JONNY L. SAUNDERS

DECENTRALIZED INFRASTRUCTURE FOR NEUROSCIENCE

Trimmings from the main document for future pieces

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

{% include toc_start.html %} 1. table of contents {:toc} {% include toc_end.html %}

PDF VERSION

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

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

[Tilson et al., 2010]

Acknowledgements in no order at all!!! (make sure to double check spelling!!! and then also double check it's cool to list them!!!):

- Lucas Ott, the steadfast
- Tillie Morris
- Nick Sattler
- Sam Mehan
- Molly Shallow
- Mike and as always ty for letting me always go rogue

- Matt Smear
- Santiago Jaramillo
- Gabriele Hayden
- Eartha Mae
- jakob voigts for participating in the glue wiki
- nwb & dandi team for dealing w/ my inane rambling
- Tomasz Pluskiewicz
- James Meickle
- Gonçalo Lopes
- Mackenzie Mathis
- Lauren E. Wool
- Gabi Hayden
- Mark Laubach & Open Behavior Team
- Os Keyes
- Avery Everhart
- Eartha Mae Guthman
- Olivia Guest
- Irene Knapp
- Nire Bryce
- Kris Chauvin
- Phil Parker
- Chris Rogers
- Danny Mclanahan
- Petar Todorov
- Jeremy Delahanty
- Andrey Andreev
- Joel Chan
- Björn Brembs

- Sanjay Srivastava & Metascience Class
- Ralph Emilio Peterson
- Manuel Schottdorf
- Ceci Herbert
- The Emerging ONICE team
- The Janet Smith House, especially Leslie Harka
- Rumbly Tumbly Lawnmower
- lmk if we talked and i missed ya!

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 [Commission, 2017] .

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[Mirowski, 2018] , 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 [Ponzi, 2020] 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 [Bietz et al., 2012] .

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.

The State of Things

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 to publish full, raw data and analysis code, often because the labor of cleaning it is too great. The "Open science" movement, roughly construed, has reached a few hard limits from present infrastructure that have forced its energy to leak from the sides as bullying leaderboards or sets of symbols that are mere signifiers of openness. "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
- Reliance on communication platforms and knowledge systems that aren'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. Often our best recourse is to make a Slack about it, which is incapable of producing a public, durable, and cumulative resource: and so the same 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 it through the pinprickiest peephole of just the most recent data you or your lab have collected rather than being able to index across the many measurements of 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.

- 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 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 [Wool and Laboratory, 2020, BARLEY and BECHKY, 1994] that doesn't have a venue for sharing but is 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 the system of extractive platforms that dominate the flow of global information. For ourselves, we stand to have the practice of science filleted at the seams into a series of mutually incompatible subscription services. 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.

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 we are embedded in a system of mutually reinforcing disincentives to cumulative infrastructure development. Our incentive systems are coproductive with a number of deeply-embedded, economically powerful entities that would really prefer owning it all themselves, thanks. Put bluntly, "we are dealing with a massively entrenched set of institutions, built around the last information age and fighting for its life" [Bowker et al., 2010]

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.

(Mis)incentives in Scientific Software

Systems Neuro specific problems for infrastructure

The incentive systems in science are complex, subject to infinite variation everywhere, so these are intended as general tendencies rather than statements of irrevocable and uniform truth.

Incentivized Fragmentation

Scientific software development favors 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 (if any) paper. 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.

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. [Howison and Herbsleb, 2013]

For an example in Neuroscience, one can browse the papers that cite the DeepLabCut paper [Mathis et al., 2018] to find hundreds of downstream projects that make various extensions and improvements that are not integrated into the main library. While the alternative extreme of a single monolithic ur-library is also undesirable, working in fragmented islands makes infrastructure a random walk instead of a cumulative effort.

After publication, scientists have little incentive to **maintain** software outside of the domains in which the primary contributors use it, so outside of the most-used libraries most scientific software is brittle and difficult to use [Mangul et al., 2019, Kumar and Dudley, 2007] .

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 [Howison and Bullard, 2016] .

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, 2018] :

> 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.

"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 of its use — what Ribes and Finholt call "The Long Now. [Ribes and Finholt, 2009] " 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 junior and senior researchers. ECRs might be interested in developing tools they'll use throughout their careers, but given the pressure to establish their reputation with publications rarely have the time to develop something fully. The time pressure never ends, and established researchers also to push enough publications through the door to be able to secure the next round of funding. The time preference of scientific software development is very short: hack it together, get the paper out, we'll fix it later.

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 most programmers *never* have a chance to learn best practices commonly accepted in software engineering. As a consequence, a lot of software tools are developed by near-amateurs with no formal software training, contributing to their brittleness [Altschul et al., 2013] .

The problem of time horizon in development is not purely a product of inexperience, and a longer time horizon is not uniformly better. We can look to the history of the semantic web, a project that was intended to bridge human and computer-readable content on the

web, for cautionary tales. In the semantic web era, thousands of some of the most gifted programmers and some of the original architects of the internet 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 (bold is mine, italics in original):

> **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. [Palmer, 2008]

Developing digital infrastructure must be both bound to fulfilling immediate, incremental needs as well as guided by a long-range vision. The technical and social lessons run in parallel: We need software that solves problems people actually have, but can flexibly support an eventual form that allows new possibilities. 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:

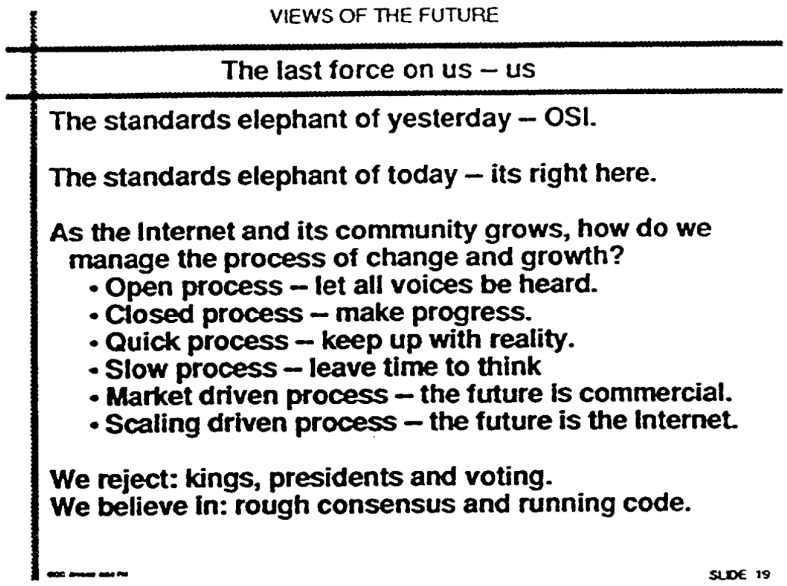
> "On the one hand, I know we have to keep it all running, but on the other, LTER is about long-term data archiving. If we want to do that, we have to have the time to test and enact new approaches. But if we're working on the to-do lists, we aren't working on the tomorrow-list" (LTER workgroup discussion 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?* [Ribes and Finholt, 2009]

"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" [Poirier, 2017] . 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. [Poirier, 2017] "

This tension is as old as the internet, where amidst the dot-com bubble a telecom spokesperson lamented that the internet wasn't controllable enough to be profitable because "it was devised by a bunch of hippie anarchists." [Hiltzik, 2001] The hippie anarchists probably agreed, rejecting "kings, presidents and voting" in favor of "rough consensus and running code." Clearly, the difference in thought styles has an unsubtle relationship with beliefs about who should be able to exercise power and what ends a system should serve [Larsen, 2012] .



A slide from David Clark's "Views of the Future"[Clark, 1992] that contrasts differing visions for the development process of the future of the internet. The struggle between engineered order and wild untamedness is summarized forcefully as "We reject: kings, presidents and voting. We believe in: rough consensus and running code

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!) [Swartz, 2013]

The outcomes of this cultural rift are subtle, but the broad strokes are clear: the "scruffies" largely diverged into the linked data community, which has taken some of the core semantic web technology like RDF, OWL, and the like, and developed a broad range of down-

stream technologies that have found purchase across information sciences, library sciences, and other applied domains[1]. 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 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.

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

[1] This isn't a story of "good people" and "bad people," as a lot of the linked data technology also serves as the backbone for abusive technology monopolies like google's acquisition of Freebase [Iain, 2019] and the profusion of knowledge graph-based medical platforms.

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 co-developers 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.

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, [Brembs et al., 2021]), and that criticism should extend to the rest of our infrastructure as information companies seek to build a for-profit platform system that spans the scientific workflow (eg. [Els, 2017]). The mode of privatization of scientific infrastructure follows the broader software market as a 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 [Mirowski, 2018])

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 [MacInnes, 2005] . 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 often profoundly unethical behavior of by funneling millions of dollars into them. Longterm, unconditional funding of wildly profitable journals has allowed conglomerates like Elsevier to become sprawling surveillance companies [REL, 2020] that are sucking as much data up as they can to market tools like algorithmic ranking of scientific productivity [Brembs, 2021] and making data sharing agreements with ICE [Biddle, 2021] . Or see our use of AWS and the laundry list of human rights abuses by Amazon [Cri, 2021] . In addition to lock-in, dependence on a constellation of SaaS allows the opportunity for platform-holders to take advantage of their limitations and *sell us additional services to make up for what the other ones purposely lack* — for example Elsevier has taken advantage of our dependence on the journal system and its strategic disorganization to sell a tool for summarizing trending research areas for tailoring maximally-fundable grants [Elsevier, a] .

Funding models and incentive structures in science are uniformly aligned towards the platformatization of scientific infrastructure. Aside from the corporate doublespeak 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 (emphases

mine):

> 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. [. . .]
>
> 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 [NIH, 2018])

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 [Buranyi, 2017] 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: the STRIDES program, with the official subdomain cloud.nih.gov, has been authorized to pay $85 million to cloud providers since 2018. In exchange, NIH hasn't received any sort of new technology, but "extramural" scientists receive a maximum discount of 25% on cloud storage and "data egress" fees as well as plenty of training on how to give control of the scientific process to platform giants [Reilly, 2021] [2]. With platforms, without exaggeration we pay them to let us pay for something that makes it so we need to pay them more later.

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] Their success stories tell the story of platform non-integration where scientists have to handbuild new tools to manage their data across multiple cloud environments: "We have been storing data in both cloud environments because we wanted the ecosystem we are creating to work on both clouds" [STR, 2020]

Protection of Institutional and Economic Power

Aside from information industries, infrastructural deficits are certainly not without beneficiaries within science — those that have already accrued power and status.

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 — one of many positive feedback loops of inequity.

More generally, incentives across infrastructures are often misaligned across strata of power and wealth. Those at the top of a

power hierarchy have every incentive to maintain the fragmentation that prevents people from competing — hopefully mostly unconsciously via uncritically participating in the system rather than maliciously reinforcing it.

This poses an organizational problem: the kind of infrastructure that unwinds platform ownership is not only unprofitable, it's anti-profitable – making it impossible to profit from its domain of use. That makes it difficult to rally the kind of development and lobbying resources that profitable technology can, requiring organization based on ethical principles and a commitment to sacrifice control in order to serve a practical need.

The problem is not insurmountable, and there are strategic advantages to decentralized infrastructure and its development within science. Centralized technologies and companies might have more concerted power, but we have *numbers* and can make tools that allow us to combine small amounts of labor from many people. A primary criticism of infrastructural overhauls is that they will cost a lot of *money,* but that's propaganda: the cost of decentralized technologies is far smaller than the vast sums of money funnelled into industry profits, labor hours spent compensating for the designed inefficiencies of the platform model, and the development of a fragmented tool ecosystem built around them.

Science, as one of few domains of non-economic labor, has the opportunity to be a seed for decentralized technologies that could broadly improve not only the health of scientific practice, but the broader information ecosystem. If we develop a plan and mobilize to make use of our collective expertise to build tools that have no business model and no means of development in commercial domains — we just need to realize what's at stake, develop a plan, and agree that the health of science is more important than the convenience of the cloud or which journal our papers go into.

The Ivies, Institutes, and "The Rest of Us"

Given these constraints 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.

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[3]:

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

[3] 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.

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[4], 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[5]. 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.

[4]

> Project Summary: Core 2, Data Science [...] 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. [...] 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. [...] https://projectreporter.nih.gov/project_info_desc

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

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 several transformative tools and datasets, including its well-known atlases

[Lein et al., 2007] 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 [Grillner et al., 2016]

> 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 [Koch and Jones, 2016]

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

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 [Mainen et al., 2016] . The most successful consortium of this kind has been the International Brain Laboratory [Abbott et al., 2017, Wool and Laboratory, 2020] , 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 [Laboratory et al., 2020a] and developing data management infrastructure to match [Laboratory et al., 2020b] (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. [Wool and Laboratory, 2020]

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.

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 [Lopes et al., 2015a] 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.

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.

The state of broader scientific deinfrastructuring is perhaps to be expected given its dependence on informational monopolies that in some part depend on it, but unlike many other industries or professions there is reason for hope in science. Science is packed with people with an enormous diversity of skills, resources, and

perspectives. Publicly funded science is relatively unique as a labor system that does not strictly depend on profit. There is widespread discontent with the systems of scientific practice, and so the question becomes how we can organize our skill, labor, and energy to rebuild the systems that constrain us.

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.

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 [Mainen et al., 2016] 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!

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.

!! need to integrate [Larsen, 2012]

Protocols, not Platforms

Much of the basic technology of the internet was developed as {% include rdfa2.html id="protocols" resource="protocols" typeof="skos:Concept" contents = "*protocols*" %} that describe the basic attributes and operations of a process. {% include rdfa2.html resource="protocols" property="skos:example" contents="A simple and common example is email over SMTP (Simple Mail Transfer Protocol)" %} [Rfc] . 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.

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" [Clark, 1988] . 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.

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 [Carpenter, 1996] . 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?" [Berners-Lee, 1998] .

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.

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 [Clark, 1988] .

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.

Infrastructure is Social

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 "community-ness," 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 [Grudin, 1994, Randall et al., 2011] .

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.

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" [Carpenter, 1996] 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) [Markoff, 1996] . *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**.

Shared Data

Formats as Onramps

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 format 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 [Team] , so a comprehensive treatment is impractical here and I will use the Neu-

rodata Without Borders:N (NWB)[Rübel et al., 2019] 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 [Rübel et al., 2021] 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.

Peer-to-peer as a Backbone

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[6] 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[7]. 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.

[6] granting agencies seem to love funding new databases, idk.

[7] As I am writing this, I am getting a (very unscientific) maximum speed of 5MB/s on the Open Science Framework

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[8], 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.

[8] peer to peer systems are, maybe predictably, a whole academic subdiscipline. See [Shen et al., 2010] for reference.

For the sake of concreteness, we can consider a (simplified) description of Bittorrent [Cohen, 2017] , 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 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 by requesting pieces of the file from each until they have the whole thing. 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 [Roettgers, 2009] . According to a database dump from 2013 [Pir, 2020] , 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[Spies, 2017a] . Despite more than 15 years of concerted effort by governments and intellectual property holders, the pirate bay is still alive and kicking [Kim, 2019] [9]. 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 [Van der Sar, 2014] – 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 [Van der Sar, 2016] . 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" [Scott, 2010] . And by groups who disseminate !! return here talking about ddosevrets.

[9] knock on wood

- 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 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 p2p system 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"[Rossi et al., 2014] when they use the bittorrent protocol, and "web seeds"[Hoffman and DeHackEd] 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." [Kahle, 2012]

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 [Kreitz and Niemela, 2010] , 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 [Andreev et al., 2021, Charles et al., 2020] , 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." [Benet, 2014] [10] Dat [Ogden, 2017] , 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 [Patsakis and Casino, 2019] , but for now sufficent to say p2p systems can handle version control.

[10] 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.

Bittorrent swarms are vulnerable to data loss if all the peers seeding a file disconnect (though the tail is longer than typically assumed, see [Zhang et al., 2011]), 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 [Clarke et al., 2001] . 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.

Solid is a project that almost exactly meets all these needs [Capadisli et al., 2020, Sambra et al., 2016, Sol] . Solid allows people to share data in Pods, which let them control access and distribution across storage system with a unified identity system. It is implementation-agnostic, and so can support peer-to-peer storage and transfer systems that comply with its protocol specification.

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, and that computer is usually connected to the internet. 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 from communities of practice, and there has been no more important and impactful decentralized archival project than internet piracy.

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[Basamanowicz, 2011, Hinduja, 2008] , 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." [Hinduja, 2008] . 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 [Basamanowicz, 2011] , 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 [Zhang et al., 2011]). 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.

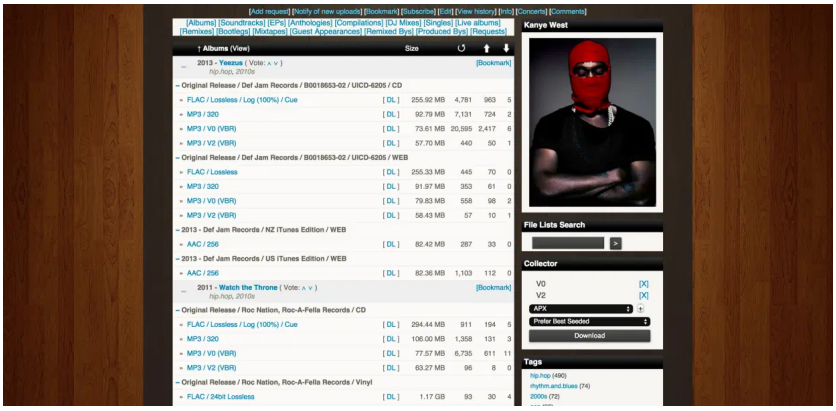
One of these private trackers was the site of one of the largest informational tragedies of the past decade: what.cd[11], which I will use as an example to describe some of these community systems.

[11] for a detailed description of the site and community, see Ian Dunham's dissertation [Dunham, 2018]

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[12] [Dunham, 2018] . 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 [Rosen, 2019] [ˆlostartists]. 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, period. One former user describes one example:

[12] 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."

> "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" [Sonnad, 2016]



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 [Meulpolder

et al.]). Invites were extremely rare, and the interview process was demanding to the point where extensive guides were written to prepare for them.

The what.cd incentive system was based on a required ratio of data uploaded vs. data downloaded [Jia et al., 2013] . 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 [Liu et al., 2010] .

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 [Kash et al., 2012, Chen et al., 2011] .

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 a library of the entirety of human music output[13], 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. Being

[13] 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?

an anonymous, gray-area community, there was of course plenty of power that was tripped on. Rather than being a messy hodgepodge of fake, low-quality uploads, though, 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. 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 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 [Fecher et al., 2017] . 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." One way to accomplish implement them is to embed a data sharing system within a social system that is designed to reward communitarian behavior.

