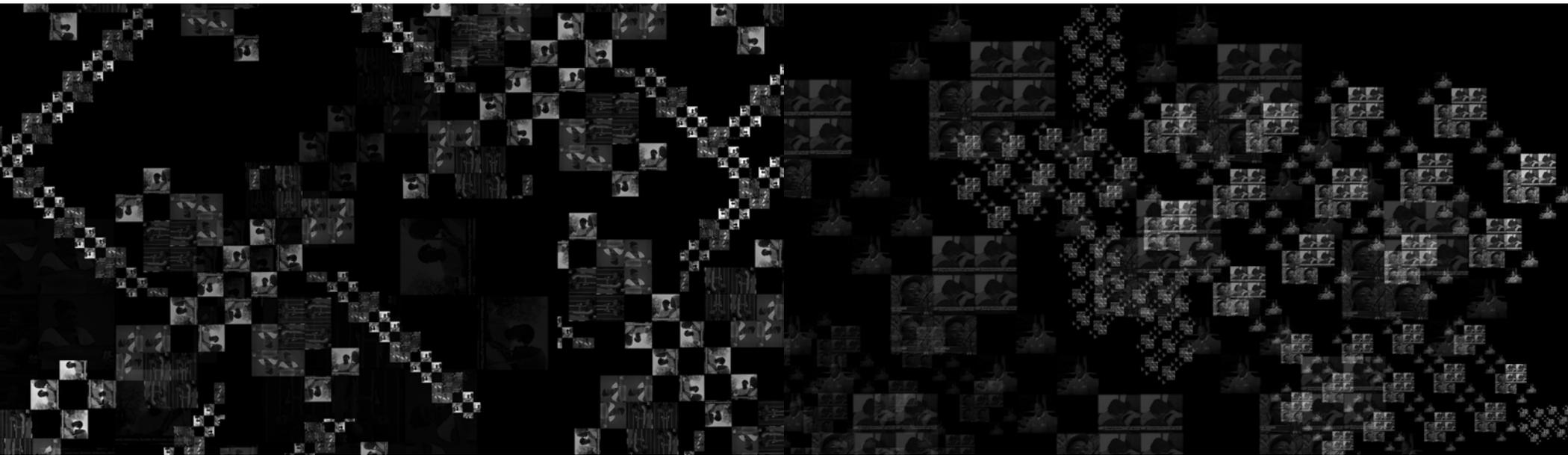
Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*



***Open Casket III***

Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*

***Open Casket IV***

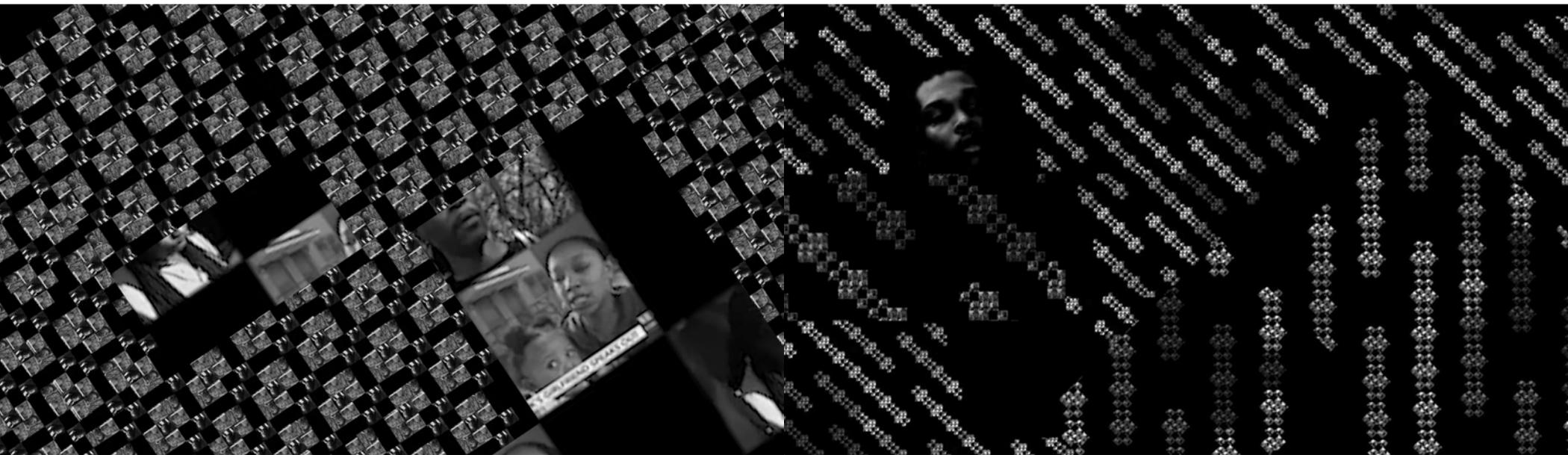
Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*



***Open Casket V***

Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*

***Open Casket VI***

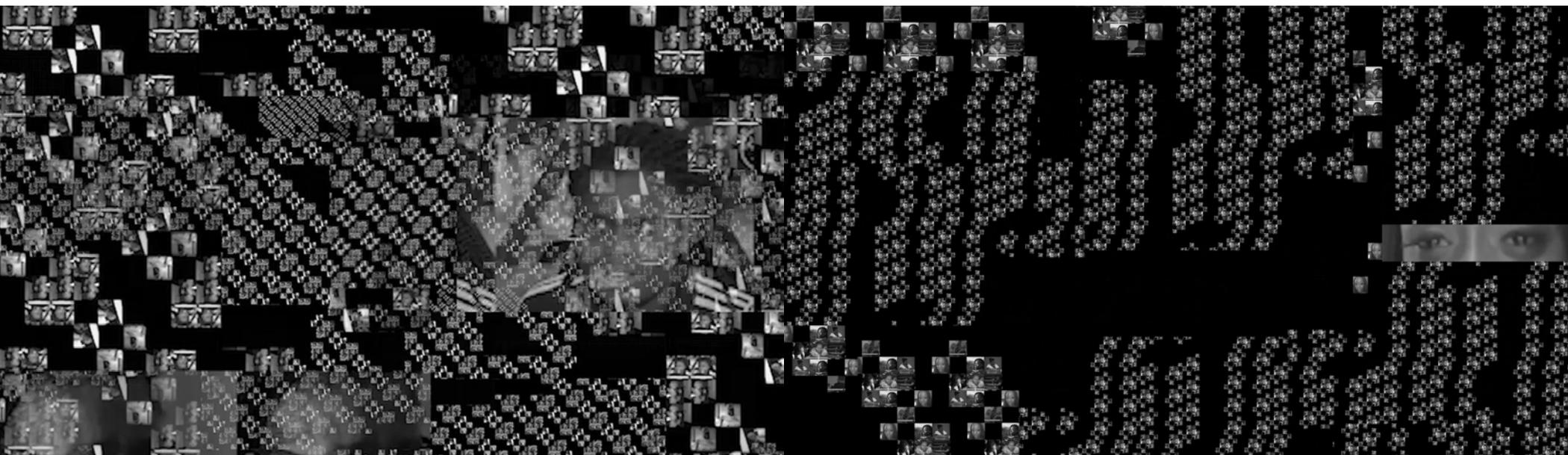
Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*



***Open Casket VII***

Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*

***Open Casket VIII***

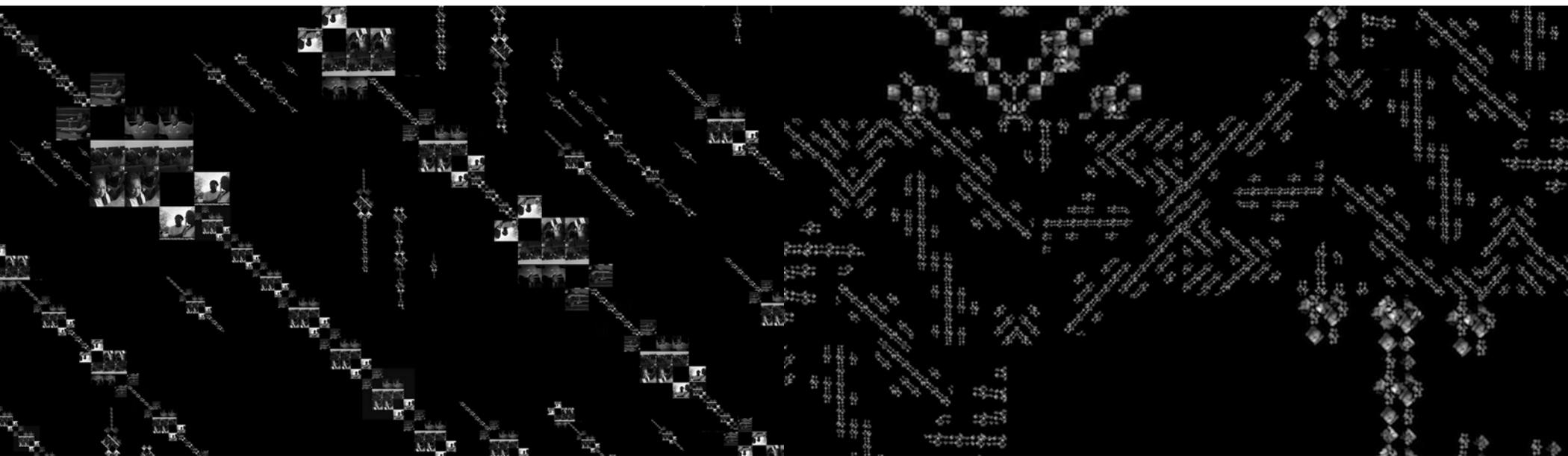
Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*



***Open Casket IX, Detail***

Weaving drawdown for crepe (casket upholstery), video,  
2500 pounds (1133 kg) hand-shoveled marble,  
dolomite, granite, concrete,  
three channel audio of families who have lost love ones to  
police violence.

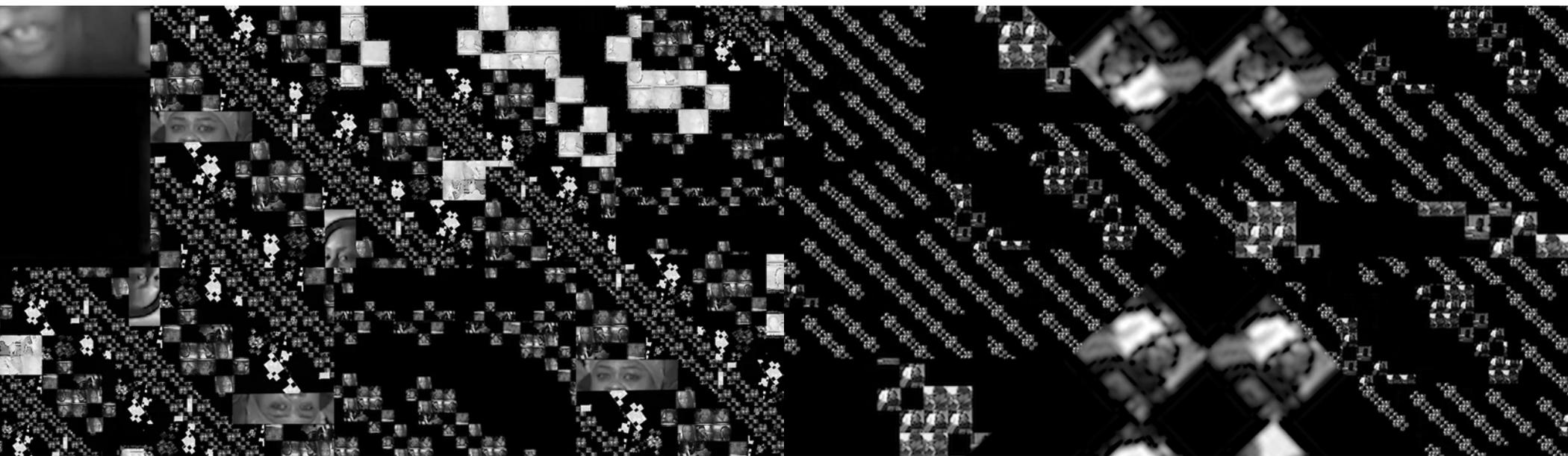
42 x 42 x 22 FT | 12.8 x 12.8 x 6.7 M  
2017

*Courtesy of the artist and Kadist Collection.*

***Open Casket X***

Weaving drawdown for crepe (casket upholstery), video  
00:30 on loop  
Dimensions vary  
2017

*Courtesy of the artist*



***Open Casket XI***

Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*

***Open Casket XII***

Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*



***Open Casket I, Detail***

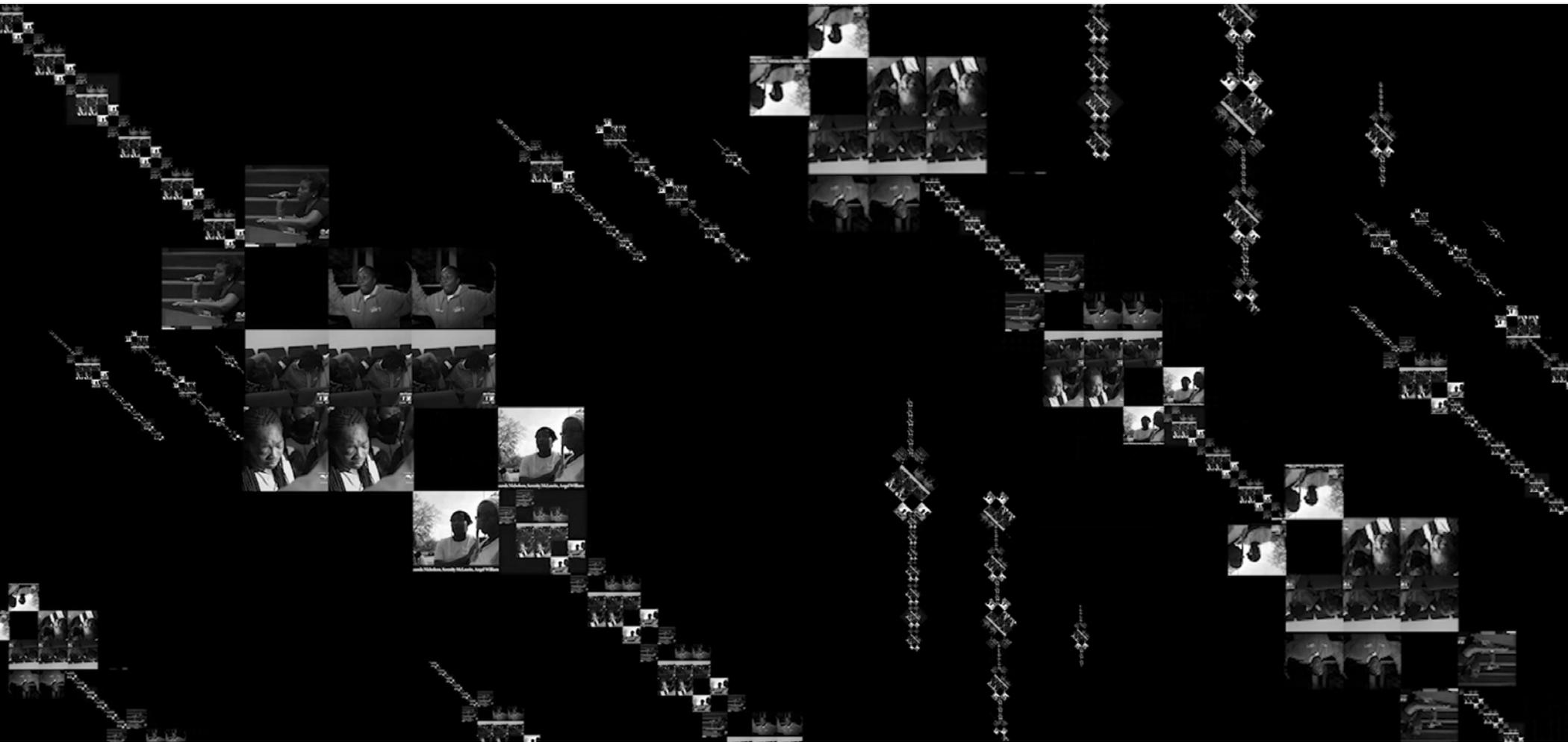
Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*



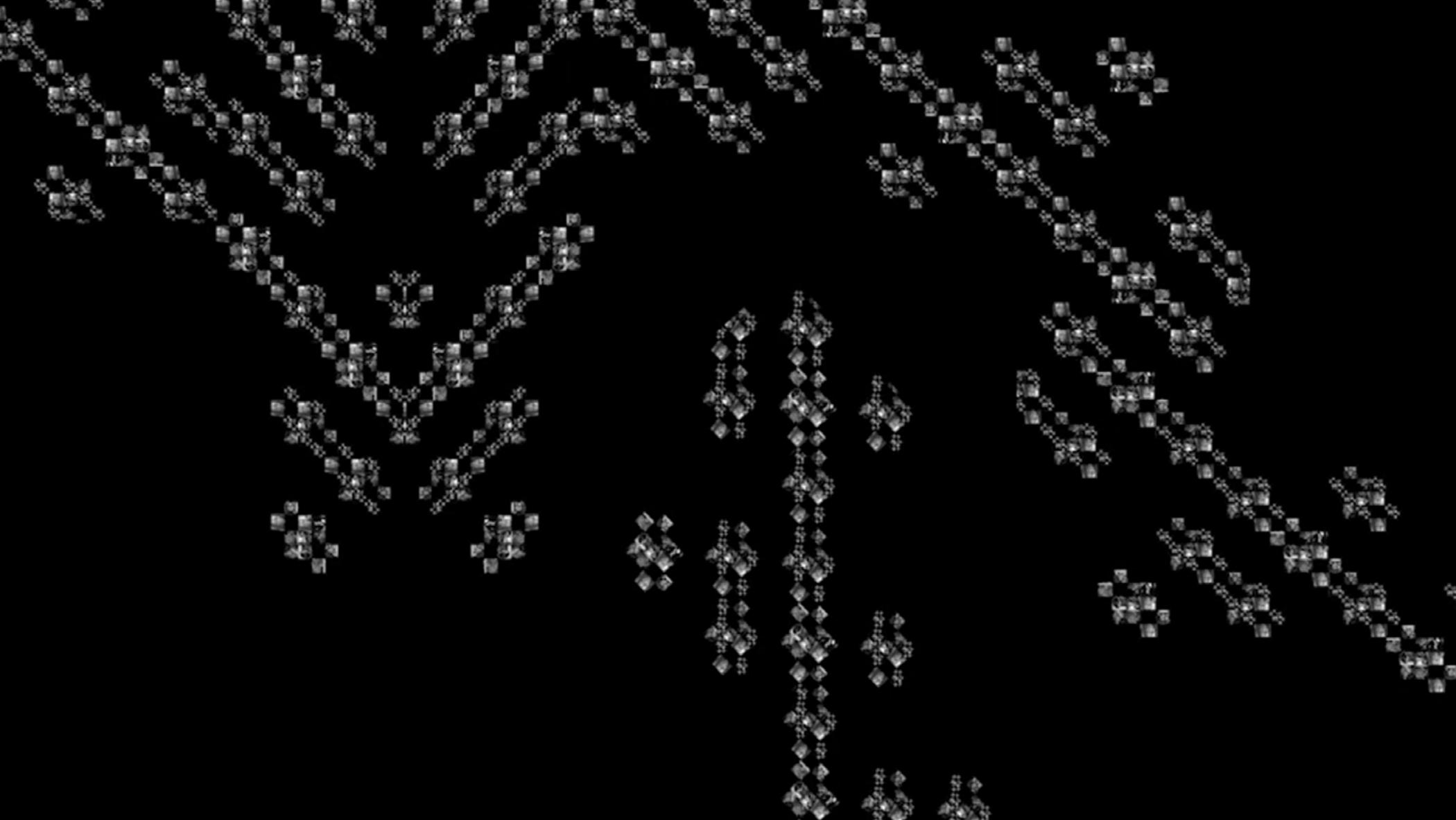
***Open Casket IX, Detail***

Weaving drawdown for crepe (casket upholstery), video,  
2500 pounds (1133 kg) hand-shoveled marble,  
dolomite, granite, concrete,  
three channel audio of families who have lost love ones to  
police violence.

