Decentralized Infrastructure for Neuro(science)

Jonny L. Saunders
October 12, 2021

Contents

{% include annotation.html %} {% include status.html %}

1. table of contents {:toc}

This is a draft document, so if you do work that you think is relevant here but I am not citing it, it's 99% likely that's because I haven't read it , not that I'm deliberately ignoring you! Odds are I'd love to read & cite your work, and if you're working in the same space try and join efforts!

> If we can make something decentralised, out of control, and of great simplicity, we must be prepared to be astonished at whatever might grow out of that new medium.
>
> Tim Berners-Lee (1998): Realising the Full Potential of the Web

> A good analogy for the development of the Internet is that of constantly renewing the individual streets and buildings of a city, rather than razing the city and rebuilding it. The architectural principles therefore aim to provide a framework for creating cooperation and standards, as a small "spanning set" of rules that generates a large, varied and evolving space of technology.
>
> RFC 1958: Architectural Principles of the Internet

> In building cyberinfrastructure, the key question is not whether a problem is a "social" problem or a "technical" one. That is putting it the wrong way around. The question is whether we choose, for any given problem, a primarily social or a technical solution
>
> Bowker, Baker, Millerand, and Ribes (2010): Toward Information Infrastructure Studies [1]

> The critical issue is, how do actors establish generative platforms by instituting a set of control points acceptable to others in a nascent ecosystem? [2]

Acknowledgements (make sure to double check spelling!!!): * Lauren E. Wool * Gaby Hayden * Eartha Mae * jakob voigts for participating in the glue wiki * nwb & dandi team for dealing w/ my inane rambling * open behavior team * metascience class for some of these ideas <3 * mike for letting me always go rogue * os & avery for STS recs * Joel Chan @ UMD * Tomasz Pluskiewicz * add rest from presentation test2

1. Introduction

We work in technical islands that range from individual researchers, to labs, consortia, and at their largest a few well-funded organizations. Our knowledge dissemination systems are as nimble as the static pdfs and ephemeral conference talks that they have been for decades (save for the godforsaken Science Twitter that we all correctly love to hate). Experimental instrumentation except for that at the polar extremes of technological complexity or simplicity is designed and built custom, locally, and on-demand. Software for performing experiments is a patchwork of libraries that satisfy some of the requirements of the experiment, sewn together by some uncommented script written years ago by a grad student who left the lab long-since. The technical knowledge to build both instrumentation and software is fragmented and unavailable as it sifts through the funnels of word-limited methods sections and never-finished documentation. And O Lord Let Us Pray For The Data, born into this world without coherent form to speak of, indexable only by passively-encrypted notes in a paper lab notebook, dressed up for the analytical ball once before being mothballed in ignominy on some unlabeled external drive.

In sum, all the ways our use and relations with computers are idiosyncratic and improvised are not isolated, but a symptom of a broader deficit in **digital infrastructure** for science. The yawning mismatch between our ambitions of what digital technology *should* allow us to do and the state of digital infrastructure hints at the magnitude of the problem: the degree to which the symptoms of digital deinfrastructuring define the daily reality of science is left as an exercise to the reader.

If the term infrastructure conjures images of highways and plumbing, then surely digital infrastructure would be flattered at the association. By analogy they illustrate many of its promises and challenges: when designed to, it can make practically impossible things trivial, allowing the development of cities by catching water where it lives and snaking it through tubes and tunnels sometimes directly into your kitchen. Its absence or failure is visible and impactful, as in the case of power outages. There is no guarantee that it "optimally" satisfies some set of needs for the benefit of the greatest number of people, as in the case of the commercial broadband duopolies. It exists not only as its technical reality, but also as an embodied and shared set of social practices, and so even when it does exist its form is not inevitable or final; as in the case of bottled water producers competing with municipal tap water on a behavioral basis despite being dramatically less efficient and more costly. Finally it is not socially or ethically neutral, and the impact of failure to build or maintain it is not equally shared, as in the expression of institutional racism that was the Flint, Michigan water crisis [3] .

Being digitally deinfrastructured is not our inevitable and eternal fate, but the course of infrastructuring is far from certain. It is not the case that "scientific digital infrastructure" will rise from the sea monolithically as a natural result of more development time and funding, but instead has many possible futures[4] , each with their own advocates and beneficiaries. Without concerted and strategic counterdevelopment based on a shared and liberatory ethical framework, science is poised to follow other domains of digital technology down the dark road of platform capitalism. The prize of owning the infrastructure that the practice of science is built on is too great, and it is not hard to imagine tech behemoths buying out the emerging landscape of small scientific-software-as-a-service startups and selling subscriptions to Science Prime.

This paper is an argument that **decentralized** digital infrastructure is the best means of realizing

the promise of digital technology for science. I will draw from several disciplines and knowledge communities like Science and Technology Studies (STS), Library and Information Science, open source software developers, and internet pirates, among others to articulate a vision of an infrastructure in three parts: **shared data, shared tools, and shared knowledge.** I will start with a brief description of what I understand to be the state of our digital infrastructure and the structural barriers and incentives that constrain its development. I will then propose a set of design principles for decentralized infrastructure and possible means of implementing it informed by prior successes and failures at building mass digital infrastructure. I will close with contrasting visions of what science could be like depending on the course of our infrastructuring, and my thoughts on how different actors in the scientific system can contribute to and benefit from decentralization.

I insist that what I will describe is *not utopian* but is eminently practical — the truly impractical choice is to do nothing and continue to rest the practice of science on a pyramid scheme [5] of underpaid labor. With a bit of development to integrate and improve the tools, **everything I propose here already exists and is widely used.** A central principle of decentralized systems is embracing heterogeneity: harnessing the power of the diverse ways we do science instead of constraining them. Rather than a patronizing argument that everyone needs to fundamentally alter the way they do science, the systems that I describe are specifically designed to be easily incorporated into existing practices and adapted to variable needs. In this way I argue decentralized systems are *more practical* than the dream that one system will be capable of expanding to the scale of all science — and as will hopefully become clear, inarguably *more powerful* than a disconnected sea of centralized platforms and services.

An easy and common misstep is to categorize this as solely a *technical* challenge. Instead the challenge of infrastructure is also *social* and *cultural* — it involves embedding any technology in a set of social practices, a shared belief that such technology should exist and that its form it not neutral, and a sense of communal valuation and purpose that sustains it [6] .

The social and technical perspectives are both essential, but make some conflicting demands on the construction of the piece: Infrastructuring requires considering the interrelatedness and mutual reinforcement of the problems to be addressed, rather than treating them as isolated problems that can be addressed piecemeal with a new package. Such a broad scope trades off with a detailed description of the relevant technology and systems, but a myopic techno-zealotry that does not examine the social and ethical nature of scientific practice risks reproducing or creating new sources of harm. As a balance I will not be proposing a complete technical specification or protocol, but describing the general form of the tools and some existing examples that satisfy them; I will not attempt a full history or treatment of the problem of infrastructuring, but provide enough to motivate the form of the proposed implementations.

My understanding of this problem is, of course, uncorrectably structured by the horizon of disciplines around systems neuroscience that has preoccupied my training. While the core of my argument is intended to be a sketch compatible with sciences and knowledge systems generally, my examples will sample from, and my focus will skew to my experience. In many cases, my use of "science" or "scientist" could be "neuroscience" or "neuroscientist," but I will mostly use the former to avoid the constant context switches. I ask the reader for a measure of patience for the many

ways this argument requires elaboration and modification for distant fields.

2. The State of Things

2.1 The Costs of being Deinfrastructured

Framing the many challenges of scientific digital technology development as reflective of a general digital infrastructure deficit gives a shared etiology to the technical and social harms that are typically treated separately. It also allows us to problematize other symptoms that are embedded in the normal practice of contemporary science.

To give a sense of the scale of need for digital scientific infrastructure, as well as a general scope for the problems the proposed system is intended to address, I will list some of the present costs. These lists are grouped into rough and overlapping categories, but make no pretense at completeness and have no particular order.

Impacts on the **daily experience** of researchers include:

- A prodigious duplication and dead-weight loss of labor as each lab, and sometimes each person within each lab, will reinvent basic code, tools, and practices from scratch. Literally it is the inefficiency of the Harberger's triangle in the supply and demand system for scientific infrastructure caused by inadequate supply. Labs with enough resources are forced to pay from other parts of their grants to hire professional programmers and engineers to build the infrastructure for their lab (and usually their lab or institute only), but most just operate on a purely amateur basis. Many PhD students will spend the first several years of their degree re-solving already-solved problems, chasing the tails of the wrong half-readable engineering whitepapers, in their 6th year finally discovering the technique that they actually needed all along. That's not an educational or training model, it's the effect of displacing the undone labor of unbuilt infrastructure on vulnerable graduate workers almost always paid poverty wages.
- At least the partial cause of the phenomenon where "every scientist needs to be a programmer now" as people who aren't particularly interested in being programmers — which is *fine* and *normal* — need to either suffer through code written by some other unlucky amateur or learn an entire additional discipline in order to do the work of the one they chose. Because there isn't more basic scientific programming infrastructure, everyone needs to be a programmer.
- A great deal of pain and alienation for early- career researchers (ECRs) not previously trained in programming before being thrown in the deep end. Learning data hygeine practices like backup, annotation, etc. "the hard way" through some catastrophic loss is accepted myth in much of

science. At some scale all the very real and widespread pain, and guilt, and shame felt by people who had little choice but to reinvent their own data management system must be recognized as an infrastructural, rather than a personal problem.

- The high cost of "openness" and the dearth of data transparency. It is still rare for systems neuroscience papers to publish the full, raw data along with all the analysis code, often because (in addition to the extraordinarily meagre incentives to do so) the data *and* analysis code are both completely homebrew and often omitted just due to the labor of cleaning it or the embarassment of sharing it[1]. The "Open science" movement, roughly construed, has made a holy mess of the social climate around openness, publicly shaming "closed scientists" on leaderboards and only occasionally recalling the relative luxury of labor or expertise to become "open." "Openness" is not a uniform or universal goal for all science, but for those for whom it makes sense, we need to provide the appropriate tooling before insisting on a change in scientific norms. We can't expect data transparency from researchers while it is still so *hard*.

Impacts on the **system of scientific inquiry** include:

- A profoundly leaky knowledge acquisition system where entire PhDs worth of data can be lost and rendered useless when a student leaves a lab and no one remembers how to access the data or how it's formatted.
- The inevitability of continual replication crises because it is often literally impossible to replicate an experiment that is done on a rig that was built one time, used entirely in-lab code, and was never documented
- Outside of increasingly archaic PDFs distributed by already archaic journals, the need to rely on communication platforms and knowledge systems that weren't designed to, and don't come close to satisfying the needs of scientific communication. In the absence of some generalized means of knowledge organization, scientists ask the void (Twitter) for advice or guidance from anyone that algorithmically stumbles by. The highest we can aspire is to make a Slack about something, which even if it reaches the rare escape velocity of participation to make it useful, is incapable of producing a public, durable, and cumulative resource: and so the questions will be asked again... and again...
- A perhaps doomed intellectual endeavor as we attempt to understand the staggering complexity of the brain by peering at the brain through the pinprickiest peephole of just the most recent data you or your lab have collected rather than being able to index across all relevant data from not only your lab, but all other labs that have measured the same phenomena. The unnecessary reduplication of experiments becomes not just a methodological limitation, but an ethical catastrophe as researchers have little choice but to abandon the elemental principle of sacrificing as few animals as possible to understand a phenomenon.
- A hierarchy of prestige that devalues the labor of multiple groups of technicians, animal care workers, and so on. Authorship is the coin of the realm, but many researchers that do work fundamental to the operation of

[1] which, to be clear, is a valid feeling and is reflective of a failure of infrastructure, not a personal failure.

science only receive the credit of an acknowledgement. We need a system to value and assign credit for the immense amount of technical and practical knowledge and labor they produce.

Impacts on the relationship between **science and society**:

- An insular system where the inaccessibility of all the "contextual" knowledge [7, 8] that is beneath the level of publication but necessary to perform experiments, like "how to build this apparatus," "what kind of motor would work here," etc. is a force that favors established and well-funded labs who can rely on local knowledege and hiring engineers/etc. and excludes new, lesser-funded labs at non-ivy institutions. The concentration of technical knowledge magnifies the inequity of strongly skewed funding distributions such that the most well-funded labs can do a completely different kind of science than the rest of us, turning the positive-feedback loop of funding begetting funding ever faster.
- An absconscion with the public resources we are privileged enough to receive, where rather than returning the fruits of the many technical challenges we are tasked with solving to the public in the form of data, tools, collected practical knowledge, etc. we largely return papers, multiplying the above impacts of labor duplication and knowledge inaccessibility by the scale of society.
- The complicity of scientists in rendering our collective intellectual heritage nothing more than another regiment in the ever-advancing armies of platform capitalism. If our highest aspirations are to shunt all our experiments, data, and analysis tools onto Amazon Web Services, our failure of imagination will be responsible for yet another obligate funnel of wealth into some of the most harmful corporations that have ever existed. For ourselves, we guarantee another triple-pay industry skimming public money: pay to store data, pay for access to the database, maybe with a premium subscription for the most in-demand datasets. For society, we squander the chance for one of the very few domains of non-economic labor to build systems to recollectivize the basic infrastrucutre of the internet: rather than providing an alternative to the information overlords and their digital enclosure movement, we will be run right into their arms.

and so on.

Considered separately, these are serious problems, but together they are a damning indictment of our role as stewards of our corner of the human knowledge project.

We arrive at this situation not because scientists are lazy and incompetent, but because the appropriate tools that fit the requirements of their discipline don't exist. The tools don't exist in part because we are embedded in a system of scientific labor that largely lack the reward mechanisms to build them, and in fact incentivize new, unintegrated, often quickly-abandoned tools rather than maintaining and expanding tools. After all, pull requests don't get get publications. We are unlikely to arrive at a set of tools that meet our needs because we are embedded in a model of scientific and digital technology production that depend on maintaining points of centralized control to guarantee continued profit extraction: put bluntly, "we are dealing with a massively entrenched set of institutions, built around the last information age and fighting for its life" [1]

There is, of course, an enormous amount of work being done by researchers and engineers on all of these problems, and a huge amount of progress has been made on them. My intention is not to shame or devalue anyone's work, but to try and describe a path towards integrating it and making it mutually reinforcing.

Before proposing a potential solution to some of the above problems, it is important to motivate why they haven't already been solved, or why their solution is not necessarily imminent. To do that, we need a sense of the social and technical challenges that structure the development of our tools.

2.2 Systems Neuroscience Specifically...

Every discipline has its own particular technical needs, and is subject to its own peculiar history and culture. Though the type of comprehensive distributed infrastructure I will describe later is a domain-general project, systems neuroscience specifically lacks some features of it that are present in immediately neighboring disciplines like genetics and cognitive psychology. I won't attempt a complete explanation, but instead will offer a few patterns I have noticed in my own limited exposure to the field that might serve as the beginnings of one. I want to be very clear throughout that I am never intending to cast shade on the work of anyone who has or does build and maintain the scientific infrastructure that exists — in fact the opposite, that y'all deserve more resources.

2.2.1 Diversity of Measurements

Molecular biology and genetics are perhaps the neighboring disciplines with the best data sharing and analytical structure, spawning and occupying the near totality of a new subdiscipline of Bioinformatics (for an absolutely fascinating ethnography, see [9]). Though the experiments are of course just as complex as those in systems neuroscience, most rely on a small number of stereotyped sequencing (meta?)methods that result in the same one-dimensional, four character sequence data structure of base pairs. Systems neuroscience experiments increasingly incorporate dozens of measurements, electrophysiology, calcium imaging, multiple video streams, motion, infrared, and other sensors, and so on. This is increasingly true as neuroscientists are attempting ever more complex and naturalistic neuroethological experiments. Even the seemingly "common" electrophysiological or multiphoton imaging data can have multiple forms — raw voltage traces? spike times? spike templates and times? single or multiunit? And these forms go through multiple intermediate stages of processing — binning, filtering, aggregating, etc. — each of which could be independently valuable and thus represented alongside their provenance in a theoretical data schema. Mainen and colleagues note this problem as well:

> The data sets generated by a functional neuroscience experiment are large. They can also be complex and multimodal in ways that, say, genomic data might not be, embracing recordings of activity, behavioural patterns, responses to perturbations, and subsequent anatomical analysis. Researchers have no agreed formats for integrating different types of information. Nor are there standard systems for curating, uploading and hosting highly multimodal data. [10]

The Neurodata Without Borders project has made

a valiant effort to unify these multiple formats, but has for reasons that I won't lay claim to knowing has yet to see widespread adoption. Contrast this with the BIDS data structure for fMRI data, where by converting your data to the structure you unlock a huge library of analysis pipelines for free. The beginnings of generalized platforms for neuroscientific data built on top of NWB are starting to happen in trickles and droplets, but they are still very much the exception rather than the rule.

We should not be so proud as to believe that our data is somehow uniquely complex. Theorizing about and reconciling the mass and heterogeneity of data in the universe is the subject of multiple full-fledged disciplines, and the conflict between simplified and centralized [11] and sprawling and distributed [12] systems is well-trodden — and we should learn from it! We could instead think of the complexity of our data and the tools we develop to address it as what we have to offer the broader human mission towards a unified system of knowledge.

2.2.2 Diversity of Preps

Though there are certain well-limbered experimental backbones like the two-alternative forced choice task, even within them there seems to be a comparatively broad diversity of experimental preparations in systems neuro relative to adjacent fields. Even a visual two-alternative forced choice task is substantially different than an auditory one, but there is almost nothing shared between those and, for example, measuring the representation of 3d space in a free-flying echolocating bat. So unlike cognitive neuroscience and psychophysics that has tools like pavlovia where the basic requirements and structure of experiments are more standardized, BioRXiv is replete with technical papers documenting "high throughput systems for this one very specific experiment" and there isn't a true experimental framework that satisfies the need for flexibility.

Mainen and colleagues note that this causes another problem distinct from variable outcome data, the even more variable and largely unreported metadata that parameterizes the minute details of experimental preps:

> Worse, neuroscientists lack standardized vocabularies for describing the experimental conditions that affect brain and behavioural functions. Such a vocabulary is needed to properly annotate functional neural data. For instance, even small differences in when a water drop is released can affect how a mouse's brain processes this event, but there is no standard way to specify such aspects of an experiment. [10]

The problem of universal annotation and metadata reporting can be reframed, not as a *barrier to developing*, but as a *design constraint* of experimental programming infrastructure. Because of the fragmentation of scientific programming infrastructure, where each experimental prep is implemented with entirely different, and often single-use software, there is no established reporting system for automatically capturing these minute details — but that doesn't mean there can't be (as I wrote previously, see section 2.3 in [13] , and coincidentally measured the effect of variable water droplets).

2.2.3 The Hacker Spirit and Celebration of Heroism

Many people are attracted to systems neuroscience precisely *because* of the... playful... attitude we take towards our rigs. If you want to do something, don't ask questions just break out the

hot glue, vaseline, and aluminum foil and hack at it until it does what you want. The natural conclusion of widespread embodiment of this lovable scamp hacker spirit is its veneration as heroism: it is a *good thing* to have done an experiment that only you are capable of doing because that means you're the best hacker. Not unrelated is the strong incentive to make something new rather than build on existing tools — you don't get publications from pull requests, and you don't get a job without publications. The initial International Brain Laboratory described the wily nature of neuroscientists accordingly:

> Simply maintaining a true collaboration between 21 laboratories accustomed to going their own way will be a major novelty in neuroscience. [14]

And yes, like the rest of the universe, perhaps the most influential forces in this domain are inertia and entropy. Once the boulder starts rolling down the hill of heroic idiosyncracy, tumbling along in a semi-stable jumble[2] that supports the experiments of a lab, retooling and standardizing that system has to be *so very cool and worth it* that it overcomes the various, uncertain, but typically substantial costs (including the valid emotional costs of wishing a peaceful voyage to well-loved handcrafted tools). More than a single moment of adoption, the universe always has room for another course of disorder, and a commitment to using communal tools must be constantly reaffirmed. As we dream up new wild experiments, it needs to be easier to implement them with the existing system and integrate the labor expended in doing so back into it than it is to patch over the problem with a quick script saved to Desktop. As people cycle through the lab, it must be easier to learn than it is to start from scratch.

Yes again, Mainen and colleagues:

> Neuroscientists frequently live on the 'bleeding' edge technologically, building bespoke and customized tools. This do-it-yourself approach has allowed innovators to get ahead of the competition, but hampered the standardization of methods essential to making experiments efficient and replicable.
>
> Remarkably, it is standard practice for each lab to custom engineer all manner of apparatus, from microscopes and electrodes to the computer programmes for analysing data. Thousands of labs worldwide use the calcium sensor GCaMP, for example, for imaging neural activity in vivo. Yet neither the microscopes used for GCaMP imaging nor the algorithms used to analyse the resulting data sets have been standardized. [10]

!! make it clearer that the hacker spirit is not a *bad* thing but another *design constraint* and that we should actually avoid the paternalistic approach that says there's a "right way" to do science, and instead honor, learn from, and support the diversity of our approaches.

2.2.4 Focus on the Science

Completely understandably... scientists want to focus on their discipline rather than spending time building infrastructure. But because infrastructure touches all of our work and very few people can only build it in their spare time (mostly for the love of the craft) we all have to build some of it. this is a classic collective action problem, and scientists are not evil or selfish for wanting to do their work.

[2] A *lovely* jumble! that probably has a lot of good qualities, it's just a little lonely maybe :(

2.2.5 Combinatorics of Recent Technology

A lot of what I will describe here is relatively new! Some ideas are very old, like the semantic web and wikis, but others like federated communication and file transfer protocols are only reaching widespread use recently. The entire universe of open source scientific hardware and software has only sprung into its full and beautiful glory in the last decade or so, from pandas and and jupyter to open ephys and miniscopes and so on. Bittorrent is cool and good but IPFS allows us to think about qualitatively different things. It's ultimately the *combination* of these recently technologies that's important, rather than any single one of them. So in some sense it wasn't *possible* to think about the type of basic infrastructure outside the traditional lens of centralized databases and individual experimental software packages.

2.3 Scientific Software Generally...

The constraints posed by the structure of systems neuroscience as a discipline are of course echos and elaborations of larger constraints in the system of scientific infrastructure production.

2.3.1 Incentivized Fragmentation

The incentive systems in science are complex, but tend to reward the production of many isolated, single-purpose software packages rather than cumulative work on shared infrastructure. The primary means of evaluation for a scientist is academic reputation, primarily operationalized by publications, but a software project will yield a single paper (if any at all). Traditional publications are static units of work that are "finished" and frozen in time, but software is never finished: the thousands of commits needed to maintain and extend the software are formally not a part of the system of academic reputation.

Shoehorning reputational rewards through traditional scientific publications has three immediate consequences: 1) Scientists are incentivized to make new, independent software that can be independently published, rather than integrating their work to extend the functionality of existing software. Howison & Herbsleb described this dynamic in the context of BLAST

> In essence we found that BLAST innovations from those motivated to improve BLAST by academic reputation are motivated to develop and to reveal, but not to integrate their contributions. Either integration is actively avoided to maintain a separate academic reputation or it is highly conditioned on whether or not publications on which they are authors will receive visibility and citation. [15]

For an example in Neuroscience, one can browse the papers that cite the DeepLabCut [16] to find hundreds of downstream projects that make various extensions and improvements that are not integrated into the main library. While the logical extreme of the alternative of a single monolithic ur-library is also undesirable, the point is that a scientist that released 10 barely working, barely documented, rapidly abandoned packages along with 10 code papers would have 10 times the academic credit than one who spent the time integrating them into a unified, well-documented frame-

work for something 1/10th[3] as useful.

2) After publication, scientists have little incentive to **maintain** software outside of the domains in which the primary contributors use it (to satisfy reputational incentives by publishing in their own discipline), so outside of the most-used libraries most scientific software is brittle and difficult to use [17, 18] .

3) Since the reputational value of a publication depends on its placement within a journal and number of citations (among other metrics), and citation practices for scientific software are far from uniform and universal, the incentive to write scientific software at all is relatively low compared to its near-universal use [19] .

2.3.2 Domain-Specific Silos

When funding exists for scientific infrastructure development, it typically comes in the form of side effects from, or administrative supplements to research grants. The NIH describes as much in their Strategic Plan for Data Science [**NIH, StrategicPlan2018**] :

> from 2007 to 2016, NIH ICs used dozens of different funding strategies to support data resources, most of them linked to research-grant mechanisms that prioritized innovation and hypothesis testing over user service, utility, access, or efficiency. In addition, although the need for open and efficient data sharing is clear, where to store and access datasets generated by individual laboratories—and how to make them compliant with FAIR principles—is not yet straightforward. Overall, it is critical that the data-resource ecosystem become seamlessly integrated such that different data types and information about different organisms or diseases can be used easily together rather than existing in separate data "silos" with only local utility.

The National Library of Medicine within the NIH currently lists 122 separate databases in its search tool, each serving a specific type of data for a specific research community. Though their current funding priorities signal a shift away from domain-specific tools, the rest of the scientific software system consists primarily of tools and data formats purpose-built for a relatively circumscribed group of scientists without any framework for their integration. Every field has its own challenges and needs for software tools, but there is little incentive to build tools that serve as generalized frameworks to integrate them.

2.3.3 "The Long Now" of Immediacy vs. Idealism

Digital infrastructure development takes place at multiple timescales simultaneously — from the momentary work of implementing it, through longer timescales of planning, organization, and documenting to the imagined indefinite future

[3] Figuratively! Non-quantitatively!

of its use — what Ribes and Finholt call "The Long Now. [20] " Infrastructural projects constitutively need to contend with the need for immediately useful results vs. general and robust systems; the need to involve the effort of skilled workers vs. the uncertainty of future support; the balance between stability with mutability; and so on. The tension between hacking something together vs. building something sustainable for future use is well-trod territory in the hot-glue and exposed wiring of systems neuroscience rigs.

Deinfrastructuring divides the incentives and interests of senior and junior researchers. Established researchers face little pressure to improve the state of infrastructure, as (very crudely) their primary incentives are to push enough publications through the door to be able to secure the next round of funding to keep their lab afloat. Their time preference is very short: hack it together, get the paper out, we'll fix it later.