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 [Bross, 2013] . 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 should think carefully about using downloading as the "cost," because downloading and analyzing huge amounts of data can be *good* and exactly what we *want* in some circumstances, but a threat to privacy and data governance in others.

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 [Langille and Eisen, 2010] or AcademicTorrents [Cohen and Lo, 2014] . 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.*

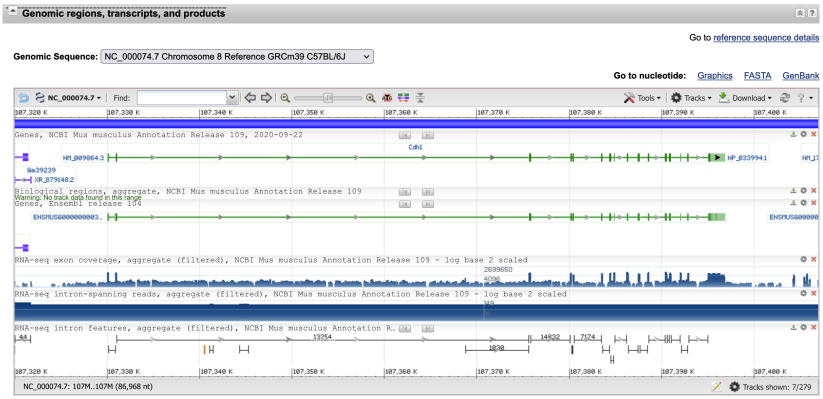
Linked Data or Surveillance Capitalism?

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[14]. 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.

[14] 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)

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[15] 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; 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.

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

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 re-structured 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.**" [Bietz and Lee, 2009]

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. [...] 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, non-existents, possibilia, and the like [Ceusters and Smith, 2010]

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. [NIH, 2018]

And NCATS elaborates it a bit more on the project "about" page:

> 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 [Consortium, 2019a] . OTA awards give the granting agency broad flexibility in to whom and for what money can be given, and consist of an initial competetive segment with possibility for indefinite noncompetitive extensions at the discretion of the agency [Fleisher, 2019] .

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 [Consortium, 2019b] . 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 [Bruskiewich et al., 2021] , 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[16][Consortium, 2019b] , 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 [Goel et al., 2021] , and descriptions from contractors [ROB, 2021] , 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 [Goel et al., 2021] — an example that is used repeatedly is 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

[16] 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

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 [ROB, 2021]) 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 [Ram et al., 2021] . 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.

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[17].

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

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 [Consortium, 2019b] , 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 [Con-

sortium, 2019b] 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 "translating data to language," with ambitions of providing a means of making algorithmic predictions for drug discovery and clinical practice rather than linking data [Hailu, 2019] Tools like these have been thoroughly problematized elsewhere, eg. [Grote and Berens, 2020, Obermeyer et al., 2019, Panch et al., 2019a,b] .

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 pre-processed 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. [Goel et al., 2021]

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 [Haendel, 2021]). Zidovudine, as a nucleoside reverse transcriptase inhibitor, does inhibit telomerase reverse transcriptase [Hukezalie et al., 2012] , but can also cause anemia and lower red blood cell counts [Zid] – 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 [ReP, 2021] . 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 [AWS, 2021] and even literally providing health care [Lerman, 2021] , and the uploading of all scientific and medical data onto AWS with entirely unenforceable promises of data privacy [Quinn, 2021] — 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.

!! It doesn't even need to be Amazon, the publishers are getting into it too! RELX owns lexisnexis, a big identity management company, and is aggressively building out its machine-learning tools for science. From their 2019 annual shareholders report:

> Elsevier serves academic and government research administrators and leaders through its Research Intelligence suite of products. SciVal is a decision tool that helps institutions to establish, execute and evaluate research strategies by leveraging bibliometric data [...] Elsevier expanded its leadership position in research institution benchmarking analytics through further investment in its SciVal Topic Prominence in Science. Big data technology takes into consideration nearly all of the articles available in Scopus since 1996 and clusters them into nearly 96,000 global, unique research topics based on citations patterns.
>
> Elsevier's flagship clinical reference platform, ClinicalKey, provides physicians, nurses and pharmacists with access to leading Elsevier and third-party reference and evidence-based medical content [...] Elsevier has developed a Healthcare Knowledge Graph, which utilises ML and Natural Language Processing (NLP) to knit together its collection of the world's foremost clinical knowledge. The Healthcare Knowledge Graph enhances ClincialKey, the portal into Elsevier's vast medical content library by providing more timely clinical results for users.
>
> [...] For healthcare professionals, Elsevier's clinical solutions include Interactive Patient Education and Care Planning. Elsevier's ClinicalPath (formerly Via Oncology) provides clinical pathways delivering personalised, evidence-based oncology guidance at the point of care. Elsevier's analytics capabilities in oncology support our ClinicalPath customers in answering increasingly complex questions around the delivery of cancer care, such as appropriate use of precision oncology and treatment adherence.

!! So not only do we risk distorting the practice of medicine, we could distort the entire trajectory of science. SciVal autoranks researchers and institutions based on how "hot" their research programs are, and helps suggest topics that are more likely to get a grant, etc. Since they also aggressively control what gets recommended, and have also recently started literally selling ads on their websites, they could easily create the same kind of informational bubbles that we are familiar with from social media. And with the combination of a biomedical knowledge graph contiguous with the pharmaceutical industry, they could steer all basic research — perhaps with us being only dimly aware — to support

the profit of their pharmaceutical partners. This isn't even speculative ! https://www.elsevier.com/solutions/biology-knowledge-graph

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.

Trans health example of potential harms

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 SmartAPI, 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 reinforcing 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[18] and its attempt to link everything — and seems ready to follow the same path of the fledgling technologies being gobbled up by technology giants to finish and privatize.

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

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 metascientists 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.

Federated Systems (of Language)

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[19] were proposed in the early 1980's [Heimbigner and McLeod, 1985] and have been developed and refined in the decades since as an alternative to either centralization or non-integration [Litwin et al., 1990, Kashyap and Sheth, 1996, Hull, 1997] . Their application to the dispersion of scientific data in local filesystems is not new [Busse et al., 1999, Djokic-Petrovic et al., 2017, Hasnain et al., 2017] , but their implementation is more challenging than imposing order with a centralized database or punting the question into the unknowable maw of machine learning.

[19] 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.

!! There is a lot of subtlety to the terminology surrounding "federated" and the typology of distributed systems generally, I am using it more in the federated messaging sense of forming groups of people, rather than the strict term federated databases which do imply a standardized schema across a federation. I am largely in line with the notion of distributed databases here [Hanke et al., 2021] .

Amit Sheth and James Larson, in their reference description of federated database systems, describe **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 **data** being managed (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).

Susanne Busse and colleagues add an additional dimension of **evolvability,** or the ability of a particular system to adapt to inevitable changing uses and requirements [Busse et al., 1999] .

In order to support such radical autonomy and evolvability, federated systems need some means of translating queries and representations between heterogeneous components. The typical conceptualization of federated databases have five layers that implement different parts of this reconciliation process [Sheth and Larson, 1990] :

- 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 [Bonifati et al., 2008] . In the strictest, "tightly coupled" federated systems, all heterogeneity in individual components has to be mapped to a single, unified federation-level schema. Loose federations don't assume a unified schema, but settle for a uniform query language, and allow multiple translations and views on data to coexist. A p2p system naturally lends itself to a looser federation, and also gives us some additional opportunities to give peers agency over schemas while also preserving some coherence across the system. I will likely make some database engineers cringe, but the emphasis for us will be more on building a system to support distributed social control over the database, rather than guaranteeing consistency and transparency between the different components.

Though there are hundreds of subtleties and choices in implementation beneath the level of detail I'll reach here, allow me to illustrate the system by example:

Let us start with the ability for a peer to choose who they are associated with 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[20]. Someone would then be able to use their `@name`space as a root, under which they could refer to their data, schemas, and so on, which will be denoted `@name:subobject` (see this notion of personal namespaces for knowledge organization discussed in early wiki culture here [Mea, c]). Let us also assume that there is no categorical difference between `@usernames` used by individual researchers, institutions,

[20] !! 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.

consortia, etc. — everyone is on the same level.

We pick up where we left off earlier with a peer who has 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:

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

Where I indicate that me, `@jonny` collected `a @nwb:NWBFile` 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).

!! pause to describe notion of using triplet links and the generality they afford us.

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 lmaoˆˆ 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. [Pirrò et al., 2012]). 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 synchronized copy of the messages sent on the servers and rooms it's federated with, and each server is capable of continuing communication if any of the others failed. We could use **ActivityPub** (AP) [Webber et al., 2018]

, 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 [Sporny et al., 2020] , so is already capable of representing linked data, and the related ActivityStreams vocabulary [Snell and Prodromou, 2017] 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.

For the moment our universe is limited only to other researchers using NWB. Conveniently, the folks at NWB have set up a federating group so that everyone who uses it can share their format extensions. Since our linking system for manipulating schemas is relatively general, we can use it to "formalize" a basic configuration for a federating group that automatically `Accept`s request to `Join` and allows any schema that inherits from their base `@nwb:NWBContainer` schema. Let's say `@fed` defines some basic properties of our federating system — it constitutes our federating "protocol" — and loosely use some terms from the ActivityStreams vocabulary as `@as`

```
<#nwbFederation>
  a @fed:Federation
  onReceive
    @as:Join @as:Accept
  allowSchema
    extensionOf @nwb:NWBContainer
```

Now anyone that is a part of the `@nwbFederation` would be able to see the schemas we have submitted, sort of like a beefed up, semantically-aware version of the existing neurodata extensions catalog. In this system, many overlapping schemas could exist simultaneously, but wouldn't become a hopeless clutter because similar schemas could be compared and reconciled based on their semantic properties.

So far we have been in the realm of metadata, but how would my computer know how to read and write the data to my disk so i can use it? In a system with heterogeneous data types and database implementations, we need some means of specifying different programs to use to read and write, different APIs, etc. Why not make that part of the file schema as well? Suppose the HDF5 group (or anyone, really!) has a namespace `@hdf` that defines the properties of an `@hdf:HDF5` file, basic operations like `Read`, `Write`, or `Select`. NWB could specify that in their definition of `@nwb:NWBFile`:

```
@nwb.NWBFile
```

```
  a @hdf:HDF5
    isVersion x.y.z
    hasDependency libhdf5==x.y.z
  usesContainer @nwb:NWBContainer
```

The abstraction around the file implementation makes it easier for others to consume my data, but it also makes it easier for *me* to use and contribute to the system. Making an extension to the schema wasn't some act of charity, it was the most direct way for me to use the tool to do what I wanted. Win-win: I get to use my fancy new instrument and store its data by extending some existing format standard, and in the process make the standard more complete and useful. We are able to make my work useful by *aligning the modalities of use and contribution.*

Now that I've got my schema extension written and submitted to the federation, time to submit my data! Since it's a p2p system, I don't need to manually upload it, but I do want to control who gets it. By default, I have all my NWB datasets set to be available to the `@nwbFederation` , and I list all my metadata on, say the Society for Neuroscience's `@sfnFederation`.

```
<#globalPermissions>
  a @fed:Permissions
  permissionsFor @jonny

  federatedWith
    name @nwbFederation
    @fed:shareData
      is @nwb:NWBFile

  federatedWith
    name @sfnFederation
    @fed:shareMetadata
```

Let's say this dataset in particular is a bit sensitive — say we apply a set of permission controls to be compliant with `@hhs.HIPAA` — but we do want to make use of some public server space run by our Institution, so we let it serve an encrypted copy that those I've shared it with can decrypt. Since we've applied the `@hhs.HIPAA` ruleset, we would be able to automatically detect if we have any conflicting permissions, but we're doing fine in this example.

```
<#datasetPermissions>
  a @fed:Permissions
  permissionsFor @jonny:cool-dataset
```

```
  accessRuleset @hhs:HIPAA
    .authorizedRecipient <#hash-of-patient-ids>

  federatedWith
    name @institutionalCloud
    @fed:shareEncrypted
```

Now I want to make use of some of my colleagues data. Say I am doing an experiment with a transgenic dragonfly and collaborating with a chemist down the hall. This transgene, known colloquially in our discipline as `"@neuro:superstar6"` (oh-so-uncreatively ripped off by the chemists as `"@chem:SUPER6"`) fluoresces when the dragonfly is feeling bashful, and we have plenty of photometry data stored as `@nwb:Fluorescence` objects. We think that its fluorescence is caused by the temperature-dependent conformational change from blushing. They've gathered NMR and Emission spectroscopy data in their chemistry-specific format, say `@acs:NMR` and `@acs:Spectroscopy`.

We get tired of having our data separated and needing to maintain a bunch of pesky scripts and folders, so we decide to make a bridge between our datasets. We need to indicate that our different names for the gene are actually the same thing and relate the spectroscopy data.

Let's make the link explicit, say we use `@skos`?

```
<#super-link-6>
  a @fed:Link

  from @neuro:superstar6
  to @chem:SUPER6
  link @skos:exactMatch
```

Our `@nwb:Fluorescence` data has the emission wavelength in its `@nwb:Fluorescence:excitation_lambda` property[21], which is the value of their `@acs:Spectroscopy` data at a particular value of its `wavelength`. Unfortunately, `wavelength` isn't metadata for our friend, but a column in the `@acs:Spectroscopy:readings` table, so for now the best we can do is indicate that `excitation_lambda` is one of the values in `wavelength` and pick it up in our analysis tools.

[21] not really where it would be in the standard, but go with it plz

```
<#imaging>
 a @fed:Link

 from @nwb:Fluorescence:excitation_lambda
 to @acs:Spectroscopy:readings
 link @fed:Subset
   valueIn "wavelength"
```

This makes it much easier for us to index our data against each other and solves a few real practical problems we were facing in our collaboration. We don't need to do as much cleaning when it's time to publish the data since it can be released as a single linked entity.

Rinse and repeat our sharing and federating process from our previous schema extension, add a little bit of extra federation with the `@acs` namespace, and in the normal course of our doing our research we've contributed to the graph structure linking two common data formats. Ours is one of many, with ugly little names like `@jonny:super-link-6`[22]. We might not have followed the exact rules, and we only made a few links rather than a single authoratative mapping, but if someone is interested in compiling one down the line they'll start off a hell of a lot further than if we hadn't contributed it!

[22] we'll return to credit assignment, don't worry! I wouldn't leave a friend out to dry.

With a protocol for how queries can be forwarded and transformed between users and federations, one could access the same kind of complex query structure as traditional databases with SPARQL [SPA, 2013] as has been proposed for biology many times before [Sima et al., 2019, Djokic-Petrovic et al., 2017, Hasnain et al., 2017] . Some division in the way that data and metadata are handled is necessary for the network to work in practice, since we can't expect a search to require terabytes of data transfer. A natural solution to this is to have metadata query results point to content addressed identifiers that are served peer to peer. A mutable/changeable/human-readable name and metadata system that points to a system of permanent, unique identifiers has been one need that has hobbled IPFS, and is the direction pointed to by DataLad [Hanke et al., 2021] .
A parallel set of conversations has been happening in the broader linked data community with regard to using ActivityPub as a way to index data on Solid.

In this example I have been implicitly treating the `@nwbFederation` users like bittorrent trackers, keeping track of different datasets in their federation, but there is no reason why queries couldn't themselves be distributed across the participating peers, though I believe tracker-like federations are useful and might emerge naturally. A system like this doesn't need the radical zero trust design of, for example, some distributed ledgers, and an overlapping array of institutional, disciplinary, interest, and so on federations would be a good means of realizing the evolvable community structure needed for sustained archives.

Extend this practice across the many overlapping gradients of cooperation and collaboration in science, and on a larger scale a system like this could serve as a way to concretize and elevate the organic, continual negotiation over meaning and practice that centralized ontologies can only capture as a snapshot. It doesn't have the same

guarantees of consistency or support for algorithmic reasoning as a top-down system would in theory, but it would give us agency over the structure of our information and have the potential to be useful for a far broader base of researchers.

I have no idea where the physicists' store their data or what format it's in, *but the chemists might,* and the best way to get there from here might be a dense, multiplicative web of actual practical knowledge instead of some sparsely used corporate API.

I have been purposefully nonprescriptive about implementation and fine details here, what have we described so far? !! short summary of preceding section !! recall that what i am describing is protocol-like, so having multiple implementations that evolve is sorta the point.

Like the preceding description of the basic peer-to-peer system, this joint metadata/p2p system could be fully compatible with existing systems. Translating between a metadata query and a means of accessing it on heterogeneous databases is a requisite part of the system, so, for example, there's no reason that an HTTP-based API like SmartAPI couldn't be queried.

DataLad [Halchenko et al., 2021, Hanke et al., 2021] and its application in Neuroscience as DANDI are two projects that are *very close* to what I have been describing here — developing a p2p backend for datalad and derivation into a protocol might even be a promising development path towards it.

!! close this section by taking a larger view - [Langille and Eisen, 2010] 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.

!! note that this is all fully compatible with existing systems and is a superset of centralized servers with centralized schemas!

Shared Tools

Straddling our system for sharing data are the tools to gather and analyze it. Experimental and analytical tools are the natural point of extension for collectively developed scientific digital infrastructure, and considering them together shows the combinatoric power of integrating interoperable domains of scientific practice. In particular, in addition to benefits from their development in isolation, we can ask how a more broadly integrated system helps problems like adoption and incentives for distributed work, enables a kind of deep

provenance from experiment to results, and lets us reimagine the form of the community and communication tools for science.

This section will be relatively short compared to shared data. We have already introduced, motivated, and exemplified many of the design practices of the broader infrastructural system. There is much less to argue against or "undo" in the spaces of analytical and experimental tools because so much more work has been done, and so much more power has been accrued in the domain of data systems. Distributed computing does have a dense history, with huge numbers of people working on the problem, but its hegemonic form is much closer to the system articulated below than centralized servers are to federated semantic p2p systems. I also have written extensively about experimental frameworks before [Saunders and Wehr, 2019] , and develop one of them so I will be brief at risk of repeating myself or appearing self-serving.

!! both these sections are also relatively unstandardized, so before jumping to some protocol just yet, we can build frameworks that start congealing the pieces en route to one.

Integrated scientific workflows have been written about many times before, typically in the context of the "open science" movement. One of the founders of the Center for Open Science, Jeffrey Spies, described a similar ethic of toolbuilding as I have in a 2017 presentation:

> 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 [Spies, 2017b]

To build an infrastructural system that enables "open" practices, *convincing* or *mandating* a change are much less likely to be successful and sustainable than focusing on building them to make doing work easier and openness automatic. To make this possible, we should focus on developing *frameworks to build* experimental and analysis tools, rather than developing more tools themselves.