*42 x 42 x 22 FT | 12.8 x 12.8 x 6.7 M*

*2017*

*Courtesy of the artist and Kadist Collection.*



***Open Casket X, Detail***

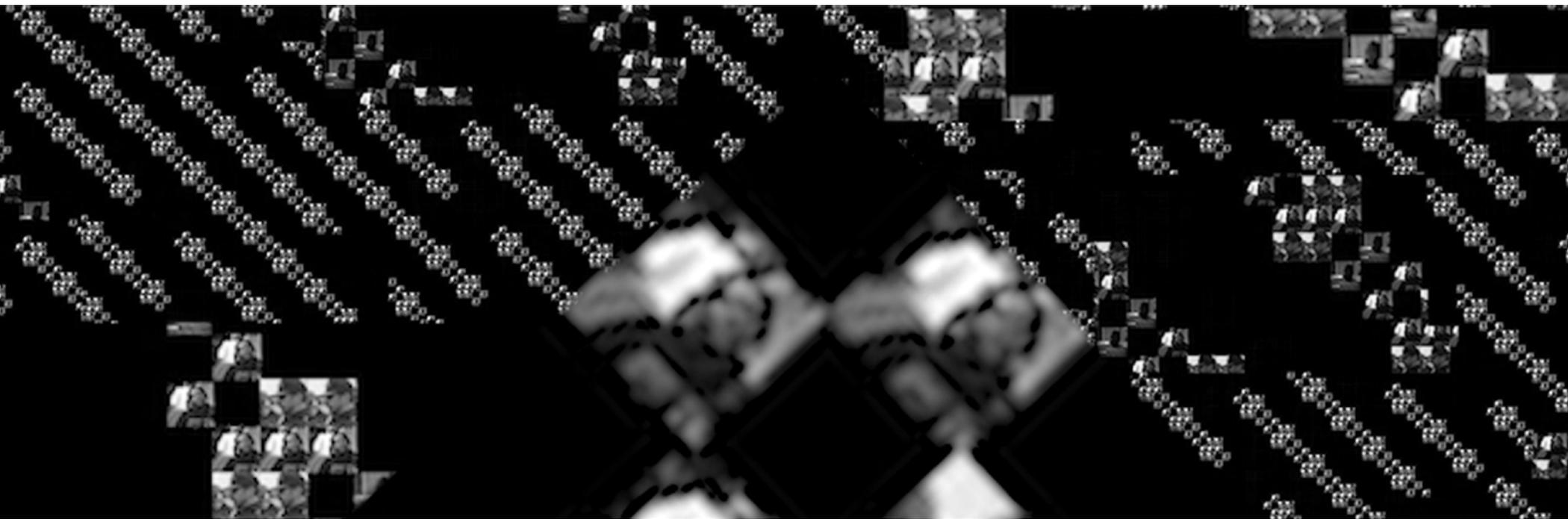
Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*



***Open Casket XII, Detail***

Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*



***Open Casket V, Detail***

Weaving drawdown for crepe (casket upholstery), video

00:30 on loop

*Dimensions vary*

2017

*Courtesy of the artist*

To learn more about the project, including an audio experience of the Mills Museum installation, please visit:

<https://www.indiraallegra.com/opencasket>

## About the artist:

Memorial as a genre is vital for its ability to hold the tension which grief creates inside crafted objects, spaces, and rituals. Indira Allegra is re-imagining what a memorial can feel like, the scale on which it can exist, and how it can function through the practices of performance, sculpture, and installation. The three practices are intertwined – with sculptures at times initiating performances, performances creating sculptures, and sculptures expanding into installation environments. Deeply informed by the ritual, relational, and performative aspects of weaving, Allegra explores the repetitive crossing of forces held under tension – be they material, social, or emotional.

Their work has been featured in exhibitions at the Museum of Arts and Design; the Arts Incubator in Chicago; John Michael Kohler Arts Center; Yerba Buena Center for the Arts; Center for Craft, Creativity, and Design; Mills College Art Museum; the Weinberg/Newton Gallery; the Museum of the African Diaspora; the Alice Gallery; and SOMArts among others. Their commissions include performances for SFMOMA, the de Young Museum, the Wattis Institute, City of Oakland, and SFJAZZ Poetry Festival. Allegra's work has been featured on BBC Radio 4 and KQED, and in *Art Journal* and *Surface Design Magazine*. They have been the recipient of the Artadia Award, the Tosa Studio Award, the Windgate-Lamar Craft Fellowship, and the Jackson Literary Award and have received support from the Mike Kelley Artist Project Grant, the MAP Fund, and the Queer Cultural Center. They are the 2019 Burke Prize winner, a Eureka Fellow, and a triennial 2019–2022 Montalvo Art Center Sally and Don Lucas Artist Fellow. Allegra teaches in the Graduate Art and Visual Culture Department at Mills College.

You can find them at [indiraallegra.com](http://indiraallegra.com) and [@indiraallegra](https://www.instagram.com/indiraallegra) on Instagram.



# Bodies into Bits

by Joy Lisi Rankin

Computerized medical education was supposed to help mend our broken healthcare system. Instead, it has reinforced some of medicine's most deeply ingrained biases.

---

One morning this June, I sat in my car at the drive-through of a local pharmacy, waiting for a COVID-19 test. Speaking through an intercom, a woman wearing a mask walked me through the process of administering the test on myself: where I could find the cotton swabs, how to open the vial of preserving saline, how to properly swab my nose, including the fifteen-second countdown for each nostril. She explained where there was an additional wipe for my protection when opening the biohazard bin in which I placed my sample, as well as how to wipe to protect others.

The woman administering all of this care was among the legions of specially trained frontline health workers needed to help manage the coronavirus crisis in the United States, under a dire lack of federal coordination and guidance. In the early stages of the pandemic, nurses around the country struggled to train

enough people to provide care, especially after officials closed schools and other usual educational avenues. Technology, however, seemed to provide a ready solution: the National Council of State Boards of Nursing and the American Association of Colleges of Nursing, among others, recommended using computer simulations and online learning platforms to upskill the nation's healthcare workers.

This turn to technological solutions for training caregivers in the face of an inadequate healthcare system is nothing new. At least since the early 1960s, when the country faced a shortage of trained nurses, computer-based education has been touted as an efficient and cost-effective way to patch holes in the nation's disastrous healthcare infrastructure. Then, as now, the rhetoric of urgency has been paired with the logic of cost savings to make online learning and computer simulations seem indispensable.

But computerized medical education has inevitably represented complex patients through grossly simplified models. Because you can't fit the diversity of human health experience into a software program, this education has always been oriented around notions of so-called "normal" or "typical" patients. In reality, these "typical" patients turn out to be composites of the sorts of people who hold power in society, particularly well-off white men. As a result, computerized medical education has helped to perpetuate the structural racism and sexism that has long pervaded the medical establishment, as well as our wider society.

Working under the promise that a computer could "dispense information just as effectively, sometimes moreso, than a human instructor," students in Illinois in the 1960s began the very first experiment in computerized medical education, learning nursing fundamentals on one of the world's earliest computer

networks. Looking back to those students and their computer-based courses demonstrates what is often overlooked, and even dangerous, with techno-care, and why that matters more than ever in our algorithmic age.

## Crushing Substernal Pain

In the early 1960s, Maryann Bitzer was pursuing her master's degree in educational psychology at the flagship campus of the University of Illinois. The university also employed her husband, Donald, who was using his engineering doctorate to investigate whether computers could be used effectively for education. Throughout the 1960s and into the 1970s, Donald led a team of researchers, including Maryann, in developing a computer network known as PLATO, Programmed Logic for Automatic Teaching Operations. PLATO comprised individual user terminals connected to a mainframe computer and, through the mainframe, to each other. The network went through several evolutions, and by the mid-1970s it included nearly 1,000 terminals around the United States, each with a flat-panel plasma touch screen, with applications including games, instant messaging, screen sharing, and email.

*“In reality, these ‘typical’ patients turn out to be composites of the sorts of people who hold power in society.”*

In a decision that ultimately benefited both of them, Maryann focused her master's thesis on how computer-based education

could work in nursing. She cited two motives for her study: a dearth of trained nursing instructors across the country, and the tremendous educational value for nursing students of working with “actual” patients. Using one of the early iterations of PLATO, which employed custom keysets and television-like cathode-ray tube screens, Maryann developed a course on treating heart attack patients. Then she delivered it to first-year nursing students at the university-associated Mercy Hospital. The course imaginatively integrated several components, immersing students in what sometimes seemed, behind the gloss of the new technology, like an actual experience of care.

First, the trainee nurse watched a short live-action film on the PLATO screen that depicted a doctor interacting with a patient, a middle-aged man. It was a clever way to present what Maryann described as “the patient’s socio-economic background, his present family situation, and his outlook on life [and] the patient’s past medical history and treatments.” Then, partway through the conversation with his doctor, the man grabbed his chest and was rushed to the hospital.

*“The simulation was limited by the level of complexity that could reasonably be programmed into the computer.”*

After this sudden turn, the terminal presented a series of screens that provided the trainee with fundamental information about different diagnostic tests and treatment courses. Then the nurse entered an extended simulation in which she could use the

PLATO keyboard to select from a limited menu of options to test out various interventions on the virtual patient and see the results. For example, after administering oxygen or nitroglycerin to her virtual patient, the nursing student could ask the computer to report her patient's pulse, temperature, blood pressure, electrocardiogram, or other test results. The nursing student could also consult a screen displaying clinical norms, such as the normal blood pressure range, to gauge the effects that her care was having on the patient.

It was a potentially engrossing but also highly circumscribed experience. The scenarios and results of the simulation were limited by the minimal level of complexity that could reasonably be programmed into the computer, especially given the fact that it took roughly forty hours of programming to create just twenty minutes of instruction. The film was designed so that each student could picture a particular living, breathing individual patient for whom they were caring as they worked through the simulation and its twenty-eight questions. But that single patient — a composite of several “typical” case studies — was the sole basis for how the computer was programmed to respond to the nurse's therapeutic interventions.

This focus on a so-called typical patient — a middle-aged man — both reflected and reinforced the prejudices of contemporary medical practice. A quarter century of research, beginning in the mid-1990s, has shown that heart attack symptoms manifest differently for women than men. For a long time, however, doctors and nurses and PLATO programs did not know how to recognize and diagnose heart attacks in women. For example, Maryann's course described its virtual patient as having “crushing” substernal pain. But women are much more likely to experience symptoms unrelated to chest pain. If they do experience chest pain, they more often describe it as discomfort or

pressure. (Despite our advances in knowledge, women are still 50–60 percent more likely to be misdiagnosed following a heart attack.) Maryann’s course was blind to this clinical reality, and as she went on to expand her PLATO-based nurse training, she helped to inculcate this bias, and others like it, in a new generation of nurses.

## Virtual Mrs. Dodd

---

Maryann’s experiment with techno-care occurred against the background of significant national investments in nursing. In 1960, the US Public Health Service created a new Division of Nursing tasked with improving patient care, increasing the number of nurses, and ensuring better nursing education. In 1963, the Surgeon General’s office published the report *Toward Quality in Nursing* which identified, among other problems, too few nursing educators, too few new nursing students, and an inadequate nursing education system. Maryann realized her experimental nursing course could be positioned as an efficient technological solution to these problems, training nurses faster and more cheaply than traditional nursing courses. In 1964, Congress enacted the far-reaching Nurse Training Act, designating the substantial sum of \$283 million (approximately \$2.3 billion in 2020 dollars) over five years to nursing education. The Nurse Training Act funded the expansion of Maryann’s PLATO project to develop a complete course on maternity nursing and a series of lessons on pharmacology.

The reliance on a single “typical” patient continued. The maternity nursing course focused on the virtual Mrs. Dodd, a secretary. Its twenty-two lessons “emphasized the normal, and presented problems which required knowledge of the normal as a basis for recognition of and action concerning the abnormal.” Students learned that “Mrs. Dodd suffers from many of the common

discomforts of pregnancy,” including nausea and swollen feet. And just as it was with the “typical” heart attack patient, the way “normal” Mrs. Dodd responded to therapeutic care was contingent on how PLATO had been programmed.

That programming was based on the standard of care for pregnancy in the 1960s, which was developed for, and applied to, white women—a bias that reinforced the invisibility of Black women to the medical establishment. (At many hospitals, including Mercy, the nurses, too, were overwhelmingly white; according to an archive at the University of Illinois, among the hospital’s hundreds of graduates until it closed in 1970, there were only ever six Black students.) For example, in the PLATO course, nurses monitored virtual Mrs. Dodd throughout all three trimesters of her pregnancy, as well as labor and delivery. But many Black women, then and now, lack sufficient access to and insurance coverage for complete prenatal and postnatal care; nurses exclusively trained to care for patients like Mrs. Dodd are poorly prepared to care for these women. Indeed, in the past few years, prominent Black women including writer and scholar Tressie McMillan Cottom and tennis superstar Serena Williams have called attention to how they and other Black women are dangerously mistreated during pregnancy, labor, and delivery. As Cottom recently wrote in *Time*: “In the wealthiest nation in the world, black women are dying in childbirth at rates comparable to those in poorer, colonized nations.”

Though severely limited, Maryann’s nursing course was nevertheless a success—in part because it reflected the limitations of the surrounding medical establishment. All of the students who completed the PLATO maternity nursing course later passed the Obstetric Nursing portion of the Illinois State Board examinations; the biases encoded in Mrs. Dodd were the same ones written into the exam. During the remainder of the 1960s,

hundreds of students at Mercy Hospital School of Nursing and nearby Parkland Community College completed PLATO nursing lessons, thus inscribing the biases into their own care.

## The Other Pandemic

---

In 1970, to commemorate its seventieth anniversary, the *American Journal of Nursing* invited prominent researchers to reflect on “Nursing in the Decade Ahead.” Under the title “Computers Have Entered Our Lives,” Maryann declared, “All indications are that computers are likely to play an integral part in assisting members of the health professions to provide comprehensive health care to people.” She has since been heralded as the influential figure who introduced simulations and computer-based courses to nursing education, where they are now widespread.

*“Virtual trainings are telling instances of the larger systems of racism and sexism that are shaping the country’s response to the pandemic.”*

But the types of sexism and racism subtly enacted on PLATO’s small screens through Maryann’s nursing courses also continues to pervade computerized medical education—even in the midst of a pandemic that is disproportionately killing Black people. In early April, the healthcare staffing provider IntelyCare reported that over 37,000 nursing professionals had completed its

COVID-19 online training, and another 310 healthcare facilities enrolled their staff in the course. When I took the course in early July, I discovered that every single patient and healthcare provider in it is white. Similarly, the online training assembled by the American Association of Critical-Care Nurses, which echoes Bitzer's heart attack course on many levels, has four units that each begin by immersing students in a critical healthcare situation with a "real" patient, all of whom appear to be white.

The exception that proves the rule is a webinar from the American Nurses Association titled "How *You* Can Have a Direct Impact on Reducing the Devastating Racial Disparities of COVID-19," which presents the case study of a forty-five-year-old Black man. Released in June amid protests over police brutality and the police murder of Black people, including George Floyd, it is the only online COVID-19 training I've seen that even begins to address American healthcare's deeply ingrained racism.

These virtual trainings are telling instances of the larger systems of racism and sexism that are shaping the country's response to the pandemic. Black people are dying from COVID-19 at a rate at least six times higher than white people. There is a multiplicity of overlapping and mutually amplifying reasons for this: the doubt with which many healthcare workers treat Black people when they report symptoms; the overall poor quality of care that Black people receive; the cumulative damage of environmental pollution that disproportionately impacts Black communities; the precarious place that many Black people occupy in our racist economy, especially in the poorly paid jobs that we now deem "essential"; the daily stress of enduring racialized discrimination and violence. As a sign carried by Dr. Jasmine Johnson, who studies maternal-fetal medicine, declared at a Black Lives Matter protest in Chapel Hill, North Carolina in June, "Racism is a pandemic, too."

Computerized medical education could be used to highlight and challenge this pandemic of racism, but only if institutions and society are willing to attend more closely to the specific needs of women and Black, brown, and Indigenous people; devote the resources necessary to creating courses that reflect the diversity of human experience; and overturn the centuries of economic exploitation that leave Black people at the bottom of America's caste system.

Ultimately, though, no computer program can sufficiently capture the extraordinary range of individual human medical experience, so we also need to invest in the sorts of hands-on medical education that doesn't seem efficient or cost-effective to policymakers and hospital administrators operating under the perverse logic of austerity. And as long as the medical establishment and medical knowledge remain hostile to the experiences of women, Black people, and other members of BIPOC communities, then no form of medical education can escape these sorts of biases.

Of course, this is true beyond medical education as well. The dramatic rise of online courses — not just in healthcare but across higher education — raises questions that were as relevant in the 1960s as they are today: What are we teaching? Who are we teaching? For whose benefit? For what larger public good? ~~~~

Joy Lisi Rankin leads the research program in Gender, Race, and Power in Artificial Intelligence at the AI Now Institute at New York University.



# Call on Me

## An Interview with an Anonymous On-Call Engineer

Software doesn't fix itself. Large and small companies that make products out of software have to have someone to call when the system breaks. At some companies, those engineers may all be on the same team together – maybe an infrastructure or DevOps team – while at other companies, the engineers who build the services are the ones who get paged when their services fail. But however you slice it, on-call engineers arrange their lives so that they can respond to pages within minutes, hopefully before the rest of us notice that their app is down.

We talked to one engineer about how they decide what counts as broken, what happens when they sleep through alarms, and how teammates take care of each other after a rough on-call shift.

---

### **How do you explain on-call to people who don't know what it is?**

I usually go broad and say something like: when something goes wrong with the product my company sells, someone has to fix it.

For stuff that I work on, my team has to fix it, no matter when it breaks. There has to be someone available 24/7 to respond. The availability piece is the thing that's hard to explain to people.

I remember there was a time when I was at the gym and my coach was asking everybody how we were doing and I was like, "I'm so tired because I was on call this week and I didn't get any sleep last night." And she was like, "Oh, are you a doctor?" Which is a very common response. And then I have to say no, I'm not a doctor. I'm on call for computers. And then people are usually pretty puzzled and ask if there's really something so important that it would warrant waking people up in the middle of the night. The answer is yes, but it's hard to explain why.

**For every hour of every day, there is a person assigned to respond if something goes down.**

When something in a system breaks, you need a 24/7 ability to respond. There's a lot more detail you could get into, like, how do you decide what's worth paging about? But that's the high-level summary.

**So how do you decide what's worth paging about?**

I'm an engineer and I work on a team that builds products that other engineering teams at my company use. The first question that we ask is, "How do we know when our products are working?" That's more complicated to figure out than it might seem.

We set up monitoring systems to examine different metrics. For an API, you might look at latency, which is how long it takes for a web request to be fulfilled. Or you might look at the error rate: in a perfect world, there would be zero errors when somebody tries to make a correct request to an API. But if there *are* errors, that could be because of a code change, or because other pieces of infrastructure aren't working.

It's not just about figuring out if something is broken, however. It's also about figuring out if it's broken *enough* to warrant human intervention. There's a general philosophy that human intervention should not be the first thing that happens. If an application tries to make an API request to an external API that has nothing to do with us or our infrastructure, and that API happens to be down—it could be GitHub, npm, or any number of services—our products should be able to retry the request. If it's the kind of thing where the request didn't work at first because GitHub was down, but the retry worked because GitHub is back up, that is something that our system should be able to just do on its own.

*“I have to say no, I'm not a doctor. I'm on call for computers.”*

But if the system can't fix itself, then we need somebody to intervene to assess how serious the problem is, and to see if there's anything we can do to mitigate the impact that it's having—the fact that this thing is broken and our customers can't use it—and then fix the thing itself.

### **What are the kinds of things you personally get paged for?**

Broadly, they fall into two categories. One, we made a change, it didn't go as planned, and it's breaking things; or, two, something external to our team is broken or unexpected, so our system doesn't work. Those are both tricky in different ways, but both of those potential failure situations inform how we build our systems and how we handle on-call.

## **What do you mean that it would affect how you build systems?**

We're on call for systems we've built, which is a very particular on-call philosophy. There are some places where these people over here create the thing, and those people over there are on call for when the thing breaks, and those are totally different teams. But we're on call for systems that we've built ourselves, so we have to expect that components of our systems will fail, and we have to integrate that anticipation of failure into what we promise our customers. And we have to think about how we architect and monitor for failure.

*“If a million people are affected and only one person is waking up to deal with it, that’s wrong.”*

## **Do you have a sense of how many people are impacted by an outage that you'd get paged for? Is there a way to measure impact?**

Absolutely. One of the first things we measure is customer impact, and that determines the severity of the incident we're dealing with. On one end of the spectrum, the least impactful end, our team will have a conversation about whether we should even be getting paged for something like this. Maybe the answer is no and we change how we're alerted, or we make the system more robust so it doesn't experience that failure anymore.

On the other end, I've responded to pages for downtime, which means that external customers cannot use our product. That's

typically measured as a percentage. So we'll say, "This outage impacted 5 percent of our customers globally" or, "10 percent of our customers in this particular region couldn't use our product for fifteen minutes."

The scariest failure I can think of that my team would be on call for is if our content delivery network (CDN) went down. That is the point of entry for customers who use our service and it handles billions of requests every day. So even if everything behind the CDN is working correctly, if there's an outage at the point of entry, that would impact a lot of people. Like, potentially all of our customers.

**I have this image of one person being woken up in the middle of the night because a million people can't access the app. It sounds like that's not how this works, though.**

If a million of our users are affected and only one person is waking up to deal with it, that's wrong. A company with a million users has hopefully put enough thought into how they do on-call that an outage of that size wouldn't happen that way. All of that said, if something like that did happen, the one person that gets paged would then page a bunch of other people once they realized that something was very broken. When there's an incident of that scale, whole teams are brought in to help and someone is the "incident commander" who coordinates the response.

Still, this doesn't happen that often. A lot of people think of outages as all or nothing. But it's not usually the case that a huge number of our customers can't use any of our services all at once. The more likely scenario is that one of our services goes down and it's part of another company's checkout system, so their customers can't pay. Or maybe their app doesn't load

properly on their customers' phones if one of our services is broken because of how the two are tied together with code. That company's customer has no idea that the problem lies with us. But they get impacted by our outage nonetheless.