ECRs are tasked with develping the tools, often interested in developing tools they'll be able to use throughout their careers, but between the pressure to establish their reputation with publications rarely have the time to develop something fully. As a consequence, a lot of software tools are developed by ECRs with no formal software training, contributing to the brittleness of scientific software and the low rates of adoption of best practices [21] . Anecdotally, the constant need to produce software that *does something* in the context of scientific programming which largely lacks the institutional systems and expert mentorship needed for well-architected software means that some programmers *never* have a chance to learn best practices commonly accepted in software engineering.

The problem of time horizon in development is not purely a product of inexperience, and a longer time horizon is not uniformly better. For an example, look no further than the history and cultural dynamics of the semantic web and linked data communities, revisted more fully in a moment as Scruffiness vs. Neatness. In the semantic web era, thousands of some of the most gifted programmers worked with an eye to the indefinite future, but the raw idealism and neglect of the pragmatic reality of the need for software to *do something* drove many to abandon the effort:

> But there was no *use* of it. I wasn't using any of the technologies for anything, except for things related to the technology itself. The Semantic Web is utterly inbred in that respect. The problem is in the model, that we create this metaformat, RDF, and *then* the use cases will come. But they haven't, and they won't. Even the genealogy use case turned out to be based on a fallacy. The very few use cases that there are, such as Dan Connolly's hAudio export process, don't justify hundreds of eminent computer scientists cranking out specification after specification and API after API.
>
> When we discussed this on the Semantic Web Interest Group, the conversation kept turning to how the formats could be fixed to make the use cases that I outlined happen. "Yeah, Sean's right, let's fix our languages!" But it's not the languages which are broken, except in as much as they are entirely broken: because it's the *mentality* of their design which is broken. You can't, it has turned out, make a metalanguage like RDF and then go looking for use cases. We thought you could, but you can't. It's taken eight years to realise. [22]

Developing digital infrastructure must be both bound to fulfilling immediate needs and a sense of incrementalism as well as guided by a long-range vision. The technical and social lessons run in parallel: We need software that solves problems peo-

ple actually have right now, but can flexibly support its eventual form. We need a long-range vision to know what kind of tools we should build and which we shouldn't, and we need to keep it in a tight loop with the always-changing needs of the people it supports. In short, to develop digital infrastructure we need to be *strategic.* To be strategic we need a *plan.* To have a plan we need to value planning as *work.* On this, Ribes and Finholt are instructive:

> "On the one hand, I know we have to keep it all running, but on the other, LTER is about long-term data archiving. If we want to do that, we have to have the time to test and enact new approaches. But if we're working on the to-do lists, we aren't working on the tomorrow-list" (LTER workgroup discussion 10/05).
>
> The tension described here involves not only time management, but also the differing valuations placed on these kinds of work. The implicit hierarchy places scientific research first, followed by deployment of new analytic tools and resources, and trailed by maintenance work. [...] While in an ideal situation development could be tied to everyday maintenance, in practice, maintenance work is often invisible and undervalued. As Star notes, infrastructure becomes visible upon breakdown, and only then is attention directed at its everyday workings (1999). Scientists are said to be rewarded for producing new knowledge, developers for successfully implementing a novel technology, but the work of maintenance (while crucial) is often thankless, of low status, and difficult to track. *How can projects support the distribution of work across research, development, and maintenance?* [20]

[23]

test

2.3.4 "Neatness" vs "Scruffiness"

Closely related to the tension between "Now" and "Later" is the tension between "Neatness" and "Scruffiness." Lindsay Poirier traces its reflection in the semantic web community as the way that differences in "thought styles" result in different "design logics" [24] . On the question of how to develop technology for representing the ontology of the web – the system of terminology and structures with which everything should be named – there were (very roughly) two camps. The "neats" prioritized consistency, predictability, uniformity, and coherence – a logically complete and formally valid System of Everything. The "scruffies" prioritized local systems of knowledge, expressivity, "believing that ontologies will evolve organically as everyday webmasters figure out what schemas they need to describe and link their data. [24] "

Practically, the differences between these thought communities impact the tools they build. Aaron Swartz put the approach of the "neat" semantic web architects the way he did:

Instead of the "let's just build something that works" attitude that made the Web (and the Internet) such a roaring success, they brought the formalizing mindset of mathematicians and the institutional structures of academics and defense contractors. They formed committees to form working groups to write drafts of ontologies that carefully listed (in 100-page Word documents) all possible things in the universe and the various properties they could have, and they spent hours in Talmudic debates over whether a washing machine was a kitchen appliance or a household cleaning device.

With them has come academic research and government grants and corporate R&D and the whole apparatus of people and institutions that

> scream "pipedream." And instead of spending time building things, they've convinced people interested in these ideas that the first thing we need to do is write standards. (To engineers, this is absurd from the start—standards are things you write after you've got something working, not before!) [25]

The "scruffies," recognizing the limitations of this approach diverged into a distinct thought community under the mantle of linked data. The linked data developers, starting by acknowledging that no one system can possibly capture everything, build tools that allow expression of local systems of meaning with the expectation and affordances for linking data between these systems as an ongoing social process.

The outcomes of this cultural rift are subtle, but the broad strokes are clear: the linked data community has taken some of the core semantic web technology like RDF, OWL, and the like, and developed a broad range of downstream technologies that have found broad use across information sciences, library sciences, and other applied domains. The vision of a totalizing and logically consistent semantic web, however, has largely faded into obscurity. One developer involved with semantic web technologies (who requested not be named), captured the present situation in their description of a still-active developer mailing list:

> I think that some people are completely detached from practical applications of what they propose. [...] I could not follow half of the messages. these guys seem completely removed from our plane of existence and I have no clue what they are trying to solve.

This division in thought styles generalizes across domains of infrastructure, though outside of the linked data and similar worlds the dichotomy is more frequently between "neatness" and "people doing whatever" – with integration and interoperability becoming nearly synonymous with standardization. Calls for standardization without careful consideration and incorporation of existing practice have a familiar cycle: devise a standard that will solve everything, implement it, wonder why people aren't using it, funding and energy dissipiates, rinse, repeat. The difficulty of scaling an exacting vision of how data should be formatted, the tools researchers should use for their experiments, and so on is that they require dramatic and sometimes total changes to the way people do science. The alternative is not between standardization and chaos, but a potential third way is designing infrastructures that allow the diversity of approaches, tools, and techniques to be expressed in a common framework or protocol along with the community infrastructure to allow the continual negotiation of their relationship.

2.3.5 Taped-on Interfaces: Open-Loop User Testing

The point of most active competition in many domains of commercial software is the user interface and experience (UI/UX), and to compete software companies will exhaustively user-test and refine them with pixel precision to avoid any potential customer feeling even a thimbleful of frustration. Scientific software development is largely disconnected from usability testing, as what little support exists is rarely tied to it. This, combined with the above incentives for developing new packages – and thus reduplicating the work of interface development – and the preponderance of semi-amateurs make it perhaps unsurprising that most scientific software is hard to use!

I intend the notion of "interface" in an expansive

way: In addition to the graphical user interface (GUI) exposed to the end-user, I am referring generally to all points of contact with users, developers, and other software. Interfaces are intrinsically social, and include the surrounding documentation and experience of use — part of using an API is being able to figure out how to use it! The typical form of scientific software is a black box: I implemented an algorithm of some kind, here is how to use it, but beneath the surface there be dragons. Ideally, software would be designed with programming interfaces and documentation at multiple scales of complexity to enable clean entrypoints for developers with differing levels of skill and investment to contribute. Additionally, it would include interfaces for use and integration with other software. Without care given to either of these interfaces, the community of codevelopers is likely to remain small, and the labor they expend is less likely to be useful outside that single project. This, in turn, reinforces the incentives for developing new packages and fragmentation.

2.3.6 Platforms, Industry Capture, and the Profit Motive

Publicly funded science is an always-irresistable golden goose for private industry. The fragmented interests of scientists and the historically light touch of funding agencies on encroaching privatization means that if some company manages to capture and privatize a corner of scientific practice they are likely to keep it. Industry capture has been thoroughly criticized in the context of the journal system (eg. recently, [26]), and that criticism should extend to the rest of our infrastructure. Another major engine for privatization of scientific infrastructure has been the preponderance of software as a service (SaaS), from startups to international megacorporations, that sell access to some, typically proprietary software without selling the software itself.

While in isolation SaaS can make individual components of the infrastructural landscape easier to access — and even free!!* — the business model is fundamentally incompatible with integrated and accessible infrastructure. The SaaS model derives revenue from subscription or use costs, often operating as freemium models that make some subset of its services available for free. Even in freemium models, though, the business model requires that some functionality of the platform is paywalled (See a more thorough treatment of platform capitalism in science in [4])

As isolated services, one can imagine the practice of science devolving along a similar path as the increasingly-fragmented streaming video market: to do my work I need to subscribe to a data storage service, a cloud computing service, a platform to host my experiments, etc. For larger software platforms, however, vertical integration of multiple complementary services makes their impact on infrastructure more insidious. Locking users into more and more services makes for more and more revenue, which encourages platforms to be as mutually incompatible as they can get away with [27] . To encourage adoption, platforms that can offer multiple services may offer one of the services – say, data storage – for free, forcing the user to use the adjoining services – say, a cloud computing platform.

Since these platforms are often subsidiaries of information industry monopolists, scientists become complicit in their ethically nightmarish behavior by funneling millions of dollars into, for example, the parent company of Elsevier and their surveillance technology agreement with ICE [28] , or AWS and the laundry list of human rights abuses by Amazon [29] .

Structurally, the adoption of SaaS on a wide scale necessarily sacrifices the goals of an integrated mass infrastructure as the practice of research is carved into small, marketable chunks within vertically integrated technology platforms. Worse, it stands to amplify, rather than reduce, inequities in science, as the labs and institutes that are able to afford the tolls between each of the weigh stations of infrastructure are able to operate more efficiently, in turn begetting more funding, and the cycle spins ever faster.

Funding models and incentive structures in science are uniformly aligned towards the platformatization of scientific infrastructure. Aside from the tragic rhetoric of "technology transfer" that pervades the neoliberal university, the relative absence of major funding opportunities for scientific software developers competitive with the profit potential from "industry" often leaves it as the only viable career path. The preceding structural constraints on local infrastructural development strongly incentivize labs and researchers to rely on SaaS that provides a readymade solution to specific problems. Distressingly, rather than supporting infrastructural development that would avoid obligate payments to platform-holders, funding agencies seem all too happy to lean into them:

> NIH will leverage what is available in the private sector, either through strategic partnerships or procurement, to create a workable Platform as a Service (PaaS) environment. [...] NIH will partner with cloud-service providers for cloud storage, computational, and related infrastructure services needed to facilitate the deposit, storage, and access to large, high-value NIH datasets.
>
> These negotiations may result in partnership agreements with top infrastructure providers from U.S.-based companies whose focus includes support for research. Suitable cloud environments will house diverse data types and high-value datasets created with public funds. NIH will ensure that they are stable and adhere to stringent security requirements and applicable law, to protect against data compromise or loss. [...] NIH's cloud-marketplace initiative will be the first step in a phased operational framework that establishes a SaaS paradigm for NIH and its stakeholders. (-NIH Strategic Plan for Data Science, 2018 [30])

The articulated plan being to pay platform holders to house data while also paying for the labor to maintain those databases veers into parody, haplessly building another triple-pay industry [31] into the economic system of science — one can hardly wait until they have the opportunity to rent their own data back with a monthly subscription.

!! this isn't a metaphor – elsevier got the deal to build the analysis pipelining system using mendeley data [32]

!! and cloud.nih.gov goes to the STRIDES program, which has cost $85 million since 2018 to establish, has a special account classification for "extramural" accounts that are researcher invoiced and managed:
[33]

!! Even on their success stories "We have been storing data in both cloud environments because we wanted the ecosystem we are creating to work on both clouds," and they are developing their own overlay on top of it to bridge them! [34]

It is unclear to me whether this is the result of the cultural hegemony of platform capitalism narrowing the space of imaginable infrastructures, industry capture of the decision-making process, or both, but the effect is the same in any case.

2.3.7 Protection of Institutional and Economic Power

The current state of deinfrastructuring certainly is not without its beneficiaries — those that have already accrued power and status within science. (I have already articulated the positive feedback loop of scientific funding, engineering costs, and prestige publishing and need to consolidate that here. The result of the protective nature of deinfrastructuring on concentrated power means that, barring some exogenous effort, we should not expect liberatory infrastructure to be developed by the places where the resources it requires are concentrated.)

2.4 Whose Job is Infrastructure? - The Ivies, Institutes, Consortia, and "The Rest of Us"

These constraints manifest differently depending on the circumstance of scientific practice. Differences in circumstance of practices also influence the kind of infrastructure developed, as well as where we should expect infrastructure development to happen as well as who benefits from it.

2.4.1 Institutional Core Facilities

Centralized "core" facilities are maybe the most typical form of infrastructure development and resource sharing at the level of departments and institutions. These facilities can range from minimal to baroque extravagance depending on institutional resources and whatever complex web of local history brought them about.

PNI Systems Core lists subprojects echo a lot of the thoughts here, particularly around effort duplication[4]:

> Creating an Optical Instrumentation Core will address the problem that much of the technical work required to innovate and maintain these instruments has shifted to students and postdocs, because it has exceeded the capacity of existing staff. This division of labor is a problem for four reasons: (1) lab personnel often do not have sufficient time or expertise to produce the best possible results, (2) the diffusion of responsibility leads people to duplicate one another's efforts, (3) researchers spend their time on technical work at the expense of doing science, and (4) expertise can be lost as students and postdocs move on. For all these reasons, we propose to standardize this function across projects to improve quality control and efficiency. Centralizing the design, construction, maintenance, and support of these instruments will increase

[4] Thanks a lot to the one-and-only stunning and brilliant Dr. Eartha Mae Guthman for suggesting looking at the BRAIN initiative grants as a way of getting insight on core facilities.

> the efficiency and rigor of our microscopy experiments, while freeing lab personnel to focus on designing experiments and collecting data.

While core facilities are an excellent way of expanding access, reducing redundancy, and standardizing tools within an instutition, as commonly structured they can displace work spent on those efforts outside of the institution. Elite institutions can attract the researchers with the technical knowledge to develop the instrumentation of the core and infrastructure for maintain it, but this development is only occasionally made usable by the broader public. The Princeton data science core is an excellent example of a core facility that does makes its software infrastructure development public[5], which they should be applauded for, but also illustrative of the problems with a core-focused infrastructure project. For an external user, the documentation and tutorials are incomplete – it's not clear to me how I would set this up for my institute, lab, or data, and there are several places of hard-coded princeton-specific values that I am unsure how exactly to adapt[6]. I would consider this example a high-water mark, and the median openness of core infrastructure falls far below it. I was unable to find an example of a core facility that maintained publicly-accessible documentation on the construction and operation of its experimental infrastructure or the management of its facility.

2.4.2 Centralized Institutes

Outside of universities, the Allen Brain Institute is perhaps the most impactful reflection of centralization in neuroscience. The Allen Institute has, in an impressively short period of time, created

[5]

> Project Summary: Core 2, Data Science Working memory, the ability to temporarily hold multiple pieces of information in mind for manipulation, is central to virtually all cognitive abilities. This multi-component research project aims to comprehensively dissect the neural circuit mechanisms of this ability across multiple brain areas. In doing so, it will generate an extremely large quantity of data, from multiple types of experiments, which will then need to be integrated together. The Data Science Core will support the individual research projects in discovering relationships among behavior, neural activity, and neural connectivity. The Core will create a standardized computational pipeline and human workflow for preprocessing of calcium-imaging data. The pipeline will run either on local computers or in cloud computing services, and users will interact with it through a web browser. The preprocessing will incorporate existing image-processing algorithms, such as Constrained Nonnegative Matrix Factorization and convolutional networks. In addition, the Core will build a data science platform that stores behavior, neural activity, and neural connectivity in a relational database that is queried by the DataJoint language. Diverse analysis tools will be integrated into DataJoint, enabling the robust maintenance of data-processing chains. This data-science platform will facilitate collaborative analysis of datasets by multiple researchers within the project, and make the analyses reproducible and extensible by other researchers. We will develop effective methods for training and otherwise disseminating our computational tools and workflows. Finally, the Core will make raw data, derived data, and analyses available to the public upon publication via the data-science platform, source-code repositories, and web-based visualization tools. To facilitate the conduct of this research, the creation of software tools, and the reuse of the data by others after the primary research has concluded, the project will adopt shared data and metadata formats using the HDF5 implementation of the Neurodata without Borders format. Data will be made public in accord with the FAIR guiding principles—findndable by a DOI and/or URL, accessible through a RESTful web API, and interoperable and reusable due to DataJoint and the Neurodata Without Borders format for data https://projectreporter.nih.gov/project_info_description.cfm?aid=9444126&icde=0

[6]Though again, this project is examplary, built by friends, and would be an excellent place to start extending towards global infrastructure.

several transformative tools and datasets, including its well-known atlases [35] and the first iteration of its Observatory project which makes a massive, high-quality calcium imaging dataset of visual cortical activity available for public use. They also develop and maintain software tools like their SDK and Brain Modeling Toolkit (BMTK), as well as a collection of hardware schematics used in their experiments. The contribution of the Allen Institute to basic neuroscientific infrastructure is so great that, anecdotally, when talking about scientific infrastructure it's not uncommon for me to hear something along the lines of "I thought the Allen was doing that."

Though the Allen Institute is an excellent model for scale at the level of a single organization, its centralized, hierarchical structure cannot (and does not attempt to) serve as the backbone for all neuroscientific infrastructure. Performing single (or a small number of, as in its also-admirable OpenScope Project) carefully controlled experiments a huge number of times is an important means of studying constrained problems, but is complementary with the diversity of research questions, model organisms, and methods present in the broader neuroscientific community.

Christof Koch, its director, describes the challenge of centrally organizing a large number of researchers:

> Our biggest institutional challenge is organizational: assembling, managing, enabling and motivating large teams of diverse scientists, engineers and technicians to operate in a highly synergistic manner in pursuit of a few basic science goals [36]

> These challenges grow as the size of the team grows. Our anecdotal evidence suggests that above a hundred members, group cohesion appears to become weaker with the appearance of semi-autonomous cliques and sub-groups. This may relate to the postulated limit on the number of meaningful social interactions humans can sustain given the size of their brain [37]

!! These institutes are certainly helpful in building core technologies for the field, but they aren't necessarily organized for developing mass-scale infrastructure.

2.4.3 Meso-scale collaborations

Given the diminishing returns to scale for centralized organizations, many have called for smaller, "meso-scale" collaborations and consortia that combine the efforts of multiple labs [10] . The most successful consortium of this kind has been the International Brain Laboratory [14, 7] , a group of 22 labs spread across six countries. They have been able to realize the promise of big team neuroscience, setting a new standard for performing reproducible experiments performed by many labs [38] and developing data management infrastructure to match [39] (seriously, don't miss their extremely impressive data portal). Their project thus serves as the benchmark for large-scale collaboration and a model from which all similar efforts should learn from.

Critical to the IBL's success was its adoption of a flat, non-hierarchical organizational structure, as described by Lauren E. Wool:

> IBL's virtual environment has grown to accommodate a diversity of scientific activity, and is supported by a flexible, 'flattened' hierarchy that emphasizes horizontal relationships over vertical management. [...] Small teams of IBL members collaborate on projects in Working Groups (WGs), which are defined around particular specializations and milestones and coordinated jointly by a chair and associate chair (typically a PI and researcher, respectively). All

> WG chairs sit on the Executive Board to propagate decisions across WGs, facilitate operational and financial support, and prepare proposals for voting by the General Assembly, which represents all PIs. In parallel, associate chairs convene on their own committee to share decisions, which are then conveyed to the entire researcher community so it may weigh in on proposals before a formal vote. The interests of PIs and researchers intersect via staff liaisons who sit on both the Executive Board and the Associate Chairs Committee, as well as an elected researcher representative, who sits on the Executive Board and is a voting member of the General Assembly. [7]

They should also be credited with their adoption of a form of consensus decision-making, sociocracy, rather than a majority-vote or top-down decisionmaking structure. Consensus decision-making systems are derived from those developed by Quakers and some Native American nations, and emphasize, perhaps unsurprisingly, the value of collective consent rather than the will of the majority. Sociocracy specifically describes consent:

> Consent means "no objections." Giving consent does not mean unanimity, agreement, or even endorsement. Decisions are made to guide actions. Can we move forward if we make this decision? Consent is given in the context of moving forward. Consent to a policy decision means you believe that it is "worth trying." Or "I can work with it." Moving forward is important for making better decisions because it provides more information. Not moving forward until a perfect decision is found, means operating in the blind. Information will always be limited to what is already known.
>
> Consent is required for all policy decisions for many reasons. The two most important are that it ensures (1) the decision will allow all members of the group to participate or produce without feeling oppressed, and (2) it will be supported by everyone. Everyone is expected to participate in the reasoning behind the decision. And no one can be excluded. https://www.sociocracy.info/what-is-sociocracy/

The central lesson of the IBL, in my opinion, is that governance matters. Even if a consortium of labs were to form on an ad-hoc basis, without a formal system to ensure contributors felt heard and empowered to shape the project it would soon become unsustainable. Even if this system is not perfect, with some labor still falling unequally on some researchers, it is a promising model for future collaborative consortia.

The infrastructure developed by the IBL is impressive, but its focus on a single experiment makes it difficult to expand and translate to widescale use. The hardware for the IBL experimental apparatus is exceptionally well-documented, with a complete and detailed build guide and library of CAD parts, but the documentation is not modularized

such that it might facilitate use in other projects, remixed, or repurposed. The experimental software is similarly single-purpose, a chimeric combination of Bonsai [40] and PyBpod scripts. It unfortunately lacks the API-level documentation that would facilitate use and modification by other developers, so it is unclear to me, for example, how I would use the experimental apparatus in a different task with perhaps slightly different hardware, or how I would then contribute that back to the library. The experimental software, according to the PDF documentation, will also not work without a connection to an alyx database. While alyx was intended for use outside the IBL, it still has IBL-specific and task-specific values in its source-code, and makes community development difficult with a similar lack of API-level documentation and requirement that users edit the library itself, rather than temporary user files, in order to use it outside the IBL.

My intention is not to denigrate the excellent tools built by the IBL, nor their inspiring realization of meso-scale collaboration, but to illustrate a problem that I see as an extension of that discussed in the context of core facilities — designing infrastructure for one task, or one group in particular makes it much less likely to be portable to other tasks and groups.

It is also unclear how replicable these consortia are, and whether they challenge, rather than reinforce technical inequity in science. Participating in consortia systems like the IBL requires that labs have additional funding for labor hours spent on work for the consortium, and in the case of graduate students and postdocs, that time can conflict with work on their degrees or personal research which are still far more potent instruments of "remaining employed in science" than collaboration. In the case that only the most well-funded labs and institutions realize the benefits of big team science without explicit consideration given to scientific equity, mesoscale collaborations could have the unintended consequence of magnifying the skewed distribution of access to technical expertise and instrumentation.

2.4.4 The rest of us...

Outside of ivies with rich core facilities, institutes like the Allen, or nascent multi-lab consortia, the rest of us are largely on our own, piecing together what we can from proprietary and open source technology. The world of open source scientific software has plenty of energy and lots of excellent work is always being done, though constrained by the circumstances of its development described briefly above. Anything else comes down to whatever we can afford with remaining grant money, scrape together from local knowledge, methods sections, begging, borrowing, and (hopefully not too much) stealing from neighboring labs.

A third option from the standardization offered by centralization and the blooming, buzzing, beautiful chaos of disconnected open-source development is that of decentralized systems, and with them we might build the means by which the "rest of us" can mutually benefit by capturing and making use of each other's knowledge and labor.

3. A Draft of Decentralized Scientific Infrastructure

Where do we go from here?

The decentralized infrastructure I will describe here is similar to previous notions of "grass-roots" science articulated within systems neuroscience [10] but has broad and deep history in many domains of computing. My intention is to provide a more prescriptive scaffolding for its design and potential implementation as a way of painting a picture of what science could be like. This sketch is not intended to be final, but a starting point for further negotiation and refinement.

Throughout this section, when I am referring to any particular piece of software I want to be clear that I don't intend to be dogmatically advocating that software *in particular*, but software *like it* that *shares its qualities* — no snake oil is sold in this document. Similarly, when I describe limitations of existing tools, without exception I am describing a tool or platform I love, have learned from, and think is valuable — learning from something can mean drawing respectful contrast!

3.1 Design Principles

I won't attempt to derive a definition of decentralized systems from base principles here, but from the systemic constraints described above, some design principles that illustrate the idea emerge naturally. For the sake of concrete illustration, in some of these I will additionally draw from the architectural principles of the internet protocols: the most successful decentralized digital technology project.

3.1.1 Protocols, not Platforms

Much of the basic technology of the internet was developed as *protocols* that describe the basic attributes and operations of a process. A simple and common example is email over SMTP (Simple Mail Transfer Protocol)[41] . SMTP describes a series of steps that email servers must follow to send a message: the sender initiates a connection to the recipient server, the recipient server acknowledges the connection, a few more handshake steps ensue to describe the senders and receivers of the message, and then the data of the message is transferred. Any software that implements the protocol can send and and receive emails to and from any other. The protocol basis of email is the reason why it is possible to send an email from a gmail account to a hotmail account (or any other hacky homebrew SMTP client) despite being wholly different pieces of software.

In contrast, *platforms* provide some service with a specific body of code usually without any pretense of generality. In contrast to email over SMTP, we have grown accustomed to not being able to send a message to someone using Telegram from WhatsApp, switching between multiple mutually incompatible apps that serve nearly identical purposes. Platforms, despite being *theoretically* more limited than associated protocols, are attractive for many reasons: they provide funding and administrative agencies a single point of contracting and liability, they typically provide a much more polished user interface, and so on. These benefits are short-lived, however, as the inevitable toll of lock-in and shadowy business models is realized.