Analytical Frameworks

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 and plot it. As a result it is often difficult to repurpose code for other contexts, so the same analysis function is rewritten in each lab's local analysis repository. Since sharing raw data and code is still a (difficult) novelty, on a broad scale this makes results in scientific literature as reliable as we imagine all the private or semi-private analysis code to be.

Analytical tools (anecdotally) make up the bulk of open source scientific software, and range from foundational and general-purpose tools like numpy [Harris et al., 2020] and scipy [Virtanen et al., 2020] , through tools that implement a class of analysis like DeepLabCut [Mathis et al., 2018] and scikit-learn [Pedregosa et al., 2011] , to tools for a specific technique like MoSeq [Wiltschko et al., 2020] and DeepSqueak [Coffey et al., 2019] . The pattern of their use is then to build them into a custom analysis system that can then in turn range in sophistication from a handful of flash-drive-versioned scripts to automated pipelines.

Having tools like these of course puts researchers miles ahead of where they would be without them, and the developers of the mentioned tools have put in a tremendous amount of work to build sensible interfaces and make them easier to use. No matter how much good work might be done, inevitable differences between APIs is a relatively sizable technical challenge for researchers — a problem compounded by the incentives for fragmentation described previously. For toolbuilders, many parts of any given tool from architecture to interface have to be redesigned with varying degrees of success each time. For science at large, with few exceptions of well-annotated and packaged code, most results are only replicable with great effort.

To be clear, we have reached levels of "not the developer's fault" to the tune of "API discontinuity" being *"the norm for 99% of software."* Negotiating boundaries between (and even within) software and information structures is an elemental part of computing. The only time it becomes a conceivable problem to "solve" is when the problem domain coalesces to the point where it is possible to articulate its abstract structure as a protocol, and the incentives are great enough to adopt it. Thankfully that's what we're trying to do here.

It's unlikely that we will solve the problem of data analysis being complicated, time consuming, and error prone by teaching every scientist to be a good programmer, but we can build experimental frameworks that make analysis tools easier to build and use.

Specifically, 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 framework doesn't need to make normative decisions like that. The system should support modularity by providing a clear set of hooks that tools can provide: eg. a clear place for a given tool to accept some input, parameters, and so on. Since data analysis can often be broken up into a series of relatively independent stages, a straightforward (and common) system for modularity is to build hooks to make a directed acyclic graph (DAG) of data transformation operations. This structure naturally lends itself to many common problems: caching intermediate results, splitting and joining multiple inputs and outputs, distributing computation over many machines, among others. Modularity is also needed within the different parts of the system itself – eg. running an analysis chain shouldn't require a GUI, but one should be available, etc.

- **Pluggable** - The framework needs to provide a clear way of incorporating external analysis packages, handling their dependencies, and exposing their parameters to the user. Development should ideally not be limited to a single body of code with a single mode of governance, but should instead be relatively conservative about requirements for integrating code, and liberal with the types of functionality that can be modified with a plugin. Supporting plugins means supporting people developing tools for the framework, so it needs to make some part of the toolbuilding process easier or otherwise empower them relative to an independent package. This includes building a visible and expressive system for submitting and indexing plugins so they can be discovered and credit can be given to the developers. Reciprocal to supporting plugins is being interoperable with existing and future systems, which the reader may have assumed was a given by now.

- **Deployable** - For wide use, the framework needs to be easy to install and deploy locally and on computing clusters. A primary obstacle is dependency management, or making sure that the computer has everything needed to run the program. Some care needs to be taken here, as there are multiple emphases in deployability that can be in conflict. Deployable for who? A system that can be relatively challenging to use for routine exploratory data analysis but can distribute analysis across 10,000 GPUs has a very circumscribed set of people it is useful for. This is a matter of balancing design constraints, but we should prioritize broad access, minimal assumptions of technological access, and ease of use over being

able to perform the most computationally demanding analyses possible when in conflict. Containerization is a common, and the most likely strategy here, but the interface to containers may need a lot of care to make accessible compared to opening a fresh .py file.

- **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. The parameterization of a pipeline or analysis DAG 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. The isolation of parameters from implementation is complementary to the separation of metadata from data and if implemented with semantic triplets would facilitate a continuous interface from our data to analysis system. This will be explored further below and in shared knowledge

Thankfully a number of existing projects that are very similar to this description are actively being built. One example is DataJoint [Yatsenko et al., 2018] , which recently expanded its facility for modularity with its recent Elements project [Yatsenko et al., 2021] . Datajoint is a system for creating analysis pipelines built from a graph of processing stages (among other features). It is designed around a refinement on traditional relational data models, which is reflected throughout the system as most operations being expressed in its particular schema, data manipulation, and query languages. This is useful for operations that are expressed in the system, but makes it harder to integrate external tools with their dependencies — at the moment it appears that spike sorting (with Kilosort [Pachitariu et al., 2016]) has to happen outside of the extracellular electrophysiology elements pipeline.

Kilosort is an excellent and incredibly useful tool, but its idiomatic architecture designed for standalone use is illustrative of the challenge of making a general-purpose analytic framework that can integrate a broad array of existing tools. It is built in MATLAB, which requires a paid license, making arbitrary deployment difficult, and MATLAB's flat path system requires careful and usual manual orchestration of potentially conflicting names in different packages. Its parameterization and use are combined in a "main" script in the repository root that creates a MATLAB struct and runs a series of functions — requiring some means for a wrapping framework to translate between input parameters and the representation expected by the tool. Its preprocessing script combines I/O, preprocessing, and plotting, and requires data to be loaded from disk rather than

passed as arguments to preserve memory — making chaining in a pipeline difficult.

This is not a criticism of Datajoint or Kilosort, which were both designed for different uses and with different philosophies (that are of course, also valid). I mean this as a brief illustration of the design challenges and tradeoffs of these systems.

We can start getting a better picture for the way a decentralized analysis framework might work by considering the separation between the metadata and code modules, hinting at a protocol as in the federated systems sketh above. Since we're considering modular analysis elements, each module would need some elemental properties like the parameters that define it, its inputs, outputs, dependencies, as well as some additional metadata about its implementation (eg. this one takes *numpy arrays* and this one takes *matlab structs*). The precise implementation of a modular protocol also depends on the graph structure of the analysis system. We invoked DAGs before, but analysis graph structure of course has its own body of researchers refining them into eg. Petri nets which are graphs whose nodes necessarily alternate between "places" (eg. intermediate data) and "transitions" (eg. an analysis operation), and their related workflow markup languages (eg. WDL or [van der Aalst and ter Hofstede, 2005]). In that scheme, a framework could provide tools for converting data between types, caching intermediate data, etc. between analysis steps, as an example of how different graph structures might influence its implementation.

Say we use `@analysis` as the namespace for our analysis protocol, and `~someone~` has provided mappings to objects in `numpy`. We can assume they are provided by the package maintainers, but that's not necessary: this is my node and it takes what I want it to!

In pseudocode, I could define some analysis node for, say, converting an RGB image to grayscale under my namespace as `@jonny:bin-spikes` like this:

```
<#bin-spikes>
  a @analysis:node
    Version >=1.0.0

  hasDescription
    "Convert an RGB Image to a grayscale image"

  inputType
    @numpy:ndarray
      # ... some spec of shape ...
```

```
outputType
  @numpy:ndarray
    # ... some spec of shape ...
```

I have abbreviated the specification of shape to not overcomplicate the pseudocode example, but say we successfully specify a 3 dimensional (width x height x channels) array with 3 channels as input, and a a 2 dimensional (width x height) array as output.

The code doesn't run on nothing! We need to specify our node's dependencies, say in this case we need to specify an operating system image `ubuntu`, a version of `python`, a system-level package `opencv`, and a few python packages on `pip`. We are pinning specific versions with semantic versioning, but the syntax isn't terribly important. Then we just need to specify where the code for the node itself comes from:

```
dependsOn
  @ubuntu:^20.*:x64
  @python:3.8
  @apt:opencv:^4.*.*
  @pip:opencv-python:^4.*.*
  @pip:numpy:^14.*.*

providedBy
  @git:repository https://mygitserver.com/binspikes/fast-binspikes.git
    @git:hash fj9wbkl
  @python:class /main-module/binspikes.py:Bin_Spikes
```

Here we can see the advantage of being able to mix and match different namespaces in a practical sense. Our `@analysis.node` protocol gives us several slots to connect different tools together, each in turn presumably provides some minimal functionality expected by that slot: eg. `inputType` can expect `@numpy:ndarray` to specify its own dependencies, the programming language it is written in, shape, data type, and so on. Coercing data between chained nodes then becomes a matter of mapping between the `@numpy` and, say a `@nwb` namespace of another format. In the same way that there can be multiple, potentially overlapping between data schemas, it would then be possible for people to implement mappings between intermediate data formats as-needed.

This node also becomes available to extend, say someone wanted to add an additional input format to my node:

```
<@friend#bin-spikes>
  a @jonny:bin-spikes
```

```
  inputType
    @pandas:DataFrame

  providedBy
    ...
```

They don't have to interact with my potentially messy codebase at all, but it is automatically linked to my work so I am credited. One could imagine a particular analysis framework implementation that would then search through extensions of a particular node for a version that supports the input/output combinations appropriate for their analysis pipeline, so the work is cumulative. This functions as a dramatic decrease in the size of a unit of work that can be shared.

This also gives us healthy abstraction over implementation. Since the functionality is provided by different, mutable namespaces, we're not locked into any particular piece of software — even our `@analysis` namespace that gives the `inputType` etc. slots could be forked. We could implement the dependency resolution system as, eg. a docker container, but it also could be just a check on the local environment if someone is just looking to run a small analysis on their laptop with those packages already installed.

We use providedBy to indicate a python class which implements the node in code. We could use an `Example_Framework` that provides a set of classes and methods to implement the different parts of the node (a la luigi). Our `Bin` class inherits from `Node`, and we implement the logic of the function by overriding its `run` method and specify an output file to store intermediate data (if requested by the pipeline) with an `output` method. We also specify a `bin_width` as a `Parameter` for our node, as an example of how a lightweight protocol could be bidirectionally specified: we could receive a parameterization from our pseudocode specification, or we could write a framework with a `Bin.export_schema()` that constructs the pseudocode specification from code.

```
from Example_Framework import Node, Param, Target

class Bin(Node):
  bin_width = Param(dtype=int, default=10)

  def output(self) -> Target:
    return Target('temporary_data.pck')

  def run(self, input:'numpy.ndarray') -> 'numpy.ndarray':
    # do some stuff
    return output
```

Now that we have a handful of processing nodes, we could then describe some `@workflow`, taking some `@nwb:NWBFile` as input, and then returning some output as a `:processed` child beneath its existing namespace. We'll only make a linear pipeline with two stages, but there's no reason more complex branching and merging couldn't be described as well.

```
<#my-analysis>
  a @analysis:workflow

  inputType
    @jonny:bin-spikes:inputType

  outputName
    .inputType:processed

  step Step1 @jonny:bin-spikes
  step Step2 @someone-else:another-step
    input Step1:output
```

Having kept the description of our data in particular abstract from the implementation of the code and the workflow specification, the only thing left is to apply it to our data! Since the parameters are linked from the analysis nodes, we can specify them here (or in the workflow). Assuming literally zero abstraction and using the tried-and-true "hardcoded dataset list" pattern, something like:

```
<#my-project>
  a @analysis:project

  hasDescription
    "I gathered some data, and it is great!"

  researchTopic
    @neuro:systems:auditory:speech-processing
    @linguistics:phonetics:perception:auditory-only

  inPaper
    @doi:10.1121:1.5091776

  workflow Analysis1 @jonny:my-analysis
    globalParams
      .Step1:params:bin_width 10

    datasets
```

```
@jonny.mydata1:v0.1.0:raw
@jonny.mydata2:^0.2.*:raw
@jonny.mydata3:>=0.1.1:raw
```

And there we are! The missing parameters like `outputName` from our workflow can be filled in from the defaults filled in the workflow node. We get some inkling of where we're going later by also being able to specify the paper this data is associated with, as well as some broad categories of research topics so that our data as well as the results of the analysis can be found.

!! brief description of the state of the system at this point, we can link from data to analyses! reapply analyses across different datasets! and so on...

So that's useful, but the faint residue of "well actually" that hangs in the air while people google the link for that xkcd comic about format expansion is not lost on me. The important part is in the way this hypothetical analysis framework and markup interact with our data system and emerging federated metadata system — The layers of abstraction here are worth unpacking, but we'll hold until the end of the shared tools section and we have a chance to consider what this system might look like for experimental tools.

Experimental Frameworks

Data that is to be analyzed has to be collected somehow. Tools to bridge the body of experimental practice are a different challenge than analyzing data, or at least so an anecdotal census of scientific software tools would suggest. *Everyone needs completely different things!* As practiced, we might imagine the practice of science as a cone of complexity: We can imagine the relatively few statistical outcomes from a family of tests and models. For every test statistic we can imagine a thousand analysis scripts, for every analysis script we might expect a hundred thousand data formats, and so the french-horn-bell convexity of complexity of experimental tools used to collect the data feels ... different.

Beyond a narrow focus of the software for performing experiments itself, the contextual knowledge work that surrounds it largely lacks a means of communication and organization. Scientific papers have increasingly marginalized methods sections, being pushed to the bottom, abbreviated, and relegated to supplemental material. The large body of work that is not immediately germane to experimental results, like animal care, engineering instruments, lab management, etc. have effectively no formal means of communication — and so little formal means of credit assignment.

Extending our ecosystem to include experimental tools has a few

immediate benefits: bridging the gap between collection and sharing data would resolve the need for format conversion as a prerequisite for inclusion in the linked system, allowing the expression of data to be a fluid part of the experiment itself; and serving as a concrete means of implementing and building a body of cumulative contextual knowledge in a creditable system.

I have previously written about the design of a generalizable, distributed experimental framework in section 2, and about one modular implementation in section 3 of [Saunders and Wehr, 2019] , so to avoid repeating myself, and since many of the ideas from the section on analysis tools apply here as well, I will be relatively brief.

We don't have the luxury of a natural formalism like a DAG to structure our experimental tools. Some design constraints on experimental frameworks might help explain why:

- They need to support a wide variety of instrumentation, from **off-the-shelf parts,** to **proprietary instruments** as are common in eg. microscopy, to **custom, idiosyncratic designs** that might make up the existing infrastructure in a lab.

- To be supportive, rather than constraining, they need to be able to **flexibly perform many kinds of experiments** in a way that is **familiar to patterns of existing practice.** That effectively means being able to coordinate heterogeneous instruments in some "task" with a flexible syntax.

- They need to be **inexpensive to implement,** in terms of both money and labor, so it can't require buying a whole new set of hardware or dramatically restructuring existing research practices.

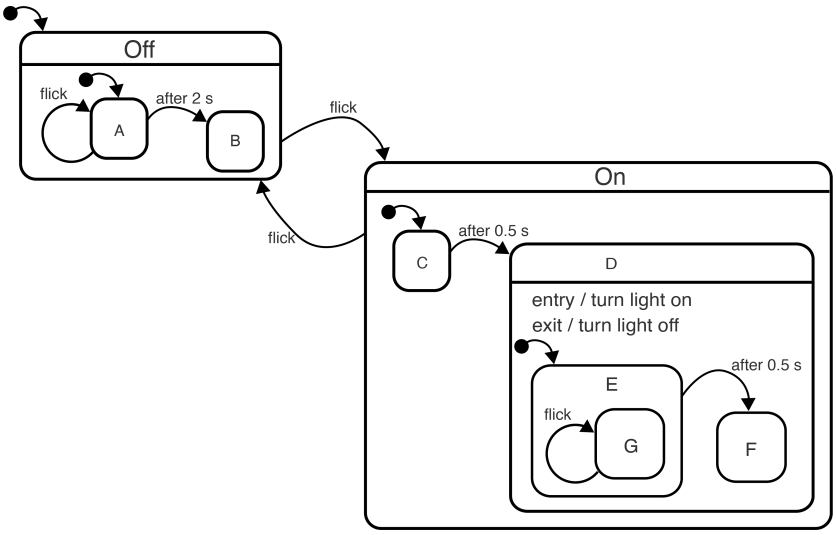
- They need to be **accessible and extensible,** with many different points of control with different expectations of expertise and commitment to the framework. It needs to be useful for someone who doesn't want to learn it to its depths, but also have a comprehensible codebase at multiple scales so that reasearchers can **easily extend** it when needed.

- They need to be designed to support **reproducibility and provenance,** which is a significant challenge given the heterogeneity inherent in the system. On one hand, being able to produce *data that is clean at the time of acquisition* simplifies automated provenance, but enabling experimental replication requires multiple layers of abstraction to keep the idiosyncracies of an experiment separable from its implementation: it shouldn't require building *exactly* the same apparatus with *exactly* the same parts connected in *exactly* the same way to replicate an experiment.

- Ideally, they need to support **cumulative labor and knowledge organization,** so an additional concern with designing abstractions between system components is allowing work to be made portable and combinable with others. The barriers to contribution should be extremely minimal, not requiring someone to be a professional programmer to make a pull request to a central library, and contributions should come in many modes — code is not the only form of knowing and it's far from the only thing needed to perform an experiment.

Here, as in the domains of data and analysis, the temptation to be universalizing is strong, and the parts of the problem that are emphasized influence the tools that are produced. A common design tactic for experimental tools is to design them as state machines, a system of states and transitions not unlike the analysis DAGs above. One such nascent project is BEADL [Wulf, 2020] from a Neurodata Without Borders working group. BEADL is an XML-based markup for standardizing a behavioral task as an abstraction of finite state machines called statecharts. Experiments are fully abstract from their hardware implementation, and can be formally validated in simulations. The working group also describes creating a standardized ontology and metadata schema for declaring all the many variable parameters for experiments, like reward sizes, stimuli, and responses [WG, 2020] . This group, largely composed of members from the Neurodata Without Borders team, understandably emphasize systematic description and uniform metadata as a primary design principle.

Personally, I *like* statecharts. The problem is that it's not necessarily natural to express things as statecharts as you would want to, or in the way that your existing, long-developed local experimental code does. There are only a few syntactical features needed to understand the following statechart: blocks are states, they can be inside each other. Arrows move between blocks depending on some condition. Entering and exiting blocks can make things happen. Short little arrows from filled spots are where you start in a block, and when you get to the end of the chart you go back to the first one. See the following example of a statechart for controlling a light, described in the introductory documentation and summarized in the figure caption:



"When you flick a lightswitch, wait 0.5 seconds before turning the light on, then once it's on wait 0.5 seconds before being able to turn it back off again. When you flick it off, wait 2 seconds before you can turn it on again.

They have an extensive set of documents that defend the consistency and readability of statecharts on their homepage, and my point here is not to disagree with them. My point is instead that tools that aspire to the status of generalized infrastructure can't ask people to dramatically change the way they think about and do science. There are many possible realizations of this task, and each is more or less natural to every person.