**Even with those smaller incidents, however, a lot of money can presumably be at stake.**

Yes, and if it's our fault, the companies that rely on our services can come to us and say, "We signed a legal contract where you promised 99.999 percent availability," or whatever percentage we promised them. There's this concept of the number of nines of availability a service has. This indicates how available you expect a service to be, because it can never be 100 percent. For instance, a service that is 99.99 percent available has four nines, while a service that is 99.9999999 percent has nine nines. You build your expectations around how close or far from 100 percent availability a service is. The more available a service is, the more other companies rely on it in building their own products.

So if we breach our obligation around availability, a company might ask us for a refund or make a decision to not use our product anymore. We might do the same thing if another product causes us downtime. When there's an outage of something we rely on, we'll go to the company and say, "We want a root cause analysis, we want to know what the fix was, and we want a refund." They can't just respond and say, "I dunno, something broke but it's good now!"

It gets trickier when you're locked into a specific vendor. In some cases, we've decided to be locked in, in part, because they promise a lot of nines and we pay a ton of money, so that when their failure affects our failure, we get details as soon as possible, we get information under NDA to understand what happened, and we can ask for more help in how our relationship works.

## Waking Up is Hard To Do

---

**If you're on call outside of a workday, what does that mean for your personal life? How does it affect your evenings and weekends?**

Well, that's definitely when I notice on-call the most. I may try to go to sleep earlier on nights when I'm on call because I can't guarantee that I'm going to sleep through the night. I can't make spontaneous plans when I'm on call unless I carry my laptop around with me. So those weeks require a lot more planning.

**Because being on call means you literally have to open your laptop and debug as soon as you get paged. We've been talking conceptually about what it is and the philosophy behind it, but that's what it literally looks like.**

Right. Let's say I get paged when I don't have my laptop or I'm out without my charger. I would escalate immediately to make sure that somebody else responds. But typically, we respond within a few minutes. We respond as if everything is urgent. If I get woken up in the middle of the night, my sleepy brain is like, "You can look at it later," but I've trained myself to not look later, to look *now* because it could be really bad. But yeah, it can really impact my life, my ability to do errands, my sleep. Bad weeks are bad.

**What do you do when you get paged?**

Almost everyone I know who's on call, regardless of the company, uses the same app to configure pager alerts. And you can configure the app to send different kinds of information each time it pages someone. We have a link to whatever metric is passing the threshold that's causing the page. Those metrics are also things that we configure.

## **Throughout your code?**

Right. All the services that make up what looks to customers like one cohesive product are owned by different teams and the teams set up the thresholds they want, and people get paged based on those thresholds. A threshold might be: “If this function fails five times in an hour, page someone.”

## **There must be dozens of those? Thousands?**

A lot. That’s why it’s important to be thoughtful about what you want to be alerted to and what the threshold for an alert is.

So coming back to your question about what I do: I first look at the metric that has passed the threshold. Then, I look for documentation about that alarm. When we create an alarm, we try to write documentation on what it is and what it’s measuring. If the docs are good, they also include context about why this piece of code or infrastructure exists and its various potential failure states.

But you don’t want to get into too much detail on each failure state. Sometimes, when people deal with failure, their instinct is to say, “If I document every single piece of information about this situation, I will know exactly how to respond when this happens again.” But if you have something that fails regularly, for a very predictable reason, you should fix the problem in the product and stop paging everyone all the time. Although sometimes that’s easier said than done. It is pretty easy to document all the ways in which something can break—it is usually much harder to build something that breaks less often.

Sometimes, I get paged for something I’ve never worked on before. That’s when I really lean on this process. I see a metric. I see some docs. Something is broken that’s potentially impacting

people. How do I use these pieces to get to an understanding of what happened and how to fix it?

**Does everybody really wake up and deal with a page they get at two in the morning? Surely, people sleep through alarms. What happens then?**

Yes, everybody really wakes up. There may be some rare case where, you know, someone got a new phone and didn't set up their notifications correctly. Or, people accidentally sleep through middle-of-the-night alarms. I certainly have. But if someone were to repeatedly refuse to respond at night, they just wouldn't last in that job. What happens when someone misses a page is that the next person gets paged. The app is configured to page a certain number of times in a certain number of ways—it'll text you, then call you, then email you—but if you don't acknowledge the page, it tries the next person.

**In the same way that computers are automated to fail over to the next system, the app will fail over to the next human if one of them is down.**

Yeah. If I sleep through an alarm, our escalation is set up to try my team members first. Then my boss, then my boss's boss, all the way up to the executives. If all of us sleep through all the alarms, the CEO would get paged. I've never seen that happen before, though.

What has happened is that I've been paged for something, didn't know how to deal with it, and then paged someone else to wake them up to help.

**How does it feel to do that?**

I mean, I wish I never had to do that. It sucks, because I know how garbage I feel after I've been woken up at that time. But this

is actually a place where team culture is important. If someone else wakes me up, I try to respond without resentment and without making somebody feel bad for needing help. We don't page each other frivolously, but if someone doesn't know what to do and I'm second in line, it is my job to respond and help that person out. It can create a really toxic culture if you're like, "Ugh, why did you wake me up for this?" And if somebody stops asking for help, that is a big potential failure scenario. That's why, when we onboard someone, we really play up the "It's totally super fine! Don't worry about it, page me anytime!" They won't actually page me anytime, but it's important for them to know that they can if they're in trouble.

**The company probably benefits from people being kind and showing up for each other in cases like that.**

Definitely. I mean, on-call can go lots of ways. What I'm describing, even if I don't love on-call, is being on a team with people I trust, knowing that I won't get yelled at or fired for unintentionally doing something that causes damage, and knowing that there's a genuine spirit of reflection around how to fail better. The thing that motivates me during on-call, much more than fixing the tech, is my teammates. There are things that are beyond our control: there's a lot of failure on the internet and we don't pick the days when a critical service goes down. If everyone is always exhausted and grumpy when they show up, that sucks for them and it sucks for me. So, almost always, if someone gets woken up in the middle of the night, another person on the team will offer to take over their shift the next night so they can get a full night's sleep. Because waking up one night sucks, but waking up two nights in a row? You're toast.

We also encourage one another to ask for help and to offer help. If we've identified something that is really disruptive to

each other's day-to-day lives, we take that seriously and make changes so that that thing doesn't happen anymore. That matters when you think about the fact that we are on call for holidays and weekends. There's a lot of motivation for us to make on-call not terrible. So we are caring for infrastructure, but ultimately we're taking care of each other. ~~~~



# The Tower and the Tent

by Nitin K. Ahuja

Contemporary megahospitals present themselves as cutting-edge technologies. A physician reflects on what happens when they fail.

---

At the university-based outpatient practice where I work as a gastroenterologist, my office overlooks the construction site of a huge new hospital that is slated to open in the summer of 2021. Since taking this job three years ago, I've watched the project develop from its steel skeleton to its curving facade, a layer cake of plate glass and copper panels. The aesthetic matches the streamlined, futuristic style of academic megahospitals from Texas to Tokyo, casting the building as a technology that's every bit as cutting-edge as the microsurgical robots and cancer immunotherapies it's designed to house. At seventeen stories high and a price tag of \$1.5 billion, it's a tower among towers (including the one where my office sits on the seventh floor), adding to an already imposing skyline of healthcare buildings near the center of Philadelphia, just west of the Schuylkill River.

The new hospital is also visible from two vinyl tents, each slightly smaller than a double-wide trailer, that were recently

set up in front of my current hospital's emergency room in the early phase of the COVID-19 pandemic. With a few dozen fellow doctors, nurses, and medical assistants, I was redeployed there for several weeks in April to test and triage patients walking in with flu-like symptoms — coughing, short of breath, but not quite sick enough to merit a hospital bed. As far as tents go, ours were pretty nice: we had Wi-Fi, some bleach-proof laptops, and sinks connected to hot water hoses that were slung like vines across the sidewalk. We also had carts that were implausibly well stocked with gloves, goggles, masks, and nasopharyngeal swabs. I was grateful for those carts, having heard terrible stories of healthcare workers dying for lack of basic supplies at the loftiest medical institutions of nearby New York City. So long as I was wrapped in the appropriate protective equipment, I was content to sit in a folding chair with rainwater pooling at the edges of our makeshift clinic.

But I was also struck by the juxtaposition between the looming megahospital, glinting in the sun like a freshly landed spaceship, and the humble tents below. It evoked an abiding paradox of modern medical care. Contemporary hospitals are like great machines: each discipline exists in its own silo, outfitted with its own technologies and specialized knowledge, often connected to other disciplines only by confusing networks of elevators, corridors, primary care physicians, and a vast, computerized bureaucracy. The proton-beam radiation equipment used to treat prostate cancer patients in the basement is far removed from the specially-trained neurology nurses working with brain-damaged patients on the floors above, but they meet in the electronic medical record, with its templated notes and billing codes. Critics have long pointed out that the enormity and complexity of this infrastructure, which is designed to literally and metaphorically convey biomedicine's healing power, frequently leaves patients feeling lost, even spiritually deadened.

The current pandemic adds another wrinkle to that longstanding contradiction. In the face of a widespread threat to our well-being, the megahospital hasn't risen to the occasion. Past social upheavals, such as postwar modernism and the rise of mass production, helped to fuel major evolutions in hospital design. Looking out from my office window at the massive structure taking shape by the river, I wondered whether, after COVID-19 had laid bare the limitations of contemporary megahospitals, these buildings might be refashioned once again.

*“Each discipline exists in its own silo, often connected to other disciplines only by confusing networks of elevators, corridors, primary care physicians and a vast, computerized bureaucracy.”*

### “It’s All in There”

More than a decade ago, as a medical student in Michigan, I began researching the origins of modern hospital design. As a soon-to-be doctor, I was interested in how my university hospital unsettled me. Months and even years into my studies there, I kept coming across new wards, offices, and laboratories. It often seemed to me at the time that miracles of healing and scientific discovery—organs transplanted, inflammatory pathways defined—must be happening around every bend. Some days, though, the hospital’s beige walls depressed me, the monotonous stretch of high-ceilinged corridors gave me vertigo, and the

ambient smell of disinfectant turned my stomach. I wanted to understand how much of this disorientation came from navigating an unfamiliar space and how much was intrinsic to the building itself.

*“The same principles underlying the proper functioning of a manufacturing plant apply to the planning of a hospital building.”*

That megahospital, like the one in Philadelphia where I’ll soon work, had its formal origins at the turn of the twentieth century. Before that time, American hospitals were more modest buildings, staffed by familiar faces from the local community, decorated like middle-class homes, with sash windows and sloping roofs. But the growing acceptance of germ theory, in which common diseases were ascribed to the spread of pathogens, correlated with a shift in hospital design toward blank, unornamented surfaces made of visibly aseptic materials like linoleum and metal.

At roughly the same time, industrial capitalism’s obsession with efficiency was exported from the factory floor to other domains, including medicine. Interest in scientific management led hospital designers to attempt to streamline patient movement in the way a factory was designed to streamline production processes. This move was reinforced by the rise of artistic modernism, which championed the now famous adage that form should follow function.

The predecessor of the Michigan hospital where I started my training was one of the prime examples of medicine's embrace of the ideals of industrial efficiency. It opened in Ann Arbor in 1925 and was designed by Albert Kahn, an architect better known for planning several of Henry Ford's automobile factories in nearby Detroit. Kahn drew clear analogies between these two building types, once telling the American Hospital Association that "the same principles underlying the proper functioning of a manufacturing plant apply to the planning of a hospital building."

Hospitals became emblems of progress and were celebrated for their rational forms. "The main purpose of the building is to function as a medical instrument," said the Finnish architect Alvar Aalto with reference to his Paimio Sanatorium, a 1933 exemplar of modernist hospital design. But the flipside of the hospital's increasing ability to anatomize patients was its capacity to dehumanize them. At the extreme, an analogy between hospitals and factories suggested an equivalence between sick people and disassembled parts.

Both the scientific power and the dehumanizing effects of the hospital have been amplified over the last several decades, as these buildings have increasingly been designed to accommodate new computing technologies. From wall-mounted monitors in every patient room to networked smartphones in the pockets of every white coat, digital interfaces have provided new avenues for clinicians to communicate with patients and each other, accelerating the pace of diagnosis and therapy. But these interfaces also risk further isolating patients, drawing clinicians' eyes toward screens and away from the person in the bed. More than once, I've asked a patient to relate the story of their illness, only to have them point to the nearest computer and tell me wearily, "It's all in there."

## An Emerging Labyrinth

---

I moved to a new state for each phase of my clinical training, spending a few years on one mammoth medical campus before graduating to the next. Though each had its spatial quirks, these tended to follow similar themes. Pavilions of various ages were connected by secret tunnels once used by hospital porters to transport equipment, now used by savvy employees to avoid walking outside in bad weather. Certain unmarked elevators skipped certain unmarked floors unless presented with a privileged ID badge. Wards were decorated with a hodgepodge of faded prints in dusty frames—geometric abstractions, a still life of flowers—apparently much easier to put up than to take down.

Irrationality emerges alongside obsolescence in hospitals, patchily, with various temporizing efforts made along the way to restore order and relevance. Inpatient units are retrofitted to new infection control guidelines requiring sinks in the hallway; radiology suites are widened to accommodate additional MRI machines. The hospital's original floor plan is carved up, each territory stewarded by a different lineage of managers, each poised for its own idiosyncratic process of redesign: the secret tunnels' access doors stay unlocked but have their signs removed; the enlarged radiology suite displaces the radiologists' workroom to another wing, estranging them still further from the patients whose scans they read. It's maturation of a sort, but also a kind of decay; slowly, monuments to rationality become labyrinths.

New hospitals are built when the obsolescence of old hospitals becomes difficult to ignore, but new hospitals can also become old hospitals rather quickly. In *Medical Nemesis*, his 1974 polemic against biomedicine, the social critic Ivan Illich describes hospitals as “concrete manifestations of those

professional prejudices which were fashionable on the day their cornerstone was laid and which were often outdated when they came into use.” Those prejudices can be as nominal as the flooring selected for intensive care units (which, at the hospital where I completed my residency, were inexplicably carpeted) or as consequential as the total number of intensive care beds available in a given city for patients in simultaneous respiratory failure.

*“Much of the new hospital will lie dormant so long as our clinical attention is preoccupied with one particular virus and the comparatively basic technologies needed to manage it.”*

All of which is to say that there’s a limit to how nimble hospitals can be in their readiness for the future, despite what their exteriors seem to promise—a point that our collective experience with COVID-19 makes plain. According to its promotional video, my university’s new building was “designed to be ready for technologies decades down the road.” This breathless rhetoric corresponds to fairly basic practices, less anticipatory than agnostic (involving, for example, multipurpose rooms designed with a lot of electrical outlets). And however flexible it may purport to be, much of the new hospital will lie dormant so long as our clinical attention is preoccupied with one particular virus and the comparatively basic technologies—masks, swabs, ventilators—needed to manage it.

## Masks in Brown Paper Bags

---

Once the pandemic recedes, where will we cast blame and seek reform? Politicians, wet markets, global supply chains — but probably not the megahospital. Most messaging from the medical-industrial complex, from drug advertisements to corporate hospital slogans, reinforces a belief in scientific progress, the reversibility of physical distress, and the usefulness of technology in both regards. These pre-pandemic ideals of biomedicine seem likely to persist in the post-COVID era, as does our tendency to celebrate them through the usual tropes of megahospital design. But these buildings' failure to prevent a great deal of death has tinged their formidable architecture with irony, and I wonder whether there are subtler ways in which our movement through these spaces might change.

Just before my stint in the emergency room tents, I finished a week-long rotation as a consultant gastroenterologist for patients admitted to the old hospital with various perennial complaints — trouble eating, painless jaundice, bloody stools. In order to keep protective equipment available for the clinical areas where they were needed most, including the tents and the ICU, respiratory masks were being carefully rationed from a central location — a control desk that oversees the building's forty or so operating rooms. Rather than discarding our masks at the end of the day, we were asked to return them in brown paper bags, writing our names on the front, like schoolchildren, so that they could be disinfected for two hours with ultraviolet light and later reused.

It was a long walk to the control desk from my office — past plaques bearing the names of erstwhile benefactors, past stairwells going up twelve flights, past rows of sleeping computers, past empty recovery bays, past unplugged fluoroscopic

imaging machines stacked against the walls. Because my practice is focused on outpatients, my hospital-based rotations are rare, and on each one I have to make an effort to remember my way around. But it's pleasant to get reacquainted with the vastness of the enterprise; the complexity of these spaces can beguile clinicians too. There's plenty to marvel at along the way to preserving a two-dollar bit of air-filtering fabric — such a ridiculously simple thing, in the grand scheme, to be holding onto so tightly. ~~~

Nitin K. Ahuja is an Assistant Professor of Clinical Medicine in the Division of Gastroenterology and Hepatology at the University of Pennsylvania.



EEEEEE LL  
EE LL  
EEEE LL  
EE LL  
EEEE LLLLL IIII

Rogerian psychotherap  
ogram was described by  
tion by Norbert Land

troubling you

YOU:  
ELIZA: Ca  
YOU: We  
ELIZA: Is  
YOU: He  
ELIZA:



# When You Say One Thing but Mean Your Motherboard

by Matthew Seiji Burns

Can computerized psychotherapy care for our mental health?

---

In early 2014, I attended the demonstration of an experimental technology funded by DARPA, the US military's advanced research and development agency. In a conference room at a Los Angeles office park, a young man sat on stage before a computer screen, with cameras trained on his face and body. On the screen, a rudimentary 3D model of a casually dressed young woman sat upright in a plush armchair of the type one might find in a psychotherapist's office. The character said her name was Ellie and that she was not a therapist, but created "to talk to people in a safe and secure environment." Another monitor, visible to the audience but not to the man, displayed real-time information from the cameras, tracking his posture, eye movement, and the changing shape of his mouth.

"I'll ask a few questions to get us started," Ellie said, regarding the man with an unnervingly neutral gaze. "And please feel free to tell me anything. Your answers are totally confidential."

As the demonstration proceeded, Ellie asked if he could recall the last time he felt truly happy. He didn't have a ready answer, and the system's computer vision software detected that he was breaking eye contact more than he had earlier in the conversation. "I noticed you were hesitant on that one," Ellie said.

Ellie's goal was to conduct conversations that would allow the software to pick up "distress indicators," from evasive glances to "negatively valenced words," that are "correlated with depression, anxiety, or post-traumatic stress disorder," researchers from the Institute for Creative Technologies at the University of Southern California, where the system was designed, wrote in a subsequent paper. To that end, Ellie was made to seem highly empathetic. "A particular concern in the design... was whether users would feel comfortable enough in the interaction to speak openly about their mental health issues to a virtual agent," the researchers wrote.

Though Ellie was obviously virtual, there were potential advantages to this. DARPA was funding ICT's research through a larger program called DCAPS, the Detection and Computational Analysis of Psychological Signals, which aimed to "develop novel analytical tools to assess psychological status of warfighters in the hopes of improving psychological health awareness and enabling them to seek timely help," the agency said. After the demo finished, one of the researchers explained to the room that speaking to a virtual avatar about mental health issues could be ideal for military personnel who are taught to be and act tough, leaving them too ashamed to open up to another human being.

Something like Ellie could be useful to the military in other ways, too. To identify and help all current and former personnel with PTSD would be a massive undertaking. Estimates from the US Department of Veterans Affairs suggest that between

11 and 20 percent of the 2.7 million service members who deployed to Iraq and Afghanistan — roughly 300,000 to 540,000 people — suffer from the disorder in any given year. Of those, DARPA says that only a small fraction seek help. It's difficult to imagine recent administrations deploying the battalions of people — therapists, trainers, outreach personnel — needed to find and care for half a million or more people with PTSD. Automation, of the kind represented by Ellie, seems to hold out the possibility of treating mental health problems at scale, or even keeping soldiers on active duty for longer periods. If successful, computerized therapy could also be applied in other circumstances where human-to-human treatment is undesirable or impractical — such as in the midst of a pandemic.

Behind this possibility lurks a larger vision, too. Though the Ellie program is in some ways crude, it seems to herald a future system that can continuously track, report, and support users' mental health on an ongoing basis. At the time of the demo, consumer devices like the Fitbit and Apple Watch were being marketed on the basis of their round-the-clock monitoring and data-collection features for physical health — information which would yield life-improving insights and interventions, the companies behind these technologies implied. More recently, researchers affiliated with Amazon published a paper describing efforts to determine a user's emotional state from their voice. If an Amazon Alexa divined you were upset, it could ask what was wrong — and maybe upsell you on some indulgent self-care items. Supporting mental health could be one more reason to justify the ambient collection and interpretation of vast streams of data from our bodies and behavior.

After seeing the DARPA demo, I was unsettled by the idea of an emotionally-aware technology ecosystem constantly reporting back to companies or governments about our mental states, and

then trying to intervene in them. But the thing I kept coming back to most often was the avatar of Ellie, sitting in her chair with her hands folded in her lap, calmly interviewing an actual human being with a potential mental illness. As a designer and writer of video games, I know that well-crafted interactive digital characters can elicit deep emotions from players, causing changes in their mood and outlook, just as powerful works in any medium can. Until I encountered Ellie, though, I hadn't imagined what it would mean for people to share their most private thoughts and feelings with a machine. I wondered whether this artificial interaction could actually help people change, or even heal. So, in a spirit of curiosity, I set out to create a sort of Ellie of my own.