3.1.2 Integration, not Invention

At the advent of the internet protocols, several different institutions and universities had already developed existing network infrastructures, and so the "top level goal" of IP was to "develop an effective technique for multiplex utilization of existing interconnected networks," and "come to grips

with the problem of integrating a number of separately administered entities into a common utility" [42] . As a result, IP was developed as a 'common language' that could be implemented on any hardware, and upon which other, more complex tools could be built. This is also a cultural practice: when the system doesn't meet some need, one should try to extend it rather than building a new, separate system — and if a new system is needed, it should be interoperable with those that exist.

This point is practical as well as tactical: to compete, an emerging protocol should integrate or be capable of bridging with the technologies that currently fill its role. A new database protocol should be capable of reading and writing existing databases, a new format should be able to ingest and export to existing formats, and so on. The degree to which switching is seamless is the degree to which people will be willing to switch.

This principle runs directly contrary to the current incentives for novelty and fragmentation, which must be directly counterbalanced by design choices elsewhere to address the incentives driving them.

3.1.3 Embrace Heterogeneity, Be Uncoercive

A reciprocal principle to integration with existing systems is to design the system to be integratable with existing practice. Decentralized systems need to anticipate unanticipated uses, and can't rely on potential users making dramatic changes to their existing practices. For example, an experimental framework should not insist on a prescribed set of supported hardware and rigid formulation for describing experiments. Instead it should provide affordances that give a clear way for users to extend the system to fit their needs [43] . In addition to integrating with existing systems, it must be straightforward for future development to be integrated. This idea is related to "the test of independent invention", summarized with the question "if someone else had already invented your system, would theirs work with yours?" [44] .

This principle also has tactical elements. An uncoercive system allows users to gradually adopt it rather than needing to adopt all of its components in order for any one of them to be useful. There always needs to be a *benefit* to adopting further components of the system to encourage *voluntary* adoption, but it should never be *compulsory*. For example, again from experimental frameworks, it should be possible to use it to control experimental hardware without needing to use the rest of the experimental design, data storage, and interface system. To some degree this is accomplished with a modular system design where designers are mindful of keeping the individual modules independently useful.

A noncoercive architecture also prioritizes the ease of leaving. Though this is somewhat tautological to protocol-driven design, specific care must be taken to enable export and migration to new systems. Making leaving easy also ensures that early missteps in development of the system are not fatal to its development, preventing lock-in to a component that needs to be restructured.

!! the coercion of centralization has a few forms. this is related to the authoritarian impulse in the open science movement that for awhile bullied people into openness. that instinct in part comes from a belief that everyone should be doing the same thing, should be posting their work on the one system. decentralization is about autonomy, and so a reciprocal approach is to make it easy and automatic.

3.1.4 Empower People, not Systems

Because IP was initially developed as a military technology by DARPA, a primary design constraint was survivability in the face of failure. The model adopted by internet architects was to move as much functionality from the network itself to the end-users of the network — rather than the network itself guaranteeing a packet is transmitted, the sending computer will do so by requiring a response from the recipient [42] .

For infrastructure, we should make tools that don't require a central team of developers to maintain, a central server-farm to host data, or a small group of people to govern. Whenever possible, data, software, and hardware should be self-describing, so one needs minimal additional tools or resources to understand and use it. It should never be the case that funding drying up for one node in the system causes the entire system to fail.

Practically, this means that the tools of digital infrastructure should be deployable by individual people and be capable of recapitulating the function of the system without reference to any central authority. Researchers need to be given control over the function of infrastructure: from controlling sharing permissions for eg. clinically sensitive data to assurance that their tools aren't spying on them. Formats and standards must be negotiable by the users of a system rather than regulated by a central governance body.

3.1.5 Infrastructures need Communities

The alternative to centralized governing and development bodies is to build the tools for community control over infrastructural components. This is perhaps the largest missing piece in current scientific tooling. On one side, decentralized governance is the means by which an infrastructure can be maintained to serve the ever-evolving needs of its users. On the other, a sense of community ownership is what drives people to not only adopt but contribute to the development of an infrastructure. In addition to a potentially woo-woo sense of socially affiliative "communityness," any collaborative system needs a way of ensuring that the practice of maintaining, building, and using it is designed to *visibly and tangibly benefit* those that do, rather than be relegated to a cabal of invisible developers and maintainers [**grudinGroupwareSocialDynamics1994 randallDistributed**
.

Governance and communication tools also make it possible to realize the infinite variation in application that infrastructures need while keeping them coherent: tools must be built with means of bringing the endless local conversations and modifications of use into a common space where they can become a cumulative sense of shared memory.

This idea will be given further treatment and instantiation in a later discussion of the social dynamics of private bittorrent trackers, and is necessarily diffuse because of the desire to not be authoritarian about the structure of governance.

3.1.6 Usability Matters

It is not enough to build a technically correct technology and assume it will be adopted or even useful, it must be developed embedded within communities of practice and *be useful for solving problems that people actually have.* We should learn from the struggles of the semantic web project. Rather than building a fully prescriptive and complete system first and instantiating it later, we should develop tools whose usability is continuously improved *en route* to a (flexible) completed

vision.

The adage from RFC 1958 "nothing gets standardized until there are multiple instances of running code" [43] captures the dual nature of the constraint well. Workable standards don't emerge until they have been extensively tested in the field, but development without an eye to an eventual protocol won't make one.

We should read the gobbling up of open protocols into proprietary platforms that defined "Web 2.0" as instructive (in addition to a demonstration of the raw power of concentrated capital) [45] . *Why* did Slack outcompete IRC? The answer is relatively simple: it was relatively simple to use. Using a contemporary example, to set up a Synapse server to communicate over Matrix one has to wade through dozens of shell commands, system-specific instructions, potential conflicts between dependent packages, set up an SQL server... and that's just the backend, we don't even have a frontend client yet! In contrast, to use Slack you download the app, give it your email, and you're off and running.

The control exerted by centralized systems over their system design does give certain structural advantages to their usability, and their for-profit model gives certain advantages to their development process. There is no reason, however, that decentralized systems *must* be intrinsically harder to use, we just need to focus on user experience to a comparable degree that centralized platforms: if it takes a college degree to turn the water on, that ain't infrastructure.

People are smart, they just get frustrated easily. We have to raise our standards of design such that we don't expect users to have even a passing familiarity with programming, attempting to build tools that are truly general use. We can't just design a peer-to-peer system, we need to make the data ingestion and annotation process automatic and effortless. We can't just build a system for credit assignment, it needs to happen as an automatic byproduct of using the system. We can't just make tools that *work,* they need to *feel good to use.*

Centralized systems also have intrinsic limitations that provide openings for decentralized systems, like cost, incompatibility with other systems, inability for extension, and opacity of function. The potential for decentralized systems to capture the independent development labor of all of its users, rather than just that of a core development team, is one means of competition. If the barriers to adoption can be lowered, and the benefits raised these constant negative pressures of centralization might overwhelm intertia.

With these principles in mind, and drawing from other knowledge communities solving similar problems: internet infrastructure, library/information science, peer-to-peer networks, and radical community organizers, I conceptualize a system of distributed infrastructure for systems neuroscience as three objectives: **shared data**, **shared tools**, and **shared knowledge**.

3.2 Shared Data

Format Standardization as an Onramp

The shallowest onramp towards a generalized data infrastructure is to make use of existing discipline-specific standardized data formats. As will be discussed later, a truly universal pandisciplinary for-

mat is effectively impossible, but to arrive at the alternative we should first congeal the wild west of unstandardized data into a smaller number of established formats.

Data formats consist of some combination of an abstract specification, an implementation in a particular storage medium, and an API for interacting with the format. I won't dwell on the particular qualities that a particular format needs, assuming that most that would be adopted would abide by FAIR principles. For now we assume that the particular constellation of these properties that make up a particular format will remain mostly intact with an eye towards semantically linking specifications and unifying their implementation.

There are a dizzying number of scientific data formats [46] , so a comprehensive treatment is impractical here and I will use the Neurodata Without Borders:N (NWB)[47] as an example. NWB is the de facto standard for systems neuroscience, adopted by many institutes and labs, though far from uniformly. NWB consists of a specification language, a schema written in that language, a storage implementation in hdf5, and an API for interacting with the data. They have done an admirable job of engaging with community needs [48] and making a modular, extensible format ecosystem.

The major point of improvement for NWB, and I imagine many data standards, is the ease of conversion. The conversion API requires extensive programming, knowledge of the format, and navigation of several separate tutorial documents. This means that individual labs, if they are lucky enough to have some partially standardized format for the lab, typically need to write (or hire someone to write) their own software library for conversion.

Without being prescriptive about its form, substantial interface development is needed to make mass conversion possible. It's usually untrue that unstandardized data had *no structure,* and researchers are typically able to articulate it – "the filenames have the data followed by the subject id," and so on. Lowering the barriers to conversion mean designing tools that match the descriptive style of folk formats, for example by prompting them to describe where each of an available set of metadata fields are located in their data. It is not an impossible goal to imagine a piece of software that can be downloaded and with minimal recourse to reference documentation allow someone to convert their lab's data within an afternoon. The barriers to conversion have to be low and the benefits of conversion have to outweigh the ease of use from ad-hoc and historical formats.

NWB also has an extension interface, which allows, for example, common data sources to be more easily described in the format. These are registered in an extensions catalogue, but at the time of writing it is relatively sparse. The preponderance of lab-specific conversion packages relative to extensions is indicative of an interface and community tools problem: presumably many people are facing similar conversion problems, but because there is not a place to share these techniques in a human-readable way, the effort is duplicated in dispersed codebases. We will return to some possible solutions for knowledge preservation and format extension when we discuss tools for shared knowledge.

For the sake of the rest of the argument, let us assume that some relatively trivial conversion process exists to subdomain-specific data formats and we reach some reasonable penetrance of standardization. The interactions with the other pieces of infrastructure that may induce and incentivize conversion will come later.

3.2.1 Peer-to-peer data sharing platform

We should adopt a *peer-to-peer* system for storing and sharing scientific data. There are, of course many existing databases for scientific data, ranging from domain-general like figshare and zenodo to the most laser-focused subdiscipline-specific. The notion of a database, like a data standard, is not monolithic. As a simplification, they consist of at least the hardware used for storage, the software implementation of read, write, and query operations, a formatting schema, some API for interacting with it, the rules and regulations that govern its use, and especially in scientific databases some frontend for visual interaction. For now we will focus on the storage software and read-write system, returning to the format, regulations, and interface later.

Centralized servers are fundamentally constrained by their storage capacity and bandwidth, both of which cost money. In order to be free, database maintainers need to constantly raise money from donations or grants[7] in order to pay for both. Funding can never be infinite, and so inevitably there must be some limit on the amount of data that someone can upload and the speed at which it can serve files[8]. In the case that a researcher never sees any of those costs, they are still being borne by some funding agency, incurring the social costs of funneling money to database maintainers. Centralized servers are also intrinsically out of the control of their users, requiring them to abide whatever terms of use the server administrators set. Even if the database is carefully backed up, it serves as a single point of infrastructural failure, where if the project lapses then at worst data will be irreversibly lost, and at best a lot of labor needs to be expended to exfiltrate, reformat, and rehost the data. The same is true of isolated, local, institutional-level servers and related database platforms, with the additional problem of skewed funding allocation making them unaffordable for many researchers.

Peer-to-peer (p2p) systems solve many of these problems, and I argue are the only type of technology capable of making a database system that can handle the scale of all scientific data. There is an enormous degree of variation between p2p systems[9], but they share a set of architectural advantages. The essential quality of any p2p system is that rather than each participant in a network interacting only with a single server that hosts all the data, everyone hosts data and interacts directly with each other.

For the sake of concreteness, we can consider a (simplified) description of Bittorrent [50] , arguably the most successful p2p protocol. To share a collection of files, a user creates a `.torrent` file which consists of a cryptographic hash, or a string that is unique to the collection of files being shared; and a list of "trackers." A tracker, appropriately, keeps track of the `.torrent` files that have been uploaded to it, and connects users that have or want the content referred to by the `.torrent` file. The uploader (or seeder) then leaves a torrent client open waiting for incoming connections. Someone who wants to download the files (a leecher) will then open the `.torrent` file in their client, which will then ask the tracker for the IP addresses of the other peers who are seeding the file, directly connect to them, and begin downloading. So far so similar to standard client-server systems, but the magic is just getting started. Say another person wants to download the same files

[7] granting agencies seem to love funding new databases, idk.
[8] As I am writing this, I am getting a (very unscientific) maximum speed of 5MB/s on the Open Science Framework
[9] peer to peer systems are, maybe predictably, a whole academic subdiscipline. See [49] for reference.

before the first person has finished downloading it: rather than *only* downloading from the original seeder, the new leecher downloads from *both* the original seeder and the first leecher. Leechers are incentivized to share among each other to prevent the seeders from spending time reuploading the pieces that they already have, and once they have finished downloading they become seeders themselves.

From this very simple example, a number of qualities of p2p systems become clear.

- First, the system is extremely **inexpensive to maintain** since it takes advantage of the existing bandwidth and storage space of the computers in the swarm, rather than dedicated servers. Near the height of its popularity in 2009, The Pirate Bay, a notorious bittorrent tracker, was estimated to cost $3,000 per month to maintain while serving approximately 20 million peers [51] . According to a database dump from 2013 [52] , multiplying the size of each torrent by the number of seeders (ignoring any partial downloads from leechers), the approximate instantaneous storage size of The Pirate Bay was ~26 Petabytes. The comparison to centralized services is not straightforward, since it is hard to evaluate the distributed costs of additional storage media (as well as the costs avoided by being able to take advantage of existing storage infrastructure within labs and institutes), but for the sake of illustration: hosting 26PB would cost $546,000/month with standard AWS S3 hosting ($0.021/GB/month).
- The **speed** of a bittorrent swarm *increases,* rather than decreases, the more people are using it since it is capable of using all of the available bandwidth in the system.
- The network is extremely **resilient** since the data is shared across many independent peers in the system. If our goal is to make a resilient and robust data architecture, we would benefit by paying attention to the tools used in the broader archival community, especially the archival communities that especially need resilience because their archives are frequent targets of governments and IP-holders[53] . Despite more than 15 years of concerted effort by governments and intellectual property holders, the pirate bay is still alive and kicking [54] [10]. This is because even if the entire infrastructure of the tracker is destroyed, as it was in 2006, the files are distributed across all of its users, the actual database of `.torrent` metadata is quite small, and the tracker software is extraordinarily simple to rehost [55] – The Pirate Bay was back online in 2 days. When another tracker, what.cd (which we will return to soon) was shut down, a series of successors popped up using the open source tools Gazelle and Ocelot that what.cd developers built. Within two weeks, one successor site had recovered and reindexed 200,000 of its torrents resubmitted by former users [56] . Bittorrent is also used by archival groups with little funding like Archive Team, who struggled – but eventually succeeded – to disseminate their historic preservation over a single "crappy cable modem" [57] . And by groups who disseminate !! return here talking about ddosevrets.
- The network is extremely **scalable** since there is no cost to connecting new peers and the users of a system expand the storage capacity of the system depending on their

[10] knock on wood

needs. Rather than having one extremely fast data center (or a privatized network designed to own the internet), the model of p2p systems is to leverage many approachable peer/servers.

Peer-to-peer systems are not mutually exclusive with centralized servers: servers are peers too, after all. A properly implemented will always be *at least* as fast and have *at least* as much storage as any alternative centralized centralized server because peers can use *both* the bandwidth of the server *and* that of any peers that have the file. In the bittorrent ecosystem large-bandwidth/storage peers are known as "seedboxes"[58] when they use the bittorrent protocol, and "web seeds"[59] when they use a protocol built on top of traditional HTTP. Archive.org has been distributing all of its materials with bittorrent by using its servers as web seeds since 2012 and makes this point explicitly: "BitTorrent is now the fastest way to download items from the Archive, because the Bittorrent client downloads simultaneously from two different Archive servers located in two different datacenters, and from other Archive users who have downloaded these Torrents already." [60]

p2p systems complement centralized servers in a number of ways beyond raw download speed, increasing the efficiency and performance of the network as a whole. Spotify began as a joint client/server and p2p system [61] , where when a listener presses play the central server provides the data until peers that have the song cached are found by the p2p system to download the rest of the song from. The central server is able to respond quickly and reliably to so the song is played as quickly as possible, and is the server of last resort in the case of rare files that aren't being shared by anyone else in the network. A p2p system complements the server and makes that possible by alleviating pressure on the server for more predictable traffic.

A peer to peer system is a particularly natural fit for many of the common circumstances and practices in science, where centralized server architectures seem (and prove) awkward and inefficient. Most labs, institutes, or other organized bodies of science have some form of local or institutional storage systems. In the most frequent cases of sharing data within a lab or institute, sending it back and forth to some nationally-centralized server is like walking across the lab by going the long way around the Earth. That's the method invoked by a Dropbox or AWS link, but in the absence of a formal one you can always revert to a low-fi p2p transfer: walking a flash drive across the lab. The system makes less sense when several people in the same place need to access the same data at the same time, as is frequently the case with multi-lab collaborations, or scientific conferences and workshops. Instead of needing to wait on the 300kb/s conference wifi bandwidth as it's cheese-gratered across every machine, we instead could directly beam it between all computers in range simultaneously, full blast through the decrepit network switch that won't have seen that much excitement in years.

!! if we take the suggestion of Andrey Andreev et al. and invest in server clusters within institutes [62] , their impact could be multiplied manyfold by being able to use them all fluidly and simultaneously for file transfer and storage. !! compatible and extends calls for more institutional support for storage liek andreev's paper, but satisfies the need for generalized storage systems that the NIH doesn't have to develop a whole new institute to handle. extra bonus! in that system each server would have to serve the entire file each time. WIth p2p then the load can be spread between all of them, decreasing costs for all institutions!!!!

So far I have relied on the Extraordinarily Simplified Bittorrent™ depiction of a peer to peer system, but there are many improvements and variants that can address different needs for scientific data infrastructure.

One obvious need that bittorrent can't currently support is version control, but more recent p2p systems do. IPFS functions like "a single BitTorrent swarm, exchanging objects within one Git repository." [63] [11] Dat [64] , specifically designed for data synchronization and versioning, handles versioning and more. A full description of IPFS is out of scope, and it has plenty of problems [65] , but for now sufficent to say p2p systems can handle version control.

Bittorrent swarms are vulnerable to data loss if all the peers seeding a file disconnect (though the tail is longer than typically assumed, see [66]), but this too can be addressed with updated p2p system design. A first-order solution to this problem is a variant of IPFS' notion of 'pinning.' Since backup to lab-level or institutional servers is already commonplace, one peer could be able to 'pin' another and automatically download all the data that they share. This concept could scale to institutes and national infrastructure as scientists can request the datasets they'd like to be saved permanently be pinned.

Another could be something akin to Freenet [67] . Peers could allocate a certain amount of their unused storage space to be used to automatically download, cache, and rehost shards of other datasets. Distributing chunks and encrypting them at rest so the rehoster can't inspect their contents would make it possible to maintain privacy and network availability for sensitive data (see, for example, ERIS). IPFS has an analogous concept – BitSwap – that is makes it into a barter system. Peers who seek to download will have to 'earn' it by finding some chunk of data that the other peers want, download, and share them, though it seems like an empirical question whether or not a barter system works or is necessary.

There are a number of additional requirements for a peer to peer scientific data infrastructure, but even these seemingly very technical problems of versioning and distributed storage show the clear need to consider the structure of the surrounding social system. What control do we give to researchers over the version history of their data? Should people that aren't the originating researcher be able to issue new versions? What structure of distributed/centralized storage works? How should we incentivize sharing of excess storage and resources?

Even before considering additional social systems, a peer to peer structure in itself implies a different relationship to a generalized data infrastructure. Scientists always unavoidably make their data available to at least one person: themselves; on at least one computer: theirs. A peer-to-peer backbone for scientific infrastructure is the unnecessarily radical notion that everyday practices like these can make up our infrastructure, rather than having it exist exogenously as something "out there." Subtly, it's the notion that our infrastructure can reflect and consist of *ourselves* instead of something out of our control that we need to buy from someone else.

Scientists don't need to reinvent the notion of distributed, community curated data archives from scratch. In addition to scholarly work on the social systems of digital infrastructure, we can learn

[11] Git, briefly, is a version control system that keeps a history of changes of files (blobs) as a Merkle DAG: files can be updated, and different versions can be branched and reconciled.

from communities of practice, and there has been no more important and impactful decentralized archival project than internet piracy.

3.2.2 Archives Need Communities

Why do hundreds of thousands of people, completely anonymously, with zero compensation, spend their time to do something that is as legally risky as curating pirated cultural archives?

Scholarly work, particularly from Economics, tends to focus on understanding piracy in order to prevent it[**basamanowiczReleaseGroupsDigital2011 hindujaDeindividuationInternetSoftware2008**] , taking the moral good of intellectual property markets as an *a priori* imperative and investigating why people behave *badly* and "rend [the] moral fabric associated with the respect of intellectual property." [68] . If we put the legality of piracy aside, we may find a wealth of wisdom and insight to draw from for building scientific infrastructure.

The world of digital piracy is massive, from entirely disorganized efforts of individual people on public sites to extraordinarily organized release groups [69] , and so a full consideration is out of scope, but many of the important lessons are taught by the structure of bittorrent trackers.

An underappreciated element of the BitTorrent protocol is the effect of the separation between the data transfer protocol and the 'discovery' part of the system — or "overlay" — on the community structure of torrent trackers (for a more complete picture of the ecosystem, see [66]). Many peer to peer networks like KaZaA or the gnutella-based Limewire had searching for files integrated into the transfer interface. The need for torrent trackers to share .torrent files spawned a massive community of private torrent trackers that for decades have been iterating on cultures of archival, experimenting with different community structures and incentives that encourage people to share and annotate some of the world's largest, most organized libraries.

One of these private trackers was the site of one of the largest informational tragedies of the past decade: what.cd[12], which I will use as an example to describe some of these community systems.

What.cd was a bittorrent tracker that was arguably the largest collection of music that has ever existed. At the time of its destruction in 2016, it was host to just over one million unique releases, and approximately 3.5 million torrents[13] [70] . Every torrent was organized in a meticulous system of metadata communally curated by its roughly 200,000 global users. The collection was built by people who cared deeply about music, rather than commercial collections provided by record labels notorious for ceasing distribution of recordings that are not commercially viable — or just losing them in a fire [71] [14]. Users would spend large amounts of money to find and digitize extremely rare recordings, many of which were unavailable anywhere else and are now unavailable anywhere,

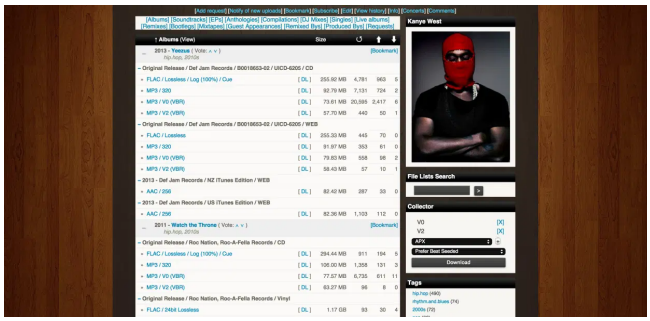
[12] for a detailed description of the site and community, see Ian Dunham's dissertation [70]

[13] Though spotify now boasts its library having 50 million tracks, back of the envelope calculations relating number of releases to number of tracks are fraught, given the long tail of track numbers on albums like classical music anthologies with several hundred tracks on a single "release."

[14]

period. One former user describes one example:

> "I did sound design for a show about Ceaușescu's Romania, and was able to pull together all of this 70s dissident prog-rock and stuff that has never been released on CD, let alone outside of Romania" [72]



The what.cd artist page for Kanye West (taken from here in the style of pirates, without permission). For the album "Yeezus," there are ten torrents, grouped by each time the album was released on CD and Web, and in multiple different qualities and formats (.flac, .mp3). Along the top is a list of the macro-level groups, where what is in view is the "albums" section, there are also sections for bootleg recordings, remixes, live albums, etc.

What.cd was a "private" bittorrent tracker, where unlike public trackers that anyone can access, membership was strictly limited to those who were personally invited or to those who passed an interview (for more on public and private tracker, see [73]). Invites were extremely rare, and the interview process was demanding to the point where

"Among the incinerated Decca masters were recordings by titanic figures in American music: Louis Armstrong, Duke Ellington, Al Jolson, Bing Crosby, Ella Fitzgerald, Judy Garland. The tape masters for Billie Holiday's Decca catalog were most likely lost in total. The Decca masters also included recordings by such greats as Louis Jordan and His Tympany Five and Patsy Cline.

The fire most likely claimed most of Chuck Berry's Chess masters and multitrack masters, a body of work that constitutes Berry's greatest recordings. The destroyed Chess masters encompassed nearly everything else recorded for the label and its subsidiaries, including most of the Chess output of Muddy Waters, Howlin' Wolf, Willie Dixon, Bo Diddley, Etta James, John Lee Hooker, Buddy Guy and Little Walter. Also very likely lost were master tapes of the first commercially released material by Aretha Franklin, recorded when she was a young teenager performing in the church services of her father, the Rev. C.L. Franklin, who made dozens of albums for Chess and its sublabels.