The problem here is really one of emphasis, BEADL seeks to solve problems with inconsistencies in terminology by standardizing them, and in order to do that seeks to standardize the syntax for specifying experiments.

This means of standardization has many attractive qualities and is being led by very capable researchers, but I think the project is illustrative of how the differing constraints of different systems and differing goals of different approaches influence the possible space of tooling. Analysis tasks are often asynchronous, where the precise timing of each node's completion is less important than the path dependencies between different nodes be clearly specified. Analysis tasks often have a clearly defined set of start, end, and intermediate cache points, rather than branching or cyclical decision paths that change over multiple timescales. Statecharts are a hierarchical abstraction of finite state machines, the primary advantage of which is that they are better able to incorporate continuous and history-dependent behavior, which causes state explosion in traditional finite-state machines.

Autopilot [Saunders and Wehr, 2019] approaches the problem

differently by avoiding standardizing *experiments* themselves, instead providing smaller building blocks of experimental tools like hardware drivers, data transformations, etc. and emphasizing understanding their use in *context*. This approach sacrifices some of the qualities of a standardized system like being a logically complete or having guaranteed interoperability of terms in order to better support integrating with existing work patterns and making work cumulative. It is a bit more humble: because we can't possibly predict the needs and limitations of a totalizing system, we split the problem along the different domains of tools and give facility for describing how they are used together.

For concrete example, we might imagine the lightswitch in an autopilot-like framework like this:

```
from autopilot.hardware.gpio import Digital_Out
from time import sleep
from threading import Lock

class Lightswitch(Digital_Out):
  def __init__(self,
    off_debounce: float = 2,
    on_delay:     float = 0.5,
    on_debounce:  float = 0.5):
    """
    Args:
      off_debounce (float):
        Time (s) before light can be turned back on
      on_delay (float):
        Time (s) before light is turned on
      on_debounce (float):
        Time (s) after turning on that light can't be turned off
    """
    self.off_debounce = off_debounce
    self.on_delay     = on_delay
    self.on_debounce  = on_debounce

    self.on = False
    self.lock = Lock()

  def switch(self):
    # use a lock to make sure if
    # called while waiting, we ignore it
    if not self.lock.acquire():
      return
```

```
    # if already on, switch off
    if self.on:
      self.on = False
      sleep(self.off_debounce)

    # otherwise switch on
    else:
      sleep(self.on_delay)
      self.on = True
      sleep(self.on_debounce)

    self.lock.release()
```

The terms `off_debounce`, `on_delay`, and `on_debounce` are certainly not part of a controlled ontology, but we have described how they are used in the docstring and how they are used is inspectable in the class itself.

The difficulty of a controlled ontology for experimental frameworks is perhaps better illustrated by considering a full experiment. In Autopilot, a full experiment can be parameterized by the `.json` files that define the task itself and the system-specific configuration of the hardware. An example task from our lab consists of 7 behavioral shaping stages that introduce the animal to different features of 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. So just for one relatively straightforward experiment, in one lab, in one subdiscipline, there are **268 parameters** – excluding all the default parameters encoded in the software.

The way Autopilot handles various parameters are part of set of layers of abstraction that separate idiosyncratic logic from the generic form of a particular `Task` or `Hardware` class. The general structure of a two-alternative forced choice task is shared across a number of experiments, but they may have different stimuli, different hardware, and so on. Autopilot `Task`s use abstract references to classes of hardware components that are required to run them, but separates their implementation as a system-specific configuration so that it's not necessary to have *exactly the same* components plugged into *exactly the same* GPIO pins, etc. Task parameters like stimuli, reward timings, etc. are similarly split into a separate task parameterization that both allow `Task`s to be generic and make provenance and experimental history easier to track. `Task` classes can be subclasses to add or modify logic while being able to reuse much of the structure and maintain

the link between the root task and its derivatives — for example one task we use that starts a continuous background sound but otherwise is the same as the root `Nafc` class. The result of these points of abstraction is to allow exact experimental replication on inexactly replicated experimental apparatuses.

In contrast, workflows in Bonsai [Lopes et al., 2015b, Lopes and Monteiro, 2021] , another very popular and very capable experimental tool, combine the pattern of nodes that constitute an experiment with idiosyncratic parameters like a crop bounding box. To be clear, I love Bonsai, and this kind of workflow reproducibility is a huge step up from the more common practice of totally lab-specific code. The flat design of Bonsai is extremely useful for prototyping and extends through to complex experiments, but would have a hard time being able to support generalizable and reusable software classes for basic experimental operations, as well as creation and negotiation over experimental terminology.

We can imagine extending the abstract specification of experimental parameters, hardware requirements, and so on to work with our federated naming system to overcome the challenges to standardizing. First, we can imagine being able to make explicit declarations about the relationship between our potentially very local terminology. Here we can declare our `Lightswitch` object and 1) link its `on_delay` to our friend `@rumbly`'s object that implements the same thing as `on_latency`, and 2) link it to a standardized `Latency` term from interlex, but since that term is for time elapsed between a stimulus and behavioral response in a psychophysical context, it's only a partial match.

```
<#Lightswitch>
  a @autopilot.hardware.Digital_Out

  param on_delay
    @skos:exactMatch @rumbly:LED:on_latency
    @skos:nearMatch @interlex:Latency

  providedBy
    @git:repository ...
    @python:class ...
```

Further, since our experimental frameworks are intended to handle off the shelf parts as well as our potentially idiosyncratic lightbulb class, we can link many implementations of a hardware controlling class to the product itself. Take for example the I2C_9DOF class that controls a 9 degree of freedom motion sensor from Sparkfun where we both indicate the specific part itself as well as the generic `ic` that it

uses:

```
<#I2C_9DOF>
  @autopilot.controlsHardware
    @sparkfun:13944
    @ic:LSM9DS1
```

This hints at the first steps of a system that would make our technical work more cumulative, as it is then easy to imagine being able to search for all the different implementations for a given piece of hardware. Since the `@sparkfun:13944` element can in turn specify properties like being an inertial motion sensor, this kind of linking becomes powerful very quickly to make bridges that allow similar work to be discovered and redeployed quickly.

We can also extend our previous connection between a dataset and the results of its analysis to also include the tools that were used to collect it. Say we want to declare the example experiment above, and then extend our `<#project-name>` project to, also above, to reference it:

```
<#example-experiment>
  a @autopilot:protocol

  level @autopilot:freeWater
    reward
      type mL
      value 5
    graduation
      a @autopilot:graduation:ntrials
      n_trials 200

  level @autopilot:Nafc
    stim
      @autopilot:stim:sound:Tone
        frequency 5000
        duration 100

  ...

  @autopilot:prefs
    @jonny:Lightswitch
      on_delay 1

<#project-name>
  a @jonny:project-name
  collectedBy @jonny:example-experiment
```

So while we sacrifice the direct declaration of standardized terminology and syntax, we gain the potential for a much denser and richer expressive structure for our experiments. Instead of a single authoritative dictionarylike meaning for a term, we instead appreciate it in the context of its use, linked to the code that implements it as well as the data it produces and the kinds of arguments that are made with different analysis chains. Of course there is no intrinsic conflict with this kind of freewheeling system and controlled vocabularies and syntaxes: in this system, they can be one of many means of expression rather than need to be singular sources of truth that depend on wide adoption. While individual instances of uncontrolled vocabularies might mean chaos, when they are integrated in a system of practice we get something much wilder but also more intricate, beautiful, and useful.

As in the case of analytical tools, the role of the experimental frameworks is also to make interacting with the rest of the system easier and doesn't involve manually editing a lot of metadata. For example, currently autopilot `Tasks` ask users to declare collected data as a pytables [Alted and Fernández-Alonso, 2003] datatypes like `target = tables.StringCol(1)` to record whether a target is `'L'` or `'R'`. If instead it was capable of specifying a Neurodata Without Borders data type like `target = '@nwb:behavior:BehavioralEvents'`, then it would be possible to directly output to a standardized format, potentially also automatically creating a `BehavioralEpochs` container or other data that are implied but otherwise have to be explicitly created. Autopilot already automatically tracks the entire behavioral history of an experimental subject, so we can also imagine it being able to automatically create a `@analysis:project` object described above that groups together multiple datasets that connected them to an analysis pathway. So in this example the elusive workflow where experimental data is automatically scooped up and incrementally analyzed that is typically a hard-won engineering battle within a single lab would become the normal mode of using the system.

The experimental framework described so far could solve some of the software challenges of doing experiments by providing a system for extending a set of reusable classes that can be combined into experiments and linked together, but we haven't described anything to address the rest of the contextual knowledge of practical scientific work. We also haven't described any sort of governance or development system that makes these packages anything more than "some repository on GitHub somewhere" with all the propensity to calcify into fiefdoms that those entail. This leads us back to a system of communication, the central piece of missingness that we have been circling around the whole piece. If you'll allow me one more delay, I

want to summarize the system so far before finally arriving there.

Abstraction & Protocol Design

This section should be split back up s.t. the parts specific to analysis/experimental tools are at the ends of those sections, and we should move the discussion about layers of abstraction congealing into a protocol in the end in the practical implementation section. I'm leaving this here until I have time to do that, but for now you probably want to skip to the next section :)

Though there are many similarities between the three domains of data, analytical, and experimental tools, the different constraints each impose on a generalizable framework for integration and interoperability are instructive. Each requires a careful consideration of the *layers of abstraction* needed to maintain the modularity of the system — this is an elemental feature of any protocol design. What are the minimal affordances needed to implement a wide array of systems and technologies within each domain? By being careful with specifying abstraction, when considered together, the linked system described so far represents a powerful step towards *collectivizing the scientific state of the art.*

There are three primary layers of abstraction in the analysis system described: the interface between the metadata description of a node and the code that implements it, the separation of individual nodes and a notion of a combined workflow, and perhaps more subtly the separation of the data applied to the workflow and the workflow itself.

!! while the analysis system seeks to make multiple software packages and environments be interoperable together, the experimental framework makes no such attempt. !! the need for careful timing and adaptation to individual systems leaves integration for the implementing codebases.

- First, the markup description of the node gives us abstraction from programming language and implementation. This lets us do stuff like use multiple tools with competing environmental needs, adapt to multiple versions of the code markup as it develops, etc. Note the interaction with the rest of the metadata system: because we required a particular type of data file, and that link should provide us some means of opening/instantiating the file with dependencies, we didn't need to write loading code. Since it's in a linked system, someone could override the implementation of my node – say someone comes up with a faster means of binning, then they just inherit from my node and replace the reference to the code. Boom we have cumulative and linked development.

- The separation of the node from the workflow means that the node can be shared and swapped and reintegrated easily, dramatically reducing the brittleness of the systme. Since there is no restriction on what constitutes a node, though, there's no reason that nodes can't be either made massive, like putting a whole library in the process method, or be packaged up together. If we made the argument and method names recursive between the workflow and the node objects then tooling could automatically traverse multiple layers of node/workflow combinations at different levels of abstraction. This being a schematic description means that there can be multiple "workflow runner" packages that eg. distribute the task across a billion supercomputers or not.
- Finally, the separation between the data applied and the workflow itself are very cool indeed given our linked and namespaced system. My workflow effectively constitutes "an unit of analysis." I have linked my data to this unit of analysis. Play out the permutations:
  - I can see all the analyses that this particular pipeline has been applied to. Since it is embedded within the same federated system as our schema system, I can draw and connect semantic links to similar analysis pipelines as well as pipeline/data combinations.
  - I can see all the different analyses that have been applied to my data: if my data is analyzed a zillion different times, in a zillion different combinations of data, I effectively get a "multiverse analysis" (cite dani) and we get to measure robustness of my data for free. It also gets to live forever and keep contributing to problems !! and i also get credited for it automatically by golly! This also applies on cases like cross-validation or evaluating different models on the same data: the versioning of it falls out naturally. Also since model weights would be an input to an analysis chain, we also get stuff like DLC's model zoo where we can share different model weights, combine them, and have a cumulative library of pretrained models as well!
  - being able to look across the landscape... we start being able to actually really make cumulative progress on best practices. 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

[Miller, 2006, Soergel, 2015, Eklund et al., 2016, Bhandari Neupane et al., 2019] , 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.

– looking forward, we might imagine our project object being linked to a DOI… we'll get there.

!! this is all extraordinarily reproducible because even though I have my portable markup description of the analysis, I can just refer to it by name in my paper (ya ya need some content based hash or archive but you get the idea)

!! since we have a bunch of p2p systems all hooked up with constantly-running daemons, to compete with the compute side of cloud technology we also should implement a voluntary compute grid akin to Folding@Home. This has the same strawmen and answers to them as the peer-to-peer system — no i'm not saying everyone puts their shitty GPU up, but it lets us combine the resources that are present at an institutional level and makes a very cheap onramp for government-level systems to be added to the mix.

!! this is all very exciting, and we can immediately start digging towards larger scientific problems, eg. what it would mean for the file drawer problem and publication bias when the barriers to analyzing data are so low you don't even need to write the null result: the data is already there, semantically annotated and all. Dreams of infinite meta-analyses across all data and all time, but hold your horses! We don't get magic for free, we haven't talked about the community systems yet that are the unspoken glue of all of this!!

The category distinction between experimental and analytical tools is, of course, a convenient ordering fiction for the purpose of this piece. Autopilot is designed to make it easy to integrate other tools, and [Kane et al., 2020]

!! so in parallel to our linking scheme is the development patterns that we use. The linking system is general enough for allcomers, and it implies the patterns of linkage that should exist, but they then need to be implemented. Much like desire pathways though, the frequent co-use of different tools gives a good idea about the direction that development should go. So the systems work reciprocally: metadata linking system connect ideas and tools, and can

!! these are examples of what happens when you relax the demanding parts of an exact ontology/knowledge graph – we don't guarantee computability across the graph itself, there's no way to automatically whiz uncritically across all datasets in the system, but as we have seen that's also not really true of the other systems either,

to the degree that it's desirable at all. Instead of having formal guarantees on the graph, we can design tools that automate certain parts of the interaction with the system to actually make our jobs easier. By being very permissive, we let the desire paths of tool use form. This is a very literal example of the 'empower people, not systems' principle.

!! reciprocally, we can also imagine the reverse: being able to develop metadata structures that are then code generators for tools that have a sufficiently sophisticated API – for example remember how we said Bonsai might have a hard time making generalizable behavioral tasks/etc? Imagine if someone made a code compilation tool that allowed people to declare abstract structures that could then be reusably reparameterzied that autocreated a bonsai workflow? In the same way that the metadata system can be used for *storage* of existing work, it can also be used to create abbreviate and abstract constructs for *use* with other tools.

!! continue the example of needing to select within datasets instead of metadata from federation section.

To take stock:

We have described a system of three component modalities: **data, analytical tools, and experimental tools** connected by a **linked data** layer. We started by describing the need for a **peer-to-peer** data system that makes use of **data standards** as an onramp to linked metadata. To interact with the system, we described an identity-based linked data system that lets individual people declare linked data resources and properties that link to **content addressed** resources in the p2p system, as well as **federate** into multiple larger organizations. We described the requirements for **DAG-based analytical frameworks** that allow people to declare individual nodes for a processing chain linked to code, combine them into workflows, and apply them to data. Finally, we described a design strategy for **component-based experimental frameworks** that lets people specify experimental metadata, tools, and output data.

This system as described is a two-layer system, with a few different domains linked by a flexible metadata linking layer. The metadata system as described is not merely *inert* metadata, but metadata linked to code that can *do something* — eg. specify access permissions, translate between data formats, execute analayis workflows, parameterize experiments, etc. Put another way, we have been attempting to describe a system that *embeds the act of sharing and curation in the practice of science.* Rather than a thankless post-hoc process, the system attempts to provide a means for aligning the daily work of scientists so that it can be cumulative and collaborative. To do this, we have tried to avoid rigid specifications of system structure, and instead

described a system that allows researchers to pluralistically define the structure themselves.

!! Now we need to consider the social tools needed to communicate within, negotiate over, and govern the system.

Shared Knowledge

The remaining set of problems implied by the infrastructural system sketched in the preceding sections is the *communication* and *organization* systems that make up the interfaces to maintain and use it. We can finally return to some of the breadcrumbs laid before: the need for negotiating over distributed and conflicting data schema, for incentivizing and organizing collective labor, and ultimately for communicating scientific results.

The communication systems that are needed double as *knowledge organization* systems. Knowledge organization has the rosy hue of something that might be uncontroversial and apolitical — surely everyone involved in scientific communication wants knowledge to be organized, right? The reality of scientific practice might give a hint at our naivete. Despite being, in some sense, itself an effort to organize knowledge, *scientific results effectively have no system of explicit organization.* There is no means of, say, "finding all the papers about a research question." The problem is so fundamental it seems natural: the usual methods of using search engines, asking around on Twitter, and chasing citation trees are flex tape slapped over the central absence of a system for formally relating our work as a shared body of knowledge.

Information capitalism, in its terrifying splendor, here too pits private profit against public good. Analogously to the necessary functional limitations of SaaS platforms, artificially limiting knowledge organization opens space for new products and profit opportunities. In their 2020 shareholder report, RELX, the parent of Elsevier, lists increasing the number of journals and papers as a primary means of increasing revenue [REL, 2020] . In the next breath, they describe how "in databases & tools and electronic reference, representing over a third of divisional[23] revenue, we continued to drive good growth through content development and enhanced machine learning [ML] and natural language processing [NLP] based functionality."

[23] RELX is a huge information conglomerate, and scientific publication is just one division.

What ML and NLP systems are they referring to? The 2019 report is a bit more revealing (emphases mine):

> Elsevier looks to enhance quality by building on its premium brands and **grow article volume** through **new journal launches,** the expansion of open access journals and growth from emerging markets; and add value to core platforms by implementing capabilities such as **advanced**

recommendations on ScienceDirect and social collaboration through reference manager and collaboration tool Mendeley.

In every market, Elsevier is applying advanced ML and NLP techniques to help researchers, engineers and clinicians perform their work better. For example, in research, ScienceDirect Topics, a free layer of content that enhances the user experience, uses **ML and NLP techniques to classify scientific content and organise it thematically,** enabling users to get faster access to relevant results and related scientific topics. The feature, launched in 2017, is proving popular, generating 15% of monthly unique visitors to ScienceDirect via a topic page. **Elsevier also applies advanced ML techniques that detect trending topics per domain,** helping researchers make more informed decisions about their research. **Coupled with the automated profiling and extraction of funding body information from scientific articles,** this process supports the whole researcher journey; from planning, to execution and funding. [REL, 2019]

Reading between the lines, it's clear that the difficulty of finding research is a feature, not a bug of their system. Their explicit business model is to increase the number of publications and sell organization back to us with recommendation services. The recommendation system might be free*, but the business is to develop dependence to sell ad placement — which they proudly describe as looking very similar to their research content [Springer Nature, Elsevier, a] .