*“It’s akin to running an algorithm on a thought until its power diminishes and eventually disappears.”*

## An Algorithm for Thoughts

When I began researching computerized therapy, virtual mental health care was already a booming category—and that was before the world was struck by the coronavirus. Following the outbreak of COVID-19, the possibility of inexpensive, scalable virtual mental health tools may very well become a necessity. Social isolation, unemployment, pervasive uncertainty, death—the pandemic and society’s response to it have created a wave of emotional distress while at the same time stripping millions of people of their jobs, healthcare, and access to therapy. “With the coronavirus pandemic causing unprecedented levels of

stress and grief, companies offering virtual mental health care say they're seeing a massive surge in interest," the medical news site *STAT* recently reported.

There's currently a bewildering array of mental health apps to download, with names such as Calm, Happify, and MindShift. Some of these programs are simply collections of user-directed exercises ("write down how you felt today"). Some seek to connect clients with human therapists over video or text chat. Others contain characters like Ellie. But most of them claim to implement elements of cognitive behavioral therapy, or CBT, a kind of therapy that emerged in the 1960s to become one of today's most heavily used and studied methods of treating depression, anxiety, and PTSD. For many years, CBT has been considered the most effective treatment for such disorders. One recent meta-analysis of mental health app trials found that over 80 percent of them used CBT in one form or another.

CBT is a heavily "manualized" form of therapy, meaning that it expects providers to stick closely to highly specific, almost scripted interactions with clients. Providers teach clients to recognize negative thoughts and "dispute" them in order to reduce their effect and replace them with positive ones. If the recurrent thought "I'm destined for failure" runs through a patient's head, for example, CBT prompts them to systematically dispute the thought with alternative interpretations: *Is this, perhaps, an exaggeration? Is there really any evidence that this is true?* It's akin to running an algorithm on a thought until its power diminishes and eventually disappears.

As a result, a common complaint from those on the receiving end of CBT is that it is cold, mechanical, and lacks empathy. Writing for *Vice* about the game I eventually made about a virtual CBT therapist, the critic Rob Zacny said he recognized

in it “that sense of emotional whiplash that comes from finally starting to confide something serious and scary, only to be met with weirdly programmatic responses from well-meaning counselors.” But the mechanistic quality of CBT also makes it especially well-suited to be supplemented by, or turned entirely into, a computer program. If your human therapist already acts and sounds like an automaton, what would be lost by replacing him with one?

*“ELIZA’s noncommittal responses were surprisingly good at convincing users that there was another human being on the other side of the teletype.”*

But virtual characters can only be effective if on some level we believe or buy into them. In a 2016 article for the *Guardian* about new questions surrounding CBT’s efficacy, the journalist Oliver Burkeman spoke with a woman who had experienced postnatal depression. “I don’t think anything has ever made me feel as lonely and isolated as having a computer program ask me how I felt on a scale of one to five, and — after I’d clicked the sad emoticon on the screen — telling me it was ‘sorry to hear that’ in a prerecorded voice,” she told him. What she realized she needed, Burkeman writes, was “real connection: that fundamental if hard-to-express sense of being held in the mind of another person, even if only for a short period each week.”

Patients aren’t the only ones who have recognized this need. In 1976, a computer scientist at MIT named Joseph Weizenbaum

declared the notion that “a computer can be programmed to become an effective psychotherapist” was nothing but “perverse.” This was a renegade position for the MIT of the era, where a powerful belief in the capabilities of artificial intelligence reigned. Weizenbaum was well aware that he was committing apostasy, but his assault on the idea of computerized therapy was all the more forceful because he had created the first virtual therapist himself.

## Alien Worlds

---

The ELIZA program that Weizenbaum created, which he released in 1966, is often called the world’s first chatbot. Cleverly, Weizenbaum designed ELIZA around the stereotype of a psychotherapist’s propensity to turn a client’s statement into a question. (People don’t like me. *Why do you think people don’t like you?*) This was an elegant way to create the effect of a computer holding its own in a conversation with the user.

Of course, ELIZA—which Weizenbaum named after Eliza Doolittle, the character in the George Bernard Shaw play *Pygmalion*—possessed no real understanding of what its human partner was saying. As the information technologist Ted Nelson wrote of the program in his classic 1974 book about hacker culture, *Computer Lib / Dream Machines*: “Now, do not be fooled. There is no Ghost in the Machine. The program does not ‘understand’ the user. THE PROGRAM IS MERELY ACTING OUT THE BAG OF TRICKS THAT JOE WEIZENBAUM THOUGHT UP.” The explanation (and emphasis) was necessary because ELIZA’s non-committal responses were surprisingly good at convincing users either that it did understand them, or that there was actually another human being on the other side of the teletype.

Indeed, ELIZA sometimes seemed to perform flawlessly, as in this example from Weizenbaum's original 1966 paper describing his creation:

*Men are all alike.*

*IN WHAT WAY*

*They're always bugging us about something or other.*

*CAN YOU THINK OF A SPECIFIC EXAMPLE*

*Well, my boyfriend made me come here.*

*YOUR BOYFRIEND MADE YOU COME HERE*

*He says I'm depressed much of the time.*

*I AM SORRY TO HEAR YOU ARE DEPRESSED*

*It's true. I am unhappy.*

*DO YOU THINK COMING HERE WILL HELP YOU NOT  
TO BE UNHAPPY*

*I need some help, that much seems certain.*

But it is also easy to have a conversation with ELIZA that goes nowhere:

*HOW DO YOU DO. PLEASE TELL ME YOUR PROBLEM*

*I have so much work to do.*

*THAT'S QUITE INTERESTING*

*Not really.*

*THAT IS INTERESTING. PLEASE CONTINUE*

*It's not.*

*DO YOU FEEL STRONGLY ABOUT DISCUSSING SUCH  
THINGS*

It usually becomes clear to modern users within minutes how fragile ELIZA's illusion of real conversation is. In the late 1960s, however, the program left a lasting impression. The astrophysicist and science popularizer Carl Sagan, in a 1975 essay for *Natural History* magazine, predicted a time in which such therapy would be commonplace, delivered through "a network of computer psychotherapeutic terminals, something like arrays of large telephone booths." Even established mental health professionals began to seriously consider the idea of computerized therapy. Perhaps most notable was the psychiatrist Kenneth Colby, who later developed his own mental health chatbots and once told a reporter, "after all, the computer doesn't burn out, look down on you, or try to have sex with you."

Weizenbaum himself believed that ELIZA only demonstrated that computers did not have to actually understand anything in order to mimic everyday conversations. The year after ELIZA was released, his more famous colleague at MIT, Marvin Minsky, declared that "within a generation, the problem of creating 'artificial intelligence' will substantially be solved." But ELIZA helped Weizenbaum to develop a more skeptical view of computer science, and of the relationship between computer and human intelligence.

Weizenbaum argued that even if a future computer were powerful enough to perform automated therapy, it would still be wrong. Human intelligence and computer logic are fundamentally different processes and wholly "alien" to one another, he wrote in his 1976 book *Computer Power and Human Reason*. As Zachary Loeb explains in the introduction to *Islands in the Cyberstream*, a posthumously published interview with Weizenbaum, "Computers excelled at tasks involving quantification, but for Weizenbaum there was much about human beings that simply could not be quantified." As tempting as it was for

computer scientists to believe that computers could model the world around them or human thought, in truth they could only create their own separate reality. “If Weizenbaum called for renunciation of the computer, in certain contexts,” Loeb continues later, “it was because the embrace of the computer in all contexts had led to a renunciation of the human.”

In her paper “Authenticity in the age of digital companions,” Sherry Turkle, a fellow MIT professor who taught classes with Weizenbaum, recounted how ELIZA’s reception informed his stance:

*Weizenbaum found it disturbing that the program was being treated as more than a parlor game. If the software elicited trust, it was only by tricking those who used it. From this viewpoint, if Eliza was a benchmark, it was because the software marked a crisis in authenticity: people did not care if their life narratives were really understood. The act of telling them created enough meaning on its own.*

Perhaps this is the most salient question when it comes to automated therapy: can the act of telling be separated from the act of listening? Does simply speaking create enough meaning on its own? If being listened to, recognized and understood by another human being is the source of therapy’s power, then there is no way bundles of code—such as ELIZA, Ellie, or the fictional app in the game I eventually created—could work. Automated mental health care would be built on a faulty premise about what therapy should be and what computers can ultimately do.

## Talking to the Dog

Almost all of today’s mental health apps are built on that potentially flawed premise. Many take the form of an instant

message-like interface with a friendly character, making comparisons to ELIZA difficult to avoid. One prominent example is Woebot, which boasts Stanford-linked founders and closed an \$8 million funding round in 2018. As with its distant ancestor, Woebot's lack of genuine human understanding is easily revealed. Telling the program I was feeling anxious about the fact that "I have so much work to do" resulted in this stilted interrogation, which reads as if it comes straight out of a CBT manual:

*Are you perhaps catastrophizing in your statement?*

No

*Are you Future Predicting with your thought, "I have so much work to do"?*

No

*You're doing awesome thinking through this by the way... we're nearly there*

*Are you 'Mind-reading' in your thought, "I have so much work to do"?*

No

*How about black-or-white thinking in this thought: "I have so much work to do"?*

Nope

Got it

*And is there a chance that you may be mistaking your feelings as evidence of the truth?*

No

*OK Matthew, you did a great job there...*

*Now for the most important bit... rewrite "I have so much work to do" without those distortions.*

I have so much work to do

*So how did your rewrite go?*

The smarmy attitude and rigid dialogue of apps like Woebot makes one sympathetic to Weizenbam's view that computerized therapy is inhumane. At the same time, I couldn't help but notice again the similarity to my own work as a game developer. Using a bag of tricks not unlike the ones Weizenbaum used to create ELIZA, many games invite players to believe that digital characters are friends who even perceive and understand them in some way. Combinations of animations, sounds, and contextually appropriate behaviors create illusions not only of life, but of genuine communication between the alien worlds of humans and computers.

***"If the feeling of connection is all that's needed for successful therapy, then human-to-human interaction may be superfluous."***

Taking advantage of "the human propensity to suspend disbelief in the presence of a persuasive dramatic presence" was the "true source of Eliza's power," the digital media pioneer Janet Murray wrote in her groundbreaking 1997 book *Hamlet on the Holodeck*. In the world of games, these persuasive dramatic presences are (usually) clear about what they really

are — characters. Game developers still rely on audiences to voluntarily buy into them at some level, in a parallel to the way the live theater becomes reality, though only temporarily. The actor who was stabbed to death during the play returns to the stage at the curtain call; the gamer eventually takes off their VR headset. What happened inside the magic circle was certainly not “true,” but was also, in a way, real, and exerted a powerful effect on those who experienced it.

Might this be all that it is needed for automated psychotherapy to work? In the nearly fifty-five years since ELIZA first appeared, very few digital characters have made an attempt to understand their users in any real way. The reason might be that they simply don’t have to in order to be compelling. And in that case, does it really matter if a therapy program understands its user or not? One of the principal investigators on the project that created Ellie, the computer science professor Louis-Pilippe Morency, compared the experience to talking to a pet. “Some people talk to their dogs,” he said, speaking to a journalist for the *Atlantic*. “Even though the dogs don’t understand it... I think there’s a little bit of that effect — just talking with *someone* makes you feel better.”

Aside from the fact that dogs aren’t, so far as we know, uploading their data anywhere, this is an important consideration. If the *feeling* of connection is all that’s needed for successful therapy, then human-to-human interaction may be superfluous. As Turkle asks, “If a person *feels* understood by an object lacking sentience, whether that object be an imitative computer program or a robot that makes eye contact and responds to touch, can that illusion of understanding be therapeutic? What is the status — therapeutic, moral, and relational — of the simulation of understanding?”

## Between a Novel and a Game

---

Perhaps because of the shifting priorities of a new administration, DARPA's interest in mental health has waned since 2014, and it isn't currently funding research into goals like treating PTSD. But witnessing Ellie at work caused me, over the course of the next several years, to design and draft a video game about the experience of computerized psychotherapy. By early 2019, the world I had slowly been building took its final shape: a present-day Seattle not so different from the real one, where gig workers called "proxies" read prompts given to them by "Eliza," a cloud-based, AI-powered virtual therapist program developed by a fictional tech giant. The company names the product after Weizenbaum's program in an attempt to ride its notoriety while, in true tech giant fashion, ignoring its creator's misgivings.

Players guide the game's main character, Evelyn Ishino-Aubrey, as she takes a job as one of these proxies, performing therapy sessions for a wide variety of people. As a proxy, all she can do is read what the Eliza algorithm has told her to read. Evelyn is warned multiple times not to deviate from the script, priming savvy gamers to believe they will immediately be able to do just that. But Evelyn performs session after session without deviating. For most of the game, I lock the character into being a "good" proxy, even at the expense of upsetting players' expectations. If Evelyn expresses doubts, she does so privately.

I deny players choice for such a long time because I couldn't accept the idea of a game where different phrases uttered by a therapist would be considered better or worse for a patient. It's easy to imagine a game in which a client comes in with certain combinations of conditions, and the challenge for the player is to pick the right things to say to match those conditions. But

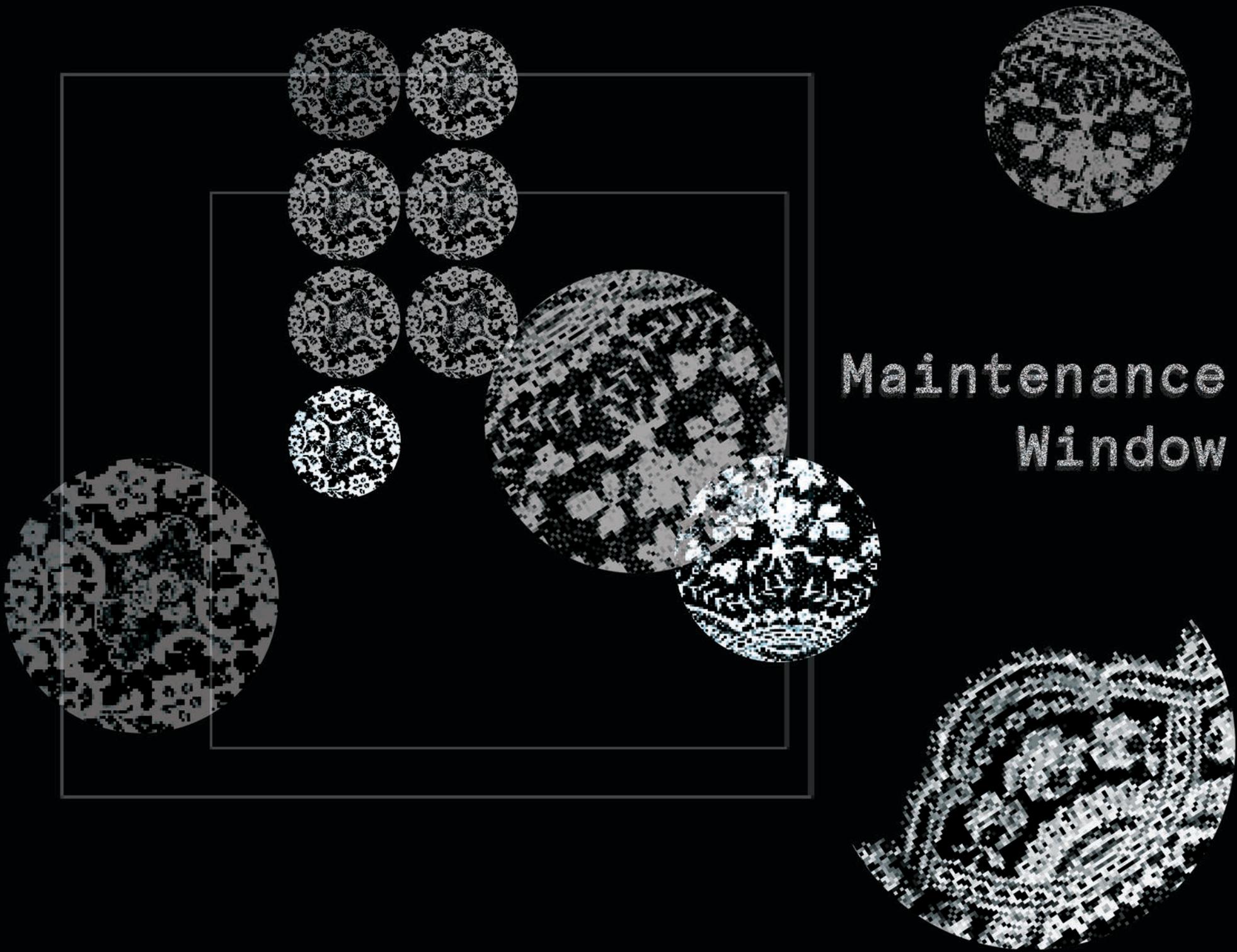
that would be to accept in advance the notion that a certain type of person needs to hear a certain type of thing in order to be guided to the best possible outcome — an assumption encoded in materials like the CBT manuals I had studied. I wanted to create therapy clients who were irreducible individuals, real fictional human beings. So in opposition to prevailing game design wisdom, I ended up with something in between a novel and a game, something that requires players to be patient and to give up control.

*“Layers of human-computer interaction help to challenge the simple binary between warm and empathetic human-to-human therapy and the encroachment of cold, computerized systems.”*

Between sessions, Evelyn has lengthy conversations with people who work, or used to work, on Eliza. Some believe the product is working well; others very much don't. Some see it only as a ladder for their own ambitions in the tech world and beyond. Evelyn must also confront her own past and struggle with depression. As the story of the game concludes, it allows players to decide the best way for Evelyn to move forward in life. Each choice includes some compromise. She could continue to contribute to Eliza. She could take her newfound interest in therapy to pursue a career as a traditional human-to-human therapist. She could turn her back on all of it. This is the first and last major choice the game asks players to make.

A human player playing a virtual human (Evelyn) whose choices are circumscribed by a computer program (Eliza) which is at once both real and a fiction created by me, its human developer: my hope is that these layers of human-computer interaction help to challenge the simple binary between warm and empathetic human-to-human therapy and the encroachment of cold, computerized systems. The reality, of course, is more complicated. Under the manualized framework of CBT and similar modes of treatment, even human therapists act robotically, whereas in ELIZA, a rudimentary but surprisingly expressive computer program emulated a much more humanistic, open-ended style of psychotherapy. It is not about a “computer” approach versus a “human” approach—it is about an indifferent versus a compassionate one. ~~~

Matthew Seiji Burns is a writer and video game designer in Seattle, Washington.



Maintenance  
Window

# Maintenance

Cubicles as far as the eye can see, each one containing a government worker clacking keys in front of an ancient, humming green screen. On the other side of those screens are millions of people trying to file for unemployment, apply for food stamps, or access any other number of government services. The website keeps crashing when they try to submit their online form, or they wait in line for hours only to reach the end and find that the piece of mail they brought to verify their identity is invalid. Try again tomorrow.

This is a familiar trope about dysfunctional government, one that emerged to serve the political goals of the right during the crisis in the 1970s that was, in the words of Ruth Wilson Gilmore, “the close of the golden age of US capitalism.” The stereotype – mindlessly bureaucratic, hopelessly outdated – was used to justify attacks on organized labor, tax cuts for capital, and the hollowing out of what little safety net the US established after the Great Depression and World War II. Historian Mar Hicks writes in their piece for this series that the implications of that hollowing out for government technology were that “as state governments have moved to slash their budgets, they’ve been less and less inclined to pay for the labor needed to maintain critical systems.” We asked a few of the people who were tasked with making those systems work anyway about their decades-long careers in government technology.

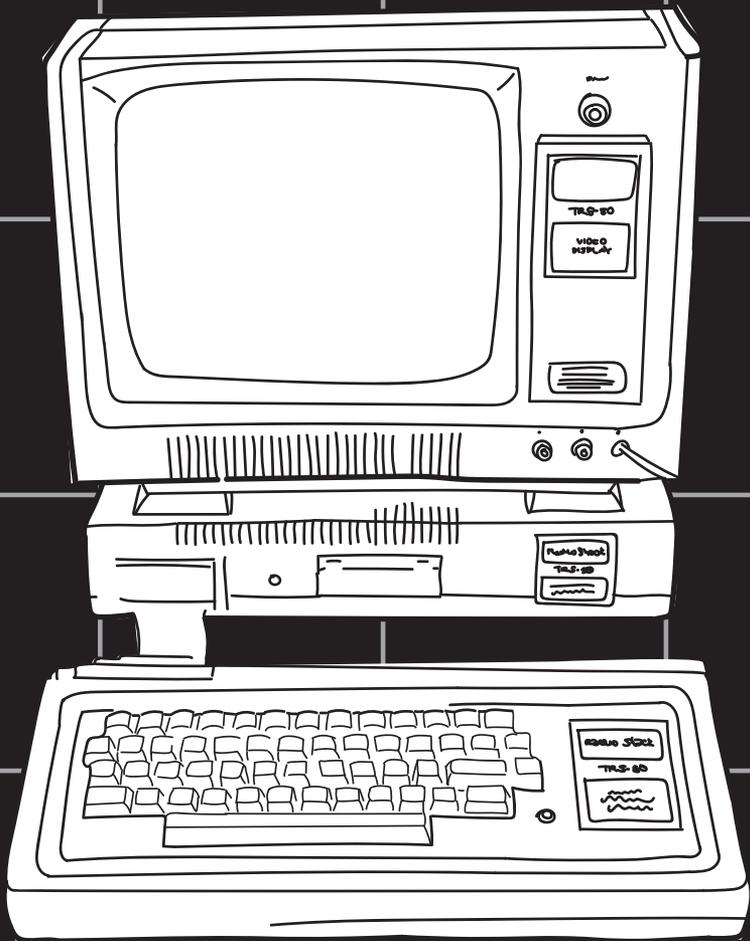
Beginning in the mid-1980s, Jed Wagner began architecting the federal system that handles veterans’ benefits appeals – a system that he alone was responsible for for three decades. Around the same time, Mike Schwab was a unionized employee

of Illinois' Central Management Services, developing and maintaining mainframe applications that other state agencies used in their day-to-day work. A few decades later, Adam Grandt-Nesher was typing Hebrew blindly into dumb terminals, the start of a winding career through taming legacy systems across private and public sectors that landed him in the middle of mainframe modernization efforts for the US federal government today. We sat down with the three of them to get a kaleidoscopic picture of how the government technology systems we have today came to be, and what they will look like in the future.