Virtually all of Buddy Holly's masters were lost in the fire. Most of John Coltrane's Impulse masters were lost, as were masters for treasured Impulse releases by Ellington, Count Basie, Coleman Hawkins, Dizzy Gillespie, Max Roach, Art Blakey, Sonny Rollins, Charles Mingus, Ornette Coleman, Alice Coltrane, Sun Ra, Albert Ayler, Pharoah Sanders and other jazz greats. Also apparently destroyed were the masters for dozens of canonical hit singles, including Bill Haley and His Comets' "Rock Around the Clock," Jackie Brenston and His Delta Cats' "Rocket 88," Bo Diddley's "Bo Diddley/I'm A Man," Etta James's "At Last," the Kingsmen's "Louie Louie" and the

The list of destroyed single and album masters takes in titles by dozens of legendary artists, a genre-spanning who's who of 20th- and 21st-century popular music. It includes recordings by Benny Goodman, Cab Calloway, the Andrews Sisters, the Ink Spots, the Mills Brothers, Lionel Hampton, Ray Charles, Sister Rosetta Tharpe, Clara Ward, Sammy Davis Jr., Les Paul, Fats Domino, Big Mama Thornton, Burl Ives, the Weavers, Kitty Wells, Ernest Tubb, Lefty Frizzell, Loretta Lynn, George Jones, Merle Haggard, Bobby (Blue) Bland, B.B. King, Ike Turner, the Four Tops, Quincy Jones, Burt Bacharach, Joan Baez, Neil Diamond, Sonny and Cher, the Mamas and the Papas, Joni Mitchell, Captain Beefheart, Cat Stevens, the Carpenters, Gladys Knight and the Pips, Al Green, the Flying Burrito Brothers, Elton John, Lynyrd Skynyrd, Eric Clapton, Jimmy Buffett, the Eagles, Don Henley, Aerosmith, Steely Dan, Iggy Pop, Rufus and Chaka Khan, Barry White, Patti LaBelle, Yoko Ono, Tom Petty and the Heartbreakers, the Police, Sting, George Strait, Steve Earle, R.E.M., Janet Jackson, Eric B. and Rakim, New Edition, Bobby Brown, Guns N' Roses, Queen Latifah, Mary J. Blige, Sonic Youth, No Doubt, Nine Inch Nails, Snoop Dogg, Nirvana, Soundgarden, Hole, Beck, Sheryl Crow, Tupac Shakur, Eminem, 50 Cent and the Roots.

Then there are masters for largely forgotten artists that were stored in the vault: tens of thousands of gospel, blues, jazz, country, soul, disco, pop, easy listening, classical, comedy and spoken-word records that may now exist only as written entries in discographies." [71]

entire guides were written to prepare for them. When I interviewed in 2009, I had to find my way onto an obscure IRC server, wait in a lobby all day until a volunteer moderator could get to me, and was then grilled on the arcana of digital music formats, spectral analysis[15], the ethics of piracy, and so on for half an hour. Getting a question wrong was an instant failure and you were banned from the server for 48 hours. A single user was only allowed one account per lifetime, so between that policy and the extremely high barriers to entries, even anonymous users were strongly incentivized to follow the sophisticated, exacting rules for contributing. While we certainly don't want such a grueling barrier to entry for scientific data infrastructure, the problem is different and arguably simpler when the system can exist in the open. For example public reputation loss can be a reasonably strong incentive to play by the rules that may trade off with the threat of banning.

The what.cd incentive system was based on a required ratio of data uploaded vs. data downloaded [74] . Peer to peer systems need to overcome a free-rider problem where users might download a torrent ("leeching") and turn their computer off, rather than leaving their connection open to share it to others (or, "seeding"). In order to download additional music, then, one would have to upload more. Since downloading is highly restricted, and everyone is trying to upload as much as they can, torrents had a large number of "seeders," and even rare recordings would be sustained for years, a pattern common to private trackers [75] .

The high seeder/leecher ratio made it so it was extremely difficult to acquire upload credit, so users were additionally incentivized to find and upload new recordings to the system. What.cd implemented a "bounty" system, where users with a large amount of excess upload credit would be able to offer some of it to whoever was able to upload the album they wanted. To "prime the pump" and keep the economy moving, highlight artists in an album of the week, or direct users to preserve rare recordings, moderators would also use a "freeleech" system, where users would be able to download a specified set of torrents without it counting against their download quantity [**kashEconomicsBitTorrentCommunities2012 chenImprovi**
.

Depending on the age of your account and the amount you had contributed, what.cd users also were given *user classes* that conferred differing degrees of prestige and abilities. This is a common tactic for publicly moderates sites like StackExchange or Genius, where users need to demonstrate a certain degree of competency and good faith before they are given the keys to the castle. User classes are both *aspirational* and incentivize additional work on the site, as well as *reputational* where a user class meant you have paid your dues and were a senior contributor.

The other half of what.cd was the more explicitly social elements: its forums, comment sections, and moderation systems. The forum was home to roiling debates that lasted years about the structure of some tagging schema, whether one genre was just another with a different name, and so on. The structure of the community was an object of constant, public negotiation, and over time the metadata system evolved to be able to support

[15]The average what.cd user was, as a result, on par with many of the auditory neuroscientists I know in their ability to read a spectrogram.

a library of the entirety of human music output[16], and the rules and incentive structures were made to align with building it. To support the good operation of the site, the forums were also home to a huge amount of technical knowledge, like guides on how to make a perfect upload, that eased new users into being able to use the system.

A critical problem in maintaining coherent databases is correcting metadata errors and departures from schemas. Finding errors was rewarded. Users were able to discuss and ask questions of the uploader in a comment section below each upload, which would allow "polite" resolution of low-level errors like typos. More serious problems could be reported to the moderation team, which caused the upload to be visibly marked as under review, and the report could then be discussed either in the comment sections or the forum. If the moderation team affirmed your report, they would usually kick back a few gigabytes of upload credit depending on the severity. Unless the problem was a repeat and malicious one, the "offender" was alerted to it, warned, and told what to do instead next time – though, being an anonymous, gray-area community, there was plenty of power that was tripped on. Rather than being a messy hodgepodge of fake, low-quality uploads, what.cd was always teetering just shy of perfection.

These structural considerations do not capture the most elusive but indisputably important features of what.cd's community infrastructure: *the sense of commmunity*. The What.cd forums were the center of many user's relationships to music. Threads about all the finest scales of music nichery could last for years: it was a rare place people who probably cared a little bit too much about music could talk to people with the same condition. What made it more satisfying than other music forums was that no matter what music you were talking about, everyone else in the conversation would always have access to it if they wanted to hear it. Independent musicians released albums in the supportive[17] Vanity House section, and people from around the world came to hold the one true album that only they knew about high aloft like a divine tablet. Beyond any structural incentives, people spent so much time building and maintaining what.cd because it became a source of community and a sink of personal investment.

Structural norms supported by social systems converge as a sort of *reputational* incentive. Uploading a new album to fill a bounty both makes the network more functional and complete, but it also *people respect you for it* because it's prominently displayed on your profile as well as in the bounty charts and that *feels good*. Becoming known on the forums for answering questions, writing guides, or even just having a good taste in music *feels good* and also contributes to the overall health of the system. Though there are plenty of databases, and even plenty of different communication venues for scientists, there aren't any databases (to my knowledge) with integrated community systems.

The tracker overlay model mirrors and extends some of the recommendations made by Benedikt Fecher and colleagues in their work on the reputational economy surrounding data sharing [76]

[16] Though music metadata might seem like a trivial problem (just look at the fields in an MP3 header), the number of edge cases are profound. How would you categorize an early Madlib casette mixtape remastered and uploaded to his website where he is mumbling to himself while recording some live show performed by multiple artists, but on the b-side is one of his Beat Konducta collections that mix together studio recordings from a collection of other artists? Who is the artist? How would you even identify the unnamed artists in the live show? Is that a compilation or a bootleg? Is it a cassette rip, a remaster, or a web release?

[17] Mostly. You know how the internet goes...

. They give three policy recommendations: Increasing reputational benefits, reducing transaction costs, and "increasing market transparency by making open access to research data more visible to members of the research community." The primary problem, in their eye, is that the reputational reward of data sharing is too small. In addition to increasing transparency, another way of increasing the reputational reward to sharing data is to embed it within a social system that is designed to reward communitarian behavior with reputational rewards. They continue to ideas like greater reward for data citations (which we will return to in credit assignment), as well as awards for good datasets. Community awards are also longstanding parts of many digital communities, like What.cd's Album of the Week, which rewarded someone who has done good work by letting them choose an album that would be freely downloadable, or Wikipedia's Barnstars.

Many features of what.cd's structure are undesirable for scientific infrastructure, but they demonstrate that a robust archive is not only a matter of building a database with some frontend, but by building a community [77] . Of course, we need to be careful with building the structural incentives for a data sharing system: the very last thing we want is another coercive leaderboard. In contrast to what.cd, for infrastructure we want extremely low barriers to entry, and be agnostic to resources — researchers with access to huge server farms should not be unduly favored. We shouldn't use downloading as the "cost," because downloading and analyzing huge amounts of data is *good* and what we *want*. A better system for science might closer to ratioless trackers that allow infinite downloads as long as they remain seeded for a certain amount of time afterwards.

These are all solvable problems, and can be worked on iteratively. They hint at a communication medium where we can discuss our experiments in the same place that they live; linking, embedding, comparing data and techniques to have the kind of longform, cumulative scientific discourse that is for now still relegated to being a fever dream. Rather than being prescriptive about one community structure, what allowed private bittorent trackers to develop and experiment with many different types of systems is the separation from the underlying data from the community overlay.

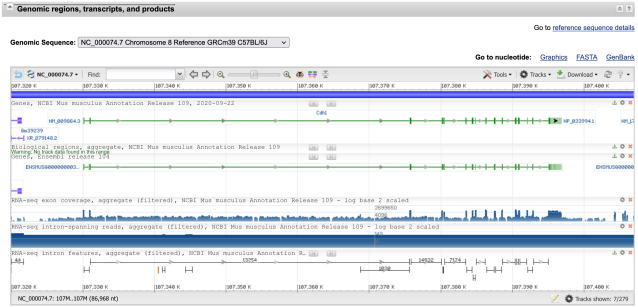
This model has its own problems, including the lack of interoperability between different trackers, the need to recreate a new set of accounts and database for each new tracker, among others. It's also been tried before: sharing data in specific formats (as our running example, Neurodata Without Borders) on indexing systems like bittorrent trackers amounts to something like BioTorrents [78] or AcademicTorrents [79] . Even with our extensions of version control and some model of automatic mirroring of data across the network, we still have some work to do. To address these and several other remaining needs for scientific data infrastructure, we can take inspiration from *federated systems*.

3.2.3 Linked Data or Surveillance Capitalism?

There is no shortage of databases for scientific data, but their traditional structure chokes on the complexity of representing multi-domain data. Typical relational databases require some formal schema to structure the data they contain, which have varying reflections in the APIs used to access

them and interfaces built atop them. This broadly polarizes database design into domain-specific and domain-general[18]. This design pattern results in a fragmented landscape of databases with limited interoperability. In a moment we'll consider *federated systems* as a way to resolve this dichotomy and continue developing the design of our p2p data infrastructure, but for now we need a better sense of the problem.

Domain-specific databases require data to be in one or a few specific formats, and usually provide richer tools for manipulating and querying by metadata, visualization, summarization, aggregation that are purpose-built for that type of data. For example, NIH's Gene tool has several visualization tools and cross-referencing tools for finding expression pathways, genetic interactions, and related sequences (Figure xx). This pattern of database design is reflected at several different scales, through institutional databases and tools like the Allen brain atlases or observatory, to lab- and project-specific dashboards. This type of database is natural, expressive, and powerful — for the researchers they are designed for. While some of these databases allow open data submission, they often require explicit moderation and approval to maintain the guaranteed consistency of the database, which can hamper mass use.



NIH's Gene tool included many specific tools for visualizing, cross-referencing, and aggregating genetic data. Shown is the "genomic regions, transcripts, and product" plot for Mouse Cdh1, which gives useful, common summary descriptions of the gene, but is not useful for, say, visualizing reading proficiency data.

General-purpose databases like figshare and zenodo[19] are useful for the mass aggregation of data, typically allowing uploads from most people with minimal barriers. Their general function limits the metadata, visualization, and other tools that are offered by domain-specific databases, however, and are essentially public, versioned, folders with a DOI. Most have fields for authorship, research groups, related publications, and a single-dimension keyword or tags system, and so don't programmatically reflect the metadata present in a given dataset.

The dichotomy of fragmented, subdomain-specific databases and general-purpose databases makes combining information from across even extremely similar subdisciplines combinatorically complex and laborious. In the absence of a formal interoperability and indexing protocol between databases, even *finding* the correct subdomain-specific database can be an act of raw experience or the raw luck of stumbling across just the right blog post list of databases. It also puts researchers who want to be good data stewards in a difficult position: they can hunt down the appropriate subdomain specific database and risk general obscurity;

[18] To continue the analogy to bittorrent trackers, an example domain-specific vs. domain-general dichotomy might be What.cd (with its specific formatting and aggregation tools for representing artists, albums, collections, genres, and so on) vs. ThePirateBay (with its general categories of content and otherwise search-based aggregation interface)

[19] No shade to Figshare, which, among others, paved the way for open data and are a massively useful thing to have in society.

use a domain-general database and make their work more difficult for themselves and their peers to use; or spend all the time it takes to upload to multiple databases with potentially conflicting demands on format.

What can be done? There are a few parsimonious answers from standardizing different parts of the process: If we had a universal data format, then interoperability becomes trivial. Conversely, we could make a single ur-database that supports all possible formats and tools.

Universalizing a single part of a database system is unlikely to work because organizing knowledge is intrinsically political. Every system of representation is necessarily rooted in its context: one person's metadata is another person's data. Every subdiscipline has conflicting *representational* needs, will develop different local terminology, allocate differing granularity and develop different groupings and hierarchies for the same phenomena. At mildest, differences in representational systems can be incompatible, but at their worst they can reflect and reinforce prejudices and become tools of intellectual and social power struggles. Every subdiscipline has conflicting *practical* needs, with infinite variation in privacy demands, different priorities between storage space, bandwidth, and computational power, and so on. In all cases the boundaries of our myopia are impossible to gauge: we might think we have arrived at a suitable schema for biology, chemistry, and physics... but what about the historians?

Matthew J Bietz and Charlotte P Lee articulate this tension better than I can in their ethnography of metagenomics databases:

> "Participants describe the individual sequence database systems as if they were shadows, poor representations of a widely-agreed-upon ideal. We find, however, that by looking across the landscape of databases, a different picture emerges. Instead, **each decision about the implementation of a particular database system plants a stake for a community boundary. The databases are not so much imperfect copies of an ideal as they are arguments about what the ideal Database should be.** [...]
>
> When the microbial ecology project adopted the database system from the traditional genomic "gene finders," they expected the database to be a boundary object. They knew they would have to customize it to some extent, but thought it would be able to "travel across borders and maintain some sort of constant identity". In the end, however, **the system was so tailored to a specific set of research questions that the collection of data, the set of tools, and even the social organization of the project had to be significantly changed.** New analysis tools were developed and old tools were discarded. Not only was the database ported to a different technology, the data itself was significantly restructured to fit the new tools and approaches. While the database development projects had begun by working together, in the end they were unable to collaborate. **The system that was supposed to tie these groups together could not be shielded from the controversies that formed the boundaries between the communities of practice.**" [9]

As one ascends the scales of formalizing to the heights of the ontology designers, the ideological nature of the project is like a klaxon (emphasis in original):

An exception is the Open Biomedical Ontologies (OBO) Foundry initiative, which accepts under its label only those ontologies that adhere to the principles of ontological realism. Where the prevailing, i.e. computer science, view of ontology is focused on the logical consistency and inferential implications of ontologies as sets of as-

> sertions, the view of the OBO Foundry is that the quality of an ontology is also - indeed primarily - determined by the accuracy with which it represents the preexisting structure of reality. Ontologies, from this perspective, are representational artifacts, comprising a taxonomy as their central backbone, whose representational units are intended to designate *universals* (such as *human being* and *patient role*) or *classes defined in terms of universals* (such as *patient,* a class encompassing *human beings* in which there inheres a *patient role*) and certain relations between them.
>
> [...]
>
> BFO is a realist ontology [15,16]. This means, most importantly, that representations faithful to BFO can acknowledge only those entities which exist in (for example, biological) reality; thus they must reject all those types of putative negative entities - lacks, absences, nonexistents, possibilia, and the like - which are sometimes postulated as artifacts of specific terminologies or of associated logical or computational frameworks [80]

Aside from unilateral standardization, another formulation that doesn't require existing server infrastructure to be dramatically changed is to link existing databases. The problem of linking databases is an old one with much well-trodden ground, and in the current regime of large server farms tend to find themselves somewhere close to metadata-indexing overlays. These overlays provide some additional tool that can translate and combine data between databases with some mapping between the terminology in the overlay and that of the individual databases. The NIH articulates this as a "Biomedical Data Translator" in its Strategic plan for Data Science:

> Through its Biomedical Data Translator program, the National Center for Advancing Translational Sciences (NCATS) is supporting research to develop ways to connect conventionally separated data types to one another to make them more useful for researchers and the public. The Translator aims to bring data types together in ways that will integrate multiple types of existing data sourcess, including objective signs and symptoms of disease, drug effects, and other types of biological data relevant to understanding the development of disease and how it progresses in patients. [30]

And NCATS elaborates it a bit more on the project "about" page (emphasis mine):

> As a result of recent scientific advances, a tremendous amount of data is available from biomedical research and clinical interactions with patients, health records, clinical trials and adverse event reports that could be useful for understanding health and disease and for developing and identifying treatments for diseases. **Ideally, these data would be mined** collectively to provide insights into the relationship between molecular and cellular processes (the targets of rational drug design) and the signs and symptoms of diseases. Currently, these very rich yet different data sources are housed in various locations, often in forms that are not compatible or interoperable with each other. - https://ncats.nih.gov/translator/about

The Translator is being developed by 28 institutions and nearly 200 team members as of 2019. They credit their group structure and flexible Other Transaction Award (OTA) funding mechanism for their successes [81] . OTA awards give the granting agency broad flexibility in to whom and for what money can be given, and consist of an ini-

tial competetive segment with possibility for indefinite noncompetitive extensions at the discretion of the agency [82] .

The project appears to be in a relatively early phase, and so it's relatively difficult to figure out exactly what it is that has been built. The projects page is currently a list of the leaders of different areas, but some parts of the project are visible through a bit of searching. They describe a registry of APIs for existing databases collected on their platform SmartAPI that are to be combined into a semantic knowledge graph [83] . There are many kinds of knowledge graphs, and we will return to them and other semantic web technologies in shared knowledge, but the Translator's knowledge graph explicitly sits "on top" of the existing databases as the only source of knowledge. Specifically, the graph structure consists of the nodes and edges of the biolink model [84] , and an edge is matched to a corresponding API that provides data for both elements. For each edge in the graph, then, a number of possible APIs can provide data without necessarily making a guarantee of consistency or accuracy.

They articulate a very similar set of beliefs about the impossibility of a unified dataset or ontology[20][83] , although arguably create one in biolink, and this problem seems to have driven the focus of the project away from linking data as such towards developing a graph-powered query engine. The Translator is being designed to use machine-learning powered "autonomous relay agents" that sift through the inhomogenous data from the APIs and are able to return a human-readable response, also generated with machine-learning. The final form of the translator is still unclear, but between SmartAPI, a seemingly-preliminary description of the reasoning engine [85] , and descriptions from contractors [86] , the machine learning component of the system could make it quite dangerous.

The intended use of the Translator seems to not be to directly search for and use the data itself, but to use the connected data to answer directed questions [85] — an example that is used repeatedly is

20

> First, we assert that a single monolithic data set that directly connects the complete set of clinical characteristics to the complete set of biomolecular features, including "-omics" data, will never exist because the number of characteristics and features is constantly shifting and exponentially growing. Second, even if such a single monolithic data set existed, all-vs.-all associations will inevitably succumb to problems with statistical power (i.e., the curse of dimensionality).9 Such problems will get worse, not better, as more and more clinical and biomolecular data are collected and become available. We also assert that there is no single language, software or natural, with which to express clinical and biomolecular observations—these observations are necessarily and appropriately linked to the measurement technologies that produce them, as well as the nuances of language. The lack of a universal language for expressing clinical and biomolecular observations presents a risk of isolation or marginalization of data that are relevant for answering a particular inquiry, but are never accessed because of a failure in translation.
>
> Based on these observations, our final assertion is that automating the ability to reason across integrated data sources and providing users who pose inquiries with a dossier of translated answers coupled with full provenance and confidence in the results is critical if we wish to accelerate clinical and translational insights, drive new discoveries, facilitate serendipity, improve clinical-trial design, and ultimately improve clinical care. This final assertion represents the driving motivation for the Translator system. [83]

drug discovery. For any given query of "drugs that could treat x disease," the system traces out the connected nodes in the graph from the disease to find its phenotypes, which are connected to genes, which might be connected to some drug, and so on. The Translator builds on top of a large number of databases and database aggregators, and so it then needs a way of comparing and ranking possible answers to the question. In a simple case, a drug that directly acted on several involved genes might be ranked higher than, say, one that acted only indirectly on phenotypes with many off-target effects.

As with any machine-learning based system, if the input data is biased or otherwise (inevitably) problematic then the algorithm can only reflect that. If it is the case that this algorithm remains proprietary (due to, for example, it being developed by a for-profit defense contractor that named it ROBOKOP [86]) harmful input data could have unpredictable long-range consequences on the practice of medicine as well as the course of medical research. Taking a very narrow sample of APIs that return data about diseases, I queried mydisease.info to see if it still had the outmoded definition of "transsexualism" as a disease [87] . Perhaps unsurprisingly, it did, and was more than happy to give me a list of genes and variants that supposedly "cause" it - see for yourself[^transgenes].

This is, presumably, the fragility and inconsistency the machine-learning layer was intended to putty over: if one follows the provenance of the entry for "gender identity disorder" (renamed in DSM-V), one reaches first the disease ontology DOID:1234 which seems to trace back into an entry in a graph aggregator Ontobee (Archive Link), which in turn lists this github repository **maintained by a single person** as its source[21].

If at its core the algorithm believes that being transgender is a disease, could it misunderstand and try to "cure" it? Even if it doesn't, won't it influence the surrounding network of entities with its links to genes, prior treatment, and so on in unpredictable ways? Combined with the online training that is then shared by other users of the translator [83] , socially problematic treatment and research practices could be built into our data infrastructure without any way of knowing their effect. In the long-run, an effort towards transparency could have precisely the opposite effect by being run through a series of black boxes.

A larger problem is reflected in the scope and evolving direction of the Translator when combined with the preceding discussion of putting all data in the hands of cloud platform holders. There is mission creep from the original NIH initiative language that essentially amounts to a way to connect different data sources — what could have been as simple as a translation table between different data standards and formats. The original funding statement from 2016 is similarly humble, and press releases through 2017 also speak mostly in terms of querying the data – though some ambition begins to creep in.

That is remarkably different than what is articulated in 2019 [83] to be much more focused on *inference* and *reasoning* from the graph structure of the linked data for the purpose of *automating drug discovery*. It seems like the original goal of making a translator in the sense of "translating data between formats" has morphed into

[21] I submitted a pull request to remove it. A teardrop in the ocean.

"translating data to language," with ambitions of providing a means of making algorithmic predictions for drug discovery and clinical practice rather than linking data [88] Tools like these have been thoroughly problematized elsewhere, eg. [**groteEthicsAlgorithmicDecisionmaking2020 obermeyerDissectingRacialBias2019 panchArtificialIntelligence**
89] .

As of September 2021, it appears there is still some work left to be done to make the Translator functional, but the early example illustrates some potential risks (emphases mine):

The strategy used by the Translator consortium in this case is to 1) identify phenotypes that are associated with [Drug-Induced Liver Injury] DILI, then 2) find genes which are correlated with these presumably pathological phenotypes, and then 3) identify drugs which target those genes' products. The rationale is that drugs which target gene products associated with phenotypes of DILI may possibly serve as candidates for treatment options.

We constructed a series of three queries, written in the Translator API standard language and submitted to xARA to select appropriate KPs to collect responses (Figure 4). **From each response, an exemplary result is selected and used in the query for the next step.**

The results of the first query produced several phenotypes, one of them was "Red blood cell count" (EFO0004305). When using this phenotype in the second step to query for genes, we identified one of the results as the telomerase reverse transcriptase (TERT) gene. This was then used in the third query (Figure 4) to identify targeting drugs, which included the drug Zidovudine.

xARA use this result to call for an explanation. The xcase retrieved uses a relationship extraction algorithm [6] fine-tuned using BioBert [7]. The explanation solution seeks previously preprocessed publications where both biomedical entities (or one of its synonyms) is found in the same article within a distance shorter than 10 sentences. The excerpt of entailing both terms is then used as input to the relationship extraction method. When implementing this solution for the gene TERT (NCBIGene:7015) and the chemical substance Zidovudine (CHEBI:10110), the solution was able to identify corroborating evidence of this drug-target interaction with the relationship types being one of: "DOWNREGULATOR," "INHIBITOR," or "INDIRECT DOWNREGULATOR" with respect to TERT. [85]

As a recap, since I'm not including the screenshots of the queries, the researchers searched first for a phenotypic feature of DILI, then selected "one of them" — red blood cell count — to search for genes that affect the phenotype, and eventually find a drug that effects that gene: all seemingly manually (an additional $1.4 million has been allocated to unify them [90]). Zidovudine, as a nucleoside reverse transcriptase inhibitor, does inhibit telomerase reverse transcriptase [91] , but can also cause anemia and lower red blood cell counts [92] – so through the extended reasoning chain the system has made a sign flip and recommended a drug that will likely make the identified phenotype (low red blood cell count) worse? The manual input will then be used to train the algorithm for future results, though how data from prior use and data from graph structure will be combined in the ranking algorithm — and then communicated to the end user — is still unclear.

Contrast this with the space-age and chromed-out description from CoVar:

> ROBOKOP technology scours vast, diverse databases to find answers that standard search technologies could never provide. It does much more than simple web-scraping. It considers

> inter-relationships between entities, such as colds cause coughs. Then it searches for new connections between bits of knowledge it finds in a wide range of data sources and generates answers in terms of these causal relationships, on-the-fly.
>
> Instead of providing a simple list of responses, ROBOKOP ranks answers based on various criteria, including the amount of supporting evidence for a claim, how many published papers reference a given fact, and the specificity of any particular relationship to the question.

For-profit platform holders are not incentivized to do responsible science, or even really make something that works, provided they can get access to some of the government funding that pours out for projects that are eventually canned - $75.5 million so far since 2016 for the Translator [93] . As exemplified by the trial and discontinuation of the NIH Data Commons after $84.7 million, centralized infrastructure projects often an opportunity to "dance until the music stops." Again, it is relatively difficult to see from the outside what work is going on and how it all fits together, but judging from RePORTER there seem to be a profusion of projects and components of the system with unclear functional overlap, and the model seems to have developed into allocating funding to develop each separate knowledge source.