It gets more sinister: Elsevier sells multiple products to recommend 'trending' research areas likely to win grants, rank scientists, etc., algorithmically filling a need created by knowledge disorganization. The branding varies by audience, but the products are the same. For pharmaceutical companies "scientific opportunity analysis" promises custom reports that answer questions like "Which targets are currently being studied?" "Which experts are not collaborating with a competitor?" and "How much funding is dedicated to a particular area of research, and how much progress has been made?" [Elsevier, b] . For academics, "Topic Prominence in Science" offers university administrators tools to "enrich strategic research planning with portfolio overviews of their own and peer institutions." Researchers get tools to "identify experts and potential cross-sector collaborators in specific Topics to strengthen their project teams and funding bids and identify Topics which are likely to be well funded." [Elsevier, b]

These tools are, of course, designed for a race to the bottom — if my colleague is getting an algorithmic leg up, how can I afford not to? Naturally only those labs that *can* afford them and the costs of rapidly pivoting research topics will benefit from them, making yet another mechanism that reentrenches scientific inequity for profit. Knowledge disorganization, coupled with a little surveillance

capitalism that monitors the activity of colleagues and rivals [Brembs et al., 2021] , has given publishers powerful control over the course of science, and they are more than happy to ride algorithmically amplified scientific hype cycles in fragmented research bubbles all the way to the bank.

The consequences are hard to overstate. In addition to literature search being an unnecessarily huge sink of time and labor, science operates as a wash of tail-chasing results that only rarely seem to cumulatively build on one another. The need to constantly reinforce the norm that purposeful failure to cite prior work is research misconduct is itself a symptom of how engaging with a larger body of work is both extremely labor intensive and *strictly optional* in the communication regime of journal publication. Despite the profusion of papers, by some measures progress in science has slowed to a crawl as the long tail of papers with very few citations grows ever longer [Chu and Evans, 2021] .

While Chu and Evans correctly diagnose *symptoms* of knowledge disorganization like the need to "resort to heuristics to make continued sense of the field" and reliance on canonical papers, by treating the journal model as a natural phenomenon and citation as the only means of ordering research, they misattribute root *causes.* The problem is not people publishing *too many papers,* or a *breakdown of traditional publication hierarchies,* but the *profitability of knowledge disorganization.* Their prescription for "a clearer hierarchy of journals" misses the role of organizing scientific work in journals ranked by prestige, rather than by the content of the work, as a potentially major driver of extremely skewed citation distributions. It also misses the publisher's stated goal of pushing algorithmic paper recommendations, as there is nothing recommendation algorithms love recommending more than things that are alreaady popular. Without diagnosing knowledge disorganization as a core part of the business model of scientific publishers, we can be led to prescriptions that would make the problem worse.

!! Another impact of the arcanity of scientific knowledge organization is that it is effectively impenetrable to people that aren't domain experts. Why is trust in science so low right now? one contributor is that they have no idea what the hell we do or how different domains of knowledge have evolved. (cite cold war peer review and journals paper)

!! Practically, this makes the quality of scientific literature constantly in question. Each paper effectively exists as an island, and engagement with prior literature is effectively optional (outside the minimum bar set by the 3-5 additional private peer reviewers, each with their own limited scope and conflicting interests). Forensic peer-

reviewers have been ringing the alarm bell, saying that there is "no net" to bad research [Heathers, 2021] , and brave and highly-skilled investigators like Elisabeth Bik have found thousands of papers with evidence of purposeful manipulation [Shen, 2020, Bik et al., 2016] . !! So our existing systems of communication and organization are woefully inadequate for our needs, and don't serve the role of guaranteeing consistency or reliability in research that they claim to.

It's hard to imagine an alternative to journals that doesn't look like, well, journals. While a full treatment of the journal system is outside the scope of this paper, the system we describe here renders them *effectively irrelevant* by making papers as we know them *unnecessary.* Rather than facing the massive collective action problem of asking everyone to change their publication practices head on, by reconsidering the way we organize the surrounding infrastructure of science we can flank journals and replace them "from below" with something qualitatively more useful.

Beyond journals, the other technologies of communication that have been adopted out of need, though not necessarily design, serve as desire paths that trace other needs for scientific communication. As a rough sample: Researchers often prepare their manuscripts using platforms like Google Drive, indicating a need for collaborative tools in perparation of an idea. When working in teams, we often use tools like Slack to plan our work. Scientific conferences reflect the need for federated communication within subdisciplines, and we have adopted Twitter as a de facto platform for socializing and sharing our work to a broader audience. We use a handful of blogs and other sites like OpenBehavior [White et al., 2019] , Open Neuroscience, and many others to index technical knowledge and tools. Finally we use sites like PubPeer and ResearchGate for comment and criticism.

!! these tools are don't really suit our needs at all, and constrain rather than support some basic things that we want to do. For example, there is really no venue where we can have sustained, longform, multiparty discussions about difficult topics in our field. Sure, it is possible to publish series of commentary pieces back and forth, or to write blog posts against one another, or to squabble on twitter, but there's nothing that truly supports a cumulative body of public understanding well.

These technologies point to a few overlapping and not altogether binary axes of communication systems. !! make this a table? with technological examples for each.

- **Durable vs Ephemeral** - journals seek to represent information as permanent, archival-grade material, but scientific communication

also necessarily exists as contextual, temporally specific snapshots.

- **Structured vs Chronological** - scientific communication both needs to present itself as a structured basis of information with formal semantic linking, but also needs the chronological structure that ties ideas to their context. This axis is a gradient from formal ontologies, through intermediate systems like forums with hierarchical topic structure that embeds a feed, to the purely chronological feed-based social media systems.

- **Messaging vs Indexing** - Communication can be person-to-person or person-to-group messaging with defined senders and recipients, or intended as a generalizable category of objects. This ranges from entirely-specific DMs through domain-specific tool indexes like OpenBehavior through the uniform indexing of Wikipedia.

- **Public vs. Private** - Who gets to read, who gets to contribute? Communication can be composed of entirely private notes to self, through communication in a lab, collaboration group, discipline, and landing in the entirely public realm of global communication.

- **Formal vs. Informal** - Journal articles and encyclopedia-bound writing that conforms to a particular modality of expression vs. a vernacular style intended to communicate with people outside the jargon culture.

- **Push vs. Pull** - Do you go to get information from a reference location, or does information come to you as an alert or message?

Clearly a variety of different types of communication tools are needed, but there is no reason that each of them should be isolated and inoperable with the others. We have already seen several of the ideas that help bring an alternative into focus. Piracy communities demonstrate ways to build social systems that can sustain infrastructure. Federated and protocol-based systems show us that we don't need to choose between a single monolithic system or many disconnected ones, but can have a heterogeneous space of tools linked by a basic protocol. The semantic web and linked data people showed us the power of triplet links as a very general means of linking disparate systems. We can bridge these lessons with some from the early wiki movement to get a more practical sense of what it takes to give people total control over the structure of their communication and knowledge systems. Together with our sketches of data, analytical, and experimental tools we can start imagining a system for coordinating them — as well as displacing some of the more intractable systems that misstructure the practice of science.

!! This is a knotty and tangled history, so I am not attempting a full recounting, but will be selectively telling the story to motivate the kinds of tools we need.

The Wiki Way

> If we take radical collaboration as our core, then it becomes clear that extending Wikipedia's success doesn't simply mean installing more copies of wiki software for different tasks. It means figuring out the key principles that make radical collaboration work. What kinds of projects is it good for? How do you get them started? How do you keep them growing? What rules do you put in place? What software do you use? [Swartz, 2006a]

> So that's it — insecure but reliable, indiscriminate and subtle, user hostile yet easy to use, slow but up to date, and full of difficult, nit-picking people who exhibit a remarkable community camaraderie. Confused? Any other online community would count each of these "negatives" as a terrible flaw, and the contradictions as impossible to reconcile. Perhaps wiki works because the other online communities don't. [Leuf and Cunningham, 2001, **?**, **?**] and in WhyWikiWorks[24]

Aside from maybe the internet itself, there is no larger public digital knowledge organization effort than Wikipedia. While there are many lessons to be learned from Wikipedia itself, it emerged from a prior base of thought and experimentation in radically permissive, self-structuring read/write — sometimes called "peer production" [Hill and Shaw, 2019] — communities. Wikis are now quasi-ubiquitous today[25], largely thanks to Wikipedia, but its specific history and intent to be an *encyclopedia* entwines it with a very particular technological and social system that obscures some of the broader dreams of early wikis.

Aaron Swartz recounts a quote from Jimmy Wales, co-founder of Wikipedia:

> "I'm not a wiki person who happened to go into encyclopedias," Wales told the crowd at Oxford. "I'm an encyclopedia person who happened to use a wiki." [Swartz, 2006b]

And further describes how this origin and mission differentiates it from other internet communities:

> But Wikipedia isn't even a typical community. Usually Internet communities are groups of people who come together to discuss something, like cryptography or the writing of a technical specification. Perhaps they meet in an IRC channel, a web forum, a newsgroup, or on a mailing list, but the focus is always something "out there", something outside the discussion itself.

[24] Interestingly, this quote is almost, but not exactly the same as that on Ward's wiki: > So that's it - insecure, indiscriminate, user-hostile, slow, full of difficult, nit-picking people, and frivolous. Any other online community would count each of these strengths as a terrible flaw. Perhaps wiki works because the other online communities do not.

I can't tell if Ward Cunningham wrote the original entry in the wiki, but in any case seems to have found a bit of optimism in the book.

[25] though their corporate manifestations would probably be unrecognizable to the project early wiki users imagined.

> But **with Wikipedia, the goal is building Wikipedia.** It's not a community set up to make some other thing, it's a community set up to make itself. And since Wikipedia was one of the first sites to do it, we know hardly anything about building communities like that. [Swartz, 2006a]

We know a lot more now than in 2006, of course, but Wikipedia still has outsized structuring influence on our beliefs about what Wikis can be. Wikipedia has since spawned a huge number of technologies and projects like MediaWiki and Wikidata, each with their own long and occasionally torrid histories. I won't dwell on the obvious and massive feat of collective organzation that the greater Wikipedia project represents — learning from its imperfections is more useful to us here, especially for things that aren't encyclopedias. The dream of a centralized, but mass-edited "encyclopedia of everything" seems to be waning, and its slow retreat from wild openness has run parallel to a long decline in contributors [Hill and Shaw, 2019, Halfaker et al., 2013] . Throughout that time, there has been a separate (and largely skeptical) set of wiki communities holding court on what radically open web communities can be like, inventing their worlds in realtime. These thought communities have histories that are continuous with one another, and in their mutual reaction and inspiration sometimes teach similar lessons from across the divides of their very different structure.

The first wiki was launched in 1995[26] (still up) and came to be known as Ward's wiki after its author WardCunningham. Technically, it was extremely simple: a handful of TextFormattingRules and use of WikiCase where if you JoinCapitalizedWords you create a link to a (potentially new) WikiPage — and the ability for anyone to edit any page. These very simple WikiDesignPrinciples led to a sprawling and continuous conversation that spanned more than a decade and thousands[27] of pages that, because of the nature of the medium, is left fully preserved in amber. Those conversations are a history of thought on what makes wiki communities work (eg. WhyWikiWorks, WhyWikiWorksNot), and what is needed to sustain them.

[26] it's complicated: http://wiki.c2.com/?WardsWikiTenthAnniversary

[27] 23,244 unique page names according to the edit history, but the edit history was also purposely pruned from time to time.

One tension that emerged early and was never fully resolved by these wikis is the balance between "DocumentMode" writing that serves as linearly-readable reference material, similar to that of Wikipedia, and "ThreadMode" writing that is a nonlinear representation of a conversation.

!! order vs contemporaneousness is a fundamental challenge of inventing culture in plaintext. The purpose of using a wiki as opposed to other technologies that existed at the time like bulletin boards, newsgroups, IRC, etc. was that it provided a means of structure (refer to previous Ward quote about providing structure for

communal development in the portland pattern repository group and maybe bring into text to make clearer). Tension was always between a desire for long now organization but always needed to relate to the immediate communicative and organizational needs. Make sure to restructure following reference to the utility of talk pages to carry through need for meta-organization in addition to any stable knowledge product: the always present need for multiple represesentations of the same thought, across timescales and interfaces.) http://wiki.c2.com/?WhyBotherToStructure

Ward Cunningham and other more organizationally-oriented contributors opposed ThreadMode (eg. ThreadModeConsideredHarmful, InFavorOfDissertation) for a number of reasons, largely due to the ThreadMess and WikiChaos it had the potential of creating.

> I occasionally suggest how this site should be used. My GoodStyle suggestions have been here since the beginning and are linked from the edit page should anyone forget. I have done my best to discourage dialog InFavorOfDissertation which offers a better fit to this medium. I've been overruled. I will continue to make small edits to pages for the sake of brevity. – WardCunningham [C2w]

Most pages are thus a combination of both, usually with some DocumentMode text at the top with ThreadMode conversations interspersed throughout without necessarily having any clean delineation between the two. Far from just being raw disorder, this mixed mode of writing gave it a peculiar character of being *both* a folk reference for a library of concepts *as well as* a history of discussion that made the contingency of that reference material plain. Beka Valentine put it well:

> c2wiki is an exercise in dialogical methods. of laying bare the fact that knowledge and ideas are not some truth delivered from On High, but rather a social process, a conversation, a dialectic, between various views and interests [valentine, 2021]

This tension and its surrounding discussions point to the need for multiple representations of a single idea: that both the social and reference representations of a concept are valuable, but aren't necessarily best served by being represented in the same place. There was relatively common understanding that the intended order of things was to have many ThreadMode conversations that would gradually be converted to DocumentMode in a process of BrainStormFirst-CleanLater. Many proposed solutions orbit around making parallel pages with similar names (like <pagename>Discussion) to clean up a document while preserving the threads (though there were plenty of interesting alternatives, eg. DialecticMode)[28].

[28] Contemporary wikis have continued this conversation, see DocumentsVsMessages on communitywiki.org

Wikipedia cut the Gordian Knot by splitting each page into a separate *Article* and *Talk* pages, with the talk page in its own **Namespace** – eg. Gordian_Knot vs Talk:Gordian_Knot. Talk pages resemble a lot of the energy of early wikis: disorganized, sometimes silly, sometimes angry, and usually charmingly pedantic. Namespaces extend the traditional "everything is a page" notion encoded in the WikiCase link system by giving different pages different roles. In addition to having parallel conversations on articles and talk pages, it is possible to have template pages that can be included on wiki pages with `{% raw %}{{double curly bracket}}{% endraw %}` syntax – eg. Template:Citation_Needed renders `{% raw %}{{Citation needed}}{% endraw %}` as [citation needed]. Talk pages have their own **functional differentiation,** with features for threading and annotating discussions that aren't present on the main article pages (see Wikipedia:Flow [Wik, 2021]).

!! from a starting point of all things being one thing, a la wikis, we frame this as functional differentiation in a namespace. We can reciprocally frame this as giving a common name to a variety of different systems. Talk pages have slowly reinvented forums, but we can also think about this problem as making forums and chatrooms and essays that are different reflections of a wiki article page.

The complete segregation of discussion to Talk pages is driven by Wikipedia's aspirations as an encyclopedia, with reminders that is the "sole purpose" peppered throughout the rules and guidelines. The presence of messy subjective discussions would of course be discordant with the very austere and "neutral" articles of an encyclopedia. There are no visible indications that the talk pages even exist in the main text, and so even deeply controversial topics have no references to the conversations in talk pages that surround them — despite this being a requested feature by both administrators and editors [Schneider et al., 2011] .

Talk pages serve as one of the primary points of coordination and conflict resolution on Wikipedia, and also provide a low-barrier entrypoint for questions posed to a space they perceive to be "an approchable community of experts" [Viegas et al., 2007] . The separation of Talk pages and the labyrinthine rules governing their use function to obscure the dialogical and collective production of knowledge at the heart of wikis and Wikipedia. The body of thought that structures Wikipedia, most of which is in its Wikipedia:* namespace, is immense and extremely valuable, but is largely hidden from most people. Since Wikipedia is "always already there" often without trace of its massively collective nature, relatively few people ever contribute to it. Reciprocally, since acknowledging personal contribution is or point of view is explicitly against some of its core policies

and traditions, there is little public credit outside the Wikipedia community itself for the labor of maintaining it.

The forking of Wards Wikis into the first SisterSites teaches a parallel strain of lessons. Ward's Wiki started as a means of organizing knowledge for the Portland Pattern Repository[29], a programming community (referred to as DesignPatterns below), and in 1998 they were overwhelmed with proponents of ExtremeProgramming, which caused the first fissure in the wiki:

> XP advocates seemed to be talking about XP at every possible opportunity and seemingly on every page with content the least bit related to software development. This annoyed a number people who were here to discuss patterns, leading to the tag XpFreeZone, as a request not to talk about ExtremeProgramming on that page.
>
> It was difficult to pick out the DesignPatterns discussion on RecentChanges[30], because most of the activity was related to ExtremeProgramming. Eventually, most of the DesignPatterns people left, to discuss patterns in a "quieter" environment, and people started referring to this site as WardsWiki instead of the PortlandPatternRepository [C2w]

One of the first and most influential Sister Sites was Meatball Wiki, described on Wards Wiki:

> SunirShah founded MeatballWiki to absorb and enlarge the discussion of what wiki and wiki like sites might be. That discussion still simmers here. But here it can take on a negative tone sounding more like complaining. On meatball, under Sunir's careful leadership, the ideas, wild or not, stay amazingly upbeat. - SisterSites

MeatballWiki became the spiritual successor to Ward's Wiki, which at that point had its own momentum of culture less interested in being the repository of wiki thought[31]. Though there are a truly monstrous number of ideas on MeatballWiki, the most relevant here here might be those concerning its very existence as a SisterSite. These were a series of discussions that melded thoughts from open source computing with social systems; in part: RightToFork, RightToLeave, EnlargeSpace, and TransClusion.