The interviews included in this series were edited by Jen Kagan, and conducted by Christa Hartsock and Julie Sutherland, who work together at Code for America, a technology nonprofit that works with governments to better deliver services like SNAP, Medicaid, and criminal record clearance. Christa is a software engineer and Julie is a designer and qualitative researcher. The interview team would like to thank Mike Schwab, Adam Grandt-Nesher, Jed Wagner, Marianne Bellotti, Philip Young, Teresa Curtis, Jason Anton, Joe Klemmer, Mike Cowden, John O'Duinn, Genevieve Gaudet, Zoe Blumenfeld, Jen Pahlka, and the many others who fielded questions, chased mainframe leads, and shaped the series.

The story does not stop here – we would like to continue chronicling stories from mainframe developers and need your help. If you have an experience to share, please reach out at [mainframes@logicmag.io](mailto:mainframes@logicmag.io).

# Window



## MAINTENANCE WINDOW

# Mike Schwab

## On Mainframe Programming for the State of Illinois

Mike Schwab's brain is an indexed list of computer models, storage capacities, and release dates. Schwab spent most of nearly thirty years as a mainframe programmer and storage expert, keeping the systems running that processed payroll for state employees and enrolled them in healthcare programs. We open our series with him because, over the course of his career in government technology that began in the mid-1980s, he saw a little bit of everything: the shift from union programmers to contractors, the budgetary constraints that made state agencies risk taking down their tech rather than pay for extra disk space, the questionable interview processes of a state police department.

He sees the machines and languages he worked with as relatives and descendants in the computing family tree. Modern mainframes descend from the IBM System/360 which descend from the Hollerith machine used for the 1890 census. He stands by COBOL, one of the tree's steadfast trunks, even as it's been blamed for failing millions of Americans who filed for unemployment in the wake of COVID-19.

Christa and Julie sat down with Mike to find out why, and hear his reflections from his time working on government technology for three decades.

---

### **Tell us a bit about your work.**

For twenty-six years, I was with Central Management Services (CMS) for the State of Illinois. CMS is like a super agency that provides services to other state agencies. They run the payroll system and do the health insurance and other benefits for most everybody that's employed by the state, including universities. There are some large agencies that do their own payroll.

I used to work on the application side developing for mainframes before I moved over to storage. On the storage side, I worked with state agencies to make sure they had enough space on the mainframe to hold their files.

### **What did the storage side entail?**

We managed equipment for state agencies and billed for CPU, disk space, and all that. If equipment was shared among a few of our agencies, we only billed each agency for what they were actually using. But there were some storage areas that were limited to one agency — since they were the only ones that used it, they got billed for all the storage, whether or not it was being used.

We set a notification level of 10 percent. So if you had less than 10 percent storage capacity remaining, you got notified to add more. But, you know, the agencies didn't want to pay for another storage volume. We'd call them up and tell them they needed more, but they'd say, "Let it drop to 9 percent, 8 percent, 7 percent — we're still working! — 5 percent! Four percent!"

Eventually, they'd say "We're not working any more. We need more space."

**Why do you think they cut it so close?**

They didn't want to pay for it. If we added a volume, which was eight gigabytes, they'd have to pay for that space even if they didn't actually need it for a file. They didn't want to drive up the computer budgets.

But when you're operating like that, any tiny little bump can stop your work from happening. And it did from time to time. One time, this agency had a log get up to the maximum number of extents. The log itself wasn't too big; it was just using too many little chunks of memory. They called me and said, "We can't do this transaction!" Everything came to a screeching halt and their system shut down until they had defragmented that log so that they could gather more space for it.

*"We'd call them up and tell them they needed more storage, but they'd say, 'Let it drop to 9 percent, 8 percent, 7 percent – we're still working!'"*

**That sounds like what's happening to these government systems that are crashing as millions of people file unemployment claims. What's your perspective on all the recent news about that, and particularly the criticisms of COBOL? Do you feel the criticisms are warranted?**

Totally unwarranted. COBOL is the backbone behind the businesses on the web. Your credit cards would not work without COBOL. American Express, Visa, MasterCard—they're all running COBOL. Airline tickets? The backbone behind that is all on mainframes. The airline industry developed software in the 1950s on specifically designed computers. The software they run on mainframes today is directly descended from COBOL code they wrote for the IBM 360, which came out in the mid-1960s.

COBOL was designed for accounting. It's designed to be exact to the penny. The IBM 360 came out with instructions that fit into that kind of paradigm. When you go to any other language, they don't often have decimal math instructions for that kind of accounting work. So any time you port COBOL over to another language, you can get your basic logic over and then you've got to test your accounting and make sure you come up to the penny again. There are often rounding errors that creep in after a while: you start losing your least significant digits and, eventually, you're no longer accurate to the penny. That does not happen with mainframes running COBOL.

**COBOL's strengths aside, these systems are struggling. Do you have any thoughts based on your experience about why they're struggling?**

None of the other recessions we've had since 1918 have come all at once like this. You start to have a recession and your unemployment systems are handling 0.1 percent more, 0.2 percent more. Your systems have time to grow and scale. You've got more workload handling the new people coming in while paying everybody else out from their last week's records.

Today, you've got 10 percent of the US population laid off and filing for unemployment within months. That's one hundred times your normal recession workload. The employees who

process those claims are completely overwhelmed. The CPU processing speed is overwhelmed. Your disk storage for all these new people is completely overwhelmed. If you're going to 15 percent unemployment and you were at 5 percent, you need three times as much storage space. And the people who work in the unemployment office are trying to handle fifty to a hundred times their normal workload. You can't just enter your information and automatically get unemployment. A person has to sit there and approve or disapprove every request.

*“Today, you’ve got 10 percent of the US population laid off and filing for unemployment within months. That’s one hundred times your normal workload in a recession.”*

**Yeah, we work a lot with government caseworkers and there’s always a human that has to do something on the other side of the screen. When I talk with friends who work at Facebook or wherever, they think it’s this perfectly oiled mechanical system. But there are humans all over it, and they do really complex things. And humans are hard to scale.**

That’s right. At CMS, we worked with the agency in charge of unemployment. The only person we ever saw from that office was a shop steward who would come down for meetings. She had a guy transfer from the Department of Corrections who had trained to work on the unemployment system. Of course, it had lots of complex rules and you have to have them all memorized. It takes weeks, if not months of training to learn all the

rules you have to follow before you can say yes or no. So this guy spent months in training, but in the end, he found it so stressful to be handling the unemployment claims that he went back to corrections.

**How do you actually sustain that when the software is so complicated? And we've seen several times in our government work that one or two people maintain ten systems. What happens when they leave?**

The complexity of these systems is a big issue. On the technical side, if there's only one person who knows a system and that last person goes to retire, there's nobody left to train new people on the software. And the mainframe software is so much more complex to use. With Linux or Windows applications, you get trained up in a week. But for mainframe applications, every installation is custom. There are tricks of the trade that new hires have to learn by osmosis, by working with the prior person. A new guy comes in, you tell him what to do, and you sit there and hold his hand while he goes through and does it. And you are still learning new tricks even after a couple of years.

Now, if you're the last person and you're not training anyone, all that knowledge goes away. A new person who comes in and starts working on a system that hasn't been maintained for three or four months. People try to write everything down but you never say a lot of these things, even to a person sitting right there next to you.

If you want to see somebody actually working on the mainframe screens to get a sense of how complicated they are, moshix—that's his username—on YouTube has about 160 video recordings up.

## **Have you watched some of these?**

Quite a few, yes. Some of them I watch because I've never used the products he's working on. Others remind me of systems I've worked on and I find it enjoyable to watch it over again.

## **Get Your CICS**

---

**You mentioned earlier that you were on the applications side at CMS before you were on the storage side. Can you tell us about when you started and what it was like?**

When they hired me, they were moving into a brand new office building. There were about 150 people there, and we had thirty or forty mainframe applications. We had systems like CICS, IMS DB, just about everything. We were a service bureau.

**Teams now tend to have product managers, designers, and programmers. What was the makeup of your team?**

We had systems analysts and we had coders. I was in-between. In terms of designing the applications, you'd start with a screen layout and a database layout, and go from there. Usually, you've got a form that you need someone to fill out. You have to replicate the form fields on screen: first name, last name, middle name, Social Security number, address, and all that. That gives you all your records.

The first IBM was this Hollerith card system they used for the 1890 census. You'd have a person's name, their address, and all this other information punched on cards. Of course, it wouldn't all fit, so you'd need a few cards for each person and you needed to know what information was on what card—card one had the name and address, card two had the spouse and job, card five had whatever. That's your set of information for that person.

Now that you've got computer files on hard drives, it's all in one record. All the transactions for that person — having a kid, changing their healthcare plan — those are different fields under that person.

**It sounds like, when you were building out features, there was programming you did on the backend and then programming in CICS. Is that more like the frontend?**

CICS itself is a multitasking transaction manager, where it takes in a transaction name that the user enters. It uses that transaction name to start the COBOL programs associated with it. A transaction in this case is like a thread. Maybe CICS runs the programs, then returns the result and exits. Or maybe it shows you a result on the screen, asks for more inputs, then does more work, and then exits. So CICS itself would not change but you'd get releases every other year with updates through the operating system.

When you found a problem with your application program, you'd put the changes in at that point. Only a few people could update the authorized production application, because you just don't want a bunch of people going in and messing up a production database, causing all kinds of problems. The real problems were bad enough.

**Did you all do any user testing where you'd have a caseworker or an HR worker go through that screen with you?**

Yes, we'd call up our end users and they'd try out the changes to the system themselves. Everybody who did the testing had a coax switch to switch between test and production.

**Like a physical switch?**

A physical switch on a coax line, yes.

**Software engineers now expect to be on call and sometimes get called in the middle of the night. Were you on call for these systems?**

Yes, I was. All thirty-some years.

We would have two or three people in a group, so we could rotate. But then we got people in who had grown up around PCs, and they just don't appreciate that computers have to operate 24/7.

**How did being on call for thirty years affect your life?**

Since I did it my whole life, it's hard to say.

I was limited in how far away I could go for a social event since I had to be able to get back to work if called. And limited on drinking since I had to be able to drive to work at any time.

Most nights I didn't get called, but then some nights would be bad and I'd get two to three calls in one night. Other times, I wouldn't be called but once a month. When I got towards the very end of my career, it was less than once a month because we kept so on top of those storage groups, made sure we had enough space on them.

**When you were working on those applications and you were working with people who were using your application, how did that feel?**

I felt a real pride and ownership in working on that screen. There were 300 to 400 people who used it to sign up 80,000 state employees for healthcare programs all across the state. It was important that it worked just right for them.

These people are all counting on me to do my thing right. I felt quite a bit of responsibility.

## Make and Model

---

### **Backing up a bit in the story – how did you first get into computers?**

I was in high school in 1979 and they got in two TRS-80 Model I computers with a cassette tape drive and sixteen kilobytes of memory. We started doing little projects on that. They had one in the electronics lab and it was empty during my lunch hour, so I would go down there and use it then. That got me interested in computers.

*“Our mainframe computer at the university had eight megabytes and when they expanded it to twelve, it took a whole refrigerator-sized unit to add that amount of memory. Every four megabytes basically meant another refrigerator.”*

You could sit there and see your results right in front of you. I had even used a teletype printer keyboard at this physics camp at the University of Wisconsin-Whitewater in June 1978. You type in a command and it's printing as you go. Then, when you press enter, the command goes in and starts printing up the results.

I got my computer science degree at Illinois State University in Normal, Illinois. I used punch cards for my first two classes in the fall of 1981 and spring of 1982. We got IBM 3270 consoles at

that time, and that was also when the IBM PC came out. I also had one class on an Apple II computer with a Zilog Z80 card in it, which was a follow-up to the Intel 8080.

**Could you describe the teletype printer keyboard more? Was there a screen?**

We used it to play this Star Trek game. You didn't have a screen; it was all printed text. You'd type your command at the printer because it was a teletype interface over an acoustic coupler modem that had cups to hold the phone headpiece with speaker and microphone. The teletype machine would output sounds that the cups would then receive and transmit through the modem. The telephone system didn't have the RJ45 phone plugs to directly connect phones into the modem yet.

**When you were going to college at Illinois State, why did you choose computer science?**

The IBM PC didn't come out until November '81 and they didn't have one at the university until fall of '82. At that time, if you wanted to use a computer, you had to work for a big business that could afford to buy one. Even the TRS-80 was a couple thousand bucks and I was earning three bucks an hour. I was not going to be able to get one. I knew a computer-oriented job would be with a big company that had an IBM mainframe or a minicomputer.

In 1981, the IBM 360 had been out fifteen years and they were still rolling out major changes to the operating system. It had thirty-two-bit registers but only used twenty-four bits for the address so it could only address sixteen megabytes. Our mainframe computer at the university had eight megabytes and when they expanded it to twelve, it took a whole refrigerator-sized unit to add that amount of memory. Every four megabytes

basically meant another refrigerator. And then the computer core itself was one or two refrigerators.

**So where did you end up working after college?**

I applied to the Illinois State Police (ISP) because they had a job posting. I went and interviewed, and they initially said they were going to hire me, but then they called back and said I wasn't going to get in because I had voted in the Democratic primary. They told me to contact my county Republican chairman and join their Young Republicans group, which I did. After that, ISP offered me a job. I started on December 3, 1984 and I worked on various applications there. One was crime statistics reports, and then I went to online applications for them. Never involved in cases; I was just making sure the computer systems were up and running to track traffic tickets and, later, investigatory files.

**What were your first impressions of that job?**

The office building was sixty years old by the time I got there. You had a desk on one side and a computer table on the other. There was a 3178 display, which is basically a CRT unit and a keyboard with a little box on the side of the computer that communicated over coax cable to the box in the closet. There wasn't a full PC there yet. It worked really well. In fact, I kind of missed it when we got our PCs. The PCs wanted everything in twenty-four lines instead of forty-three, so we had half the amount of screen we'd had before, but it seemed like even less. But other than that, you know, it's all the same information.

**Could you talk about the applications that you worked on while you worked there and who used them?**

It was the dispatchers at the state police headquarters who talked to the troopers over the radio. Entering traffic tickets,

entering a license plate to do a query on it. Find out who the owner was, if they had any warrants out on them.

That system had been around over a decade by the time I got there in '84. It was about '69 or '70 when they got their first computer going to communicate with the FBI and offices in other states.

### **It was a federal project?**

There was a state project called LEADS that communicated with the FBI's computer and then each state had their own computer and applications, just like every state has their own state police.

These were applications written in COBOL, Assembler, and IDEAL. Then I did a reporting system on a CICS with Datacom/DB. For a drug traffic stop system, I copied an application we had to a new application called Valkyrie. It was to track what they found during traffic stops: drugs, weapons, whatever was found. I think we were the first state to come up with an application to do something like that, and that was one of the last projects I got to do over at the state police about 1989. The headquarters for that was going to be in El Paso.

### **We work on technology to help clear people's criminal records. So I've talked to a number of people at the FBI and Nlets about making sense of criminal record data across state lines.**

Yes, it's complicated. The FBI has their own network for sending electronic law enforcement data nationally, but the states can't use it. The states had to create their own organization called Nlets to communicate criminal justice information, like out-of-state arrests, across state lines.

There are other barriers too. You know the 1-800 number you have to call to do the background check for a firearms purchase? By US law, all that information has to be stored on paper. So when you call that number, somebody takes the person's information and goes and looks it up on paper and then calls you back with the information. You can't just type it in and get the information back, or even just have a person in FBI headquarters type the name in and find out right there. The FBI person doing the lookup has to go physically search through paper for that person.

And Congress doesn't want the information released. So they just put up all these obstacles in people's ways.

### **Why did you leave ISP?**

Well, that was the asbestos. We were up on the sixth floor and we had like skylights up there and the caulk would leak from time to time. Sometimes, we'd have a leak and you'd smell the mold. And then that started dropping on the asbestos and it all started falling off the pipes and onto people's desks.

One person had a four-inch-high conical pile of white powder on their desk every day for about two weeks. We thought the janitor was pulling a prank or something because they'd sweep it off in the morning and there'd be nothing there all day. Then we'd come in the next day and there would be another pile there.

When we found out it was asbestos, we were concerned. We reported it to building maintenance and they covered it with duct tape and started taking air samples. This air sample machine was sitting there every day making noise. We lived with it. They moved us to another floor while they fixed it up, and when they were done, we went back up there.

## **You went back?**

Yup. But that was the last straw for me. Repairs were going so slow. They were there for another twenty-five years in that building. It was 2009 before they bought the Franklin Life buildings in town and got the people out of there. It was 2012 before they got the computers out. The rest of the building still has to be cleared out and they haven't done that. It's been sitting empty for close to a decade now.

## **Union Dilution**

---

### **Were you covered by the union at that time? And was the union present in those conversations about the asbestos?**

Most of my time at state police, I was in a merit comp, a non-union position. The “programmer analysts” were in the union, but we were just “programmers” so we weren't. Right before I left, around '88, they brought us in.

### **Do you know why they did that?**

When the union wants to bring in a particular title, they send out letters to everybody in that position. When the union gets back so many letters, they can go to the National Labor Relations Board and call for an election where everyone votes on whether they want union representation. Enough of us voted to join.

*“And then that started dropping on the asbestos and it all started falling off the pipes and onto people's desks.”*

**By the time you moved to CMS, were you always in the union going forward?**

After that point, yes. I told ISP in spring of 1990 that I was going to go over to CMS. The US Supreme Court was sitting on a case called *Rutan v. Republican Party of Illinois* over the hiring practices by the state of Illinois. There was a transfer freeze and hiring freeze during that whole period, so I sat there for six months working on a project not knowing when my transfer would go through. Once that decision came down, all the transfers went through all at once.

**What union were you a part of?**

AFSCME. Different locals represent different agencies. There was one for the police officers in the state police department and then there was another for several agencies that have employees in Springfield, just the employees in Springfield. There was Local 2224 that covered CMS and five other agencies in Sangamon county. And then Council 31 has all the other state employees in different locals. So they were running the show whenever they were negotiating contracts. In about 1994 or so, I actually stood election for negotiator and won, so I was involved in those negotiations for about five months. We'd go in for a Tuesday, Wednesday, Thursday negotiation session. And then three weeks later we'd have the next one. Each local could send five delegates, so we had something like 250 people total for our side and then there'd be maybe a dozen agency heads for the management. We'd meet in convention centers.

**What else would the union work on?**

For example, my coworker was applying for other positions and she had the seniority and the skills to take these other positions. But she would never hear back about her applications. Most

people, when they don't hear anything, they just assume that they didn't get it. But she would follow up, find out who got the position, and then compare their record to hers. If she had seniority on them, she'd file a grievance. She'd get the raise she should have gotten before, though it wasn't backdated to make up for all the time she had been underpaid. She was stuck and trying to go someplace else for fifteen years. I know that happened to her at least three times. And it just seemed so cheap, you know. Her seniority should have applied—your seniority is supposed to give you priority within your bargaining unit, within that county—but the state wanted somebody else and it didn't cost them much.

The state basically just hires whoever they want. And if the person who they should have hired eventually found out, then and only then would the state have to give them a small raise to match what should have already been their pay. It's a low-cost violation for them.

### **When did you first experience contractors in your government work?**

When I was first hired on at the state police, they already had some contractors and we basically treated them the same. The main thing was, on the phone list, they all had an asterisk by their names. And if they did something wrong, then boom, they were gone and there was nothing they could do about it.

The big thing about being a state employee is you get a pension and health insurance. The state has to do that to keep people there, and if you as an employee want to keep those things, you have to stay with the state of Illinois. But if you were a contractor, you didn't get anything else. They were paid twice what we were, but no pension or healthcare. If they found better wages or

something more interesting somewhere else, they could be gone in two weeks. Lots of people stuck around, but they also might be gone in a year or two when their project wrapped up.

### **What sorts of roles did contractors fill?**

Application programming for the most part. The union had put into the contracts that agencies couldn't do a personal services contract. So that meant that they could never interview an individual contractor. The contractor would have to incorporate into their own company or be hired through some other company. But they were basically just employees.