The risk with this project is very real because of the context of its development. After 5 years, it still seems like the the Translator is relatively far from realizing the vision of biopolitical control through algorithmic predictions, but combined with Amazon's aggressive expansion into health technology [94] and even literally providing health care [95] , and the uploading of all scientific and medical data onto AWS with entirely unenforceable promises of data privacy — the notion of spending public money to develop a system for aggregating patient data with scientific and clinical data becomes dangerous. It doesn't require takeover by Amazon to become dangerous — once you introduce the need for data to train an algorithm, you need to feed it data, and so the translator gains the incentive to suck up as much personal and other data as it can.

Even assuming the Translator works perfectly and has zero unanticipated consequences, the development strategy still reflects the inequities that pervade science rather than challenge them. Biopharmaceutical research, followed by broader biomedical research, being immediately and extremely profitable, attracts an enormous quantity of resources and develops state of the art infrastructure, while no similar infrastructure is built for the rest of science, academia, and society.

I think it is important to pause and appreciate the potential for harm in the data infrastructural system describes so far, continuing to use structural transphobia as one example among many possible harms. First, a brief recap:

Through STRIDES, cloud providers like AWS, Google Cloud, and Microsoft Azure are intended to become the primary custodians of scientific data. Regardless of contracts and assurances, since their system is opaque and proprietary, there is no way to ensure that they will not crawl this data and use it to train their various algorithms-as-a-service — and they seem all too happy to do so, as evidenced by GitHub Co-Pilot reproducing copyrighted code and code with licenses that explicitly forbade its use in that context. Given that Amazon is expanding aggressively into health technology[94] , including wearables and literally providing health care [95] , primary scientific data is a

valuable prize in their mission to cement dominance in algorithmic health.

The effort to unify data across the landscape of databases, patient data, and so on is built atop a rickety pile of SaaS so fragile that a *single person* with a *single repository* can have ripple effects across the aggregators that impact the whole knowledge graph. In the above example, an outdated set of terminology classifies a subset of human gender as a disease, which then is linked to candidate genes and other nodes in the knowledge graph. Since there is a preponderance of misguided research about about the etiology and "biological mechanisms" of transgender people, the graph neighboorhood around transness is rich with biomarkers and functional data.

All of the above is known to be true now, but let's see how it could play out practically in an all-too-plausible thought experiment.

Though the translator system now is intended for basic research and drug discovery, there is stated desire for it to eventually become a consumer/clinical product [88] . Say a cloud provider rolls out a service for clinical recommendations for doctors informed by the full range of scientific, clinical, wearable, and other personal data they have available — a trivial extension of existing patient medical aggregation and recommendation services that express their biopolitical control as a slick wristband with app. It's very "smart" and is very "private" in the sense that only the algorithm ever sees your personal data.

Since these cloud providers as a rule depend on developing elaborate personal profiles for targeted advertising algorithmically inferred from available data[22], that naturally includes diagnosed or inferred disease — a practice they explicitly describe in the patents for the targeting technology[96] , gone to court to defend [**SmithFacebookInc2018 krashinskyGoogleBrokeCanada20** , formed secretive joint projects with healthcare systems to pursue [97] , and so on. Nothing too diabolical here, just a system wherein **your search results and online shopping habits influence your health care in unpredictable and frequently inaccurate [98] ways.**

Imagine, through some pattern in your personal data, **Amazon diagnoses you as trans.**Whether their assessment is true or not is unimportant. Since the Translator works as a graph-based knowledge engine, your algorithmic transness, with its links through related genes, "symptoms," and whatever other uninspectable network links the knowledge graph has, influences the medical care you receive. All part of the constellation of personalized information that constitutes "personalized medicine."

[22] A patent from Google is telling about how they view privacy concerns: whatever we can't get explicitly, we'll infer to sell better ads! > One possible method to improve ad targeting is for ad targeting systems to obtain and use user profiles. For example, user profiles may be determined using information voluntarily given by users (e.g., when they subscribe to a service). This user attribute information may then be matched against advertiser specified attributes of the ad (e.g., targeting criteria). Unfortunately, user profile information is not always available since many Websites (e.g., search engines) do not require subscription or user registration. Moreover, even when available, the user profile may be incomplete (e.g., because the information given at the time of subscription may be limited to what is needed for the service and hence not comprehensive, because of privacy considerations, etc.). Furthermore, advertisers may need to manually define user profile targeting information. In addition, even if user profile information is available, advertisers may not be able to use this information to target ads effectively. [96]

The Translator assures us that it will give doctors understandable provenance by being able to explain how it arrived at its recommendation. Let's assume from prior experience with neural net language models that part of the process doesn't work very well, or at least doesn't give a fully exhaustive description of every single relevant graph entity. Now let's further assume based on the above DILI example that the knowledge graph is not able to reliably "understand" the complex cultural-technological context of transness, and since it is classified as a "disease" decides that you need to be "cured." Since it has access to a diverse array of biomedical data, it might even be able to concoct a very effective conversion therapy regimen *personalized just for you.* The algorithm could prescribe your conversion therapy *without you or the doctor knowing it.*

Transphobic behavior that impacts treatment is common [87, 99] . Since the Translator's algorithm is designed to learn from feedback and use[83] , transphobic practices could easily reinforce and magnify the algorithm's initial guess about what transness being a disease should mean for trans people in practice. Combined with the limitations on provision of care from insurance systems [99] , on a wide scale transphobic medical practices could be transmuted into a "scientifically justified" standard of care.

Scaling out further, the original intention of the tool is to guide drug discovery and pharmaceutical research, so harm could be encoded into the indefinite future of biomedical research — imperceptibly guiding the array of candidate drugs to test based on an algorithmically biased perception of biology and medical prerogative. Even in the case that society changes and we attempt to make amends in our institution for outdated and harmful notions, the long tail of ingrained learning in a proprietary algorithm could be hard to unlearn if the proprieter is inclined to try at all. So even many years into the future when we "know better," the ghosts of algorithmically guided medical reserach and practice could still unknowingly guide our hands.

The pathologizing of transgender people is just one example among many demonstrated instances of algorithmic bias like race, disability, and effectively any other marginalized group. The critical issue is that **we might not have any idea** how the algorithm is influencing research and practice at scales large and small, immediate and indefinite. The impacts don't have to be as dramatic as this particular thought experiment to be harmful. The subtlety of having dosages, prescriptions, and candidate drugs jittered by a massive integrated machine learning system is harm in itself: our medical care becomes training data. The point is that we *can't know* the effects of letting the course of our medical research and clinical care be steered by an algorithm embedded within a platform that has *any* incentive that conflicts with our collective health.

How did we get here? How could an effort to link biomedical data become an instrument of mass surveillance and harm?

I have no doubt that everyone working on the Translator is doing so for good reasons, and they have done useful work. Forming a consortium and settling on a development model is hard work and this group should be applauded for that. Unifying APIs with Smart-API, drafting an ontology, and making a knowledge graph, are all directly useful to reducing barriers to desiloing data and shared in the vision articulated here.

The problems here come in a few mutually rein-

forcing flavors, I'll group them crudely into the constraints of existing infrastructure, centralized models of development, and a misspecification of what the purpose of the infrastructure should be.

Navigating a relationship with existing technology in new development is tricky, but there is a distinction between integrating with it and embodying its implications. Since the other projects spawned from the Data Science Initiative embraced the use of cloud storage, the constraint of using centralized servers with the need for a linking overlay was baked in the project from the beginning. From this decision immediately comes the impossibility of enforcing privacy guarantees and the rigidity of database formats and tooling. Since the project started from a place of presuming that the data would be hosted "out there" where much of its existence is prespecified, building the Translator "on top" of that system is a natural conclusion. Further, since the centralized systems proposed in the other projects don't aim to provide a means of standardization or integration of scientific data that doesn't already have a form, the reliance on APIs for access to structured data follows as well.

Organizing the process as building a set of tools as a relatively large, but nonetheless centralized and demarcated group pose additional challenges. I won't speculate on the incentives and personal dynamics that led there, but I also believe this development model comes from good intention. While there is clearly a lot of delegation and distributed work, the project in its different teams takes on specific tools that *they* build and *we* use. This is broadly true of scientific tools, especially databases, and contributes to how they *feel*: they feel disconnected with our work, don't necessarily help us do it more easily or more effectively, and contributing to them is a burdensome act of charity.

This is reflected in the form of the biolink ontology, where rather than a tool for scientists to *build* ontologies, it is intended to be *built towards*. There is tension between the articulated impossibility of a grand unified ontology and the eventual form of the algorithm that depends on one that, in their words, motivated the turn to machine learning to reconcile that impossibility. The compromise seems to be the use of a quasi-"neutral" meta-ontology that instantiates its different abstract objects depending on the contents of its APIs. A ranking algorithm to parse the potentially infinite results follows, and so too does the need for feedback and training and the potential for long-lived and uninterrogatable algorithmic bias.

These all contribute to the misdirection in the goal of the project. Linking *all* or *most*biomedical data in single mutually coherent system drifted into an API-driven knowledge-graph for pharmaceutical and clinical recommendations. Here we meet a bit of a reprise of the #neat mindset, which emphasizes global coherence as a basis for reasoning rather than providing a means of expressing the natural connections between things in their local usage. Put another way, the emphasis is on making something logically complete for some dream of algorithmically-perfect future rather than to be useful to do the things researchers at large want to do but find difficult. The press releases and papers of the Translator project echo a lot of the heady days of the semantic web[23] and its attempt to link everything — and seems ready to follow the same

[23] not to mention a sort of enlightenment-era diderot-like quest for the encyclopedia of everything

path of the fledgling technologies being gobbled up by technology giants to finish and privatize.

I think the problem with the initial and eventual goals of the translater can be illustrated by problematizing the central focus on linking "all data," or at least "all biomedical data." Who is a system of "all (biomedical) data" for? Outside of meta-scientists and pharmaceutical companies, I think most people are interested primarily in the data of their colleagues and surrounding disciplines. Every infrastructural model is an act of balancing constraints, and prioritizing "all data" seems to imply "for some people." Who is supposed to be able to upload data? change the ontology? inspect the machine learning model? Who is in charge of what? Who is a knowledge-graph query engine useful for?

Another prioritization might be building systems for *all people* that can *embed with existing practices* and *help them do their work* which typically involves accessing *some data.* The system needs to not only be designed to allow anyone to integrate their data into it, but also to be integrated into how researchers collect and use their data. It needs to give them firm, verifiable, and fine-grained control over who has access to their data and for what purpose. It needs to be *multiple,* governable and malleable in local communities of practice. Through the normal act of making my data available to my colleague and vice versa, build on a cumulative and negotiable understanding of the relationship between our work and its meaning.

Without too much more prefacing, let's return to the scheduled programming.

3.2.4 Federated Systems

When last we left it, our peer-to-peer system needed some way of linking data together. Instead of a big bucket of files as is traditional in torrents and domain-general databases, we need some way of exposing the metadata of disparate data formats so that we can query for and find the particular range of datasets appropriate to our question. !! For this section, I want to develop a notion of data linking that's a lot closer to natural language than an engineering specification.

Each format has a different metadata structure with different names, and even within a single format we want to support researchers who extend and modify the core format. Additionally, each format has a different implementation, eg. as an hdf5 file, binary files in structured subdirectories, SQL-like databases.

That's a lot of heterogeneity to manage, but fret not: there is hope. Researchers navigate this variability manually as a standard part of the job, and we can make that work cumulative by building tools that allow researchers to communally describe and negotiate over the structure of their data and the local relationships to other data structures. We can extend our peer-to-peer system to be a *federated database* system.

Federated systems consist of *distributed*, *heterogeneous*, and *autonomous* agents that implement some minimal agreed-upon standards for mutual communication and (co-)operation.

Federated databases[24] were proposed in the early 1980's [100] and have been developed and refined in the decades since as an alternative to either centralization or non-integration [**litwinInteroperabilityMultipleAutonomous1990 kashyapSemanticSchemaSimilarities1996 hullManagingSe**. Their application to the dispersion of scientific data in local filesystems is not new [**busseFederatedInformationSystems1999 djokic-petrovicPIBASFedSPARQLWebbased2017 hasnainBioFedFeder**, but their implementation is more challenging than imposing order with a centralized database or punting the question into the unknowable maw of machine learning.

!! we also have to give up the golden idol of perfect computability, reasoning, and integrity in favor of transparency, autonomy. not just as a "it's hard in federated dbs" but also because we *shouldn't try* to do them because those projects are fundamentally coercive. !! the goal is to make something that *works* and *does the things that scientists want to do* instead of something *magic*. Those constraints are mostly from the needs of corporate DBs who don't need people to access and modify the database. We have different needs – something to support our research instead creating a database for the sake of the database.

Amit Sheth and James Larson, in their reference description of federated database systems, describe the *design autonomy* as one critical dimension that characterizes them:

> Design autonomy refers to the ability of a component DBS to choose its own design with respect to any matter, including
>
> (a) The **semantics** (i.e., the Universe of Discourse),
>
> (b) The **representation** (data model, query language) and the **naming** of the data elements,
>
> (c) The conceptualization or **semantic interpretation** of the data (which greatly contributes to the problem of semantic heterogeneity),
>
> (d) **Constraints** (e.g., semantic integrity constraints and the serializability criteria) used to manage the data,
>
> (e) The **functionality** of the system (i.e., the operations supported by system),
>
> (f) The **association and sharing with other systems**, and
>
> (g) The **implementation** (e.g., record and file structures, concurrency control algorithms).

[24] though there are subtleties to the terminology, with related terms like "multidatabase," "data integration," and "data lake" composing subtle shades of a shared idea. I will use federated databases as a single term that encompasses these multiple ideas here, for the sake of constraining the scope of the paper.

Susanne Busse and colleagues add an additional dimension of **evolvability**: "Following"natural" tendencies, autonomous components will inevitably develop heterogeneous structures. It is the task of the federation layer to cope with the different types of heterogeneity." [101] .

!! critically we also need *communication, association, and execution autonomy* to control the use of data.

The typical conceptualization of federated databases that follow typically have five layers [102] :

- A **local schema** is the representation of the data on local servers, including the means by which they are implemented in binary on the disk
- A **component schema** serves to translate the local schema to a format that is compatible with the larger, federated schema
- An **export schema** defines permissions, and what parts of the local database are made available to the federation of other servers
- The **federated schema** is the collection of export schemas, allowing a query to be broken apart and addressed to different export schemas. There can be multiple federated schemas to accomodate different combinations of export schemas.
- An **export schema** can further be used to make the federated schema better available to external users, but in this case since there is no notion of "external" it is less relevant.

This conceptualization provides a good starting framework and isolation of the different components of a database system, but a peer-to-peer database system has different constraints and opportunities [103] .

!! drafting the sketch here so i can refine it later i am very tired!

Beneath this description is a tremendous amount of subtlety and contingency in implementation, but bear with me through a conceptual description.

Let us start with the ability for a peer to choose who they are associated with and what they share at multiple scales of organization: a peer can directly connect with another peer, but peers can also federate into groups, groups can federate into groups of groups, and so on. Within each of these grouping structures, the peer is given control over what data of theirs is shared. Clearly, we need some form of *identity* in the system, let's make it simple and flat and denote that in pseudocode as `@username` — in reality, without any form of distributed uniqueness checking, we would need to have some notion of where this username is "from," so let's say we actually have a system like `username@name-provider` but for this example assume a single name provider, say ORCID.

!! now would be the time blockchain ppl are like "but wait! that's centralization! how can you trust ORCID??" Those kinds of systems are designed for zero-trust environments, but we don't need absolute zero trust in this system since we are assuming we're operating with visible entities in a system already bound to some degree by reputation.

Let us also assume that there is no difference between `@usernames` used by individual researchers, institutions, consortia, etc.

The peer starts with their data in some discipline-specific format, which let us assume for the sake of concreteness has a representation as an OWL

schema.

That schema could be "owned" by the `@username` corresponding to the standard-writing group — eg `@nwb` for neurodata without borders. In a turtle-ish pseudocode, then, our dataset might look like this:

```
<#dataset-name>
    @nwb:general:experimenter @jonny
    @nwb:ElectricalSeries
        .electrodes [1, 2, 3]
        .rate 30000
        .data [...]
```

Where I indicate that me, `@jonny` collected a dataset (indicated with `<#dataset-name>` to differentiate an application/instantiation of a schema from its definition) that consisted of an `@nwb:ElectricalSeries` and the relevant attributes (where a leading `.` is a shorthand for the parent schema element.)

I have some custom field for my data, though, which I extend the format specification to represent. Say I have invented some new kind of solar-powered electrophysiological device and want to annotate its specs alongside my data.

```
<@jonny:SolarEphys < @nwb:NWBContainer>
    ManufactureDate
    InputWattageSeries < @nwb:ElectricalSeries
        newprop
        -removedprop
```

!! think of a better example lmao^^ and then annotate what's going on.

There are many strategies for making my ontology extension available to others in a federated network. We could use a distributed hash table, or **DHT**, like bittorrent, which distributes references to information across a network of peers (eg. [104]). We could use a strategy like the **Matrix** messaging protocol, where users belong to a single home server that federates with other servers. Each server is responsible for keeping a copy of the messages sent on the servers and rooms it's federated with. We could use **ActivityPub** (AP) [105] , a publisher-subscriber model where users affiliated with a server post messages to their 'outbox' and are sent to listening servers (or made available to HTTP GET requests). AP uses JSON-LD [106] , so is already capable of representing linked data, and the related ActivityStreams vocabulary [107] also has plenty of relevant action types for creating, discussing, and negotiating over links (also see cpub). We'll return to ActivityPub later, but for now the point is to let us assume we have a system for distributing schemas/extensions/links associated with an identity publicly or to a select group of peers.

!! Now ppl can query by my additional datatype alongside nwb datatypes

!! Now I want to make use of my colleagues data. In order to bridge them together and compare like terms, I could describe some link between them. Say I use some vocabulary someone has set up to describe different links, say we use `@skos`

```
<@jonny:ElectricalSeries < @nwb:ElectricalSeries>
    @skos:closeMatch .frequency @chemistry:ReagentMo
```

!! making this mapping lets me *use* their data, rather than being the drudgery of linking — hold on till the next section. This is something that happens all the time in the normal course of reserach, but by integrating it into the system of our data's representation and giving me agency over it, I contribute to the rest of it.

!! ah but what of different data implementations? how do i actually *read* the data if it's in hdf and i'm used to SQL? well there's no reason that the implementation cant' itself be a schema that instructs us how to read the data (in turn allowing us to overwrite it and make our own mapping between eg. a video file held externally.) !! same thing with linking up with existing databases – if they aren't p2p, just indicate that in the representational schema. !! to enable p2p with arbitrary data structures (we just have to add a step to hash the addressable pieces of data)

!! illustrate incentive to fix and link terminology to be discoverable as well as do your own work.

!! example of search between two schemas – find me all the ways that these schemas have been related and let me pick them! Browse all the ways this schema has been extended. !! federated querying like SPARQL are complicated and involve breaking up the request and executing it, but let's also assume that's possible. !! separate the querying part which refers to content-addresses hashes and the p2p part

!! again being purposefully nonprescriptive in implementation here – eg. you could have aggregators built on top that make queries speedier, and so on.

!! Now to search and download many other schemas – query your federation – again also could have trackerlike sites sitting on top to make indexing faster, or DHT system, implementation not important here. The point is we *want* lots of space to make different kinds of community overlays, interfaces, and tools. You do a query to find the unique IDs of the dataset that you want, then start the transfer.

!! extend this beyond just pairwise interactions. I don't know what the physicists are up to, but the *chemists might.* so I could try traversing the graph and find out what links have been made to get there. If they don't exist, i can make my own.

!! long-range schema resolution is what's needed, that's shared knowledge section

!! what we're *not* trying to do is make an automated, web-crawlable system to serve corporate clients, but make something that is social, communal, evolving (and maybe eventually it will reach a point of coherency where that is possible, but it really isn't the goal – we'd rather have some of the data and understand where it came from than access to all of the data with some of it potentially being junk and off-target but we just machine-learned through it anywaay).

!! we don't *need* to appeal to a single unifying graph because we have many overlapping ones at multiple scales. Ya it's noisier, but that's the point – it's about deducing graph structure from use rather than imposing it. If you don't like a particular structure, you're free to fork it (if they export their schemas) or make your own. If people want to use your data, they can write links from their format to yours, and vice versa. You *should* be able to determine which dialect your data is in in the same way you get to choose how you speak, and it doesn't necessarily need to be mutually compatible with others. If translations are possible, then great we should make them, but universality is not part of this system.

In the case of federated database systems, the federation layer provides a uniform way to mediate differences in schemas and formats between individual databases in the system. To share data between subdisciplines and fields we need to be able to perform some *mapping* between the differ-

ent data formats and standards that they use: we need some way of translating the neuroscientist's `GENOTYPE` to the geneticists `GENETIC_SEQUENCE`.

!! I will be purposefully vague about the means of implementing these mappings until we reach the shared knowledge section, but first we need a brief practical example of how a system like this might work.

Say I'm a neuroscientist who just collected a dataset that consists of a few electrophysiological recordings from a cluster of Consciousness Cells in some obscure midbrain nucleus, and then sectioned the brain and imaged their positions. I deposit my dataset on my local in-lab server, which I have set up to federate with the fancy new Neurophysiologist's Extravagant, Undying, Repository of Open data (NEUROd). All servers in this federation are required to have their data in the standardized NWB format, and since mine already is (go me!) my server announces to the others that we have some new data available! Some enterprising group of neuroscientific programmers has built a website that allows its users to search, browse, and do all the fancy visualization of data they would expect from a modern database, so I go and see how my new dataset has changed some standard aggregated analysis of all the Concscious Cells from all the other labs participating in the federation. Hang on, I say, a question mark appearing over my head like a cartoon caricature of a curious scientist – I wonder if these Consciousness Cells are in the same place in the evolutionary neighbors of my model organism!? I then run a query for all datasets that have positional data for Consciousness Cells. NEUROd has chosen to federate with the Evolutionary Volitional data sharing Operation (EVO), a federation of evolutionary biologists, some of whom study the origins of Consciousness Cells. They have their data in their own evolutionary biologist-specific format, but since there is some mapping between fields in the NWB standard and theirs, that's no problem. My search then returns data from not only all the other neuroscientists in NEUROd, then, but also matching data from EVO — and my cross-disciplinary question then becomes trivial to answer.

(figure of federated databases here).

The federated database system extends the peer to peer systems discussed earlier and provides a direct means of solving the problems of database fragmentation by subdiscipline. Since the requirements for being a 'node' in the federation are minimal, individual, local servers work seamlessly with institutional servers and large, national servers to take advantage of all available storage and bandwidth of the participating servers — a promising way to solve the problems posed by the "big data" of contemporary science (eg. one articulation by [108]). While mappings between schemas are not magical and require work, they provide a single point of mediation between the data formats of different disciplines. Federation gets us the best of both worlds: the flexibility and domain-specific tools of subdisciplinary databases with the availability of domain-general databases. The radical autonomy of federated systems dramatically lowers the barriers to standardization: rather than requiring everyone to do *the same thing in the same way* and fundamentally change how they do things, researchers need to just build the bridges to connect their existing systems to the federated standard. These bridges can be created gradually. Since nodes in a federated system are free to choose whether they connect to others, there do not need to be mappings between *all* types of data in a federation, and there is no need for creating the oft-fabled *"one true standard"* for all data. Researchers that are interested in interfacing their data with others are strongly incentivized to write the mappings that

permit it, and so they can emerge as they are demanded. Researchers are also given far more control over their own data than is afforded by traditional databases: it is entirely possible to have fine-grained permissions controls that allow researchers to share only the data they want to with the rest of the system while still taking advantage of, for example, locally federated servers that make their data available to other collaborating labs.

It's difficult to overstate how fundamentally a widely-adopted federated database system would be for all domains of science: when designing a behavioral experiment to study the circadian cycle, rather than relying on rules of thumbs or a handful of papers, one could directly query data about the sleep-wake cycles of animals recorded by field biologists in their natural habitats, cross reference that with geophysical measurements of daylight times and temperatures in those locations, and normalize the intensity of light you plan to give your animals by estimating tree-canopy coverage from LIDAR data from the geographers. One could make extraordinarily biophysically realistic models of neural networks by incorporating biophysical data about the properties of ion channels and cell membranes, tractography data from human DTI fMRI images, and then compare some dynamical measurement of your network against other dynamic systems models like power grids, telecommunications networks, swarming ants, and so on. Seemingly-intractable problems like the "file drawer" problem simply dissolve: null results are self-evident and don't *need* publication when researchers asking a question are able to see it themselves by analyzing all previous data gathered. Without exaggeration, they present the possibility of making *all* experiments multidisciplinary, making use of our collected human knowledge without disciplinary barriers. Indeed nearly all scientific literature is already available on a federated database system to anyone with an internet connection — arguably the largest expansion of scientific knowledge accessibility ever.

The fundamental tradeoff between centralized and decentralized database systems is that of flexibility vs. coherence: centralized systems can simply enforce a single standard for data and assume that everything it works with will have it. Federated systems require some means of maintaining the mappings between schemas that allow their fluid translation. They also require some means of representing and negotiating data that is unanticipated by existing schemas. The fine details of implementing a federated database system are outside the scope of this paper, but we will return to a means of distributed maintenance of mappings between schemas by taking advantage of semantic web technologies in shared knowledge. Before we do though, we need to discuss the shared tools to analyze and generate the data for the system in this section.

!! federated systems let us bridge the gap between localized server technology like datajoint and mass server technology like databases. If you let people federate at a local scale to share data between an institute, a consortium, etc. and then let those things scale to federate together you have a plausible means by which slowly a generalized database system could be accumulated over time.