What can be done when the internal divisions in a wiki community and the weight of its history make healthy contribution impossible? The first place to start is with the RightToLeave, where it is always possible to just stop being part of a community. This approach is clearly the most destructive, as it involves abandoning the emotional bonds of a community, prior work (see the WikiMindWipe where a user left and took all their contributions with them), and doesn't necessarily provide an alternative that alleviates the cause of the tension. The next idea is to *fork* the community, where the body of a community — in the case of wikis the pages and history — can

[29] The initial motivations are actually stunningly close to the kinds of communication and knowledge organization problems we are still solving today (even in this piece) > Cunningham had developed a database to collect the contributions of the listserv members. He had noticed that the content of the listserv tended to get buried, and therefore the most recent post might be underinformed about posts which came before it. The way around this problem was to collect ideas in a database, and then edit those ideas rather than begin anew with each listserv posting. Cunningham's post states that "The plan is to have interested parties write web pages about the People, Projects and Patterns that have changed the way they program. Short stories that hint at patterns are welcome too." As to the rhetorical expectations, Cunningham added "The writing style is casual, like email or netnews, but doesn't have to be so repetitive since the things being discussed don't disappear. Think of it as a moderated list where anyone can be moderator and everything is archived. It's not quite a chat, still, conversation is possible." - [Cummings, 2009]

[30] Recent Changes was the dominant, if not controversial means of keeping track with recent wiki traffic, see RecentChangesJunkie

[31] There seems to have been an overriding belief that theoretical ideas about wikis and wiki culture belong on Meatball Wiki, from WikiWikiWebFaq: > Q: Do two separate wikis ever merge together to create one new wiki? Has this happened before? Keep in mind that I don't just mean two different pages within a wiki. (And for that matter, where is an appropriate page where I can post questions about the history of all wikis, not just this one?) > > A1: I don't know of any such wiki merge, nor of any discussion of the history of all wikis. Such a discussion should probably reside (if created) on MeatballWiki.

be duplicated so that it can proceed along two parallel tracks. Exercising the right to fork is, according to Meatball, "people exercising their RightToLeave whilst maintaining their emotional stake" [Mea, d] .

The discussion around the Right to Fork on Meatball is far from uniformly positive, and is certainly colored by the strong presence of its BenevolentDictator Sunir Shah who viewed it as a last resort after all attempts at ConflictResolution have failed. They point to the potentially damaging effects of a fork, like bitterness, disputes over content ownership (see MeatballIsNotFree), and potentially an avoidance of conflict resolution that is a normal and healthy part of any community. Others place it more in the realm of a radical *political* action rather than a strictly social action. Writing about the fork of OpenOffice to LibreOffice, Terry Hancock writes:

> [In] proprietary software [a] political executive decision can kill a project, regardless of developer or user interest. But with free software, the power lies with the people who make it and use it, and the freedom to fork is the guarantee of that power. [...] The freedom to fork a free software project is [a] "tool of revolution" intended to safeguard the real freedoms in free software. [Hancock, 2010]

Forking digital communities can be much less acrimonious than physically-based communities because of the ability to EnlargeSpace given by the medium:

> In order to preserve GlobalResources, create more public space. This reduces limited resource tension. Unlike the RealWorld, land is cheap online. In effect, this nullifies the TragedyOfTheCommons by removing the resource pressure that created the "tragedy" in the first place. **You can't overgraze the infinity.** - [Mea, a]

It is always possible to duplicate digital resources, create more spaces to resolve tensions over shared resources, and so on. Enlarging space has the natural potential to make the broader social scene bewildering with a geyser of pages and communities, but can be further made less damaging by having mechanisms to link histories, trace their divergence, and potentially resolve a fork as is common in open source software development. Forking is then a natural process of community regeneration, allowing people to regroup to make healthier spaces when needed, where the fork is itself part of the history of the community rather than an unfathomable rift.

Forking communities is not the same as forking community resources: "you can't fork a community [...] what you can do is fork the content and to *split* the community" [Mea, b] . As described so far, a fork divides people into unreconciled and separate communities. In some cases this makes forking difficult, in others it makes it impossible: the prime example, again, is Wikipedia. It is

simply too large and too culturally dominant to fork. Even though it is technically possible to fork Wikipedia, if you succeeded, then what? Who would come with you to build it, and who would that be useful for? This is partly a product of its totalizing effort to be an encyclopedia of everything (what good would *another* encyclopedia of everything be?) but also the weight of history: you won't get enough long-encultured Wikipedians to join you.

The last major effort to fork Wikipedia was in 2002 with an effort led by Edgar Enyedy to move the Spanish Wikipedia to The Enciclopedia Libre Universal en Español [Tkacz, 2011, 2014] . Though it was brief and unsuccessful, Enyedy claims that because Jimmy Wales was worried about other non-English communities following their lead, he and the other admins capitulated to the demands for no advertising and a transfer to a .org domain, among others[32]. Even a politically symbolic fork is dependent on the perceived threat to the original project, and that window seems to have been closed after 2002.

[32] Jimmy Wales, naturally, disputes this characterization of events.

The cultural tensions and difficulties that lead other wikis and projects to fork have taken their toll on the editorship and culture of Wikipedia. The community is drawn into dozens of conflicting philosophical camps: the Deletionists[33] vs. the Inclusionists, Eventualists vs. Immediatists, Mergists vs. Separatists, and yes even a stub page for Wikisecessionism. Editorship has steadily declined from a peak in 2007. Its relatively invisible community systems make it mostly a matter of chance or ideology that new contributors are attracted in the first place. In its calcification of norms, largely to protect against legitimate challenges to the integrity of the encyclopedia, any newcomers that do find their way into editing now have little chance to catch a foothold in the culture before they are frustrated by (sometimes algorithmic) rejection [Hill and Shaw, 2019, Halfaker et al., 2013] .

[33] Also see Association of Wikipedians Who Dislike Making Broad Judgments About the Worthiness of a General Category of Article, and Who Are in Favor of the Deletion of Some Particularly Bad Articles, but That Doesn't Mean They Are Deletionists

Arguably all internet communities have some kind of life cycle, so the question becomes how to design systems that support healthy forking without replicating the current situation of fragmentation. Wikis, including Meatball and MediaWiki, as well as other projects like Xanadu often turn to **transclusion** — or being able to reference and include the content of one wiki (or wiki page) in another. Rather than copying and pasting, the remote content is kept updated with any changes made to it.

Transclusion naturally brings with it a set of additional challenges: Who can transclude my work? Whose work can I transclude? Can my edits be propagated back to their work? What can be transcluded, at what level of granularity, and how? While before we had characterized splitting communities as an intrinsic part of a fork, that need not

be the case in a system built for transclusion. Instead relationships post-fork are then made an *explicit social process* within the system, where even if a community wants to work as separate subgroups, it is possible for them to arrive at some agreement over what they want to share and what they want to keep separate. This kind of decentralized work system resembles radical organizing tactics like hub-and-spoke models or affinity groups where many autonomous groups fluidly work together or separately on an array of shared projects without aspiring to create "one big movement" [Klein, 2001] . Murray Bookchin describes:

> The groups proliferate on a molecular level and they have their own "Brownian movement." Whether they link together or separate is determined by living situations, not by bureaucratic fiat from a distant center. [. . .]
>
> [N]othing prevents affinity groups from working together closely on any scale required by a living situation. They can easily federate by means of local, regional or national assemblies to formulate common policies and they can create temporary action committees (like those of the French students and workers in 1968) to coordinate specific tasks. [. . .] As a result of their autonomy and localism, the groups can retain a sensitive appreciation of new possibilities. Intensely experimental and variegated in lifestyles, they act as a stimulus on each other as well as on the popular movement. [Bookchin, 1969]

To cherrypick a few lessons from more than 25 years of thought from tens of thousands of people: The differing models of document vs. thread modes and separate article vs. talk pages show us that using **namespaces** is an effective way to bridge multimodal expression on the same topic across percerceived timescales or other conflicting communicative needs. This is especially true when the namespaces have **functional differentiation** like the tools for threading conversations on Wikipedia Talk pages and the parsing and code generation tools of Templates. These namespaces need to be **visibly crosslinked** both to preserve the social character of knowledge work, but also to provide a means of credit assignment and tool development between namespaces. Any communication system needs to be designed to **prioritize ease of leaving** and **ease of forking** such that a person can take their work and represent it on some new system or start a new group to encourage experimentation in governance models and technologies. One way of accomplishing these goals might be to build a system around **social transclusion** such that work across many systems and domains can be linked into a larger body of work without needing to create a system that becomes too large to fork. The need for communication across namespaces and systems, coupled with transclusion further implies the need for **bidirectional transclusion** so that in addition to being able to transclude something in a

document, there is visible representation of work being transcluded (eg. commented on, used in an analysis, etc.) by allowed peers and federations.

These lessons, coupled with those from private bittorrent trackers, linked data communities, and the p2p federated system we have sketched so far give us some guidelines and motivating examples to build a varied space of communication tools to communicate our work, govern the system, and grow a shared, cumulative body of knowledge.

!! on to talking about the work being done in this domain, some ux ideas from this, and a continuation of the sketch of the system!

Rebuilding Scientific Communication

!! need introduction!

It's time to start thinking about interfaces. We have sketched our system in turtle-like pseudocode, but directly interacting with our linking syntax would be labor intensive and technically challenging. Instead we can start thinking about tools for interacting with it in an abstract way. Beneath every good interface we're familiar with, a data model lies in wait. A .docx file is just a zipped archive full of xml, so a document with the single word "melon" is actually represented (after some preamble) like:

```
<w:body>
  <w:p
    w14:paraId="0667868A"
    w14:textId="50600F77"
    w:rsidR="002B7ADC"
    w:rsidRDefault="00A776E4">
    <w:r>
        <w:t>melon</w:t>
    </w:r>
  </w:p>
</w:body>
```

Same thing with jupyter notebooks, where a block of code:

```
>>> rating = 100
>>> print(f'I rate this dream {rating}')
'I rate this dream 100'
```

is represented as JSON (simplified for brevity):

```
{
  "cell_type": "code",
```

```
  "id": "thousand-vermont",
  "outputs": [{
    "name": "stdout",
    "output_type": "stream",
    "text": [
      "I rate this dream 100\n"
    ]
  }],
  "source": [
    "rating = 100\n",
    "print(f'I rate this dream {rating}')"
  ]
}
```

So we are already used to working with interfaces to data models, we just need to think about what kind of interfaces we need for a scientific communication system.

Let's pick up where we left off with our linked data and tools. Recall that we had a `project` named `#my-project` that made reference to our experiment, a few datasets that it produced, and an analysis pipeline that we ran on it. We *could* just ship the raw numbers from the analysis, wash our hands of it, and walk straight into the ocean without looking back, but usually scientists like to take a few additional steps to visualize the data and write about what it means.

Notebooks (JSON-LD) Say we have a means of downloading the results of some analysis we have already run as a result of `#my-project`. Recall that the data system we described was a system that links names under our `@jonny` namespace to a content-addressed p2p system, but someone has built a package to handle that under the hood. We might do a quick writeup in a notebook like this:

!! insert jupyter notebook here!

The .json inside our notebook file would look something like this:

```
{
   "cell_type": "code",
   "execution_count": 2,
   "id": "rapid-information",
   "metadata": {
    "scrolled": true
   },
   "outputs": [
    "..."
   ],
   "source": [
```

```
    "x, y, sizes = get_data('@jonny:my-project:Analysis1')"
   ]
}
```

We could make use of another linked data technology, JSON-LD, that is an extension and format that is interoperable with the RDF links we have been using implicitly throughout, to note that this cell contains a reference to our dataset. Say we use a `@comms` ontology to denote the various parts of our communication system, and put that in the `metadata` field:

```
"metadata": {
  "scrolled": true,
  "@comms:usesData": "@jonny:my-project:Analysis1"
}
```

Now say we have another little interface to declare links inline in our notebook using magic commands. We might declare the name of our notebook like

`%%docId @jonny:my-project:Writeup`

and then in the cell we indicate that we have plotted our data like this:

```
%%cellId Plotty
%%cellLink @comms:plotsData @jonny:my-project:Analysis1
```

So then, say, we indicate in `@jonny:my-project` that this document is related to it, and the links embedded within the notebook indicate that it has cells that use a specific result and plot it. If I enable sharing from my namespace, it becomes a creditable and discoverable part of my scientific work — a straightforward means of breaking up the scientific paper as the unit of knowledge work. Recall that our sharing rules weren't just a binary switch, but can indicate different people and groups, so we can communicate the intention of publication and status of the document[34] on an analogue scale from a private demo to our lab, a presentation to an institute or conference, or a public part of the scientific discourse.

!! also brief nod to other document systems like https://dokie.li/

[34] While we're at it, why not make it explicit by declaring its `creativeWorkStatus` as `Draft`

Forums & Feeds Communication doesn't need to be (and shouldn't be) exclusively unidirectional statements of fact. Our linked data system that allows us to directly references the subcomponents of an experiment, including analysis results and visualizations, naturally lends itself to use in a **forum.** In between feed-only mediums like most social media platforms and the indexical permanence of a wiki or publication, forums are a currently missing piece in most scientific

communication systems: a way to have longform discussions about science in a public and semipermanent environment.

We can start by imagining a forum where people in our discipline go to present their work and solicit feedback. We think we really have something, and it challenges some widely held previous results:

> hi everyone it is me, take a look at my analysis: `[[@forum:showImage @jonny:my-project:Writeup:Plotty]]` !!render inline
>
> I think it raises a number of interesting questions, in particular about `@rival`'s long-standing argument `@rival:hillsToDieOn:earthIsInsideTheSun` I also wonder what this means about this conversation we've been having more broadly about `@discipline:whereAreThePlanets`. Anyway, write back soon, xoxo

Our rival is polite and professional, so they take the criticism in stride and do their own analysis:

> Interesting results! I think I will have to revisit that, as well as something else I have been working on, `@rival:projects:escapeTheSun`. I wonder what it would look like if we used my analysis pipeline instead. I wrote a few conversion nodes (`@rival:nodes:newNode`) that could make our work easier to synchronize in the future.
>
> `[[@forum:rerunAnalysis @jonny:my-project:Analysis1 @rival:newAnalysis]]`
>
> `[[@forum:completeGraph @rival:newAnalysis @jonny:my-project:Writeup:Plotty]]`
>
> `[[@forum:showImage @rival:newAnalysis:Writeup:Plotty]]`

They have their own compute server set up that listens for commands like `@forum:rerunAnalysis` and so once they post, their server downloads the container and re-runs the analysis. `rerunAnalysis` is a link between our two analysis pipelines, so it is also possible to cross-apply the other parts of my analysis chain to their reanalysis. In this case say my `@rival` was careful to ensure their pipeline returned exactly the same data format as mine did, so it's possible to use something like `completeGraph` to retrace the steps in between the results and the plots that were generated. These are, of course, speculative features of a speculative forum, but they serve as examples of how this kind of federated naming system allows for new kinds of tools.

Sharing results, communicating them to the people that might be interested, reconsidering and re-analyzing work is an extremely normal part of science, but in this parallel universe we have the tools to also contribute to a cumulative body of knowledge that is explicit and public. If we allowed it, people that were interested in our data would be able to find the other ways it was analyzed, visualized, and discussed. We have recontextualized ours and our `@rival`'s previously published work and enriched the discussion surrounding

our discipline's ongoing struggle to understand `whereAreThePlanets`. And we managed to do it incrementally, with a smaller document than an occasionally-titanic manuscript might be.

Traditional forums like phpBB are housed on a single domain and server, and have fixed moderation and structure. A forum built on top of a p2p system of linked data designed for **transclusion** and **ease of forking** could look a little different. Rather than independent web service, we could build a forum as another peer in our p2p swarm, and the forum could operate as an *interface* to the linked data system.

For concreteness, let's call our forum `@neurochat`. We join the forum with our existing identity by sending them a `@as.Join` request from their login portal, which gives them permission to issue certain links and activity on our behalf. `@neurochat` is a minimal forum, a glassy reflection of a platonic ideal projected against the cave wall of our laptop. It has a few broad categories like "Neuromodulation" and "Sensory Neuroscience," within which are collections of threads full of chronologically-sorted posts. This organization is reflective of their internal concept model, so, for example, threads within the Neuromodulation category are represented as members of `@neurochat:categories:Neuromod` and so on. When we post through their web interface, we create a few links with shared custody: We create a `@as:Note` that is `@as:attributedTo` us, has the `@as:context` of the thread we're posting in, and is linked as `@as:inReplyTo` the preceding post or any we've quoted. The forum is thus represented as a *discourse graph* whose structure is encoded as triplet links, but also provides a set of UX tools for viewing and interacting with it. Our humble `@jonny:myproject` now also carries with it references to the places where it is discussed.

In the simplest case, the content of our posts could be mirrored between the `@neurochat` server and our own namespace. Say our post `@neurochat:posts:<post_id>` is mirrored as `@jonny:neurochat:...`. The embedding within our linked system give us a much richer space of negotiation over permissions and the status of our writing, though. Since this is a public forum, the server might set posts to be able to be seen and re-represented by default. We could then imagine a set of federated forums where a single post to one of them is then crossposted to several different communities: eg. if our work was an interdisciplinary project that was also releant to some people from `@linguisticsChat`. If we have need for a bit more privacy, our forum could take into account our own blocks of users and federations, eg. if we never wanted our data/posts to be used by any `@amazon`-affiliated federations or by known troll users. `@neurochat` is a very barebones forum, so it would also be possible for someone to create

their own *fork* of the *interface* to provide additional functionality, ux improvements, etc. We could then trivially make a *fork* of the *community* by picking up our corner of the discourse graph and associating it with a new forum in the event of, eg. disagreements with the moderation team, the strictures of the category system, etc. Since our posts are in our own namespace, we could then transclude them wherever we wanted, eg. in a wiki page about a topic as in agora's twitter bot.

We have been considering `@neurochat` as a distinct site with its own code and features, presumably located at something like `neurochat.com`, but we can further imagine it in conversation with the parallel namespaces of wiki `Talk:` pages. If we think of a paper or some other primary text as the "Article" page, we can imagine being able to have a `Forum:` attached to it for further discussion. This isn't far-fetched at all: this paper has its own gitter chatroom, which is a primarily web-based Matrix client [Hodgson, 2020a,b] . Combined with transclusion between instances of forums, we could imagine the forum for our particular project being indexed in a larger system of scientific forums. So rather than a collection of empty rooms and new logins to make, our forum is part of a broader scientific conversation, but remains under our control.

Forums are just one point in a continuous feature space of communication media: nested, chronological, feedlike collections of threads within categories. If we were to take forum threads out of their categories, pour them into our water supply, and drink whatever came our way like a dog drinking out of an algorithmic fire hydrant, we would have Twitter. Algorithmic, rather than purposefully organized feed systems have their own sort of tachycardic charm. They are effective at what they aim to do, presenting us whatever maximizes the amount of time we spend looking at them in a sort of hallucinatory timeless now of infinite disorganization — at the expense of desirable features of a communication system like a sense of stable, autonomously chosen community, perspective on broader conversation, and cumulative collective memory.