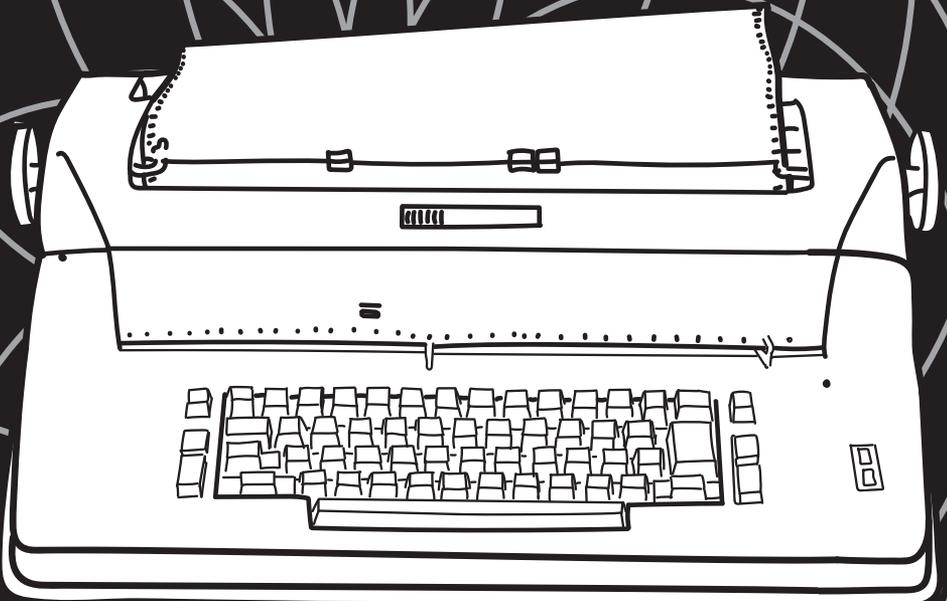
We got rid of those for the most part about 2005. The union got the state to recognize what state agencies were doing. Now, if you want an employee, you've got to hire an employee. You can't just hire a contractor through a third party and say it's not personal services. The state had a year to hire replacements for those people. That was quite a bit of turmoil: a lot of people to hire and a lot of expertise was lost. I mean, some of the contractors had been in there for decades.

### **Did it seem like that was one for one, like one employee was hired to replace every contractor, or were there some contractors who were never replaced?**

There was considerable shrinkage. It's worth noting that by this time, the state of Illinois had cut taxes, yet kept up their expenditures, so they were behind on paying bills and falling further behind every year. They were borrowing out of our pension funds just trying to catch up on bills. Today, Illinois has something like five billion in unpaid bills with about six months delay.

Back to your question about contractors specifically: the agencies tried hiring, but they didn't get many replacements. Many contractors didn't apply for their own jobs as employees because

the pay was lower. I saw the job postings go up on the bulletin boards. Normally, there would be enough surface area on the board for each listing to have its own place. The listings were so thick you had to leaf through pages and pages. Even after a year, they had all these vacancies they were still trying to fill. ~~~



## MAINTENANCE WINDOW

# Adam Grandt-Nesher

## On Modernizing Infrastructure

Adam Grandt-Nesher built a career transforming computing systems – first with the Israeli army, then for fintech conglomerates, and now for the US federal government. Over the years, he has seen and tried many different ways of making seemingly anachronistic computing infrastructure work, learning much about the human value these systems provide and the importance of addressing rather than obscuring their core technical problems along the way.

After spending two years trying to transition from fintech to more morally fulfilling work, he landed at the US General Services Administration, where he now works with a variety of federal agencies to get the most out of their infrastructure. He was hired as a “termed” employee, a practice adopted by government technology transformation agencies like the United States Digital Service (USDS) and 18F, which means that he will transition out of his job in only a handful of years – a timeline shorter than most transformational projects require for completion.

Christa and Julie chatted with Adam to learn more about strategies for modernizing legacy infrastructure, transitioning from the private to public sector, and the challenges and rewards of government technology.

---

### **When did you first start learning to program?**

I've been programming since I can remember. I grew up in Israel so I couldn't actually speak or read English, but I could write elementary BASIC code.

### **What was it like not knowing English when you were learning BASIC?**

I memorized the commands in Hebrew, which were overlaid on top of the standard QWERTY English keyboard. So to draw a circle I spelled the word 'circle' in the Hebrew letters. It's not an actual word, obviously. I didn't recognize or understand English letters.

I've been reading fluently in English since I was eleven or twelve, and learning English definitely helped me learn to code because I could recognize the shape of the letters and something about their meaning earlier than the smarter kids did.

### **Do you feel that the prevalence of English in coding languages is a barrier for non-native English speakers?**

I stopped using Hebrew when working on computers when I was fourteen or fifteen, as soon I had access to control my own computers. Growing up, I used the Hebrew version of Windows, which is built to support right-to-left languages and tries to translate. But, supporting both Hebrew and English languages is really hard and is never done well, and so I gave up on Hebrew as a working language at a very young age.

So, the short answer is yes, absolutely it's a barrier. And the solution is that most of us just give up. My phone and computers have been in English for about as long as I can remember, because the experience is horrible otherwise.

**When you were young, what did you think that you wanted to do with your life?**

I was very, very certain until my late teens that I was going to be a doctor, like a good Jewish son of a good Jewish mother. I have some medical challenges though, and when I joined the Israeli Army for my mandatory service I was assigned to be a truck driver, which was less cool.

At that point, I had already developed some level of programming knowledge. I remember an interesting discussion with my HR non-commissioned officer, where I expressed discontent with being a truck driver. She tried to explain that it's a fantastic gateway to a career in life, and I—maybe slightly rudely—expressed that as an eighteen-year-old, I had slightly higher aspirations than being a truck driver.

And so I failed the truck driving class. I failed it horribly.

**Was that intentional?**

Yes, very much so. Somewhere in my files, I still have that rejection letter, which more or less says that I am too stupid to drive a truck. It's one of the things that I'm most proud of.

That got me out of that black hole and into the general assignment. Then I mentioned that I could code and they made me a small computer operator in the Salary Analysis unit of the HR division of the Israeli army, which is now called the Manpower Directorate.

This job started on an actual terminal, which is basically a keyboard attached to a printer. You typed blind, hit enter, and got a response, which is usually something like “error on line three.” Then you have to guess what line three is, because the end of line submits a query so everything you’re typing is on one single line. So, to debug, you just type it over and over again.

The language used on the terminals was a Hebrew query language called **אמת 3**, or EMET 3. In Israel, speaking Hebrew has a nationalistic value. It’s a revived language that was part of the creation of the country as a country. So a lot of the motivation behind the creation of **אמת 3** was that it had to be in Hebrew.

*“Like every other piece of legacy software, when you work with it long enough it shapes the organization around it.”*

The query language was written in Hebrew but compiled into English—because everything is compiled in English. Brackets had to be reversed because figuring out how to write brackets in reverse does not work in that compiler installed in pre-Unix VMS. They used the ASCII code that corresponded with the letter that was on the keyboard when you typed in Hebrew, and while compiling reversed the Hebrew letters into those ASCII codes. It was essentially gibberish English, basically what I did when I was learning BASIC. The language was written in 1973, and I started working on it in 1999.

But, that was my life. You’d get a question like, “How many people served between 1998 and 2000 as nurses in this unit? And

how much money did they get paid?” And then they would run an analysis to get a sense of what kind of modification to their salary they should have because they were practicing emergency medicine in a medical combat unit. My job was to run those reports for them, because translating from the logic of the query to that stupid language was hard.

Like every other piece of legacy software, when you work with it long enough it shapes the organization around it. Workflows are created to handle the limitations and people like me start building bits and pieces on top of it to make it better so that the workflow can become better. And what you end up with is a whole bunch of legacy code held together with duct tape.

### **How did you feel about your job at the time?**

Initially, I was very grateful because I wasn't driving a truck. Being the clever guy who can do these reports when everybody else gets stuck was exciting for a while. But then I got tired of doing the same thing over and over again. You just have to copy and paste, change the dates. It gets very boring.

### **Outside of the desire to do interesting things, did you feel any responsibility to your job or a sense of civic duty? I'm curious what motivated you.**

That's a discussion that's really interesting for my current job. But, for that job? You need to remember that I was eighteen and serving in the Israeli army is a complicated thing. Everybody has to do it. Caring about the people that I was affecting by supporting the system clearly isn't something that eighteen-year-olds do. Or at least, I didn't at that age. So, no. I was there because it was fun and interesting and it didn't involve driving a truck.

Now, it's very different. My current work affects five million federal employees and two and a half million military units. What

I'm doing directly improves their lives and improves the life of the people who serve them, and that is awesome. That's why I'm doing my job now. I don't think that any of those considerations existed back then.

### **What made you leave that job? Where did you go next?**

I moved from the Salary Analysis unit, to the HR division IT. At the end of my service, I was given the opportunity to work on a tool that was supposed to remove the need for anybody accessing or using **3תמא** ever again. I was so full of resentment of that system that it sounded like an awesome idea.

The first RFP for that project came out in 1981. I was born in 1981. In 2001, I was essentially the tech lead on it. The intention was to wrap this system in a modern—2001 modern—UI system where people could go into a graphical interface, type in the ID number of the soldier or the query, and get the results they needed to make their decisions. I decided to volunteer for an extra six months of army service to see this project through.

*“Wrapping a UI around a legacy system doesn't pay the technical debt. It adds more technical debt.”*

The workflow that we're trying to solve is the following: we have something called lonely soldiers in the Israeli army. They are soldiers who usually are American volunteers. Their family does not live in Israel. They get housing, food, and extra money. For the record, when you join the Israeli army, as a soldier, you basically make twenty-four bucks a month. You don't get paid

for your service for the first three years. But, lonely soldiers get a bunch of services and more money because they need to eat.

So, let's pretend that a lonely soldier is in a combat unit on the border somewhere. When they submit their application for benefits, there's a couple-months-long process of passing paper up the chain of command, at the end of which, someone runs a bunch of queries and updates the record in the mainframe system. Now, let's pretend that there's a typo in the form being passed. That form would then travel back another two and a half months to get back to the soldier to be corrected.

What we wanted to do was to shorten that process by offering a graphical interface accessible to the local HR officer so they could submit the applications form electronically and have it approved by the central command HR staff. That was our goal. Take a six month process, turn it into two weeks. To make that possible, we have to build that UI system, wrap it around the actual mainframe system, and make the connection.

Now, that is a fairly common modernization approach to mainframes, but it's really not modernization. Today, I am generally angry at this approach.

### **How come?**

Because what I'm doing now is cleaning up after people who made that decision twenty years ago. I'm also twenty years older and I understand technical debt a lot better now. Wrapping a UI around a legacy system doesn't pay the technical debt. It adds more technical debt. My job now is literally cleaning up after that situation.

There's a bunch of parts of the industry where people basically make their living out of building wrapper systems around mainframes, because it means that, despite whatever is currently

running, you appear to have a shiny new modern system six months later. But you don't, not really, because you can't actually change the underlying workflow in any way—which works for a while, until you have to change something and you can't.

## **Making Moves**

---

### **Where did you go after the army?**

I got to New York and as I was about to head out on a month-long cross-country roadtrip, a relative of an army buddy of mine said he needed a website to write quotes for his moving company. I said okay, I'll work for a couple weeks and make a little bit of money and I can do a two month long trip... And that was nineteen years ago.

Eventually, around 2010-ish, I got pulled into fintech by somebody that I was working with on the moving side, introducing me to the idea of taking programming algorithms, hosting them on a stable system, and running it.

Essentially, the foreign exchange (forex) market is ginormous, and has many, many, many, many traders, with each of them trading tiny amounts with insane leverage. It's a very, very bad industry, morally speaking. There's a tool called MetaTrader, which was the primary go-to tool for retail trading in the forex industry—this is a Windows application running thirty-two-bit code in 2016. You wrote your algorithm and you ran it on this application on your desktop. If you had a power outage or one of your kids came over to play a game and closed your application, you would lose money as a result of it.

Fairly early on, we came up with the concept of taking a machine and running it on a Linux server using Wine, to make things more stable. I ended up doing a lot of work with Wine to make

it run this trading algorithm properly online. Today I would just do this with virtualization or containerization, but since we're talking about the late 2000s, none of these were options. I spent the next four years getting that up and running, having up to 11-12,000 people using the system at some point, and then selling that. That was the first and only software that sold successfully. It was my primary success in fintech.

Then, the company I worked for got bought by a Slovak conglomerate of foreign exchange brokerages. After buying our company, they just kept buying more and more brokerages. Each brokerage would hand us — hand me — their technology team and their technology stack. We would work to bring their team into our environment, taking their technology and combining everything to create a unified service. We became a technology service company.

As we bought new companies, we became really good at wrapping around other people's projects — regardless of which stack they were using — and getting them to talk to each other. We'd take complete projects, wrap them, and turn them into services supporting our unified frontend and unified APIs. Over time, we got into refactoring stuff and breaking them into different pieces and we eventually started building lean microservices, running on AWS, things like that.

Now that I'm talking to you about this, it's very, very sad. The story of my career is that I absorb other people's legacy software and then kick it until it becomes useful again. Which is, to an extent, what I'm doing now for the federal government.

### **What was your experience transitioning to the public sector?**

Well, at first I went looking for jobs at nonprofits as opposed to government because government is — in my head, at

least — morally ambiguous. But nonprofits don't like people who don't have degrees. Government doesn't like people who don't have degrees. Anywhere else in tech, nobody cares about what certificates you have; people care about what you've done. You only need certificates or degrees if you haven't done enough yet.

I spent two and a half years trying to find a way to transition my career over to something meaningful. At the end of the day, the problem is that the nonprofits jobs are few and far between. I applied for dozens of jobs. All of that failed miserably. My search for more morally fulfilling work ended up taking two years.

**Wow, two years is a long time.**

Eventually I discovered the United States Digital Service on Twitter. I applied to USDS and I went through their very tedious six-interview process. Apparently I passed that. But, because I'm a former officer in a foreign army and they do most of their work with the Department of Veterans Affairs and the Department of Defense, they felt I would never pass their security checks and decided not to hire me.

But there's a circle of government technology organizations where they all know each other and work together — USDS, the General Services Administration (GSA), Technology Transformation Services (TTS) which includes 18F, etc. They forwarded my resumé to them.

I interviewed for GSA and then waited three months to hear they wanted to hire me. And then, three months after that, they told me I'd need to start the background check process. It took something like eight months altogether from the time I interviewed for GSA to the time I started to work for them.

**I'm interested in how you feel about that government hiring process. My understanding is that one of the reasons why**

**there's an insistence on pedigrees or a specific number of years of experience is to try to level the playing field. For example, testing someone on specific topics in an interview can privilege certain backgrounds and it's easier to let bias slip in. But, if you look just at years of experience, in theory that will make it a bit more equal. That doesn't account for your experience with needing a specific academic pedigree, though.**

Yeah. I understand the need to try to remove bias from hiring. But access to higher education is a biased measurement. And in tech, it's also kind of irrelevant to skill.

One of the things that I have done many, many, many times is hire people who have just come out of their bachelors or masters programs and then spent six months paying them salaries while I'm undoing what academia did to them. People come out of academia with an interesting theoretical background. But honestly, let me ask you, have you ever done Big O analysis during your professional work as an engineer?

**I've had to think about it, but not deeply or often.**

Exactly. You think about complexity and you think about efficiency and you measure it, but Big O analysis as an interview tool is only useful to measure how much attention you paid in your second year of college. It's one of my pet peeves. It's a symptom of a disease.

## **Nobody Puts COBOL in the Corner**

---

**As I understand it, you're not currently a permanent employee with the federal government. How would you describe your position?**

My position right now is a termed position. Technically, in eight months, I am supposed to go back to the private sector and move on with my life. I probably won't because this is what I want to do with my life. So I am going to fight to find a way to stay beyond my term. For the government, every change requires years.

As an example, take my current effort to replace a specific mainframe overseeing the benefits and livelihoods of two and a half million people. Getting away from that system will take more than five years, because of the amount of technical debt that has to be paid.

*“To create sustainable change, we need angry people in the same chair for more than two years.”*

We have led a successful acquisition of a mainframe that took three months and we're very proud of that. Usually it takes more like seven months to finish an acquisition process in government. We bought a new mainframe, which is a horrible thing to say, but that was the right thing to do because the old system was really about six, seven years past the end of life, and we needed five years to pay the technical debt to fix the system. So, we had to buy a new one and we did it in three months.

That's what I think makes the GSA's Centers of Excellence (CoE) so special; because the CoE is a part of the federal acquisition service, we can cut that time down to get a new system. But there is value in people like me, people who are angry at the way

government works, sticking around for five years, for ten years to see these things through.

Right now, government is shortcutting the hiring process with termed employees like me who have short two-year stints to try to make change. I don't think that's effective because you need someone to stay and see it through. Especially for an organization like the federal government where, every four years, regardless of what happens from the mission perspective, leadership goes away. Targets change. To create sustainable change, we need angry people in the same chair for more than two years.

**What is a termed position? And what are the other positions available in government, whether salary or contract?**

The federal government has four positions that I'm aware of. You have general service (GS) employees, who are salaried employees who work for the federal government. They're somewhere between GS-1 and GS-15, which is a salary scale, and they are competitive service positions. Essentially, that means they have the ability to apply to any position within government, compete for it, and get it as a long-term position.

Then there are people like me, termed service. To make my position valuable, they let me slide into GS-15 step 8, which is two steps below the maximum salary of GS-15 step 10. They try to match our salaries from the private sector. They couldn't—they could only match two thirds of what you make in fintech in New York. But they tried, which I appreciate. I was completely aware that I was going to take a salary hit by switching.

In my case, the termed position is two plus two, meaning I have a two-year term and would be up for renewal for another two years. They can fire me within the first three months and they

can decide whether or not to renew my term at the end of the first two years. Terms can be anywhere from a year to eight years.

Termed employees are essentially civilians—I can't apply for other competitive positions outside of my current role. When I'm done with this role, I have to apply again to a GS position and all of my government experience doesn't count.

**When you say it doesn't count, do you mean toward the qualifications for other positions or for benefits?**

All of the above. That experience essentially doesn't exist. If I apply and get into a permanent position that is open to the public, which is some small percentage of the perm job positions, it could contribute to the pension calculation, if and when I retire. But that's about it.

*“This is kind of the sad thing about technology – the more you look into it, the more you understand how little the technology means.”*

It also doesn't count for experience for other positions. My position right now is nonsupervisory. For accessing the third kind of position, you have to have a year of supervisory experience. I can manage as many people as I want right now, it doesn't count. I have to start from scratch from that perspective.

The third kind of position is Senior Executive Service, or SES. Most people with senior titles—CIO, CEO, directors—tend to be SES. There is a subset of management criteria that you have to

reach to become SES, and you also have to have the supervisory experience as a competitive employee that I mentioned earlier.

My goal in life now if I'm going to state government is to become SES. But I can't even start working towards that until I am rehired. Practically speaking, from a career perspective I may as well be a civilian right now.

### **Why is your goal to be SES?**

Because whichever system I'm in, I require myself to accomplish the highest level possible. Right? Also, because that's where change is. I'm trying to effect change.

The more you spend time in technology, the more you understand that technology is not the problem. I can build the shiniest system in the world, but it doesn't mean that anyone will use it. Change is always about people, and people are controlled by management.

For example, if I want to get an agency to start working in a different way, I have to be near enough to the top to be able to make decisions that affect how people function. Being at the top doesn't make it easy, but it gives you the option of trying to accomplish it. This is kind of the sad thing about technology—the more you look into it, the more you understand how little the technology means.

### **We haven't yet talked about contractors. From your perspective, what is the role of contractors within these teams and systems? And why does government often procure or contract technology versus building in-house?**

That's a really complicated question. I mentioned there are four classifications, right? Termed, GS, SES—and contractors are the fourth one.

It takes me somewhere between two to three months to get a contractor in the door, whereas a new hire takes about eight months to a year. So as a product manager, if I'm trying to get something done in government, contractors are usually my go-to solution.

The other piece here is that for political reasons or other reasons, hiring freezes are a thing in government. A bunch of agencies have had eight to ten years of hiring freezes and, as a result, have basically gotten to the point where only 20 percent of the workforce are federal employees. Because they can't hire, they just buy more contracting because there's no monitoring of that. That is actually fairly common.

As an example of one of these freezes, say a lot of the main-frame support people across government started to work as federal employees, but then the federal government can't hire any new ones. The federal government will go out and buy contractor support to help maintain this legacy system, and the contractor support will turn to the salaried employee and say, "we'll pay you 150 percent of your salary and give you better benefits if you come work for us." Federal employees in those positions get to essentially keep the same job, but have better quality of life as a contractor.

The law says that federal contractors can't manage contractors, so a federal employee has to be the contracting officer representative who manages that contract. So you have people whose job used to be technologist or system administrators, but now they are federal employees who are middle management and as a result all they do is manage those acquisition contracts. We've seen this pattern a lot.

**How does that impact the maintainability of software?**

Theoretically, it should improve it because every single person is replaceable. Practically, it presents interesting challenges.

For example, VB6 was decommissioned in 2012, I think, but there are many VB6 systems across government because VB6 presents a fairly low barrier to entry for development. There's a whole bunch of these lying around.

Say I have a contract with the vendor and that contract says I need VB6 experts to manage these VB6 systems. Awesome. The contractor gets to move those experts around, and they're supposed to be interchangeable. But what you learn from *managing* technology as opposed to *doing* technology is that knowledge management is really hard to do. Organizational knowledge dissipates even if we get a new VB6 expert, because that expert has never seen *this* system and the person who was an expert in *this* system gets moved to another project.

Documentation is hard — and if it doesn't exist, I end up paying the same contracting company over and over to learn the business logic and reasoning behind a specific system. That's the flaw of this approach.