!! lots of times this has been proposed before [**simaEnablingSemanticQueries2019 djokic-petrovicPIBAS**

!! DataLad [109] and its application in Neuroscience as DANDI are two projects that are conceptually and practically much closer to the kinds of systems that I am describing here (a peer-to-peer backend for DataLad is, I think, a promising development path). !! brief explanation of datalad !! problem is that it slices the problem in a differ-

ent place, and needs two extensions: federation for affiliating into larger networks, and federation for negotiating distributed queries across linked datasets

!! finally and perhaps most importantly we need to build the UI and communication tools to let us fluidly negotiate over the schema, including how to make multiple overlapping schemas associated eg. with a username. across several levels, eg. a format can have an export schema, and there can be canonical mappings between export schemas, which can the in turn be combined into higher-level common standards for combinations of formats, and so on. We want to be able to flexibly use these mappings without necessarily participating in the entire system, and we also want to make it possible for our local changes to be accessed by the larger system.

!! close this section by taking a larger view - [78] DANDI is in on the p2p system, as is kachery-p2p!! p2p systems already plenty in use, academic torrents, biotorrents, libgen on IPFS !! the proof of their utility is in the pudding, arguably when i've been talkiung about 'centralized servers' what i'm actually talking about content delivery networks, which are effectively p2p systems – they just own all the peers.

3.3 Shared Tools

!! talk about Plugin Oriented Programming (pypi POP page) as a design philosophy

!! has been spoken about in the context of reproducibility and openness [110] , embedding openness in workflow, which reaches a number of similar conclusions as I do here:

> Open Workflow: 1. Meet users where they are 2. Respect current incentives 3. Respect current workflow
>
> We could... demonstrate that it makes research more efficient, of higher quality, and more accessible. Better, we could... demonstrate that researchers will get published more often. Even better, we could... make it easy Best, we could... make it automatic - Jeffrey Spies, A workflow-centric approach to increasing reproducibility and data integrity (2017) [110]

If we're building infrastructure to allow us to build on each other's labor by sharing data, why not do the same for the tools that analyze and collect the data while we're at it? The benefits of distributed infrastructure that allow us to preserve our collected labor and knowledge compound when applied in multiple domains. The benefits of shared data, analytical, and experimental infrastructure are far more than the sum of their parts. Each is useful on its own, but as additional components of the system are developed they make the incentive to develop the rest even stronger <- this para is dogshit. rewrite with a clear head.

This section will be relatively short as I feel like a shared analytical framework is relatively uncontroversial, just a matter of putting labor in the right place. I also don't want to give the impression of self-promotion, as I have spent the last several years designing an experimental framework, autopilot. I will discuss it because, unsurprisingly, I designed it based on the same thoughts that have since developed into this paper, but I want to be clear that as with the rest of the paper, my focus is on the *kind* of tools we need rather than promoting

one specific tool.

3.3.1 Analytical Framework

The first natural companion of shared data infrastructure is a shared analytical framework. A major driver for the need for everyone to write their own analysis code largely from scratch is that it needs to account for the idiosyncratic structure of everyone's data. Most scientists are (blessedly) not trained programmers, so code for loading and negotiating loading data is often intertwined with the code used to analyze it, so it is often difficult to adapt another lab's analysis code for use in other contexts. If instead neuroscientists had all their data in a standardized format, then it would be possible to write an analysis method once and allow the rest of the community to benefit from it.

A shared analytical framework should be

- *modular* - Rather than implementing an entire analysis pipeline as a monolith, the system should be broken into minimal, composable modules. The threshold of what constitutes "minimal" is of course to some degree a matter of taste, but the overriding design principle should be to minimize the amount of duplicated labor. Rather than implementing a "peri-stimulus time-histogram" module, we should implement a "binning" module for counting spikes, connect it to an "alignment" module that splits the recording into chunks aligned at the stimulus onset, and so on. Higher-order analysis methods are relatively trivially composed from component parts, but extracting component parts from a frankenstein do-everything script is not. I expect this point to be relatively uncontroversial as it is a general principle of program design.
- *deployable* - For wide use, the framework needs to be easy to install and deploy locally and on computing clusters. The primary obstacle is dependency management, or making sure that the computer has everything needed to run the program. Anecdotally, more than the complexity of using the package itself, the primary barrier for nonprogrammer scientists using a particular software package is managing to get it installed. Luckily containerization and package management is a widespread and increasingly streamlined practice, so I expect this too to be uncontroversial.
- *pluggable* - The framework needs to provide a clear way of incorporating external analysis packages, handling their dependencies, and exposing their parameters to the user.
- *reproducible* - The framework should separate the *parameterization* of a pipeline, the specific options set by the user, and its *implementation*, the code that constitutes it. Implicit in a modularly constructed analysis framework is the notion of a "pipeline," or a specification of a tree (or, specifically, a DAG) of successive stages that process, merge, or split the data from the previous stage. The parameterization of a pipeline should be portable such that it, for example, can be published in the supplementary materials of a paper and reproduced exactly by anyone using the system.

Thankfully, DataJoint already does most of this, and is expanding its modularity with its recent Elements project.

!! need to revisit this in light of the paper: [111]

Though it currently uses a MySQL, relational database as its backend, extending it to incorporate with the peer to peer database system de-

scribed above would be an early, concrete development goal for this program. I have heard rumors they are considering adopting a decentralized traditional relational database like CockroachDB, which is not the same thing as a p2p federated semantic database system as I describe here, but is certainly a step in that direction. The rest is in the minutiae of normal software development, as well as building a user interface and collaboration platform for curation and management of shared pipelines. Thank you DataJoint team for making this section so simple.

The combined benefits of a unified data sharing and analytical system have a far greater reach than just saving redundant development time:

Papers published with a concise, inspectable description of their analytical pipeline sidestep the vagueries of methods section prose and allow widescale independent replication of published analyses. A system of documenting and discussing the countless hyperparameters and preprocessing tricks, often as much art as science, could operate as a means of implementing the countless papers describing best practices in analysis. If made easily expandable, so that the developers had a clear way to integrate their tools, access to the state of the art in analysis would be radically democratized, rather than limited to those with finely-tuned twitter feeds and patience to wade through seas of errors and stackexchange posts to get them to work.

A common admonishment in cryptographically-adjacent communities is to "never roll your own crypto," because your homebrew crypto library will never be more secure than reference implementations that have an entire profession of people trying to expose and patch their weaknesses. Bugs in analysis code that produce inaccurate results are inevitable and rampant [**millerScientistNightmareSoftware2006 soergelRampantS eklundClusterFailureWhy2016a bhandarineupaneCharacte**, but impossible to diagnose when every paper writes its own pipeline. A common analysis framework would be a single point of inspection for bugs, and facilitate re-analysis and re-evaluation of affected results after a patch.

Perhaps more idealistic is the possibility of a new kind of scientific consensus. Scientific consensus is subtle and elusive, but to a very crude approximation two of the most common means of its expression are review papers and meta-analyses. Review papers make a prose argument for a consensus interpretation of a body of literature. Meta analyses do the same with secondary analyses, most often on the statistics reported in papers rather than the raw data itself. Both are vulnerable to sampling problems, where the author of a review may selectively cite papers to make an argument, and meta-analyses might be unable to recover all the relevant work from incomplete search and data availability. Instead if one could index across all data relevant to a particular question, and aggregate the different pipelines used to analyze it, it would be possible to make statements of scientific consensus rooted in a full provenance chain back to the raw data.

More fundamentally, a shared data and analysis framework would change the nature of secondary analysis. Increasing rates of data publication and the creation of large public datasets like those of the Allen Observatory make it possible for meta-scientists and theoreticians to re-analyze existing data with new methods and tools. There is now such a need for secondary analysis that the NIH, among other organizations, is providing specific funding opportunities to encourage it. Secondary analyses are still (unfortunately) treated as second-class research, and are limited to analyzing one or a small number of datasets due to the labor

involved and the diversity of analytical strategies that makes a common point of comparison different. If, say some theoretician were to develop some new analytical technique that replaced some traditional step in a shared processing pipeline, in our beautiful world of infrastructure it would be possible to not only aggregate across existng analyses, as above, but apply their new method across an entire category of research.

In effect, analytical infrastructure can at least partially "decouple" the data in a paper from its analyis, and thus the interpretations offered by the primary researchers. For a given paper, if it was possible to see its results as analyzed by all the different processing pipelines that have been applied to it, then a set of observations remains a living object rather than a fixed, historical object frozen in carbonite at the time of publication. In addition to statements of consensus that can programmatically aggregate *existing* results as described by the primary researchers, it also becomes possible to make *fluid* statements of consensus, such that a body of data when analyzed with some new analysis pipeline can yield an entirely *new* set of outcomes unanticipated by the original authors. I think many scientists would agree that this is how an ideal scientific process would work, and this is one way of dramatically lowering the structural barriers that make it deviate from that ideal.

I'll give one more tantalizing possibility here: at the point when we have a peer-to-peer federated system of data-sharing servers integrated with some easily deployable analysis pipelining framework, then we also get a distributed computing grid akin to Folding@Home where users donate some of the computing power of their servers to analyze pieces of some large analysis job with very little additional development.

3.3.2 Experimental Framework

On the other side of data from its analysis are the tools used for its collection. A unifying experimental framework is seemingly a different kind and scale of complexity compared to a unifying data framework. *Everyone needs completely different things!* I have previously written about the design of a generalizable, distributed behavior framework in section 2, and about one modular implementation in section 3 of [13] , and so I will first abbreviate and extend the discussion found there and then consider the role of an experimental framework in broader scientific infrastructure. I designed Autopilot with many of the same fundamental motivations as I articulate here, so being dredged from the same well it should be far from surprising that I see it as a natural example. My intention is not as a self-serving advertisement for *everyone to use my software,* but to use it as an *example* of the *kind* of tool that I think would fit a particular role in a broader set of scientific infrastructure (!! redundant, pick a framing).

I first want to clarify what i'm talking about as an 'experimental framework' – not talking about projects **that we love** like open ephys/etc that develop specific hardware. Those are strictly complementary (and should be given more resources!) I'm talking about something to unify them, to combine the excellent pieces that implement differnt parts of experiments into a unified system.

The most basic requirement of a piece of shared experimental infrastructure is that it must be capable of expressing and being adapted to **perform any experiment.** The "any" there is a hard-ish "any," the reason for which should become clearer

soon. At an extremely abstract level, this means that the framework needs to be able to **control potentially high numbers of independent hardware components,** record measurements from them, and coordinate them together in some logical system that constitutes a "task" (or more broadly an "experiment"). In order to be widely adoptable, it needs to be able to **integrate with the instrumentation that researchers already use** rather than requiring researchers to reoutfit their entire rigs. That means, in turn, that it needs to provide a clear means for users to **extend its functionality** and contribute their extensions to the framework. At the same time as providing a clear entrypoint for researcher-developers to interact with the code, it needs to provide a **simple user inferface** so that regular use doesn't require extensive programming knowledge. In other words, if it ain't usable by everyone, it ain't infrastructure, and the same can be said for expense: it must be **inexpensive to implement.** Finally, it needs to be purpose-built for **reproducibility and replication** by preserving a full chain of **provenance** across the wandering path of parameter tuning and experimental design in a clear, **standardized data format** and providing a means of **replicating experiments** even in rigs that are only an approximate match to the original.

Autopilot attempts to achieve these lofty goals by embracing a distributed, modular architecture. Autopilot is built as a system of modules that each represent fundamental parts of experiments in general: hardware control, stimulus generation, data management, and so on. Everything is networked, so everything can talk to anything, even and especially across computers: in practice this means that it is capable of coordinating arbitrary numbers of experimental hardware components by just *using more computers.* It is built around the Raspberry Pi, a low-cost single-board computer with an enormous support community and library of off-the-shelf components, but can be used on any computer. Autopilot imposes few limitations on the structure of tasks and experiments, but also gives users a clear means of defining the parameters they require, the data that will be produced, how to plot it, and so on, such that any task has a portable, publishable representation that is not dependent on the local hardware used to implement it. Its modular hierarchy already provides structure that makes it easy for researchers to modify existing components to suit their needs, and some of its co-developers and I are currently implementing a generalized plugin system that will allow users to replace any component of the system in such a way that their work can be made available and referenceable by any other user of the system. Information about the state of the system, the plugins used, the history of tasks and parameters that an experimental subject experiences, are all obsessively documented, and the data it produces is clean at the time of acquisition. Portable task descriptions, referenceable plugins, and exact documentation of provenance make Autopilot capable of facilitating replication while still supporting extreme heterogeneity in its use. In sum, we designed Autopilot to be flexible, efficient, and reproducible enough for use as general experimental infrastructure.

When compared, the preceding reads as a rephrasing of the design principles articulated in (!! link to section). Autopilot is of course far from a finished project, and many of its design goals remain aspirational due to the small number of contribu-

tors[25]. I would be remiss in failing to mention Bonsai, which I love and have learned a lot from. I view Bonsai as a somewhat complementary project and would one day love to merge efforts. The primary differences between Bonsai and Autopilot, besides the massive and obvious difference in number of users and maturity of the library, are a) Autopilot is written in Python, a high-level programming language, and "glues" together fast, low-level library, where Bonsai is written in C#, which is also quite fast but is comparatively less accessible to a broad number of users. Relatedly, Autopilot's documentation describes how the library works down to the lowest levels while Bonsai's is more focused on the user level. b) Autopilot emphasizes communication between objects and their use in a distributed architecture, while Bonsai provides an excellent means of chaining objects together on a single system. c) Autopilot makes comparatively more nudges, and provides a few more features for making reproducible tasks and standardizing data. Again my intention is not a self-serving advocacy for my software, but to say that Bonsai is another extremely capable and widely-used system, and we need systems *like* them capable of serving the role in broader infrastructure that I will turn to now.

In addition to the benefits of reduced duplication of labor and greater access to the state of the art that runs through this whole argument, a standardized experimental framework multiplies the benefits of the data and analytical systems described previously.

When we talk about standardizing data, we talk in the parlance of "conversion," but conversion is only necessary because reserachers collect data in local, idiosyncratic formats. The reason researchers rely on idiosyncratic formats is that it is far from straightforward to directly collect data from their heterogeneous tools in a standardized format. The need for data conversion leaves an

[25] I am the first to admit Autopilot's shortcomings, which I document extensively in its development roadmap and github issues.

airgap between the ideal of universal data access and its labor-intensive practical reality: only those that are most ideologically committed and have enough resources to convert & share their data will do so. We could (and should) lessen the chore of data conversion with continued development of intuitive conversion tools, but an experimental framework that collected data that was *clean at the time of acquisition* then we could shortcircuit the need for conversion altogether. It would also completely dissolve the need for researchers to interact with the peer-to-peer sharing system described previously by automatically dumping standardized data directly into it. In short, an experimental framework could make all the steps between collecting and sharing data completely seamless, and by doing so make the dream of universal data availability possible.

Neuroscience has made substantial progress standardizing an ontology of common terms for cells, chemicals, etc. (see the Neuroscience Information Framework's Ontology) but an ontology for the many minute parameters that define a behavioral experiment's operation has proven elusive. Creating a standardized language for expressing and communicating behavioral experiments is the object of one of the Neurodata Without Borders working groups, in collaboration with the BEADL project, and they've done admirable work there. They have an in-progress terminology for certain parameters like `Reward`, `Guess`, etc., as part of a state-machine (!!define in margin) based representation of a task. The model of standardization would then be to define some extensible terminology, and then either build some software that implements the state machine descriptions of tasks or else ask existing software developers to incorporate them in their systems.

This path to standardization has many attractive qualities, like the formal verification possible with state machines, but may have trouble reaching universal adoption: at even modest complexities, experiments that are simple to explain in prose can be awkward and complicated to express as state machines (eg. section 3.1 in [13] , though the proposed statecharts model is a bit friendlier than traditional state machines). If it is difficult to express a particular feature of some experiment in some formalism, and easier to implement it as some external software, unintegrated with the behavioral framework, then much of the appeal of standardization is lost.

---

bigg redundancy from here...

Uniform standardization is desirable in the circumstances where it is possible, but the scale of variability in the parameters and designs of behavioral neuroscience experiments is truly on a different scale than the already-perplexing case of measurement data standardization. For example, a standard experiment in our lab implemented in Autopilot can be fully described by the parameters that define the experimental protocol itself, and those that parameterize the raspberry pi and the experimental hardware (here they are). The training protocol consists of 7 shaping stages that gradually introduce a mouse to a fairly typical auditory categorization task, each of which includes the parameters for at most 12 different stimuli per stage, probabilities for presenting lasers, bias correction, reinforcement, criteria for advancing to the next stage, etc. The rest of the parameterization includes details for configuring, calibrating and operating the rest of the system – and this is the minimal set of parameters for replicating this experiment that excludes all the defaults, implicit behavior, and well, the rest of the system. For this one relatively straightforward experiment, in one

lab, in one subdiscipline, there are 268 different parameters. It's not really about the *number* of parameters per se, but their unpredictability: one needs to parameterize every electrode on a neuropixel probe, but they are shared across a comparatively small number of things of their kind.

Asking people to change the entire way they think about, describe, down to the very mental model that they use to think about it is actually a huge ask. Even if some reasonable standardized lexicon was proposed, it will face the same difficulties as, well, normal lexicons: there is no neutral 'name' for anything, and any word is dependent on the way we conceptualize its use and meaning. This isn't woo-woo unknowability shit: one person's sensory response latency is another person's time of delayed gratification suppression. Even assuming that, getting everyone to start re-expressing all their experiments in a probably very different way than they have been thinking about them for 20-30 years with all the entrenched hardware decisions made over that time is just rounding the bend ready to beat u up for ur lunch money.

Another, complementary way of approaching this problem is to focus on giving people a way to express themselves in a 'safe' environment, focus on the way they *use* them rather than try to define all of them a-priori. sorta like lameguage lmao.

a behavioral framework designed for reproducibility, that preserves a complete history of task parameters as well as the code that uses them, solves both the problems of external inspection and replication without needing to prescribe a specific formalization or uniform ontology. It doesn't matter *what* terms you use if it's trivial to see *how* they're used. Importantly, this strategy punts on the goal of interoperability, but does not forsake it: we will revisit standardized ontologies in the next section. Asking large numbers of people to change the way that they think about their experiments and the words they use to describe them is, ultimately, a pretty big ask. Providing people a tool that allows themselves to express themselves in whatever form is natural to them and make their terminology meaningful by preserving its context might be easier. (put people in the same system and give them a space to express the terms they use and let them standardize among themselves rather than imposing.)

... to here

Replication is seriously hard. designing a software system that's smart enough about the division between the logical structure of the task and the implementation is seriously hard. the raspi is general purpose enough that was can incorporate pretty general purpose hardware control systems with nontrad components as well, so it balances being an approachable "start from somewhere" (actually in a really good place) with general still byo-hardware. replication needs to basically be incorporated from the ground up, as most behavioral packages that exist tend to rely on local script files that are still labor-intensive to create and are rarely shared, because they're not really intended to be made sharable. « point im' trying to make here is that it can't be an afterthought, the ways that it's easy to go wrong.

But for systems that do link code to a portable task description, where the documentation for each parameter is also good (like wat if that documentation was linked to the semantic wiki... return to in next section), then it is entirely possible to download a system that you point to whatever parts you have around and let er rip. (this doesn't address the technical complexity, but that's also a tease for the next section).

It is already occasionally possible to follow the trail of provenance back to some experimental code, but when all code is developed independently, is any of it reliable [112] ? Like bugs in analytical software, bugs in experimental control software are likely rife, but unless they are present in the few pieces of commonly used open-source software they are almost entirely undiagnosable. Conversely, maybe more positively, a shared experimental framework gives a place to gather reference implementations of the many common algorithms, routines, and hardware controllers used in neuroscientific experiments. (!! the tiny details matter, but they almost never make it into methods sections. eg. we use bias correction methods, but the way we do it might be different than the way you do it. people do lots of great work optimizing over different training regimens, but that usually gets left as text. Algorithms for hardware control and sensor fusion are split across a zillion adafruit libraries, and they aren't modularized or split up or even documented that well)

As an example, intertial motion sensors (IMUs) are an increasingly common tool for neuroscientists interested in studying unrestrained, freely moving behavior. In our case, we were working with an IMU with three, 3-dimensional sensors: an accelerometer, gyroscope and magnetometer. The raw signals from these IMUs (linear acceleration, angular velocity) are rarely useful on its own, and researchers are usually after some derived value like orientation, position, etc. Since the readings are also noisy, these transformed signals typically rely on some sensor fusion algorithms to condition and combine them. We were interested in measuring "absolute" geocentric vertical velocity to control a motorized platform in a closed-loop experiment (as I described in my NMC3 talk). Adafruit provides a basic Python library to control the IMU, but it was relatively undocumented, slow, and didn't expose all the functionality of the chip, so we adapted it to Autopilot. We were able to find a number of whitepapers that described a sensor fusion algorithms, but no implementations. The algorithm we eventually landed on uses a Kalman filter to combine accelerometer and gyroscope readings to estimate orientation [113] . In this case were lucky to find Roger Labbe's excellent filterpy library [114, 115] , and with a few performance and syntax tweaks were also able to adapt it to autopilot, extended it to implement the orientation transformation, and built it into the IMU Object. (!! - go back through and give names to each of the objects and methods for reference below)

OK cool so you programmed an accelerometer, what's the big deal? First, from the developer's perspective: we needed to implement some `hardware` object and teach it about some geometric `transformation`. The autopilot `hardware` and `transform` modules give a clear place to implement both. The minimal expected structure of these modules make it straightforward to adapt existing code to the library, but we can also copy and modify (or, "inherit from") some existing hardware object to avoid having to write basic operations from scratch (eg. the I/O operations) and extend their functionality (eg. they are now networked, can take advantage of autopilot's unified logging system, etc.). To the degree that the framework is widely adopted, it gives credit to and provides a direct means of making the algorithms and tools they develop available to users. (!! they don't even need to integrate with the system wholesale, just expose some API and write a quick wrapper for this, we're wrking on it!!) This is, in some sense, the essence of what i mean by a behavior "framework" — a minimal "spanning set" of rules for how the system works that gives clear points of extension.

From the user's perspective: We could have implemented the sensor fusion algorithm and geomet-

ric transform directly "in" the `IMU` hardware object, but instead we separated them out as several independent `transform` objects. Rather than extending the functionality of a single hardware object, we instead gained several basic algorithms. Their generality is *noncoercive* — The problem of getting absolute orientation from an IMU is solved for *everyone*, even those that don't want to adopt any other part of the system. The rotation algorithm is generic and modular: it can be used as just a trigonmetric transformation of accelerometer readings without a Kalman filter, incorporate gyroscopic readings, or use an entirely different timeseries filter altogether. By integrating them in an existing library of `transform` objects, they are made combinatorically more useful — so far all I have discussed has been a means of estimating orientation, we still need to *use* that orientation estimate to extract "absolute" vertical acceleration.

In autopilot, we can express the rest of what we need as a series of `transform` objects that can be added together with + and +=:

```
from autopilot import transform as t

# we start with some measurement of
# - subjective acceleration (x, y, z)
# - rotation (roll, pitch)
# then we...

# rotate the acceleration around the x and y axes
z_velocity = t.geometry.Rotate('xy')
# select the "z," or vertical component,
# aka accelerometer[2]
z_velocity += t.selection.Slice(slice(2,3))
# subtract the constant acceleration due to gravity
z_velocity += t.math.Add(-9.8)
# and then integrate the acceleration measurements
# over time to get velocity
z_velocity += t.timeseries.Integrate(dt_scale = 1.0)
```

So then when used with the IMU object...

```
from autopilot.hardware.i2c import IMU_9DOF

# create the sensor object to read from it
sensor = IMU_9DOF()

# get accelerometer readings by accessing its prope
>>> sensor.acceleration
array([0,0,9.8])

# apply our transformation by giving the accelerome
# and orientation readings to our transform object
>>> z_velocity.process((sensor.acceleration, sensor.
1.6095 # or whatever m/s
```

This transformation could itself be built into the sensor object as an additional `IMU_9DOF.velocity` property, as was done for `.rotation`, reconfigured to add additional processing stages, and so on.

(!! add example of adding DLC to position estimate? yes. to show how things don't need to be built into autopilot, just given some API, as was done with deeplabcut, at the end of the developer section)

From the often-overlooked perspective of some downstream "reader": when everything is integrated into an extensible experimental framework, complete retrospective provenance becomes possible. Autopilot exhaustively logs all local parameters like hardware configuration, as well as references to all versions of all code used to generate a dataset *in the dataset itself* automatically. A reader can then trace all the data presented in a paper back through a standardized analysis pipeline to the raw data, and then continue to inspect every part of how it was generated. Since the code is all available by default, it

becomes possible to audit experimental code on a broad scale: if a reader were to find a bug, they could raise an issue, patch it — and flag all datasets that were effected.

At this point we have largely closed the loop of science: starting with standardized data, shared in a scalable p2p system, with some federated interface structure, through modularized analysis parts, published alongside the means to directly reproduce the experiment and re-generate the data... and when we start considering these technologies as an ensemble some things that truly sound like science fiction compared to scientific reality start to become possible. In addition to allowing all of the above features of standardized output data being cross-indexable, what about making the literal fine-grained parameters a way of indexing The All Knowledge Base (go back to previous section and make clearer that only the output data is indexable)? Doing a simultaneous optimization over all of our parameters is basically impossible, and we have all these heuristics for hopping and skipping over it, but what if the behavioral system could query all other times the experiment has been performed, cross reference with published outcome data from the parameterization, and recommend the optimal parameterization for whatever you are studying? The compounding nature of making systems that preserve and respect the diversity of labor to make it coproductive is tectonic: at every stage, from implementation to tweaking, to understanding a science with appropriate infrastructure would move at light speed compared to the way we do it now.

... the major part that's missing is some means of negotiating our schemas and data ... transition to next section

3.4 Shared Knowledge

!! jimmy wales on wikipedia:

!! why is it that literally every project is organized on google docs and slack? we can do better for collective organization

!! https://www.dbpedia.org/

!! why is public trust in scientists so low? could it be that there is an alternative to scientists seeing themselves as cloistered experts? re: cold war peer review paper

The (part of the system that's most needed and potentially transformative) is a system of scientific communication.

Except for certain domain-specific exceptions, the scientific communication system consists of the two ancient monoliths groaning with the dust of their obsolescence: the dead and static papers of the traditional journal system, and the ephemeral halo of insider knowledge shared at conferences. The remainder of the gigantic overflowing franzia bag of scientific discourse is funelled ingloriously

onto Twitter[26] — and it *sucks*.