Still, the emergence of a recognizable "Science Twitter" demonstrates the depth of need for rapid, informal communication systems in science. We should embrace the plurality of registers in scientific communication, that there needs to be space for near-amateurs to pose naive questions alongside careful and considered formal scholarship. That is just to say that we should reflect the division of formality from scientific value in what we build, and build systems to support the implicit communicative labor of science like whisper networks, mailing lists, and groupchats that have always existed. The blending of digital cultures, and broadly 'non-academic scientists'

with traditional scientific communication streams is healthy: with appropriate caveats for abuse, strawmen, et al. I don't think it takes that much critical analysis to argue that "shitposts are good, actually, for science."

A federated, multi-interface, autonomously-hosted system of social media systems already exists, and we've been talking about it: the roughly construed "Fediverse" based (largely) on ActivityPub. !! check rest of document and see how much explanation of activitypub is needed here/what can be consolidated. but in any case provide some other examples like peertube and agora, dokieli, funkwhale

Mastodon already implements most of the forum example described above: it has its own protocol that extends activitypub, but it functions as an interface to a protocol-based threaded communication. For example this post is represented in (abbreviated) JSON:


```
{
    "id": "107328829457619549",
    "created_at": "2021-11-23T22:52:49.044Z",
    "in_reply_to_id": "107328825611826508",
    "in_reply_to_account_id": "274647",
    "sensitive": false,
    "spoiler_text": "",
    "visibility": "public",
    "url": "https://social.coop/@jonny/107328829457619549",
    "content": "<p>and making a reply to the post to show the in_reply_to and context fields</p>",
    "account":
    {
        "id": "274647",
        "username": "jonny",
        "fields":
        [ ... ]
    },
    "media_attachments": [],
    "mentions": [],
    "tags": [],
}
```


and then rendered by the particular version of Mastodon implemented on the host, social.coop. As long as the host sends and receives post (and other) data in a compatible format, it can render it however it wants, add tools, etc. It becomes trivial to imagine, then, a continuum of communication tools between and around microblogging sites like Twitter and Mastodon and forums: just add categorization, tagging, or systems for whatever need is revealed by the normal dynamics of use.

The problem with an endless homogenous feed is filtering and prioritizing what to show. The lack of control over feed content is not an accident: it's the product — ready access to a hundred million hamsters on personalized content wheels with whatever combination of micro and macrotargeting you could want. Nothing seems out of the ordinary when you have no control over what you see. Reciprocally, there's no way aside from herding a flock of alternate accounts to direct what you say to different audiences. Mastodon can filter posts at a federation level[35], with hashtags, and lets users make lists of peers, but is a proudly chronological feed. No algorithms allowed. Using it has a learning curve, as when you start you see nothing, but before you know it you can't find anything in the pile. Forums threads, within categories are also typically chronologically sorted, but because they are identified with a *subject* rather than by the *person* who started the thread typically have longer lifespans and more findable.

[35] Only other servers that the host server federates with are listed

!! There is no single answer to systems of discovery, but somewhere between explicit categorical organization, person and subject-centric threads, semantic annotation, and making smaller p2p federations is a recipe for a broad, continuous, and cumulative scientific discussion. Instead of casting about for advice within our information bubbles, we might aspire to having a *place* to *ask* the people who *might know*. Instead of starting another new slack with a few hundred posts that then vanishes entirely, we might imagine being able to fluidly form and dissolve communities and be able to build on their history.

Annotation & Overlays We can't expect the entire practice of academic publishing to transition to cell-based text editors anytime soon. In the same way that we discussed frameworks for integrating heterogeneous analytical and experimental tools, we need some means of **bridging** communication tools and **overlays** for interacting with communication formats. Bridging communication protocols is a relatively well-defined project, eg. the many ways to use Matrix with Slack, email, Signal, etc. The overlays for websites, pdfs, and other more static media that we'll discuss are means for annotation and bidirectional transclusion: including pieces of the work elsewhere, and representing inclusions elsewhere on the work. In representing the intrinsically interactive and social nature of reading (eg. see [Jackson, 2001]), overlays naturally lend themselves to imagining new systems to replace traditional mechanisms for peer-review and criticism. We don't need to look far to find a well-trod interface for annotation overlays: we shouldn't underrate the humble highlighter.

Hypothes.is, enabled on this page, lets readers highlight and annotate any webpage with a browser extension or javascript bookmarklet. At its heart is a system for making anchors, references to specific places in a text, and the means of matching them even when the text changes or the reference is ambiguous [csillag, 2013] . For example, this anchor has three features, a `RangeSelector` that anchors it given the position within the paragraph, an absolute `TextPositionSelector`, and a contextual `TextQuoteSelector` that you can see with an API call.

On its own, it serves to give a `Talk:` page to every website. With an integration into a system of linked data and identity, it also serves as a means of extending the notion of bidirectional transclusion described above to work that is not explicitly formatted for it. Most scientific work is represented as `.pdfs` rather than `.html` pages, and hypothes.is already supports annotating PDFs. With an integration into pdf reading software, for example Zotero's (currently beta) PDF reader, there would be a relatively low barrier to integrating collaborative annotation into existing workflows and practices.

The dream of public peer review with public annotation is relatively straightforward, but so are the nightmares of a scientific literature swamped with trolls. Talking about our work on a forum with a "forward" reference, of the work linked to by the forum or on PubPeer feels fine, but the "reverse" reference of an annotation appearing on your page that links to a forum discussion is significantly more challenging — "who gets to annotate my work?"

Framed as an annotation system, the answer given by the current model of peer review is "usually three, usually anonymous people." Except the document and annotations are usually private until the author revises the document to the point where no annotations remain, and the peer reviewers become invisible along with the social nature of the review. The notion that the body of scientific knowledge is best curated by passing each paper through a gauntlet of three anonymous reviewers, after which it becomes Fact and nearly as a rule never changed is ridiculous on its face.

Digital publishing makes imagining the social regulation of science as a much more broadly based and continuous process much easier, but the problem of moderation remains. Some movement has been made towards public peer review: eLife has integrated hypothes.is since 2016 [ELi, 2016] , and bioRxiv had decided to integrate it as well in 2017 [dwhly, 2017] before getting cold feet about the genuinely hard problem of moderation (among others [heatherstaines, 2018]) and instead adopting the more publisher-friendly TRiP system of refereed peer-reviews [nateangell, 2019] .

Asking every author to become a forum moderator and constantly

patrolling the annotations of their papers sounds bad, as does the work of maintaining block and allowlists for every individual account. While a full description of the norms and tools needed to maintain healthy public peer review is out of scope here, the system of autonomous users being able to organize into overlapping federations described throughout *provides a space for having that conversation.* Authors could, for example, allow the display of annotations from a professional society like `@sfn` that has a code of conduct and moderation team, or annotations associated with comments on `@pubpeer`, or from a looser organization of colleagues and other `@neurofriends`. Conversely, being able to make annotations and comments from different federations gives us a rough proxy to different registers of communication and preserves the plurality of our expression. While my official `@university`-affiliated federation is restrained and academic, my `@neurotrans` alt might be a little more freewheeling. A protocol for federating peers that we first described in the context of sharing data has the more general consequence of creating a means of negotiating and experimenting with different systems of social norms and governance.

Social tools like these are in the hypothes.is team's development roadmap, but I intend it as a well-developed and mature example of a general type of technology[36]. Many scientists are already familiar with another implementation: the comment and review features of Google Docs and Microsoft Word. We already use these tools to work together to improve our work, but there's no reason the process should stop at the time of publication. Combined with a system for valuing and publishing smaller units of work, the process of public peer review starts to look a lot healthier as a continuous process of communication and collective mentorship instead of the current system of a gladitorial thumbs up/down indictment on years of your life.

[36] cf. the genius.com overlay.

Trackers & Wikis The final set of social interfaces is effectively the "body" of social technology. So far our infrastructural systems have an unsatisfyingly hollow center: there's a lot of talk about tool frameworks and protocols for linked data, but *where is it? what does it look like?* We can pick up the threads left hanging from our description of bittorrent trackers and knit them in with those from the wiki way and describe how systems for surfacing procedural and technical knowledge work can also serve as a basis of searching, indexing, and governing the rest of the system.

Bittorrent trackers serve to index data and organize a curation community — we need that too, let's start there. Say we have a tracker that indexes a particular format of data, as `@dandihub` does

with `@nwb`. We can search for data using all the fields of NWB, but don't want to rely just on the peers that are active, so the role of the tracker is to maintain a searchable index of metadata that refers to the datasets shared by peers. We want to be interoperable with other trackers that index compatible data, so let's say that's implemented as a database that supports SPARQL federated queries[37] where requests can be spread across many databases. For concreteness, let's assume that the results of our search are some content-addressed reference to a resource on a p2p network like a magnet link.

We need some kind of *client* that can download files and run in the background to share them. We can start with the image of a bittorrent client like qBittorrent that does just that, but we also need a means of making the link declarations that we did before in pseudocode, and it makes sense for the client to handle that as well. Let's say our client handles our identity, either by a self-created cryptographic hash as in IPFS[Benet, 2014] , or attested by some trusted third party as in ActivityPub. Instead of our identity being tied to the services provided by the server, however, we can think of this as a peer-to-peer ActivityPub where we can directly send and receive messages containing our links and negotiating our connections. As an interface, say we have a typical file browser that we can set permissions for files, group them into projects, and share them with others. Since the system consists of links, an editor that allows users to visualize and edit a hierarchical graph of nodes and (typed) edges:

!! input network editing React figure from presentation here!

So say it's time for us to share a dataset. We click the 'share' button in our client which sends an ActivityPub-style message saying we have `@as:Created` a new resource to the other peers indicated in our permission settings. This message both uploads the metadata for our dataset to the, say, `@dandihub` tracker, but since `@dandihub` is an equivalent peer in our system, and modeling off ActivityPub we are able to have "friends," we can notify other researchers directly. The tracker can host our metadata pointing to our data so it's available from any other peer that's hosting it even if we go offline, but peers can query us directly to enumerate all the links, datasets, etc. we have allowed them to.

What about handling format extensions not included in the base `@nwb` format? Since we own the representation of our data, we can imagine a strict base `@nwb`-only tracker, but also think of `@dandihub` that has built tools to handle extensions. So alongside our dataset we can upload an extension like our `@jonny:SolarEphys` example that derives from `@nwb:ElectricalSeries`, and the tracker then can display our extension as well as all the other extensions that branch off the various points of the standard. At this point we can imagine

[37] Tim Berners-Lee describes the distinction between traditional relationship databases and RDF databases: > 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 sophisticated operations to be performed across them. https://www.w3.org/DesignIssues/RDFnot.html

a spray of thousands of trivially different extensions to handle overlapping data types, which is where most data stores typically stop, but let's explore community systems built on forums and wikis for schema resolution as an example of *distributed governance.*

!! figure of lots of leaf nodes hanging off ElectricalSeries

Wikis are not magical systems of infinite pluralistic knowledge, but one thing they do well is provide the means of developing durable but plastic systems norms and policies for a wide variety of social systems. Butler, Joyce and Pike, emphasis mine:

> Providing tools and infrastructure mechanisms that support the development and management of policies is an important part of creating social computing systems that work. [...]
>
> When organizations invest in [collaborative] technologies, [...] their first step is often to put in place a collection of policies and guidelines regarding their use. **However, less attention is given to the policies and guidelines created by the groups that use these systems which are often left to "emerge" spontaneously.** The examples and concepts described in this paper highlight the complexity of rule formation and suggest that support should be provided to help collaborating groups create and maintain effective rulespaces.
>
> [...] **The true power of wikis lies in the fact that they are a platform that provides affordances which allow for a wide variety of rich, multifaceted organizational structures.** Rather than assuming that rules, policies, and guidelines are operating in only one fashion, wikis allow for, and in fact facilitate, the creation of policies and procedures that serve a wide variety of functions [Butler et al., 2008]

So between discussion on the forum or in `Talk:`-like pages, we can imagine a set of norms and policies evolving from the community on this particuar tracker, perhaps unlike other trackers. In this case we can imagine someone wanting to clean up some near-equivalent extensions by starting a thread in the forum to discuss the proposed changes. Say we want to merge `@jonny:Extension1` and `@rumbly:Extension2` – the forum notifies us that someone is talking about our extension so we have a chance to weigh in. If we reach some sort of amicable consensus where we agree to supercede it with a merged `@forum:Extension3` type, the forum could send us a `@as:Offer` to `@as:Update` our extension, which should we `@as:Accept` from our client then notifies all the downstream consumers of our data and extension that its format has changed.

What if consensus fails? Since every link in the system is underneath a `@namespace`, links never have a pretense of "correctness," but have the ontological status of a linguistic gesture: links are "something someone said" that we're free to disagree with[38]. In that case, the `@forum:Extension3` exists as "someone said these are equivalent, but I don't necessarily agree" and the forum is free to represent its

[38] "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." - https://www.w3.org/DesignIssues/RDB-RDF.html

cleaned up representation while preserving the plurality of expression in our data format. If I want to go to greener pastures to a forum that has policies and culture closer to mine, it's relatively straightforward to federate with a new tracker and move my data there since I still own it all.

Let's pick up scientific communication in linked data forums in conversation with the social incentives for curation of trackers. This system as described is a forum where everyone in the conversation has access to the data and results in question reminiscent of What.cd and access to music. While upload/download ratio might not be the best social incentive system for scientific trackers, there are plenty of others.

For example, we briefly mentioned a Folding@Home-like system of donated computing resources, and separately described embedding analyses in a forum by calling our own compute resources. Together, a tracker could implement a compute ratio where to use shared computing resources you need to contribute a certain amount of your own. The bounty system where peers would donate their excess upload in exchange for uploading a rare album on what.cd could translate to one where someone who has donated a lot of excess compute time could donate it for someone uploading or collecting a particular dataset.

Another tracker more focused on sharing and reviewing results might make a review ratio system, where for every review your work receives you need to review n other works. This would effectively function as a **reviewer co-op** that can make the implicit labor of reviewing explicit, and develop systems for tying the reviews required for frequent publication with explicit norms around reciprocal reviewing.

Forum and feedlike media are good for organizing continuous conversation, but wikis serve as a more durable knowledge store for cumulative reference information. We don't need to imagine wikis as being text-only, with wiki formatting used just to change the appearance of text, but as a means of declaring and manipulating semantic links. For example, Semantic MediaWiki is an extension to Wikipedia's wiki system that extends `[[Wikilinks]]` to be able to declare semantic links like `[[linkType::Target]]`. For example, if our project had a wiki page like `[[My Project]]` we could say it `[[hasType::@analysis:project]]` and `[[usesDataset::@jonny:mydata1]]` etc. These wikis have the capability to not only organize knowledge, but also serve as a flexible means of declaring new programming interfaces and assigning credit.

As a live example, let's consider the Autopilot Wiki at `https://wiki.auto-pi-lot.com`. This wiki has a set of categories, proper-

ties, templates, and forms for describing the additional contextual technical knowledge needed to use Autopilot, a framework for behavioral experiments [Saunders and Wehr, 2019] . The semantic structure of the links is useful for designing interfaces based on complex queries, for example "find me all the `passive electronic components` that have a `guide` that describes `using` a `soldering iron` to build `lighting` for a behavioral `enclosure`". Each page can have a rich semantic description with multimodal links describing tools, CAD diagrams, associated DOIs, software dependencies, etc. Links can be declared `[[linkModality::inline]]` as a fluid part of writing, but also can be submitted by using forms (eg for new Parts) with structured, autocompleting properties to lower syntax barriers for new users.

The "soft durability" of wikis makes space to discuss "off-label" uses for hardware common across many disciplines that typically exists as lab lore rather than documented. For example, an early-adopter of Autopilot sent me a message saying they weren't able to get ultrasound from an amplifier that was advertised up to 192kHz. Upon further study, we found there was a 20kHz low-pass output filter and were able to find and remove the components and leave a trail of breadcrumbs for future users. Though this is a simple example, it is emblematic of the kind of knowledge work that currently has no good means of communication or professional valuation.

The blend of programmatic and natural language descriptions makes it easy to contribute to, but also makes knowledge organization improve the software that uses it. The Amp2 page lists which of the GPIO pins of a raspberry pi it depends on, so Autopilot will be extended to check for conflicting hardware configurations[39]. Better: since it's possible for anyone to make new templates, forms, categories, and pages, the wiki can be used to build new programming interfaces entirely. Autopilot's plugin system is built this way, where one submits a plugin with a form which then makes it immediately available to any Autopilot user.

[39] for example, pin 7 mutes the board, but is still exposed in the 40-pin header. We powered an LED with pin 7 and were absolutely baffled why the sound would mute every time the light went on for a week or so.

The addition of structured contextual knowledge to our system gives us an almost comical degree of provenance: from conversations in a forum that reference a paper, that links to its analysis, data, experimental software, all the way back to the properties of the solenoids used in the experiment. It's not just provenance for provenance's sake as extra labor, every step is *useful* to the experimenter. I give the example of the Autopilot wiki for concreteness, but the broader point is that forums and wikis can serve the role of negotiating systems of expression for different parts of the system.

The same combination of trackers, forums, and wikis has a natural application to analysis pipelines. Ideally, to move beyond fragile

code reduplicated in every lab, we need some means of reaching consensus on a few canonical implementations of fundamental analysis operations. Given a system where analysis chains are linked to the formats and subdisciplines they are used with, we can map a semantically dense map of the analysis paths used in a research domain. In neurophysiology: "What are the different ways spikes are extracted and analyzed from extracellular electrophysiology recordings?" Having the ability to discuss and contextualize different analytical methods elevates all the exasperated methods critiques and exhortations to "not use this statistically unsound technique" into something *structurally expressed in the practice of science.* See all the `@neurotheory` threads about this specific analysis chain, or the `@methodswiki` page that summarizes this general category of techniques.

We're now in a place where we can address the problem of a cumulative knowledge system for science directly. In many (most?) scientific epistemologies, scientific results do not directly reflect some truth about reality, but instead instead are embedded in a system of meaning through a process of active interpretation (eg. [Meehl, 1978]). The interpretation of every scientific result is left as the responsibility of the authors to recreate and a few reviewers to evaluate, which would be a monumental amount of labor given the velocity of papers, so researchers do the best they can engaging with a small amount of research. Since the space of argumentation is built from scratch each time from incomplete information, there's no guarantee of making cumulative progress on a shared set of theories, and most fall far from the supposed ideal of hard refutation and can have long lives as "zombie theories." van Rooij and Baggio describe the "collecting seashells" approach of gathering many results and leaving the theory for later with an analogy:

> "In a sense, trying to build theories on collections of effects is much like trying to write novels by collecting sentences from randomly generated letter strings. Indeed, each novel ultimately consists of strings of letters, and theories should ultimately be compatible with effects. Still, the majority of the (infinitely possible) effects are irrelevant for the aims of theory building, just as the majority of (infinitely possible) sentences are irrelevant for writing a novel." [van Rooij and Baggio, 2021]

They and others (eg. [Guest and Martin, 2021]) have argued for an iterative process of experiments informed by theory and modeling that confirm or constrain future models. Their articulation of the need for multiple registers of formality and rigidity is particularly resonant here. van Rooij and Baggio again:

> "The first sketch of an f need not be the final one; what matters is how

> the initial f is constrained and refined and how the rectification process can actually drive the theory forward. Theory building is a creative process involving a dialectic of divergent and convergent thinking, informal and formal thinking." [van Rooij and Baggio, 2021]

Let's turn our provenance chain into a circle: a means of linking theories to analytical results and interpretation as well as experimental design and tooling. Say the theorists have a wiki. They start making some loose schematic descriptions of their theories and linking them to different experimental results that constrain, affirm, refute, or otherwise interact with them. These could be forward or backlinks: declared by the original author or by someone else describing their results.