**To your point about people, COBOL and mainframe systems have been in the news lately as state unemployment applications struggle under the historic load of applicants from the pandemic. A lot of folks are demonizing the languages, but my opinion is that it's not just about the language or the technical systems — there's an important social component, too. What is your take on that?**

I actually got into a very angry discussion with a few of the COBOL defenders. I described what happens to COBOL in its lifecycle. Every language has a lifecycle for large systems. COBOL has a very specific one.

Mainframe systems have one primary benefit, which is a horrible trap: you can set the system up, leave it alone, and it will keep running, forever. What you get is forty-year-old codebases that keep doing their job and management doesn't bother paying for it. Why would you? I can get away with paying less money and the system will keep producing, so I don't need to bother maintaining it. But then you have code that was written forty years ago by people who have since retired.

*“I’m not saying that COBOL is a bad language. I’m saying that it’s so good that it gets abandoned over and over again.”*

When the time comes to make a change, you see one of the following. I'm looking at a system that had 900,000 lines of code. It was obviously not man-made. Who made it? Multiple times in the past, IBM introduced new hardware. To make the migration to the new hardware easier, they built tools that would automatically rewrite code. So I have a piece of code that was a hundred thousand lines long to begin with that has now been rewritten and rewritten by other pieces of code. It works, but now there's no documentation to speak of. There is no understanding of which piece of the code does what. And what I have is people — and they're really good people — who've been maintaining it from the edges, writing around pieces of code that are too big to change like I did back in 2001. But the system is still there. And so you have massive systems that are completely monolithic. There is no understanding which piece

of the code is what, or what I can do to kill it or make it better. I have no way to maintain it.

And then something happens like COVID-19. Suddenly my system that can handle  $x$  amount of calls per day needs to handle  $400x$ , and I have no idea how to approach it because that is part of the underlying system itself. It's not an edge problem.

I'm not saying that COBOL is a bad language. I'm saying that it's so good that it gets abandoned over and over again.

## May the Pipeline Be Unbroken

---

**One of the things we've struggled with in this series is finding a diverse set of people who have held technical roles with mainframe systems for a long time – and are willing to speak with us. So, given the breadth of your career and where you are now, I'm curious if that lack of diversity matches your experience or if it's just a problem with our sourcing?**

I think it's an instance of the fake "pipeline problem" for people who are not white men into technology roles. I call the pipeline problem "fake" because I don't think that's the actual problem. General technology in GSA has the largest number of queer people, the largest number of women I have ever worked with. For people who shoulder the joys of unpaid work as part of their lives, government does what is actually necessary to make it feasible for those people to stay at their jobs.

For people still maintaining these systems, though, you're looking at the pipeline from thirty to forty years ago. There are new people who are pouring into the larger pool of workers, but the mainframe teams haven't changed. That's the way it was forty years ago and that's the way it is now, because these are the same people.

**What we've heard from people maintaining these systems on under-resourced teams is that they've been basically expected to be on call for their entire career. One of the things we've been trying to understand is: What impact does it have on their personal life? What you are describing gives some color to that, as well.**

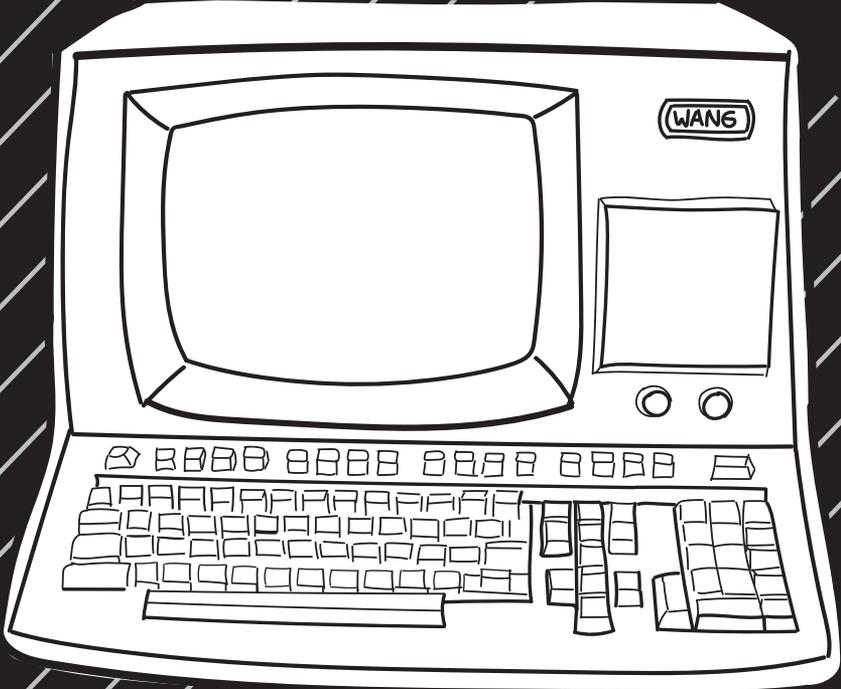
Ah, that question brings me back to the joys of working in fin-tech. Foreign exchange is open 24/7. The market opens Sunday at 4:00 p.m. and closes on Friday at 4:00 p.m. and is open 24/5 in the middle. This means that if you're in charge of the system, you're on call during the week because you have to be there if things break, and the weekend is when you do all your maintenance. So, you just work constantly.

In terms of the impact on people's personal lives: you really shouldn't talk to me, you should talk to my partner and ask him. The fact that I can't make plans, the fact that I have to walk out of dinner with his parents and go deal with work calls, I'm sure that has an impact. And yeah, people working maintaining government systems have been doing that sort of on-call work forever. That is what they do.

By the way, these people that are on call all the time, the same people who weren't willing to talk to you when you asked for an interview: part of the reason they refused is that they are still GS-12s and 13s. They're low in the hierarchy, and yet so essential. Let's say you publish something and it somehow comes out as negative about GSA. In my position, I would get a slap on the wrist. If they said anything that they shouldn't say about their agency however, then they're done. That is their career. So, I have a kind of privilege; I'm coming at this from a position where I can talk to you and I'm not risking that much.

These people who've been here for forty years — and they're completely essential — know for a fact that they'll be railroaded if anything happens.

I'm sorry, I'm not trying to guilt you. I'm trying to expose the dynamic. 



## MAINTENANCE WINDOW

# Jed Wagner

## On Single-Handedly Building and Maintaining the Veterans Appeals System

The Veterans Appeals Control and Locator System, or VACOLS, starts and ends with Jed Wagner. In the late 1980s, Wagner was hired as a part-time contractor by the Department of Veterans Affairs. He started single-handedly building the system that he was ultimately hired to work on full-time. Now, thirty years later, VACOLS is processing its last appeals, and when when the system is sunsetted, he will retire.

Originally, the system's primary job was to know the locations of the physical files of the 30,000 or so veterans who appealed their benefits decisions. Over time, VACOLS grew and became more complex as Wagner iterated and added modules like courtroom scheduling and video hearings. He would gather requirements directly from judicial review officers, judges, and administrators, deploy a prototype, get feedback, and deploy again – years before Agile became a popular approach to software development.

The Veterans Appeals Improvement and Modernization Act of 2017 created infrastructure and funding for a new appeals system that hopes to resolve appeals in six months as

opposed to three to seven years. In a pivot away from the old paradigm where one long-term employee spends their career caring for the system they built themselves, the new appeals system will be overseen by a termed employee and then handed off to the next termed employee after that.

Christa and Julie sat down with Jed to talk what he built, and what's next.

---

**Tell us about your job.**

My official title is Computer Specialist. I've been the lead developer on a team of one for over thirty years. I created and maintain a computer system called VACOLS, the Veterans Appeals Control and Locator System, which manages the appeals process for the US Department of Veterans Affairs.

**That's an interesting title, "computer specialist." Why is that your title as opposed to "software engineer"?**

When I first got started, that was what we were: specialists. We did everything from hardware to running cables to configuring and maintaining switches and routers. We maintain our own Exchange servers, which are Microsoft systems used for email and calendaring. So when I started at the VA, I was involved with all of that. Then, in 1986 or 1987, I started developing VACOLS as a side project. Eventually, more and more people started using the application, and it became more and more my primary task.

I don't know if the government has "software engineers," I imagine they do. But "specialist" was my official title thirty years ago, and that's what it still is today.

**You started out as a contract worker at the VA. What was that like?**

I was on a full-time contract at the time with the State Department's diplomatic security unit, and they were ten blocks away from the VA. At lunchtime, I'd run over to the VA, do my VACOLS stuff, and then head back to my regular job after that.

The VA had me working in the computer room, which was about sixty-two degrees. Remember, this was pre-PC so we had what were called dumb terminals: a keyboard and screen all attached directly to the server with a coax cable. No processor or anything. At the VA, the coax cable ran back to these Wang minicomputers. I was a team of one back then, too, so it was just me in there.

There was a lot of learning to do. It's fairly complicated, that whole appeals process. A lot of different legal regulations, things you have to follow. So it was a bit of a challenge in figuring out everything that goes on there.

**Can you tell us what an appeal is and why someone might go through the appeal process?**

The majority of appeals are for benefits. Let's say there's a vet who returns from service and he has a knee condition that's rated as 30 percent disability. As time goes on, he might get worse, and he can appeal his original decision and say, hey this is a lot worse. I'd like to get 50 percent, or 100 percent, or whatever. So those are called increased rating appeals, which is probably 50 percent of what we do.

The other appeals are for service-connected compensation. Someone says, "I have PTSD resulting from my service." He goes to his local office, puts in a PTSD claim, and it gets denied. He can say, "I want to appeal my original grading decision and appeal to the Board of Veterans' Appeals." At that point, he can have a hearing before a judge at the Board, and the Board will issue a second decision either affirming the original or

overturning and granting his request for benefits or to increase his rating. Basically, it's an appellate court.

**I don't think that I had fully understood that appeals in this context could be a case of a disability worsening over time.**

At any point he can appeal his initial rating up to 100 percent disability if that's warranted. But the appeal also involves presenting new evidence, new exams and everything. A judge reviews all the new evidence and has a hearing face-to-face or via video with the appellate and then writes a decision.

**So a veteran continues interacting with this system over the course of their life?**

Not every decision that the VA makes is appealed. Only about 10 percent of the original decisions made by the local offices get appealed to the Board. In 90 percent of the cases, the veteran is either satisfied with the decision or decides not to pursue the claim further.

*“He could be rated 30 percent for an elbow injury, 10 percent for a knee injury, 30 percent for the head injury.”*

**How many appeals does the current system process per year?**

Last year, they decided about 90,000 legacy appeals. When the Board was 400 people, they'd do about 30,000 a year. Now that they're 1,200 people, they can process about 90,000 a year.

### **What is the increase from?**

More people, more judges, more attorneys. Over the last twenty years, the board has tripled in size. The more people they have drafting decisions, the more judges they have holding hearings, the more cases they can get out the door. Basically just increased manpower.

### **And how does the system that you built fit into the appeals process?**

It tracks all the elements of the appeal. If a veteran was in an accident, he could've injured his knee, his elbows, his head—so there could be different issues on appeal. He could be rated 30 percent for an elbow injury, 10 percent for a knee injury, 30 percent for the head injury. Typically, each appeal has an average of three issues. VACOLS keeps track of all the issues he's appealing, keeps track of all the forms he has to submit during the process.

Like for the first step, he's going to file a notice of disagreement to indicate that he disagreed with the VA's original decision. And then there's four or five other forms to fill out during the process, depending on how far it progresses. And all those forms are kept in the database.

Then, if he requests a hearing, it keeps track of that request and the dates, as well as any mail or evidence we get in correspondence that they sent to the VA. That gets loaded into VACOLS and tracked.

The appeal will start out at the regional office and it'll go to the decision review officer, then it might go to the appellant's rep at the Veterans of Foreign Wars, American Legion, whoever his rep is. The same information will go to an attorney and a judge, so there might be ten or fifteen people touching the appeal at any given point. VACOLS keeps track.

## **What was the process before they decided to build a technology system to handle it?**

Paper. Up until about five years ago, all the claims files with all their service records and military records and everything were in these big twenty-four-inch-thick folders.

For the first twenty-five years of VACOLS, one of the main things it tracked was exactly where these big claims folders were — who physically had them in their possession. Was it with a judge, was it with an attorney, was it out in the regional office, was it with a veteran service organization? We used barcode technology to track all the claims folders. We had thousands of these moving through the building.

So that was the main purpose of VACOLS to begin with: let's track where these folders are and who has them and the outcome of the appeal. Since then, we've added on module after module to schedule hearings, hold virtual hearings, and do all sorts of other things that are part of the appeals process.

Originally, the system kept track of a physical folder. Now, it's keeping track of virtually who has the claim and what the status is and what part of the process it's in.

### **That sounds complicated.**

Yeah, and I've only touched on a few of the areas. There's a lot of information it keeps track of.

## **Punch Out**

### **What originally led you to computers?**

I graduated from Penn State back in the '70s. I got a business degree coming out of school. I didn't really take any IT classes.

There wasn't really any offered back in the '70s. I got my first job with a contractor here in the DC area and the guy that hired me worked for a small business and he was looking for programmers, but he really wanted non-technical people. He hired about a dozen people, with different backgrounds, and then he taught us how to program. My first programming job was working on COBOL applications for large naval contracts.

*“Up until about five years ago, all the claims files with all their service records and military records and everything were in these big twenty-four-inch-thick folders.”*

**Did you have any experience with computers before you landed that job?**

Not really. I took one computer course in college and we had to write our programs on punch cards, go to the computer lab at night, and wait in line for two hours to submit the big deck of punch cards. Then it would compile that as your program, and you'd find out that you missed a comma, so you'd have to go correct that and stand in line again for two hours. That was my initial exposure to programming: writing COBOL programs on punch cards.

**Do you remember how you felt about it at the time?**

I felt like it was not a field I wanted to get into.

**So what led you to take that first job?**

School loans, rent, all the usual things. I'd had a lot of blue-collar jobs, worked in factories, and I knew I didn't wanna go back to small-town Pennsylvania working the factories again. So it wasn't as much a matter of falling in love with programming; it was a matter of avoiding the alternatives. The job market was tight back in the early '80s, so if you got an offer coming out of school, you jumped on it.

**At that time, did you imagine that you would do different things with your career afterwards?**

Yeah, but I showed an aptitude. I was pretty good at programming, and after a few years I realized it was a growing market, so it looked like a good field to get into.

There still weren't PCs and all that at that point, but if you had strong COBOL skills back in the '70s, early '80s, there were a lot of job opportunities. So I bounced around on a number of different contracts. One of them happened to be with the VA.

**How did that first contract at the VA end up turning into your career?**

After I developed VACOLS, I maintained it part-time for about five years. Then, in the early '90s, they wanted to migrate the system off their old Wang computers and onto PCs and a relational database. That's when they asked me if I wanted to come on board and take on the project. I did that migration, and then we started developing with a rapid application development tool called PowerBuilder, which was big back in the early '90s. And that's been the formula for the last twenty-five years. We've used Oracle as a database and PowerBuilder as the window into that database, to create a client-server solution.

**Have you ever thought about leaving for another job or another opportunity?**

Yeah, I've thought about it. But I feel a responsibility. I've always been loyal. I've always believed strongly in the VA's mission. My dad was a veteran. I have uncles that are veterans. And my dad, he's in his eighties, and still goes to the VA every month. He loves the VA. You only hear the bad stories, but there are lots of people out there like my dad. They love the care and the attention. He goes there and they give him stuff he never even asks for. He raves about the care he gets at the VA. I've always felt a loyalty and a strong commitment to the mission of the VA. So, yeah, I've thought about other jobs, but the VA treats you well.

*“I knew I didn't wanna go back to small-town Pennsylvania working the factories again.”*

## Go It Alone

**When you were first starting to build out the system, were there other good technical models you looked to, or were you just creating this thing from scratch?**

No, I was winging it. I picked up PowerBuilder off the shelf and learned it on my own. It was a great tool back in the '90s, although it's since lost market share. Sybase eventually purchased it, and I think SAP owns it now. There's a few people still using it. At the time, it was really easy to pick up and learn.

I was working directly with the end-user base to get all the user requirements and everything—the attorneys and the judges and the admin people that held hearings and docketed the appeals cases.

We did an Agile-type development back then before Agile was the big thing. We would just go in and get the requirements and knock them out. We'd try to put something in production in a couple weeks like they do today, move on, and then add onto it a couple of weeks later.

It was a good place to work in that regard because you're working without any constraints. Being a development team of one and working directly with the end users and being there in the same office — you could really get a lot done fast.

**That's interesting to hear, because we've talked with a lot of government folks who are trying to move away from Waterfall towards Agile.**

Yeah, we were ahead of our time, but it worked out.

**Were you still working on things other than VACOLS?**

I not only did the programming, but we did all the Windows upgrades, as well as the wiring and cabling. I'd do programming on a Friday, then come in Saturday and run cables through the ceiling. You had to do everything back then.

Every shop — the Veterans Benefits Association, Veterans Health Administration, the Board of Veterans' Affairs — all had their own IT guys, and as an IT guy you did pretty much what you wanted in terms of technology. Like, the Board was running Windows NT, while all the other organizations within the VA chose to go Windows 95, Windows 98. So even the version of Windows used within the VA was not standardized at the time. We had a lot of freedom as far as what type of servers we wanted to use.

Today, everything's more consolidated and organized and everybody's on the same platforms. But back then, it was the Wild

West. You did what you wanted, and as long as it worked, nobody complained.

**What were your responsibilities to that system? If it were to go down, were you on call?**

Yeah, I've always been the primary point of contact if we ever have server issues or database issues.

**Has that been difficult at all?**

Not as long as you keep your cell phone on-hand. I always take short break vacations—don't want to be away too long.

**What about before you had a cell phone?**

They had my home number back then. There is always a way to find me if they need me. We had pagers back in the '80s. I think I still have one in a drawer here somewhere.

*“I'd do programming on a Friday, come in Saturday and run cables through the ceiling. You had to do everything back then.”*

**Was it daunting to be working on that system alone for decades? It sounds like a lot to handle.**

Yeah. It started out being a system just for the Board of Veterans' Appeals, which was 400 people at the time, and now it's grown to over 17,000 users as we pushed it out—VBA, VHA, and all these other organizations. As we developed more applications and interfaces for more kinds of users, it became more and more my baby. It became a full-time job—I painted myself into a corner.

**Who are those users? Who do they work for? You said a couple of government agencies, but I'm curious what all those acronyms mean.**

All the appeals start at the VBA, Veterans Benefits Administration, and then they may move to the Board, but not necessarily. About 14,000 of VACOLS users work for VBA. The appellant could decide to withdraw before the board sees it or the VBA can decide, hey, we made a mistake when we originally denied this, and they can grant benefits before it gets to the board. So a lot of things go on before an appeal ever gets to the Board. The majority of VA people using VACOLS are VBA.

Then we have the Board, which is about 1,200 people now. It's more or less tripled in size. I also got about a thousand veterans service reps — American Legion, VFW, Disabled American Vets, Paralyzed Veterans of America — about thirty, forty different organizations that we call service organizations that have access to VACOLS. And then VHA is the fourth group. We don't get a lot of medical appeals; there's probably 500 or so VHA users. If you combine the VBA, BVA, VHA, and the veterans service orgs then you come up with roughly 17,000 users.

**When you were working on VACOLS, was there ever a point when you said, "I actually need more people on this"?**

Not really. It was client-server, so deployment was easy. It's not like a lot of web applications today where you need a lot of people to maintain them. If it worked, it worked; you threw it out on the server and everybody accessed it from the same point. And, knock on wood, we haven't had much downtime. We've been using Oracle databases for thirty years and they never go down. The hardest thing was that, a couple years ago, they decided that all VA applications had to have two-factor authentication. We learned that 17,000 users hadn't been

logging in correctly and wouldn't be able to get into their own accounts, so I had to quickly create a new authentication database for those 17,000 users. Other than that, things went pretty smoothly the last few years.

### **Do you think of yourself as a craftsperson?**

Yeah, I guess. It's nice to sit back and look at what you've created. Right now, VACOLS is a suite of twenty different applications. There's one for scheduling hearings, HR has their own application, quality review has an application, the intake team has their own application. The dispatch team has an application. They're separate versions, VACOLS for the VSOs (Veterans Service Organizations), those service reps, and for the regional offices. So it's a lot like the VA's Caseflow software, how they're developing different modules, sort of copying what we did. Just like there's a VACOLS dispatch program, there's a Caseflow dispatch program. Once they get all their apps finished, I'm looking at retirement here soon.

## **The Last Appeals**

**We recently went through a project where we were doing research on how people get Supplemental Security Income and disability insurance. Getting evidence for a claim in those cases is often an insurmountable obstacle for people. The appeals process that VACOLS handles seems even more complicated than those.**

That's why they came up with this new Appeals Modernization Act, where they're revamping the whole appeal process to make it a lot simpler and faster. The problem with the old appeals system was that it was an open-ended process. At any point in the process, the guy could submit new evidence, new exams, so things were constantly changing. If he filed a claim two years

ago, the records could be stale by the time it reached the judge. He might have to go out and get new exams. The new process is a lot tighter. You submit all the evidence up front and then you're locked in. They try to review in a more timely manner, within six months or so, so you don't have all these issues with stale records. They've streamlined it.