Since the advent of the contemporary journal system, communication technology has been

[26] no citation needed, right? if there is some other bastion of scientific discourse i would love to know about it.

stripped to its very atoms and rebuilt — and it has managed to dig in and *persist* while all the letterman jackets and beatniks of its era have become vape teens on tiktok. A reconsideration of the entire scientific publishing system is strictly out of scope for this paper, but the communication system I will describe exists in the gaps of need it leaves unfilled. Criticisms of the scientific communication system typically start by imaging much of the contemporary journal system as etched as fact on the face of reality, and tweaking at a few of its more ticklish knobs (eg. [116]). Instead let's try it the other way: to trace the outlines of how a scientific communication system *should* work, given the basis of holistic infrastructure described so far. I will argue that a communication system, and more specifically the community it supports, is the blood that must pump through any of these digital systems that aspire to call themselves infrastructure. To arrive at a proposed form for a system, I'll start by laying the basic axes of communication technology, and then load the scales with the empirical girth of the largest knowledge systems that have ever existed: Wikipedia and internet piracy.

There simply isn't a place to have longform, thoughtful, durable discussions about science. The direct connection between the lack of a communcaition venue to the lack of a way of storing technical, contextual knowledge is often overlooked. Because we don't have a place to talk about what we do, we don't have a place to write down how to do it. Science needs a communcation platform, but the needs and constraints of a scientific communication platform are different than those satisfied by the major paradigms of chatrooms, forums etc. By considering this platform as another infrastructure project alongside and integrated with those described in the previous sections, its form becomes much clearer, and it could serve as the centerpiece of scientific infrastructure.

I will argue that we should build a semantically-enabled communication and knowledge-base system on top of activitypub to unify the preceding digital infrastructure elements. !!importantly, should also have means of ingest for existing tools and elements – easy to import existing papers and citation trees, plugins for existing data sharing systems.

!! description of its role as a schema resolution system – currently we implement all these protocols and standards in these siloed, centralized groups that are inherently slow to respond to changes and needs in the field. instead we want to give people the tools so that their the knowledge can be directly preserved and acted on.

!! descrption of its role as a tool of scientific discussion – integrated with the data server and standardized analysis pipelines, it could be possible to have a discussion board where we were able to pose novel scientific questions, answerable with transparent, interrogatable analysis systems. Semantic linking makes the major questions in the field possible to answer, as discussions are linked to one another in a structured way and it is possible to literally trace the flow of thought.

!! let's tour through wikipedia for a second and see how it's organized. Look at these community incentive structures and the huge macro-to-micro level organization of the wiki projects. The infinitely mutable nature of a wiki is what makes it powerful, but the SaaS wikis we're familiar with don't capture the same kind of 'build the ground you walk on' energy of the real wiki movement.

!! what's critically different here between other projects is that we are explicitly considering the incentives to join each of these efforts, and by integrating them explicitly, each of them is more ap-

pealing. so while there are lots of databases, lots of analysis systems, lots of wikis, and so on, there aren't many that are linked with one another such that participating in one part of the system makes the rest of the system more powerful as well as makes it more useful to the user.

3.4.1 Axes of Communication Systems

3.4.2 Semantic Wikis - Technical Knowledge Preservation

[117]

!! Read and cite! [118]

!! the word for communally curated schemas is https://en.wikipedia.org/wiki/Folksonomy

!! [119]

!! wikibase can do federated SPARQL queries https://wikiba.se/ - and has been used to make folksonomies https://biss.pensoft.net/article/37212/

> I can see my bank statements on the web, and my photographs, and I can see my appointments in a calendar. But can I see my photos in a calendar to see what I was doing when I took them? Can I see bank statement lines in a calendar? https://www.w3.org/2001/sw/

!! lots of scientific wikis - https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Molecular_Biology/Genetics/Gene_Wi - https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Molecular_Biology/Genetics/Gene_Wiki

3.4.3 Semantic Wikis - Schema Resolution & Communication platform

!! bids is doing something like this https://nidm-terms.github.io/

!! interlex

> The Semantic Web is about two things. It is about common formats for integration and combination of data drawn from diverse sources, where on the original Web mainly concentrated on the interchange of documents. It is also about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing. https://www.w3.org/2001/sw/

!! Semantic combination of databases in science are also not new [120, 121] . We need both though! semantic federated databases!

Part of what is missing and a place where we could learn from librarians is the notion of governance over a knowledge schema. People have a lot of trouble with NWB because they doubt if it could account for all the idiosyncracies in the types of data that we have to represent. But instead if we have a way of capturing all that thought and insight and practical experience in a governance and decisionmaking structure then we could flexibily work our way to a set of schemas that work for everyone. Part of what needs to be done is to move from SQL queries to a more expressive abstract system of schema creation that more people can participate in – that's what infrastructure building is, making things that seem impossible or difficult routine. Practically, this can mean an explicit versioning system that not only specifies different versions of a data representation, but for every transition between state there is some notion of making that transition in the data structure. (give example of the subject upgrade system). If that was possible, then the notion of data structure would entirely evaporate, best of both worlds. we get ev-

erything and the game is over forever. This is also the distinction between centralized and decentralized systems. we can just make the changes and since they're done against a background of unified intent and expression they can exist simulataneously, commune with one another, while being forwardly productive as their contradictions are resolved.

3.4.4 Linked communication platform

We all hate science twitter, why does it exist?

> Though frequently viewed as a product to finish, it is dynamic ontologies with associated process-building activities designed, developed, and deployed locally that will allow ontologies to grow and to change. And finally, the technical activity of ontology building is always coupled with the background work of identifying and informing a broader community of future ontology users. [1]

> Two essential features coordinate this information to better serve our organizational decision-making, learning, and memory. The first is our constellation of Working Groups that maintain and distribute local, specialized knowledge to other groups across the network. [...] A second, more emergent property is the subgroup of IBL researchers who have become experts, liaisons, and interpreters of knowledge across the network. These members each manage a domain of explicit records (e.g., written protocols) and tacit information (e.g., colloquialisms, decision histories) that are quickly and informally disseminated to address real-time needs and problems. A remarkable nimbleness is afforded by this system of rapid responders deployed across our web of Working Groups. However, this kind of internalized knowledge can be vulnerable to drop-out when people leave the collaboration, and can be complex to archive. An ongoing challenge for our collaboration is how to archive both our explicit and tacit processes held in both people and places. This is not only to document our own history but as part of a roadmap for future science teams, whose dynamics are still not fully understood. [7]

importantly, semantic wiki can be accessed progrtammatically, so you don't need to use the service and can build your own interface to it.

good science community infra - https://www.zooniverse.org

Relational database systems, manage RDF data, but in a specialized way. In a table, there are many records with the same set of properties. An individual cell (which corresponds to an RDF property) is not often thought of on its own. SQL queries can join tables and extract data from tables, and the result is generally a table. So, the practical use for which RDB software is used typically optimized for soing operations with a small number of tables some of which may have a large number of elements.

RDB systems have datatypes at the atomic (unstructured) level, as RDF and XML will/do. Combination rules tend in RDBs to be loosely enforced, in that a query can join tables by any comlumns which match by datatype – without any check on the semantics. You could for example create a list of houses that have the same number as rooms as an employee's shoe size, for every employee, even though the sense of that would be questionable.

The Semantic Web is not designed just as a new data model - it is specifically appropriate to the linking of data of many different models. One of the great things it will allow is to add information relating different databases on the Web, to allow sophisti-

> cated operations to be performed across them. https://www.w3.org/DesignIssues/RDFnot.html

in addition to a wiki, we need some conversational engine – talk pages are ok, but they're too fragmented and all hard to keep up to date with. Realtime, chatlike interfaces don't preserve information well, so we should use some intermediate medium like a forum or stack exchange that allows conversations to be tagged and searched and sorted and organized.

Social incentive structure is huge here.

Compared to RDBMS https://www.w3.org/DesignIssues/RDB-RDF.html – rather than individual schemas, groupings of properties, we have 'relationships.' this example is good:

> For example, one person may define a vehicle as having a number of wheels and a weight and a length, but not foresee a color. This will not stop another person making the assertion that a given car is red, using the color vocabular from elsewhere.

We're talking about a collaboration medium here... we need a way of organizing open questions in the field and discussing them in a straightfoward way. Why is it that every scientist needs to figure out their own completely gray-area way of discovering papers?

Bad APIs have killed projects with shitloads of funding like NWB and IPFS https://macwright.com/2019/06/08/again.html - usability needs to be *the first priority* - you can develop all the fancy shit that you want, if no one can install and unse it in 10 minutes then it's totally useless. This is why the community also has to be collaborative, not just the technology, hends the shared governance idea... ppl note that IPFS has no economic model – that's like true, because there has to be some other incentive system for using it – it makes your work more powerful, it plugs you into a community, etc. https://blog.bluzelle.com/ipfs-is-not-what-you-think-it-is-e0aa8dc69b

3.4.5 Credit Assignment

depth of linking is combinatoric – if you have a paper ecosystem where the numbers are linked to the data, and then the data is annotated, then it's possible to index information across papers not just by textual similarity metrics but on similarity of the structure of experiment and data.

the work of maintaining the system can't be invisible, read & cite [**classeDistributedInfrastructureSupport201**

4. Conclusion

Shared Governance

!! just make this a final note in the conclusion

In addition to like a wiki... need some way of having conversations and arguments about what means what. like some proposal system for linking certain tags together or pointing one to the other...so shared knowledge and shared gover-

nance can be a fluid entity.

to avoid the coersion described in [9] we must make any metadata schema collaborative and mutually beneficial – there is no such thing as 'required' data as long as we design a system that preserves as much information as possible on collection, designing infrastructure is an act of community trust.

Dont want to be prescriptive here, but that we can learn from previous efforts like - https://en.wikipedia.org/wiki/Evergreen_(software) , - IBL, - etc. ab

4.1 Contrasting visions for science

4.1.1 The worst platform capitalist world

4.1.2 What we could hope for

As a break from doomsaying, imagine the positive vision of doing neuroscience with all the power of basic infrastructure.

You have some new research question, and so you turn to the standard Python (or whatever) library that allows you to query data from yours and all other labs who share their data with this system. You're immediately able to filter through to find all the recordings from a particular subtype of cell in a particular region being exposed to some particular set of stimuli across some particular manipulation. Since you have access to decades of labor by thousands of scientists, even with that complex filter you still find, say for the sake of having a round cool-sounding number, a million recordings. Because they're all in some standardized format, over the years a common analysis pipeline has been developed, so you're also immediately able to perform the analyses to confirm the hunch for your new question — and it's time to implement it.

You don't need to implement the whole thing from scratch because you can check out a similar experiment from the standardized experimental software framework, read the communally maintained documentation, make the minor tweaks you need for your experiment, and you're off and running. You need to build some brand new component, but you also have a practical knowledge repository where other scientists working on similar problems have described the basic components, circuits, and have even uploaded some 3d-printable components for you to use. Because the repository was designed for ease of use and has a robust system of community incentives for contribution, as you build you document what you learn, and when you're finished upload the schematics and write instructions for your new component. The experimental software framework was designed to incorporate custom components, so you extend some similar hardware control code and integrate it with your experiment without needing to resort to some patchwork system of TTL synchronization pulses and serial port arcana.

You did it! Experiment Over! The experimental framework produces data that is clean, annotated, and standardized at the time of acquisition, and automatically integrates it with the analysis pipeline you built when your experiment was just a budding baby hypothesis, so your analysis is finished shortly after the experiment is. You have the "auto-upload" setting on, so without any additional effort your work has been firehosing information back the global knowledge pool. You do a pull request for the improvements you've made to the experimental software, write the paper, and the loop is complete: a closed knowledge system where nothing is wasted and everyone is more capable and

empowered by drawing from and contributing to it.

OK Here's the moment at the end of 2001.

end with the more radical vision — science post papers. Information is semantically organized, so it is possible to ask and answer questions through the medium in which information is represented. Discussion forums exist to describe particular kinds of questions, and a robust discussion of primary scientific data is made possible. Scientists lost their role as arbiters of all reality, but instead are just the comrades closest to the questions, capable of answering open questions in the community, able to design the experiments proposed.

The notion of the filedrawer problem dissappearing, we don't need to publish null results when the data is all always available.

The fractal nature of provenance — where if one can trace an intellectual lineage through its data, one solves credit assignment as centrality within a network.

High school biology classrooms are able to directly interface with the fundament of science, open questions are directly open to students,

!! ethical magnitude can't be lost, the information monopolists are 7 or the 10 largest companies in the world, and they got that way by buying industries like ours. https://2020.internethealthreport.org/slideshow-internet-health/#3

5. References

References

[1] Geoffrey C. Bowker et al. “Toward Information Infrastructure Studies: Ways of Knowing in a Networked Environment”. In: *International Handbook of Internet Research*. Ed. by Jeremy Hunsinger, Lisbeth Klastrup, and Matthew Allen. Dordrecht: Springer Netherlands, 2010, pp. 97–117. ISBN: 978-1-4020-9789-8. DOI: 10.1007/978-1-4020-9789-8_5. URL: https://doi.org/10.1007/978-1-4020-9789-8_5 (visited on 03/07/2021) (cit. on pp. 2, 7, 73).

[2] David Tilson, Kalle Lyytinen, and Carsten Sørensen. “Digital Infrastructures: The Missing IS Research Agenda”. In: *Information Systems Research* 21.4 (Dec. 2010), pp. 748–759. ISSN: 1047-7047, 1526-5536. DOI: 10.1287/isre.1100.0318. URL: http://pubsonline.informs.org/doi/abs/10.1287/isre.1100.0318 (visited on 05/02/2021) (cit. on p. 2).

[3] Michican Civil Rights Commission. *The Flint Water Crisis: Systemic Racism Through the Lens of Flint*. Tech. rep. Michican Civil Rights Commission, Feb. 2017. URL: https://web.archive.org/web/20210518020755/https://www.michigan.gov/documents/mdcr/VFlintCrisisRep-F-Edited3-13-17_554317_7.pdf (visited on 05/18/2021) (cit. on p. 3).

[4] Philip Mirowski. “The Future(s) of Open Science”. In: *Social Studies of Science* 48.2 (Apr. 2018), pp. 171–203. ISSN: 0306-3127. DOI: 10.1177/0306312718772086. URL: https://doi.org/10.1177/0306312718772086 (visited on 03/27/2021) (cit. on pp. 3, 17).

[5] Charleeze Ponzi. *Is Science a Pyramid Scheme? The Correlation between an Author’s Position in the Academic Hierarchy and Her Scientific Output per Year*. Jan. 2020. DOI: 10.31235/osf.io/c3xg5. URL: https://osf.io/preprints/socarxiv/c3xg5/ (visited on 09/16/2021) (cit. on p. 4).

[6] Matthew J. Bietz, Toni Ferro, and Charlotte P. Lee. “Sustaining the Development of Cyberinfrastructure: An Organization Adapting to Change”. In: *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*. CSCW ’12. New York, NY, USA: Association for Computing Machinery, Feb. 2012, pp. 901–910. ISBN: 978-1-4503-1086-4. DOI: 10.1145/2145204.2145339. URL: https://doi.org/10.1145/2145204.2145339 (visited on 09/21/2021) (cit. on p. 4).

[7] Lauren E. Wool and The International Brain Laboratory. “Knowledge across Networks: How to Build a Global Neuroscience Collaboration”. In: (July 2020). DOI: 10.1016/j.conb.2020.10.020. URL: https://psyarxiv.com/f4uaj/ (visited on 03/07/2021) (cit. on pp. 7, 21, 23, 73).

[8] STEPHEN R. BARLEY and BETH A. BECHKY. “In the Backrooms of Science: The Work of Technicians in Science Labs”. In: *Work and Occupations* 21.1 (Feb. 1994), pp. 85–126. ISSN: 0730-8884. DOI: 10.1177/0730888494021001004. URL: https://doi.org/10.1177/0730888494021001004 (visited on 03/15/2021) (cit. on p. 7).

[9] Matthew J. Bietz and Charlotte P. Lee. “Collaboration in Metagenomics: Sequence Databases and the Organization of Scientific Work”. In: *ECSCW 2009*. Ed. by Ina Wagner et al. London: Springer,

2009, pp. 243–262. ISBN: 978-1-84882-854-4. DOI: 10.1007/978-1-84882-854-4_15 (cit. on pp. 8, 41, 76).

[10] Zachary F. Mainen, Michael Häusser, and Alexandre Pouget. "A Better Way to Crack the Brain". In: *Nature News* 539.7628 (Nov. 2016), p. 159. DOI: 10.1038/539159a. URL: http://www.nature.com/news/a-better-way-to-crack-the-brain-1.20935 (visited on 03/09/2021) (cit. on pp. 8–10, 21, 26).

[11] Thomas Baker. "Maintaining Dublin Core as a Semantic Web Vocabulary". In: *From Integrated Publication and Information Systems to Information and Knowledge Environments: Essays Dedicated to Erich J. Neuhold on the Occasion of His 65th Birthday*. Ed. by Matthias Hemmje, Claudia Niederée, and Thomas Risse. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 61–68. ISBN: 978-3-540-31842-2. DOI: 10.1007/978-3-540-31842-2_7. URL: https://doi.org/10.1007/978-3-540-31842-2_7 (visited on 03/12/2021) (cit. on p. 9).

[12] TIM BERNERS-LEE, JAMES HENDLER, and ORA LASSILA. "THE SEMANTIC WEB". In: *Scientific American* 284.5 (2001), pp. 34–43. ISSN: 0036-8733. URL: https://www.jstor.org/stable/26059207 (visited on 03/12/2021) (cit. on p. 9).

[13] Jonny L. Saunders and Michael Wehr. "Autopilot: Automating Behavioral Experiments with Lots of Raspberry Pis". In: *bioRxiv* (Oct. 2019), p. 807693. DOI: 10.1101/807693. URL: https://www.biorxiv.org/content/10.1101/807693v1 (visited on 03/12/2021) (cit. on pp. 9, 62, 65).

[14] Larry F. Abbott et al. "An International Laboratory for Systems and Computational Neuroscience". In: *Neuron* 96.6 (Dec. 2017), pp. 1213–1218. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2017.12.013. URL: https://www.sciencedirect.com/science/article/pii/S0896627317311364 (visited on 03/15/2021) (cit. on pp. 10, 21).

[15] James Howison and James D. Herbsleb. "Incentives and Integration in Scientific Software Production". In: *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*. CSCW '13. New York, NY, USA: Association for Computing Machinery, Feb. 2013, pp. 459–470. ISBN: 978-1-4503-1331-5. DOI: 10.1145/2441776.2441828. URL: https://doi.org/10.1145/2441776.2441828 (visited on 09/21/2021) (cit. on p. 11).

[16] Alexander Mathis et al. "DeepLabCut: Markerless Pose Estimation of User-Defined Body Parts with Deep Learning". In: *Nature Neuroscience* 21.9 (Sept. 2018), pp. 1281–1289. ISSN: 1546-1726. DOI: 10.1038/s41593-018-0209-y. URL: https://www.nature.com/articles/s41593-018-0209-y (visited on 09/22/2021) (cit. on p. 11).

[17] Serghei Mangul et al. "Improving the Usability and Archival Stability of Bioinformatics Software". In: *Genome Biology* 20.1 (Feb. 2019), p. 47. ISSN: 1474-760X. DOI: 10.1186/s13059-019-1649-8. URL: https://doi.org/10.1186/s13059-019-1649-8 (visited on 09/22/2021) (cit. on p. 12).

[18] Sudhir Kumar and Joel Dudley. "Bioinformatics Software for Biologists in the Genomics Era". In: *Bioinformatics* 23.14 (July 2007), pp. 1713–1717. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btm239. URL: https://doi.org/10.1093/bioinformatics/btm239 (visited on 09/22/2021) (cit. on p. 12).

[19] James Howison and Julia Bullard. “Software in the Scientific Literature: Problems with Seeing, Finding, and Using Software Mentioned in the Biology Literature”. In: *Journal of the Association for Information Science and Technology* 67.9 (2016), pp. 2137–2155. ISSN: 2330-1643. DOI: 10.1002/asi.23538. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23538 (visited on 09/21/2021) (cit. on p. 12).

[20] David Ribes and Thomas Finholt. “The Long Now of Technology Infrastructure: Articulating Tensions in Development”. In: *Journal of the Association for Information Systems* 10.5 (May 2009), pp. 375–398. ISSN: 15369323. DOI: 10.17705/1jais.00199. URL: https://aisel.aisnet.org/jais/vol10/iss5/5/ (visited on 05/02/2021) (cit. on pp. 13, 14).

[21] Stephen Altschul et al. “The Anatomy of Successful Computational Biology Software”. In: *Nature Biotechnology* 31.10 (Oct. 2013), pp. 894–897. ISSN: 1546-1696. DOI: 10.1038/nbt.2721. URL: https://www.nature.com/articles/nbt.2721 (visited on 09/22/2021) (cit. on p. 13).

[22] Sean B. Palmer. *Ditching the Semantic Web?* Mar. 2008. URL: http://inamidst.com/whits/2008/ditching (visited on 09/24/2021) (cit. on p. 13).

[23] Annabelle Gawer. “Bridging Differing Perspectives on Technological Platforms: Toward an Integrative Framework”. In: *Research Policy* 43.7 (Sept. 2014), pp. 1239–1249. ISSN: 0048-7333. DOI: 10.1016/j.respol.2014.03.006. URL: https://www.sciencedirect.com/science/article/pii/S0048733314000456 (visited on 05/02/2021) (cit. on p. 14).

[24] Lindsay Poirier. “A Turn for the Scruffy: An Ethnographic Study of Semantic Web Architecture”. In: *Proceedings of the 2017 ACM on Web Science Conference*. WebSci ’17. New York, NY, USA: Association for Computing Machinery, June 2017, pp. 359–367. ISBN: 978-1-4503-4896-6. DOI: 10.1145/3091478.3091505. URL: https://doi.org/10.1145/3091478.3091505 (visited on 07/01/2021) (cit. on p. 14).

[25] Aaron Swartz. “Aaron Swartz’s A Programmable Web: An Unfinished Work”. In: *Synthesis Lectures on the Semantic Web: Theory and Technology* 3.2 (Feb. 2013), pp. 1–64. ISSN: 2160-4711, 2160-472X. DOI: 10.2200/S00481ED1V01Y201302WBE005. URL: http://www.morganclaypool.com/doi/abs/10.2200/S00481ED1V01Y201302WBE005 (visited on 06/30/2021) (cit. on p. 16).

[26] Björn Brembs et al. “Replacing Academic Journals”. In: (Sept. 2021). DOI: 10.5281/zenodo.5526635. URL: https://zenodo.org/record/5526635 (visited on 09/24/2021) (cit. on p. 17).

[27] Ian MacInnes. “Compatibility Standards and Monopoly Incentives: The Impact of Service-Based Software Licensing”. In: *International Journal of Services and Standards* 1.3 (Jan. 2005), pp. 255–270. ISSN: 1740-8849. DOI: 10.1504/IJSS.2005.005799. URL: https://www.inderscienceonline.com/doi/abs/10.1504/IJSS.2005.005799 (visited on 09/24/2021) (cit. on p. 17).

[28] Sam Biddle. *LexisNexis to Provide Giant Database of Personal Information to ICE*. Apr. 2021. URL: https://theintercept.com/2021/04/02/ice-database-surveillance-lexisnexis/ (visited on 09/24/2021) (cit. on p. 17).

[29] “Criticism of Amazon”. In: *Wikipedia* (Sept. 2021). URL: https://en.wikipedia.org/w/index.php?title=Criticism_of_Amazon&oldid=1043543093 (visited on 09/25/2021) (cit. on p. 17).

[30] *NIH Strategic Plan for Data Science*. Tech. rep. National Institutes of Health, June 2018. URL: https://web.archive.org/web/20210907014444/https://datascience.nih.gov/sites/default/files/NIH_Strategic_Plan_for_Data_Science_Final_508.pdf (visited on 09/22/2021) (cit. on pp. 18, 43).

[31] Stephen Buranyi. "Is the Staggeringly Profitable Business of Scientific Publishing Bad for Science?" In: *The Guardian* (June 2017). ISSN: 0261-3077. URL: https://www.theguardian.com/science/2017/jun/27/profitable-business-scientific-publishing-bad-for-science (visited on 09/25/2021) (cit. on p. 18).

[32] *Elsevier and Seven Bridges Receive NIH Data Commons Grant for Biomedical Data Analysis*. Nov. 2017. URL: https://www.elsevier.com/about/press-releases/archive/science-and-technology/elsevier-and-seven-bridges-receive-nih-data-commons-grant-for-biomedical-data-analysis (visited on 10/06/2021) (cit. on p. 18).

[33] R. Todd Reilly. *NIH STRIDES Initiative*. Jan. 2021. URL: https://web.archive.org/web/20211006011408/https://ncihub.org/resources/2422/download/21.01.08_NIH_STRIDES_Presentation.pptx (visited on 10/06/2021) (cit. on p. 18).

[34] *STRIDES Initiative Success Story: University of Michigan TOPMed | Data Science at NIH*. Oct. 2020. URL: https://web.archive.org/web/20210324024612/https://datascience.nih.gov/strides-initiative-success-story-university-michigan-topmed (visited on 10/06/2021) (cit. on p. 19).

[35] Ed S. Lein et al. "Genome-Wide Atlas of Gene Expression in the Adult Mouse Brain". In: *Nature* 445.7124 (Jan. 2007), pp. 168–176. ISSN: 1476-4687. DOI: 10.1038/nature05453. URL: https://www.nature.com/articles/nature05453 (visited on 03/15/2021) (cit. on p. 21).

[36] Sten Grillner et al. "Worldwide Initiatives to Advance Brain Research". In: *Nature Neuroscience* 19.9 (Sept. 2016), pp. 1118–1122. ISSN: 1546-1726. DOI: 10.1038/nn.4371. URL: https://www.nature.com/articles/nn.4371 (visited on 03/15/2021) (cit. on p. 21).

[37] Christof Koch and Allan Jones. "Big Science, Team Science, and Open Science for Neuroscience". In: *Neuron* 92.3 (Nov. 2016), pp. 612–616. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2016.10.019. URL: https://www.sciencedirect.com/science/article/pii/S0896627316307206 (visited on 03/15/2021) (cit. on p. 21).

[38] The International Brain Laboratory et al. "Standardized and Reproducible Measurement of Decision-Making in Mice". In: *bioRxiv* (Oct. 2020), p. 2020.01.17.909838. DOI: 10.1101/2020.01.17.909838. URL: https://www.biorxiv.org/content/10.1101/2020.01.17.909838v5 (visited on 03/15/2021) (cit. on p. 21).

[39] The International Brain Laboratory et al. "Data Architecture for a Large-Scale Neuroscience Collaboration". In: *bioRxiv* (Feb. 2020), p. 827873. DOI: 10.1101/827873. URL: https://www.biorxiv.org/content/10.1101/827873v2 (visited on 03/15/2021) (cit. on p. 21).

[40] Gonçalo Lopes et al. "Bonsai: An Event-Based Framework for Processing and Controlling Data Streams". In: *Frontiers in Neuroinformatics* 9 (2015). ISSN: 1662-5196. DOI: 10.3389/fninf.2015.00007. URL: https://www.frontiersin.org/articles/10.3389/fninf.2015.00007/full (visited on 03/15/2021) (cit. on p. 24).

[41] *Rfc5321 - Simple Mail Transfer Protocol*. URL: https://datatracker.ietf.org/doc/html/rfc5321#section-3 (visited on 09/27/2021) (cit. on p. 26).

[42] D. Clark. "The Design Philosophy of the DARPA Internet Protocols". In: *Symposium Proceedings on Communications Architectures and Protocols*. SIGCOMM '88. New York, NY, USA: Association for Computing Machinery, Aug. 1988, pp. 106–114. ISBN: 978-0-89791-279-2. DOI: 10.1145/52324.52336. URL: https://doi.org/10.1145/52324.52336 (visited on 03/15/2021) (cit. on pp. 27, 28).

[43] Brian E. Carpenter. *RFC 1958 - Architectural Principles of the Internet*. June 1996. URL: https://tools.ietf.org/html/rfc1958 (visited on 03/15/2021) (cit. on pp. 27, 29).

[44] Tim Berners-Lee. *Principles of Design*. 1998. URL: https://www.w3.org/DesignIssues/Principles.html#Decentrali (visited on 03/15/2021) (cit. on p. 27).

[45] John Markoff. "Tomorrow, the World Wide Web!;Microsoft, the PC King, Wants to Reign Over the Internet". In: *The New York Times* (July 1996). ISSN: 0362-4331. URL: https://www.nytimes.com/1996/07/16/business/tomorrow-world-wide-web-microsoft-pc-king-wants-reign-over-internet.html (visited on 10/12/2021) (cit. on p. 29).

[46] Archive Team. *Scientific Data Formats - Just Solve the File Format Problem*. URL: http://justsolve.archiveteam.org/wiki/Scientific_Data_formats (visited on 09/27/2021) (cit. on p. 30).

[47] Oliver Rübel et al. "NWB:N 2.0: An Accessible Data Standard for Neurophysiology". In: *bioRxiv* (Jan. 2019), p. 523035. DOI: 10.1101/523035. URL: https://www.biorxiv.org/content/10.1101/523035v1 (visited on 03/15/2021) (cit. on p. 30).

[48] Oliver Rübel et al. *The Neurodata Without Borders Ecosystem for Neurophysiological Data Science*. Mar. 2021. DOI: 10.1101/2021.03.13.435173. URL: https://www.biorxiv.org/content/10.1101/2021.03.13.435173v1 (visited on 09/27/2021) (cit. on p. 30).

[49] Xuemin Shen et al. *Handbook of Peer-to-Peer Networking*. Springer Science & Business Media, Mar. 2010. ISBN: 978-0-387-09751-0 (cit. on p. 31).

[50] Bram Cohen. *The BitTorrent Protocol Specification*. Feb. 2017. URL: https://www.bittorrent.org/beps/bep_0003.html (visited on 09/28/2021) (cit. on p. 31).

[51] Janko Roettgers. *The Pirate Bay: Distributing the World's Entertainment for $3,000 a Month*. July 2009. URL: https://gigaom.com/2009/07/19/the-pirate-bay-distributing-the-worlds-entertainment-for-3000-a-month/ (visited on 09/28/2021) (cit. on p. 32).

[52] *The Pirate Bay - Archiveteam*. Sept. 2020. URL: https://wiki.archiveteam.org/index.php?title=The_Pirate_Bay&oldid=45467 (visited on 09/28/2021) (cit. on p. 32).

[53] Jeffrey Spies. *Data Integrity for Librarians, Archivists, and Criminals: What We Can Steal from Bitcoin, BitTorrent, and Usenet*. Mar. 2017. URL: https://www.cni.org/topics/digital-curation/data-integrity-for-librarians-archivists-and-criminals-what-we-can-steal-from-bitcoin-bittorrent-and-usenet (visited on 10/01/2021) (cit. on p. 32).

[54] Eddie Kim. *After 15 Years, the Pirate Bay Still Can't Be Killed*. May 2019. URL: https://melmagazine.com/en-us/story/after-15-years-the-pirate-bay-still-cant-be-killed (visited on 09/28/2021) (cit. on p. 32).

[55] Ernesto Van der Sar. *The Open Bay: Now Anyone Can Run A Pirate Bay 'Copy'*. Dec. 2014. URL: https://torrentfreak.com/open-bay-now-everyone-can-run-pirate-bay-copy-141219/ (visited on 09/28/2021) (cit. on p. 32).

[56] Ernesto Van der Sar. *What.Cd Is Dead, But The Torrent Hydra Lives On*. Dec. 2016. URL: https://torrentfreak.com/what-cd-is-dead-but-the-torrent-hydra-lives-on-161202/ (visited on 03/18/2021) (cit. on p. 32).

[57] Jason Scott. *Geocities Torrent Update*. Dec. 2010. URL: http://ascii.textfiles.com/archives/2894 (visited on 09/30/2021) (cit. on p. 32).

[58] Dario Rossi et al. "Peeking through the BitTorrent Seedbox Hosting Ecosystem". In: *Traffic Monitoring and Analysis*. Ed. by Alberto Dainotti, Anirban Mahanti, and Steve Uhlig. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2014, pp. 115–126. ISBN: 978-3-642-54999-1. DOI: 10.1007/978-3-642-54999-1_10 (cit. on p. 33).

[59] John Hoffman and DeHackEd. *HTTP-Based Seeding Specification*. URL: http://www.bittornado.com/docs/webseed-spec.txt (visited on 09/30/2021) (cit. on p. 33).

[60] Brewster Kahle. *Over 1,000,000 Torrents of Downloadable Books, Music, and Movies*. Aug. 2012. URL: http://blog.archive.org/2012/08/07/over-1000000-torrents-of-downloadable-books-music-and-movies/ (visited on 09/30/2021) (cit. on p. 33).

[61] G. Kreitz and F. Niemela. "Spotify – Large Scale, Low Latency, P2P Music-on-Demand Streaming". In: *2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P)*. Delft, Netherlands: IEEE, Aug. 2010, pp. 1–10. ISBN: 978-1-4244-7140-9. DOI: 10.1109/P2P.2010.5569963. URL: http://ieeexplore.ieee.org/document/5569963/ (visited on 10/01/2021) (cit. on p. 33).

[62] Andrey Andreev et al. "Biologists Need Modern Data Infrastructure on Campus". In: *arXiv:2108.07631 [q-bio]* (Aug. 2021). arXiv: 2108.07631 [q-bio]. URL: http://arxiv.org/abs/2108.07631 (visited on 09/24/2021) (cit. on p. 33).

[63] Juan Benet. "IPFS - Content Addressed, Versioned, P2P File System". In: *arXiv:1407.3561 [cs]* (July 2014). arXiv: 1407.3561 [cs]. URL: http://arxiv.org/abs/1407.3561 (visited on 03/20/2021) (cit. on p. 34).

[64] Maxwell Ogden. *Dat - Distributed Dataset Synchronization And Versioning*. Preprint. Open Science Framework, Jan. 2017. DOI: 10.31219/osf.io/nsv2c. URL: https://osf.io/nsv2c (visited on 10/01/2021) (cit. on p. 34).

[65] Constantinos Patsakis and Fran Casino. "Hydras and IPFS: A Decentralised Playground for Malware". In: *International Journal of Information Security* 18.6 (Dec. 2019), pp. 787–799. ISSN: 1615-5262, 1615-5270. DOI: 10.1007/s10207-019-00443-0. arXiv: 1905.11880. URL: http://arxiv.org/abs/1905.11880 (visited on 10/01/2021) (cit. on p. 34).

[66] C. Zhang et al. "Unraveling the BitTorrent Ecosystem". In: *IEEE Transactions on Parallel and Distributed Systems* 22.7 (July 2011), pp. 1164–1177. ISSN: 1558-2183. DOI: 10.1109/TPDS.2010.123 (cit. on pp. 34, 35).

[67] Ian Clarke et al. “Freenet: A Distributed Anonymous Information Storage and Retrieval System”. In: *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability Berkeley, CA, USA, July 25–26, 2000 Proceedings*. Ed. by Hannes Federrath. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2001, pp. 46–66. ISBN: 978-3-540-44702-3. DOI: 10.1007/3-540-44702-4_4. URL: https://doi.org/10.1007/3-540-44702-4_4 (visited on 03/18/2021) (cit. on p. 34).

[68] Sameer Hinduja. “Deindividuation and Internet Software Piracy”. In: *CyberPsychology & Behavior* 11.4 (Aug. 2008), pp. 391–398. ISSN: 1094-9313. DOI: 10.1089/cpb.2007.0048. URL: https://www.liebertpub.com/doi/abs/10.1089/cpb.2007.0048 (visited on 10/01/2021) (cit. on p. 35).

[69] Jonathan Robert Basamanowicz. “Release Groups and Digital Copyright Piracy”. Thesis. Arts & Social Sciences: School of Criminology, May 2011. DOI: 10/etd6644_JBasamanowicz.pdf. URL: https://summit.sfu.ca/item/11710 (visited on 10/01/2021) (cit. on p. 35).

[70] Ian Dunham. “What.CD: A Legacy of Sharing”. PhD thesis. Rutgers University - School of Graduate Studies, 2018. DOI: 10.7282/T3V128F3. URL: https://rucore.libraries.rutgers.edu/rutgers-lib/58981/ (visited on 03/16/2021) (cit. on p. 35).

[71] Jody Rosen. “The Day the Music Burned”. In: *The New York Times* (June 2019). ISSN: 0362-4331. URL: https://www.nytimes.com/2019/06/11/magazine/universal-fire-master-recordings.html (visited on 03/18/2021) (cit. on pp. 35, 36).

[72] Nikhil Sonnad. *A Eulogy for What.Cd, the Greatest Music Collection in the History of the World—until It Vanished*. Nov. 2016. URL: https://qz.com/840661/what-cd-is-gone-a-eulogy-for-the-greatest-music-collection-in-the-world/ (visited on 03/16/2021) (cit. on p. 36).

[73] M Meulpolder et al. “Public and Private BitTorrent Communities: A Measurement Study”. In: (), p. 5 (cit. on p. 36).

[74] Adele Lu Jia et al. “How to Survive and Thrive in a Private BitTorrent Community”. In: *Distributed Computing and Networking*. Ed. by Davide Frey et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013, pp. 270–284. ISBN: 978-3-642-35668-1. DOI: 10.1007/978-3-642-35668-1_19 (cit. on p. 37).

[75] Z. Liu et al. “Understanding and Improving Ratio Incentives in Private Communities”. In: *2010 IEEE 30th International Conference on Distributed Computing Systems*. June 2010, pp. 610–621. DOI: 10.1109/ICDCS.2010.90 (cit. on p. 37).

[76] Benedikt Fecher et al. “A Reputation Economy: How Individual Reward Considerations Trump Systemic Arguments for Open Access to Data”. In: *Palgrave Communications* 3.1 (June 2017), pp. 1–10. ISSN: 2055-1045. DOI: 10.1057/palcomms.2017.51. URL: https://www.nature.com/articles/palcomms201751 (visited on 10/01/2021) (cit. on p. 38).

[77] Jordan Bross. “Community, Collaboration and Contribution: Evaluating a BitTorrent Tracker as a Digital Library.” M.S. in Library Science. UNC Chapel Hill, Dec. 2013. URL: https://doi.org/10.17615/g1cw-kw06 (cit. on p. 39).

[78] Morgan G. I. Langille and Jonathan A. Eisen. "BioTorrents: A File Sharing Service for Scientific Data". In: *PLoS ONE* 5.4 (Apr. 2010). ISSN: 1932-6203. DOI: 10.1371/journal.pone.0010071. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2854681/ (visited on 04/01/2021) (cit. on pp. 39, 59).

[79] Joseph Paul Cohen and Henry Z. Lo. "Academic Torrents: A Community-Maintained Distributed Repository". In: *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment*. XSEDE '14. New York, NY, USA: Association for Computing Machinery, July 2014, pp. 1–2. ISBN: 978-1-4503-2893-7. DOI: 10.1145/2616498.2616528. URL: https://doi.org/10.1145/2616498.2616528 (visited on 10/04/2021) (cit. on p. 39).

[80] Werner Ceusters and Barry Smith. "Foundations for a Realist Ontology of Mental Disease". In: *Journal of Biomedical Semantics* 1.1 (Dec. 2010), p. 10. ISSN: 2041-1480. DOI: 10.1186/2041-1480-1-10. URL: https://doi.org/10.1186/2041-1480-1-10 (visited on 10/07/2021) (cit. on p. 43).

[81] The Biomedical Data Translator Consortium. "The Biomedical Data Translator Program: Conception, Culture, and Community". In: *Clinical and Translational Science* 12.2 (2019), pp. 91–94. ISSN: 1752-8062. DOI: 10.1111/cts.12592. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cts.12592 (visited on 10/05/2021) (cit. on p. 43).

[82] Stacia Fleisher. "Other Transaction Award Policy Guide - Biomedical Data Translator Program". In: (Sept. 2019), p. 38 (cit. on p. 44).

[83] The Biomedical Data Translator Consortium. "Toward A Universal Biomedical Data Translator". In: *Clinical and Translational Science* 12.2 (2019), pp. 86–90. ISSN: 1752-8062. DOI: 10.1111/cts.12591. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cts.12591 (visited on 10/05/2021) (cit. on pp. 44, 45, 50).

[84] Richard Bruskiewich et al. *Biolink/Biolink-Model: 2.2.5*. Zenodo. Sept. 2021. URL: https://zenodo.org/record/5520104 (visited on 10/05/2021) (cit. on p. 44).

[85] Prateek Goel et al. "Explanation Container in Case-Based Biomedical Question-Answering". In: Sept. 2021, p. 10. URL: https://web.archive.org/web/*/https://gaia.fdi.ucm.es/events/xcbr/papers/ICCBR_2021_paper_100.pdf (cit. on pp. 44, 46).

[86] *ROBOKOP - CoVar*. Oct. 2021. URL: https://web.archive.org/web/20211006030919/https://covar.com/case-study/robokop/ (visited on 10/06/2021) (cit. on pp. 44, 45).

[87] A Ram et al. "Transphobia, Encoded: An Examination of Trans-Specific Terminology in SNOMED CT and ICD-10-CM". In: *Journal of the American Medical Informatics Association* ocab200 (Sept. 2021). ISSN: 1527-974X. DOI: 10.1093/jamia/ocab200. URL: https://doi.org/10.1093/jamia/ocab200 (visited on 10/07/2021) (cit. on pp. 45, 50).

[88] Ruth Hailu. *NIH-Funded Project Aims to Build a 'Google' for Biomedical Data*. July 2019. URL: https://www.statnews.com/2019/07/31/nih-funded-project-aims-to-build-a-google-for-biomedical-data/ (visited on 10/06/2021) (cit. on pp. 46, 49).

[89] Trishan Panch, Heather Mattie, and Leo Anthony Celi. "The "Inconvenient Truth" about AI in Healthcare". In: *npj Digital Medicine* 2.1 (Aug. 2019), pp. 1–3. ISSN: 2398-6352. DOI: 10.1038/s41746-

019-0155-4. URL: https://www.nature.com/articles/s41746-019-0155-4 (visited on 10/07/2021) (cit. on p. 46).

[90] Melissa A Haendel. *A Common Dialect for Infrastructure and Services in Translator*. Feb. 2021. URL: https://reporter.nih.gov/project-details/10330632 (visited on 10/07/2021) (cit. on p. 47).

[91] Kyle R. Hukezalie et al. "In Vitro and Ex Vivo Inhibition of Human Telomerase by Anti-HIV Nucleoside Reverse Transcriptase Inhibitors (NRTIs) but Not by Non-NRTIs". In: *PLoS ONE* 7.11 (Nov. 2012), e47505. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0047505. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3499584/ (visited on 10/06/2021) (cit. on p. 47).

[92] *Zidovudine - Patient | NIH*. URL: https://clinicalinfo.hiv.gov/en/drugs/zidovudine/patient (visited on 10/06/2021) (cit. on p. 47).

[93] *RePORT ⊠ RePORTER "Biomedical Data Translator"*. Oct. 2021. URL: https://reporter.nih.gov/search/kDJ97zGUFEaIBIltUmyd_Q/projects?sort_field=FiscalYear&sort_order=desc (visited on 10/05/2021) (cit. on p. 48).

[94] *AWS Announces AWS Healthcare Accelerator for Startups in the Public Sector*. June 2021. URL: https://aws.amazon.com/blogs/publicsector/aws-announces-healthcare-accelerator-program-startups-public-sector/ (visited on 10/06/2021) (cit. on p. 48).

[95] Rachel Lerman. "Amazon Built Its Own Health-Care Service for Employees. Now It's Selling It to Other Companies." In: *Washington Post* (Mar. 2021). ISSN: 0190-8286. URL: https://www.washingtonpost.com/technology/2021/03/17/amazon-healthcare-service-care-expansion/ (visited on 10/06/2021) (cit. on p. 48).

[96] Krishna Bharat, Stephen Lawrence, and Mehran Sahami. "Generating User Information for Use in Targeted Advertising". US20050131762A1. June 2005. URL: https://patents.google.com/patent/US20050131762A1/en (visited on 10/07/2021) (cit. on p. 49).

[97] Marc Bourreau et al. "Google/Fitbit Will Monetise Health Data and Harm Consumers". In: 107 (2020), p. 13 (cit. on p. 49).

[98] Laila Rasmy et al. "Med-BERT: Pretrained Contextualized Embeddings on Large-Scale Structured Electronic Health Records for Disease Prediction". In: *npj Digital Medicine* 4.1 (May 2021), pp. 1–13. ISSN: 2398-6352. DOI: 10.1038/s41746-021-00455-y. URL: https://www.nature.com/articles/s41746-021-00455-y (visited on 10/07/2021) (cit. on p. 49).

[99] Chase Strangio. "Can Reproductive Trans Bodies Exist?" In: *City University of New York Law Review* 19.2 (July 2016), p. 223. ISSN: 2572-7788. URL: https://academicworks.cuny.edu/clr/vol19/iss2/3 (cit. on p. 50).

[100] Dennis Heimbigner and Dennis McLeod. "A Federated Architecture for Information Management". In: *ACM Transactions on Information Systems* 3.3 (July 1985), pp. 253–278. ISSN: 1046-8188. DOI: 10.1145/4229.4233. URL: https://doi.org/10.1145/4229.4233 (visited on 03/25/2021) (cit. on p. 53).

[101] Susanne Busse et al. "Federated Information Systems: Concepts, Terminology and Architectures". In: (1999), p. 40 (cit. on p. 54).

[102] Amit P. Sheth and James A. Larson. “Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases”. In: *ACM Computing Surveys* 22.3 (Sept. 1990), pp. 183–236. ISSN: 0360-0300. DOI: 10.1145/96602.96604. URL: https://doi.org/10.1145/96602.96604 (visited on 03/25/2021) (cit. on p. 54).

[103] Angela Bonifati et al. “Distributed Databases and Peer-to-Peer Databases: Past and Present”. In: *ACM SIGMOD Record* 37.1 (Mar. 2008), pp. 5–11. ISSN: 0163-5808. DOI: 10.1145/1374780.1374781. URL: https://doi.org/10.1145/1374780.1374781 (visited on 10/11/2021) (cit. on p. 54).

[104] Giuseppe Pirrò, Domenico Talia, and Paolo Trunfio. “A DHT-Based Semantic Overlay Network for Service Discovery”. In: *Future Generation Computer Systems* 28.4 (Apr. 2012), pp. 689–707. ISSN: 0167-739X. DOI: 10.1016/j.future.2011.11.007. URL: https://www.sciencedirect.com/science/article/pii/S0167739X11002305 (visited on 10/11/2021) (cit. on p. 55).

[105] Christopher Webber et al. *ActivityPub*. W3C Recommendation. W3C, Jan. 2018. URL: https://www.w3.org/TR/2018/REC-activitypub-20180123/ (cit. on p. 55).

[106] Manu Sporny et al. *JSON-LD 1.1 - A JSON-Based Serialization for Linked Data*. July 2020. URL: https://www.w3.org/TR/json-ld/ (visited on 10/12/2021) (cit. on p. 55).

[107] James M Snell and Evan Prodromou. *Activity Streams 2.0*. May 2017. URL: https://www.w3.org/TR/activitystreams-core/ (visited on 10/12/2021) (cit. on p. 55).

[108] Adam S. Charles et al. “Toward Community-Driven Big Open Brain Science: Open Big Data and Tools for Structure, Function, and Genetics”. In: *Annual Review of Neuroscience* 43 (July 2020), pp. 441–464. ISSN: 1545-4126. DOI: 10.1146/annurev-neuro-100119-110036 (cit. on p. 57).

[109] Yaroslav O. Halchenko et al. “DataLad: Distributed System for Joint Management of Code, Data, and Their Relationship”. In: *Journal of Open Source Software* 6.63 (July 2021), p. 3262. ISSN: 2475-9066. DOI: 10.21105/joss.03262. URL: https://joss.theoj.org/papers/10.21105/joss.03262 (visited on 10/02/2021) (cit. on p. 58).

[110] Jeffrey Spies. “A Workflow-Centric Approach to Increasing Reproducibility and Data Integrity”. In: (Aug. 2017). URL: https://scholarworks.iu.edu/dspace/handle/2022/21729 (visited on 10/01/2021) (cit. on p. 59).

[111] Dimitri Yatsenko et al. “DataJoint Elements: Data Workflows for Neurophysiology”. In: *bioRxiv* (Mar. 2021), p. 2021.03.30.437358. DOI: 10.1101/2021.03.30.437358. URL: https://www.biorxiv.org/content/10.1101/2021.03.30.437358v1 (visited on 04/30/2021) (cit. on p. 60).

[112] Matthew Wall. “Reliability Starts with the Experimental Tools Employed”. In: (Nov. 2018). DOI: 10.31234/osf.io/upynr. URL: https://psyarxiv.com/upynr/ (visited on 03/20/2019) (cit. on p. 67).

[113] Fatemeh Abyarjoo et al. “Implementing a Sensor Fusion Algorithm for 3D Orientation Detection with Inertial/Magnetic Sensors”. In: *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*. Ed. by Tarek Sobh and Khaled Elleithy. Lecture Notes in Electrical Engineering. Cham: Springer International Publishing, 2015, pp. 305–310. ISBN: 978-3-319-06773-5. DOI: 10.1007/978-3-319-06773-5_41 (cit. on p. 67).

[114] Roger Labbe. *Kalman and Bayesian Filters in Python*. Commit 24b9fb3. Oct. 2020. URL: https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python (visited on 04/21/2021) (cit. on p. 67).

[115] Roger Labbe. *Rlabbe/Filterpy*. Apr. 2021. URL: https://github.com/rlabbe/filterpy (visited on 04/22/2021) (cit. on p. 67).

[116] Remco Heesen and Liam Kofi Bright. "Is Peer Review a Good Idea?" In: *The British Journal for the Philosophy of Science* 0.0 (May 2020), pp. 000–000. DOI: 10.1093/bjps/axz029. eprint: https://doi.org/10.1093/bjps/axz029. URL: https://doi.org/10.1093/bjps/axz029 (cit. on p. 71).

[117] Maged Kamel Boulos. "Semantic Wikis: A Comprehensible Introduction with Examples from the Health Sciences". In: *Journal of Emerging Technologies in Web Intelligence* 1 (Aug. 2009). DOI: 10.4304/jetwi.1.1.94-96 (cit. on p. 72).

[118] Tadeu Classe et al. "A Distributed Infrastructure to Support Scientific Experiments". In: *Journal of Grid Computing* 15.4 (Dec. 2017), pp. 475–500. ISSN: 1572-9184. DOI: 10.1007/s10723-017-9401-7. URL: https://doi.org/10.1007/s10723-017-9401-7 (visited on 03/09/2021) (cit. on p. 72).

[119] Benjamin M. Good, Joseph T. Tennis, and Mark D. Wilkinson. "Social Tagging in the Life Sciences: Characterizing a New Metadata Resource for Bioinformatics". In: *BMC Bioinformatics* 10.1 (Sept. 2009), p. 313. ISSN: 1471-2105. DOI: 10.1186/1471-2105-10-313. URL: https://doi.org/10.1186/1471-2105-10-313 (visited on 04/08/2021) (cit. on p. 72).

[120] Kei-Hoi Cheung et al. "Semantic Web Approach to Database Integration in the Life Sciences". In: *Semantic Web*. Ed. by Christopher J. O. Baker and Kei-Hoi Cheung. Boston, MA: Springer US, 2007, pp. 11–30. ISBN: 978-0-387-48436-5. DOI: 10.1007/978-0-387-48438-9_2. URL: http://link.springer.com/10.1007/978-0-387-48438-9_2 (visited on 03/30/2021) (cit. on p. 72).

[121] Ana Claudia Sima et al. "Enabling Semantic Queries across Federated Bioinformatics Databases". In: *Database* 2019.baz106 (Jan. 2019). ISSN: 1758-0463. DOI: 10.1093/database/baz106. URL: https://doi.org/10.1093/database/baz106 (visited on 03/30/2021) (cit. on p. 72).