**Has that changed over time? Now that you're working with the United States Digital Service, I'm curious how that has changed your development process.**

No, it's still basically the same thing. There are less requirements now that VACOLS is getting ready to be sunset, but there are some new things they're doing. Within the last couple of months, they've introduced virtual hearings where the veteran can have a hearing with a judge from his home, just like we're doing here with a Skype or a Zoom meeting. We have to start tracking those for the legacy appeals. I'll have to make some modifications just to indicate that these are virtual hearings, as opposed to face-to-face or video hearings held in the office. So there's still a few new requirements that I have to put into VACOLS. Most of the new development is in the Caseflow system. I used to work closely with those guys, showing them all the requirements and what VACOLS did, and what they'd have to account for.

**What was the most surprising thing about working on that project with them?**

How young they all were. They were all younger than my kids, so that took some getting used to it. But they were all great to work with, really sharp. They came in and got things done quickly. I mean, that whole digital service thing was thrown together by Obama in his last year. The VA was one of the first organizational digital services and so far it's worked out pretty good.

**When the discussion started on your end to start replacing that system, did you indicate you wanted to retire and they were like, wait, we've got to replace this thing – or how did that conversation come about?**

It wasn't so much replacing VACOLS as replacing the appeals process with the Appeals Modernization Act to make it faster and simpler. They were gonna have to develop a new system for that anyway, so it just made sense. VACOLS had run its course. There's going to be legacy appeals for another couple of years, but VACOLS can be sunset after that.

**How does that feel for you?**

Sounds good to me! Sounds good to my wife. She's got a lot of places she wants to go. Get to do a little traveling, do things that you put off all these years. So long as none of the kids move back home, we'll be fine.

**That's exciting! What is the timeline for that?**

Maybe two years. The legacy appeals, there's probably 200,000 left. And they do about 100,000 a year. They could get through most of them in the next couple of years.

**One of the reasons that we're doing this series now is because COBOL has been in the news lately with New Jersey making a call for COBOL programmers to help with their unemployment systems struggling under the load of applicants impacted by COVID-19. I think a lot of folks are realizing for the first time that COBOL systems are still a part of government technology. I was curious if you've heard of this happening, and what your thoughts are about that and the criticisms of these legacy mainframe systems?**

I'm surprised they're still out there. I haven't been following the news that closely, but I wasn't aware that New Jersey was having

an issue with them. I mean, there are tons of legacy systems within VA. They have their own programming system, VistA, and they have something like 180 different VistA applications that are used in the VA hospitals and medical centers. So the Board is just a minute little part of the VA. There are a lot of legacy programmers at the VA that work on these VistA systems that have been there longer than I have, and they've got the same situation where they've now painted themselves into a corner. They're the expert on the system and probably the only person who knows how to maintain them. Nobody's coming into the job market these days saying, "I want to be a VistA programmer or a COBOL programmer or a PowerBuilder programmer." But somebody has to do it.

I know the VA still has what they call stovepipe systems, which are basically standalone systems that don't interface with anything else. That was VACOLS for a while, although now we've integrated with Caseflow and other systems. But I'm sure there's still a lot of these old stovepipe systems out there. They're probably running in some background that nobody's even aware of. ~~~



```

AGY0155. DEMO. SRCLIB (PROGRAM
==>
***** Top of
1 IDENTIFICATION DIVISION.
2 PROGRAM-ID. QUASAR.
3 x
4 ENVIRONMENT DIVISION.
5 IDENTIFICATION SECTION.
6 COMPUTER. DELL.
7 DELI

```

# Built to Last

by Mar Hicks

When overwhelmed unemployment insurance systems malfunctioned during the pandemic, governments blamed the sixty-year-old programming language COBOL. But what really failed?

---

At the time of this writing, in July 2020, the COVID-19 pandemic has killed over 133,000 people in the United States. The dead are disproportionately Black and Latinx people and those who were unable, or not allowed by their employers, to work remotely. During the pandemic, we've seen our technological infrastructures assume ever more importance—from the communications technology that allows people with the means and privilege to telecommute, to the platforms that amplify public health information or deadly, politicized misinformation. We've also seen some of the infrastructure that runs the social safety net break down under an increasing load. This includes state unemployment systems that pay workers the benefits they've contributed to for decades through taxes. In a global pandemic, being able to work from home, to quit and live on savings, or to be laid off and

draw unemployment benefits has literally become a matter of life and death.

The cracks in our technological infrastructure became painfully evident in the spring, as US corporations responded to the pandemic by laying off more and more workers. So many people had to file for unemployment at once that computerized unemployment claim systems started to malfunction. Around the country, phone lines jammed, websites crashed, and millions of people faced the possibility of not being able to pay for rent, medicine, or food.

*“For many reasons, COBOL was an easy scapegoat in this crisis – but in reality what failed wasn’t the technology at all.”*

As the catastrophe unfolded, several state governments blamed it on aged, supposedly obsolete computer systems written in COBOL, a programming language that originated in the late 1950s. At least a dozen state unemployment systems still run on this sixty-one-year-old language, including ones that help administer funds of a billion dollars or more in California, Colorado, and New Jersey. When the deluge of unemployment claims hit, the havoc it seemed to wreak on COBOL systems was so widespread that many states apparently didn’t have enough programmers to repair the damage; the governor of New Jersey even publicly pleaded for the help of volunteers who knew the language.

But then something strange happened. When scores of COBOL programmers rushed to offer their services, the state governments blaming COBOL didn't accept the help. In fact, it turned out the states didn't really need it to begin with. For many reasons, COBOL had been an easy scapegoat in this crisis—but in reality what had failed wasn't the technology at all.

## A “Dead” Language is Born

---

One of the most remarkable things about the unemployment claims malfunction wasn't that things might suddenly go terribly wrong with COBOL systems, but that they never had before. Many computerized government and business processes around the world still run on and are actively written in COBOL—from the programs that reconcile almost every credit card transaction around the globe to the ones that administer a disability benefits system serving roughly ten million US veterans. The language remains so important that IBM's latest, fastest “Z” series of mainframes, have COBOL support as a key feature.

For six decades, systems written in COBOL have proven highly robust—which is exactly what they were designed to be. COBOL was conceived in 1959, when a computer scientist named Mary Hawes, who worked in the electronics division of the business equipment manufacturer Burroughs, called for a meeting of computer professionals at the University of Pennsylvania. Hawes wanted to bring industry, government, and academic experts together to design a programming language for basic business functions, especially finance and accounting, that was easily adaptable to the needs of different organizations and portable between mainframes manufactured by different computer companies.

The group that Hawes convened evolved into a body called CODASYL, the Conference on Data Systems Languages, which included computer scientists from the major computing hardware manufacturers of the time, as well as the government and the military. CODASYL set out to design a programming language that would be easier to use, easier to read, and easier to maintain than any other programming language then in existence.

The committee's central insight was to design the language using terms from plain English, in a way that was more self-explanatory than other languages. With COBOL, which stands for "Common Business-Oriented Language," the goal was to make the code so readable that the program itself would document how it worked, allowing programmers to understand and maintain the code more easily.

*“They set out to design a programming language that would be easier to use, easier to read, and easier to maintain than any other programming language then in existence.”*

COBOL is a “problem-oriented” language, whose structure was designed around the goals of business and administrative users, instead of being maximally flexible for the kind of complex mathematical problems that scientific users would need to solve. The main architect of COBOL, Jean Sammet, then a researcher at Sylvania Electric and later a manager at IBM, wrote,

“It was certainly intended (and expected) that the language could be used by novice programmers and read by management.” (Although the pioneering computer scientist Grace Hopper has often been referred to as the “mother of COBOL,” her involvement in developing the specification for the language was minimal; Sammet is the one who really deserves the title.)

Other early high-level programming languages, such as FORTRAN, a language for performing complex mathematical functions, used idiosyncratic abbreviations and mathematical symbols that could be difficult to understand if you weren't a seasoned user of the language. For example, while a FORTRAN program would require programmers to know mathematical formula notation, and write commands like:

```
TOTAL = REAL(NINT(EARN * TAX * 100.0))/100.0
```

users of COBOL could write the same command as:

```
MULTIPLY EARNINGS BY TAXRATE GIVING SOCIAL-SECUR  
ROUNDED.
```

As you can tell from the COBOL version, but probably not from the FORTRAN version, this line of code is a (simplified) example of how both languages could compute a social security payment and round the total to the penny. Because it was designed not just to be written but also to be read, COBOL would make computerized business processes more legible, both for the original programmers and managers, and for those who maintained these systems long afterwards.

A portable, easier-to-use programming language was a revolutionary idea for its time, and prefigured many of the programming languages that came after. Yet COBOL was almost pronounced dead even before it was born. In 1960, the year that

the language's specification was published, a member of the CODASYL committee named Howard Bromberg commissioned a little "COBOL tombstone" as a practical joke. Bromberg, a manager at the electronics company RCA who had earlier worked with Grace Hopper on her FLOW-MATIC language, was concerned that by the time everybody finally got done designing COBOL, competitors working on proprietary languages would have swept the market, locking users into programming languages that might only run well on one manufacturer's line of machines.



**COBOL tombstone, copyright Mark Richards.  
Courtesy of the Computer History Museum in Mountain View, CA.**

But when COBOL came out in 1960, less than a year after Mary Hawes's initial call to action, it was by no means dead on arrival. The earliest demonstrations of COBOL showed the language could truly be universal across hardware. "The significance of this," Sammet wrote, with characteristic understatement, was that it meant "compatibility could really be achieved." Suddenly, computer users had a full-featured cross-platform programming

language of far greater power than anything that came before. COBOL was a runaway success. By 1970 it was the most widely used programming language in the world.

## Scapegoats and Gatekeepers

---

Over the subsequent decades, billions and billions of lines of COBOL code were written, many of which are still running within financial institutions and government agencies today. The language has been continually improved and given new features. And yet, COBOL has been derided by many within the computer science field as a weak or simplistic language. Though couched in technical terms, these criticisms have drawn on a deeper source: the culture and gender dynamics of early computer programming.

During the 1960s, as computer programming increasingly came to be regarded as a science, more and more men flooded into what had previously been a field dominated by women. Many of these men fancied themselves to be a cut above the programmers who came before, and they often perceived COBOL as inferior and unattractive, in part because it did not require abstruse knowledge of underlying computer hardware or a computer science qualification. Arguments about which languages and programming techniques were “best” were part of the field’s growing pains as new practitioners tried to prove their worth and professionalize what had been seen until the 1960s as rote, unintellectual, feminized work. Consciously or not, the last thing many male computer scientists entering the field wanted was to make the field easier to enter or code easier to read, which might undermine their claims to professional and “scientific” expertise.

At first, however, the men needed help. Looking back, we see many examples of women teaching men how to program, before



**Programmer Juliet Muro Oeffinger teaching students  
Gerald Griepenstroh, Ron Holander, Von Hale, and Ed Tombaugh  
how to program in COBOL c. 1967.  
(Photo from the personal collection of Juliet Muro Oeffinger.)**

women gradually receded from positions of prominence in the field. Juliet Muro Oeffinger, one of about a dozen programmers I interviewed for this piece, began programming in assembly language in 1964 after graduating college with a BA in math. “When COBOL became the next hot thing,” she said, “I learned COBOL and taught it for Honeywell Computer Systems as a Customer Education Rep.” In the images to the left, Oeffinger teaches a room full of men at the Southern Indiana Gas and Electric Company how to program in the language. Within a short time, these trainees — who had no prior experience with computer work of any kind — would have been programming in COBOL.

Another retired programmer I spoke to named Pam Foltz noted that good COBOL was great long-term infrastructure, because it was so transparent. Almost anyone with a decent grasp of programming could come into a COBOL system built by someone else decades earlier and understand how the code worked. Foltz had a long career as a programmer for financial institutions, retraining in COBOL soon after getting her BA in American studies from the University of North Carolina at Chapel Hill in the 1960s. Perhaps this dual background is one reason why her code was so readable; as one of her supervisors admiringly told her, “You write COBOL like a novel! Anyone could follow your code.”

Ironically, this accessibility is one of the reasons that COBOL was denigrated. It is not simply that the language is old; so are many infrastructural programming languages. Take the C programming language: it was created in 1972, but as one of the current COBOL programmers I interviewed pointed out, nobody makes fun of it or calls it an “old dead language” the way people do with COBOL. Many interviewees noted that knowing COBOL is in fact a useful present-day skill that’s still taught in many community college computer science courses in the US, and many colleges around the world.

But despite this, there's a cottage industry devoted to making fun of COBOL precisely for its strengths. COBOL's qualities of being relatively self-documenting, having a short onboarding period (though a long path to becoming an expert), and having been originally designed by committee for big, unglamorous, infrastructural business systems all count against it. So does the fact that it did not come out of a research-oriented context, like languages such as C, ALGOL, or FORTRAN.

In a broader sense, hating COBOL was — and is — part of a struggle between consolidating and protecting computer programmers' professional prestige on the one hand, and making programming less opaque and more accessible on the other. There's an old joke among programmers: "If it was hard to write, it should be hard to read." In other words, if your code is easy to understand, maybe you and your skills aren't all that unique or valuable. If management thinks the tools you use and the code you write could be easily learned by anyone, you are eminently replaceable.

*“Gatekeeping is not only done by people and institutions; it's written into programming languages themselves.”*

The fear of this existential threat to computing expertise has become so ingrained in the field that many people don't even see the preference for complex languages for what it is: an attempt to protect one's status by favoring tools that gate-keep rather than those that assist newcomers. As one contemporary

programmer, who works mainly in C++ and Java at IBM, told me, “Every new programming language that comes out that makes things simpler in some way is usually made fun of by some contingent of existing programmers as making programming too easy—or they say it’s not a ‘real language.’” Because Java, for example, included automatic memory management, it was seen as a less robust language, and the people who programmed in it were sometimes considered inferior programmers. “It’s been going on forever,” said this programmer, who has been working in the field for close to thirty years. “It’s about gatekeeping, and keeping one’s prestige and importance in the face of technological advancements that make it easier to be replaced by new people with easier to use tools.” Gatekeeping is not only done by people and institutions; it’s written into programming languages themselves.

In a field that has elevated boy geniuses and rockstar coders, obscure hacks and complex black-boxed algorithms, it’s perhaps no wonder that a committee-designed language meant to be easier to learn and use—and which was created by a team that included multiple women in positions of authority—would be held in low esteem. But modern computing has started to become undone, and to undo other parts of our societies, through the field’s high opinion of itself, and through the way that it concentrates power into the hands of programmers who mistake social, political, and economic problems for technical ones, often with disastrous results.

## **The Labor of Care**

---

A global pandemic in which more people than ever before are applying for unemployment is a situation that COBOL systems were never designed to handle, because a global catastrophe on this scale was never supposed to happen. And yet, even in the

midst of this crisis, COBOL systems didn't actually break down. Although New Jersey's governor issued his desperate plea for COBOL programmers, later investigations revealed that it was the website through which people filed claims, written in the comparatively much newer programming language Java, that was responsible for the errors, breakdowns, and slowdowns. The backend system that processed those claims—the one written in COBOL—hadn't been to blame at all.

So why was COBOL framed as the culprit? It's a common fiction that computing technologies tend to become obsolete in a matter of years or even months, because this sells more units of consumer electronics. But this has never been true when it comes to large-scale computing infrastructure.

This misapprehension, and the language's history of being disdained by an increasingly toxic programming culture, made COBOL an easy scapegoat. But the narrative that COBOL was to blame for recent failures undoes itself: scapegoating COBOL can't get far when the code is in fact meant to be easy to read and maintain.

That said, even the most robust systems need proper maintenance in order to fix bugs, add features, and interface with new computing technologies. Despite the essential functions they perform, many COBOL systems have not been well cared for. If they had come close to faltering in the current crisis, it wouldn't have been because of the technology itself. Instead, it would have been due to the austerity logic to which so many state and local governments have succumbed.

In order to care for technological infrastructure, we need maintenance engineers, not just systems designers—and that means paying for people, not just for products. COBOL was never meant

to cut programmers out of the equation. But as state governments have moved to slash their budgets, they've been less and less inclined to pay for the labor needed to maintain critical systems. Many of the people who should have been on payroll to maintain and update the COBOL unemployment systems in states such as New Jersey have instead been laid off due to state budget cuts. As a result, those systems can become fragile, and in a crisis, they're liable to collapse due to lack of maintenance.

It was this austerity-driven lack of investment in people—rather than the handy fiction, peddled by state governments, that programmers with obsolete skills retired—that removed COBOL programmers years before this recent crisis. The reality is that there are plenty of new COBOL programmers out there who could do the job. In fact, the majority of people in the COBOL programmers' Facebook group are twenty-five to thirty-five-years-old, and the number of people being trained to program and maintain COBOL systems globally is only growing. Many people who work with COBOL graduated in the 1990s or 2000s and have spent most of their twenty-first century careers maintaining and programming COBOL systems.

*“Constantly building new technological systems at the expense of existing infrastructure is not progress is among the most regressive paths a society can take.”*

If the government programmers who were supposed to be around were still on payroll to maintain unemployment systems, there's a very good chance that the failure of unemployment insurance systems to meet the life-or-death needs of people across the country wouldn't have happened. It's likely those programmers would have needed to break out their Java skills to fix the issue, though. Because, despite the age of COBOL systems, when the crisis hit, COBOL trundled along, remarkably stable.

*“When systems are built to last for decades, we often don't see the disaster unfolding until the people who cared for those systems have been gone for quite some time.”*

Indeed, present-day tech could use more of the sort of resilience and accessibility that COBOL brought to computing — especially for systems that have broad impacts, will be widely used, and will be long-term infrastructure that needs to be maintained by many hands in the future. In this sense, COBOL and its scapegoating show us an important aspect of high tech that few in Silicon Valley, or in government, seem to understand. Older systems have value, and constantly building new technological systems for short-term profit at the expense of existing infrastructure is not progress. In fact, it is among the most regressive paths a society can take.

As we stand in the middle of this pandemic, it is time for us to collectively rethink and recalculate the value that many

so-called tech innovations, and innovators, bring to democracy. When these contributions are designed around monetizing flaws or gaps in existing economic, social, or political systems, rather than doing the mundane, plodding work of caring for and fixing the systems we all rely on, we end up with more problems than solutions, more scapegoats instead of insights into where we truly went wrong.

There are analogies between the failure of state unemployment systems and the failure of all sorts of public infrastructure: Hurricane Sandy hit the New York City subway system so hard because it, too, had been weakened by decades of disinvestment. Hurricane Katrina destroyed Black lives and neighborhoods in New Orleans because the levee maintenance work that was the responsibility of the federal government was far past due, a result of racist resource allocation. COVID-19 continues to ravage the United States more than any other nation because the federal infrastructure needed to confront public health crises has been hollowed for decades, and held in particular contempt by an Administration that puts profits over people, and cares little, if at all, about the predominantly Black and Latinx people in the US who are disproportionately dying.

If we want to care for people in a pandemic, we also have to be willing to pay for the labor of care. This means the nurses and doctors who treat COVID patients; the students and teachers who require smaller, online classes to return to school; and the grocery workers who risk their lives every day. It also means making long-term investments in the engineers who care for the digital infrastructures that care for us in a crisis.

When systems are built to last for decades, we often don't see the disaster unfolding until the people who cared for

those systems have been gone for quite some time. The blessing and the curse of good infrastructure is that when it works, it is invisible: which means that too often, we don't devote much care to it until it collapses. 

Mar Hicks is a historian of technology, professor, former sysadmin, and the author of the book *Programmed Inequality*.

o hi

## We're a small magazine, and we pay our writers.

Writing is hard work, and we believe we can't have a better discourse around technology without compensating the people who are working to improve it.

But we can't do this without your help.

**Subscribing is one way to support this project.**

**Making a tax-deductible donation is another.**

Logic Magazine is published by the Logic Foundation, a California nonprofit public benefit corporation with 501(c)(3) status.

Your contributions will enable us to keep building a project that's committed to recognizing and rewarding creative labor.

For more information about the Logic Foundation, including how to donate electronically or by check, visit [logicmag.io/donate](https://logicmag.io/donate).

Other questions? Email [donate@logicmag.io](mailto:donate@logicmag.io).

thxbye

# LOGIC

## upcoming

### **ISSUE 12: COMMONS**

**WINTER 2020**

The information age began with dreams of a common language. The internet would let any network speak to any other. It has since been described as many kinds of places to gather: a public square, a mall, a street, a garden, or utopia, which is another way of saying no (real) place at all. In this issue we will explore shared standards and protocols that make possible new forms of civic life, ownership, aid, and their opposites, at the same time that they differentiate us ever more precisely. Online, you are unique like everybody else.

### **ISSUE 13: DISTRIBUTION**

**SPRING 2021**

The internet was invented for the purpose of redistribution: to move computing power from one place to another. Today, the cloud both has and has not fulfilled this dream. On the one hand, users anywhere can borrow cycles from servers deep in the forests of Oregon or high on the mountains of Guizhou. On the other, those servers belong to just a handful of companies. Data analytics and machine learning have made it possible to optimize supply chains linking every part of the world. But they have not spread production or profits evenly. As blockchain evangelists aim to distribute trust, and distributed-cognition theorists describe a world of matter vibrating with consciousness, this issue will explore the distributive aspects of digital technologies. New futures are always arriving; they are never evenly distributed.

subscribe @ <https://logicmag.io>