- theorists have a wiki
- structurally express theory, but even if not, link to different experimental results variations:
- see how the evidence for a theory is collected – what papers, datasets, and analysis chains were used to support it?
- see what happens to the impact of a dataset when analyzed with some new method
-
- recall this is a fluid consensus, and each different wiki can be interpreted in context as the product of the particular community it comes from.

!! remember these are all just interfaces to our linked data protocol.

What we've described is a nonutopian, fully realizable path to making a scientific system that is fully negotiable through the entire theoretical-empirical loop with minor development of existing tools and minimal adjustment of scientific practices. No clouds, no journals, a little rough around the edges but collectively owned by all scientists.

We still need a little more strategy...

Credit Assignment

!!The critical anchor of the entire scientific communication system is the system of professional incentives that make it so nothing outside of a journal counts as science. Blog posts are nice and all, but they aren't *science.* One way of approaching this problem is convincing, en masse, a majority of researchers to boycott journals or value other

mediums in hiring decisions. That seems pretty unlikely for all the reasons all collective action problems are. Instead of approaching it prestige-side first, we can approach from the credit assignment side. !! make it easy for someone else to use your work and then by using it you have some verifiable record that other people like and use your stuff! !! this appeals to a much broader base of people not traditionally in the scientific value system, so they might be interested. !! also lets us value different kinds of scientific labor, like mentorship, advice, debugging, etc. without necessarily needing to gamify it. !! Deep linking and long provenance lets us see our impact on the broader scientific world, which is a much more valuable and informative than shitty journal rankings. !! people always talk about how shitty journal metrics are, and so that's an opening! !!

the work of maintaining the system can't be invisible, read & cite [Classe et al., 2017, Bowker et al., 2010]

!! essentially all questions about "changing the system of science" inevitably lead to credit assignment, but in our system it is the same as provenance. We can give credit to all work from data production, analysis tooling, technical work, theoretical work, and so on that we currently do with just author lists. brief nod to semantic publishing, though a treatment of the journal system is officially out of scope.

!! cite eLife EIC's comments on scientific credit assignment systems

Conclusion

!! summary of the system design

!! description of a new kind of scientific consensus *en toto*

Shared Governance

!! the broad and uncertain future here is how this system will be goverened and how it will be oeprated. Though we design a system that decentralizes its operation, decentralizing power is not an automatic guarantee of the technology, so we need to remember the main question here is a refocusing of our culture *along with* refocusing our technology. We need to reconceptualize what we demand from our communication systems, how much power and agency we have over them, and how we relate with other scientists.

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

!! multiplicity is in itself a form of governance, where rather than needing to canalize things into a single decision, it is possible to have

all the options exist simultaneously and let history sort them out. http://meatballwiki.org/wiki/VotingIsEvil http://meatballwiki.org/wiki/EnlargeSpace

!! what 'whoops we created a bureaucracy' teaches us is that wikis and wikilike technologies are important means of realizing multiple systems of governance and norms. see also the tyranny of structurelessness

> For everyone to have the opportunity to be involved in a given group and to participate in its activities the structure must be explicit, not implicit. The rules of decision-making must be open and available to everyone, and this can happen only if they are formalized. This is not to say that formalization of a structure of a group will destroy the informal structure. It usually doesn't. But it does hinder the informal structure from having predominant control and make available some means of attacking it if the people involved are not at least responsible to the needs of the group at large.
>
> The end of consciousness-raising leaves people with no place to go, and the lack of structure leaves them with no way of getting there. The women the movement either turn in on themselves and their sisters or seek other alternatives of action. There are few that are available. Some women just "do their own thing." This can lead to a great deal of individual creativity, much of which is useful for the movement, but it is not a viable alternative for most women and certainly does not foster a spirit of cooperative group effort. Other women drift out of the movement entirely because they don't want to develop an individual project and they have found no way of discovering, joining, or starting group projects that interest them. [Freeman, 1970]

Tactics & Strategy

!! How do we make this happen? Practical recommendations for various stakeholders

!! Some of the tactical vision for this is embedded in the structure and serial order of the piece. There is no reason that the metadata framework described here needs to be intrinsically linked to the p2p data sharing system, and there is no inherent need to first arrive at some state of quasi-standardization, but because many data standards are already in OWL or other RDF system and need some mechanism for making extensions, there is an immediate practical problem solved by implementing a linked data layer on top of a data standard and sharing system. There is little reason for a developer of an experimental library to declare a rich metadata system, but if it was possible to use it to make data output easier and make the system more powerful in the process, we then have a strong incentive.

Contrasting visions for science

!! through this text I have tried to sketch in parallel the vision of scientific practice as I see it heading now, into a platform capitalist hell, and an alternative, which is not a utopia but it is a place where we save a shitload of labor and (revisit the harms in the introduction).

The worst platform capitalist world

!! ahh huh you know what it is

What we could hope for

!! ya remake this description only less ivy and rosewaters and reintroduce some of the frustrations that might occur in the system. yno there are limitations but shit would actually genuinely be useful.

Limitations

- identity!
- interaction of p2p and linked data system – lightweight linked metadata can be reproduced more easily than massive raw data, but it needs to be possible to apply permissions and access regulation with more verifiability than just being able to access a unique tracker ID or being pointed to a UUID.

Two outstanding problems on Mastodon hint at a few open challenges to development: feed organization and the fluidity of federation formation, dissolution, and interaction.

By default, and affirmed by maybe an understandable reaction against algorithmic feed organization, Mastodon is a mostly chronological list of posts from people that you follow and that are in your host server's federated networks. While this transparency is reassuring that we aren't being microtargeted for advertising, it does make the system overwhelming to navigate, and splitting accounts multiple times to accomodate is common. A system of semantic organization is a distinct third way between algorithmic and chronological organization. Building a system that goes beyond moderator-specified category systems familiar in forums towards a sensible interface for navigating tangled concept hierarchies is an open challenge, as far as I'm aware.

An intermediate goal might be to give finer control over groups, but groups are currently a complicated question between fediverse implementations [Sta, 2021] .

!! portability of identity, necessary negotiation over partitioning of communities.

Bibliography

C2wiki - Wiki History. http://wiki.c2.com/?WikiHistory.

Meatball Wiki: EnlargeSpace.
http://meatballwiki.org/wiki/EnlargeSpace, a.

Meatball Wiki: ForkingOfOnlineCommunities.
http://meatballwiki.org/wiki/ForkingOfOnlineCommunities,
b.

Meatball Wiki: PersonalCategories.
http://meatballwiki.org/wiki/PersonalCategories, c.

Meatball Wiki: RightToLeave.
http://meatballwiki.org/wiki/RightToLeave, d.

Rfc5321 - Simple Mail Transfer Protocol.
https://datatracker.ietf.org/doc/html/rfc5321#section-3.

Solid - P2P Foundation. https://wiki.p2pfoundation.net/Solid.

Zidovudine - Patient | NIH.
https://clinicalinfo.hiv.gov/en/drugs/zidovudine/patient.

SPARQL 1.1 Federated Query. https://www.w3.org/TR/sparql11-federated-query/, March 2013.

eLife partners with Hypothes.is to advance open scholarly annotation.
https://elifesciences.org/for-the-press/7e722of6/elife-partners-with-hypothes-is-to-advance-open-scholarly-annotation, September 2016.

Elsevier and Seven Bridges receive NIH Data Commons grant for biomedical data analysis. https://www.elsevier.com/about/press-releases/archive/science-and-technology/elsevier-and-seven-bridges-receive-nih-data-commons-grant-for-biomedical-data-analysis, November 2017.

NIH Strategic Plan for Data Science. Technical report, National Institutes of Health, June 2018.

RELX Annual Report 2019, 2019.

The Pirate Bay - Archiveteam.
https://wiki.archiveteam.org/index.php?title=The_Pirate_Bay&oldid=45467,
September 2020.

RELX Annual Report 2020. page 196, 2020.

STRIDES Initiative Success Story: University of
Michigan TOPMed | Data Science at NIH.
https://web.archive.org/web/20210324024612/https://datascience.nih.gov/strides-
initiative-success-story-university-michigan-topmed, October
2020.

AWS announces AWS Healthcare Accelerator for startups in the
public sector. https://aws.amazon.com/blogs/publicsector/aws-
announces-healthcare-accelerator-program-startups-public-sector/,
June 2021.

Criticism of Amazon. *Wikipedia*, September 2021.

ROBOKOP - CoVar.
https://web.archive.org/web/20211006030919/https://covar.com/case-
study/robokop/, October 2021.

RePORT ⟩ RePORTER "Biomedical Data Translator".
https://reporter.nih.gov/search/kDJ97zGUFEaIBIltUmyd_Q/projects?sort_field=FiscalYear&sort_order=desc,
October 2021.

Standardizing on ActivityPub Groups - ActivityPub.
https://socialhub.activitypub.rocks/t/standardizing-on-
activitypub-groups/1984, August 2021.

Wikipedia:Flow. *Wikipedia*, June 2021.

Larry F. Abbott, Dora E. Angelaki, Matteo Carandini, Anne K.
Churchland, Yang Dan, Peter Dayan, Sophie Deneve, Ila Fi-
ete, Surya Ganguli, Kenneth D. Harris, Michael Häusser, Sonja
Hofer, Peter E. Latham, Zachary F. Mainen, Thomas Mrsic-
Flogel, Liam Paninski, Jonathan W. Pillow, Alexandre Pouget,
Karel Svoboda, Ilana B. Witten, and Anthony M. Zador. An In-
ternational Laboratory for Systems and Computational Neuro-
science. *Neuron*, 96(6):1213–1218, December 2017. ISSN 0896-6273.
https://doi.org/10.1016/j.neuron.2017.12.013.

Francesc Alted and Mercedes Fernández-Alonso. PyTables: Processing
And Analyzing Extremely Large Amounts Of Data In Python. In
PyCon2003, page 9, April 2003.

Stephen Altschul, Barry Demchak, Richard Durbin, Robert Gentleman, Martin Krzywinski, Heng Li, Anton Nekrutenko, James Robinson, Wayne Rasband, James Taylor, and Cole Trapnell. The anatomy of successful computational biology software. *Nature Biotechnology*, 31(10):894–897, October 2013. ISSN 1546-1696. https://doi.org/10.1038/nbt.2721.

Andrey Andreev, Tom Morrell, Kristin Briney, Sandra Gesing, and Uri Manor. Biologists need modern data infrastructure on campus. *arXiv:2108.07631 [q-bio]*, August 2021.

STEPHEN R. BARLEY and BETH A. BECHKY. In the Backrooms of Science: The Work of Technicians in Science Labs. *Work and Occupations*, 21(1):85–126, February 1994. ISSN 0730-8884. https://doi.org/10.1177/073088849402100104.

Jonathan Robert Basamanowicz. *Release Groups and Digital Copyright Piracy*. Thesis, Arts & Social Sciences: School of Criminology, May 2011.

Juan Benet. IPFS - Content Addressed, Versioned, P2P File System. *arXiv:1407.3561 [cs]*, July 2014.

Tim Berners-Lee. Principles of Design. https://www.w3.org/DesignIssues/Principles.html#Decentrali, 1998.

Jayanti Bhandari Neupane, Ram P. Neupane, Yuheng Luo, Wesley Y. Yoshida, Rui Sun, and Philip G. Williams. Characterization of Leptazolines A–D, Polar Oxazolines from the Cyanobacterium Leptolyngbya sp., Reveals a Glitch with the "Willoughby–Hoye" Scripts for Calculating NMR Chemical Shifts. *Organic Letters*, 21 (20):8449–8453, October 2019. ISSN 1523-7060. https://doi.org/10.1021/acs.orglett.9b03216.

Sam Biddle. LexisNexis to Provide Giant Database of Personal Information to ICE, April 2021.

Matthew J. Bietz and Charlotte P. Lee. Collaboration in Metagenomics: Sequence Databases and the Organization of Scientific Work. In Ina Wagner, Hilda Tellioğlu, Ellen Balka, Carla Simone, and Luigina Ciolfi, editors, *ECSCW 2009*, pages 243–262, London, 2009. Springer. ISBN 978-1-84882-854-4. https://doi.org/10.1007/978-1-84882-854-4_15.

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, pages 901–910, New York, NY, USA, February 2012. Association for Computing Machinery. ISBN 978-1-4503-1086-4. https://doi.org/10.1145/2145204.2145339.

Elisabeth M. Bik, Arturo Casadevall, and Ferric C. Fang. The Prevalence of Inappropriate Image Duplication in Biomedical Research Publications. *mBio*, 7(3):e00809–16, June 2016. https://doi.org/10.1128/mBio.00809-16.

Angela Bonifati, Panos K. Chrysanthis, Aris M. Ouksel, and Kai-Uwe Sattler. Distributed databases and peer-to-peer databases: Past and present. *ACM SIGMOD Record*, 37(1):5–11, March 2008. ISSN 0163-5808. https://doi.org/10.1145/1374780.1374781.

Murray Bookchin. A Note on Affinity Groups. page 2, 1969.

Geoffrey C. Bowker, Karen Baker, Florence Millerand, and David Ribes. Toward Information Infrastructure Studies: Ways of Knowing in a Networked Environment. In Jeremy Hunsinger, Lisbeth Klastrup, and Matthew Allen, editors, *International Handbook of Internet Research*, pages 97–117. Springer Netherlands, Dordrecht, 2010. ISBN 978-1-4020-9789-8. https://doi.org/10.1007/978-1-4020-9789-8_5.

Björn Brembs. Algorithmic employment decisions in academia?, September 2021.

Björn Brembs, Philippe Huneman, Felix Schönbrodt, Gustav Nilsonne, Toma Susi, Renke Siems, Pandelis Perakakis, Varvara Trachana, Lai Ma, and Sara Rodriguez-Cuadrado. Replacing academic journals. September 2021. https://doi.org/10.5281/zenodo.5526635.

Jordan Bross. *Community, Collaboration and Contribution: Evaluating a BitTorrent Tracker as a Digital Library.* M.S. in Library Science, UNC Chapel Hill, December 2013.

Richard Bruskiewich, Deepak, Sierra Moxon, Chris Mungall, Harold Solbrig, cbizon, Matthew Brush, Kent Shefchek, Lance Hannestad, YaphetKG, Nomi Harris, bbopjenkins, diatomsRcool, Patrick Wang, Jim Balhoff, Kevin Schaper, JIWEN XIN, Phil Owen, Gregory Stupp, JervenBolleman, The Gitter Badger, Vincent Emonet, and vdancik. Biolink/biolink-model: 2.2.5. Zenodo, September 2021.

Stephen Buranyi. Is the staggeringly profitable business of scientific publishing bad for science? *The Guardian*, June 2017. ISSN 0261-3077.

Susanne Busse, Ralf-Detlef Kutsche, Ulf Leser, and Herbert Weber. Federated Information Systems: Concepts, Terminology and Architectures. page 40, 1999.

Brian Butler, Elisabeth Joyce, and Jacqueline Pike. Don't look now, but we've created a bureaucracy: The nature and roles of policies and rules in wikipedia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1101–1110, New York, NY, USA, April 2008. Association for Computing Machinery. ISBN 978-1-60558-011-1. https://doi.org/10.1145/1357054.1357227.

Sarven Capadisli, Tim Berners-Lee, Ruben Verborgh, Kjetil Kjernsmo, Justin Bingham, and Dmitri Zagidulin. Solid Protocol. https://solidproject.org/TR/protocol, December 2020.

Brian E. Carpenter. RFC 1958 - Architectural Principles of the Internet. https://tools.ietf.org/html/rfc1958, June 1996.

Werner Ceusters and Barry Smith. Foundations for a realist ontology of mental disease. *Journal of Biomedical Semantics*, 1(1):10, December 2010. ISSN 2041-1480. https://doi.org/10.1186/2041-1480-1-10.

Adam S. Charles, Benjamin Falk, Nicholas Turner, Talmo D. Pereira, Daniel Tward, Benjamin D. Pedigo, Jaewon Chung, Randal Burns, Satrajit S. Ghosh, Justus M. Kebschull, William Silversmith, and Joshua T. Vogelstein. Toward Community-Driven Big Open Brain Science: Open Big Data and Tools for Structure, Function, and Genetics. *Annual Review of Neuroscience*, 43:441–464, July 2020. ISSN 1545-4126. https://doi.org/10.1146/annurev-neuro-100119-110036.

X. Chen, X. Chu, and Z. Li. Improving Sustainability of Private P2P Communities. In *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–6, July 2011. https://doi.org/10.1109/ICCCN.2011.6005944.

Johan S. G. Chu and James A. Evans. Slowed canonical progress in large fields of science. *Proceedings of the National Academy of Sciences*, 118(41), October 2021. ISSN 0027-8424, 1091-6490. https://doi.org/10.1073/pnas.2021636118.

D. Clark. The design philosophy of the DARPA internet protocols. In *Symposium Proceedings on Communications Architectures and Protocols*, SIGCOMM '88, pages 106–114, New York, NY, USA, August 1988. Association for Computing Machinery. ISBN 978-0-89791-279-2. https://doi.org/10.1145/52324.52336.

David Clark. A Cloudy Crystal Ball - Visions of the Future. In *Proceedings of the Twenty-Fourth Internet Engineering Task Force*, pages 539–543, July 1992.

Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability Berkeley, CA, USA, July 25–26, 2000 Proceedings*, Lecture Notes in Computer Science, pages 46–66. Springer, Berlin, Heidelberg, 2001. ISBN 978-3-540-44702-3. https://doi.org/10.1007/3-540-44702-4_4.

Tadeu Classe, Regina Braga, José Maria N. David, Fernanda Campos, and Wagner Arbex. A Distributed Infrastructure to Support Scientific Experiments. *Journal of Grid Computing*, 15(4):475–500, December 2017. ISSN 1572-9184. https://doi.org/10.1007/s10723-017-9401-7.

Kevin R. Coffey, Russell G. Marx, and John F. Neumaier. DeepSqueak: A deep learning-based system for detection and analysis of ultrasonic vocalizations. *Neuropsychopharmacology*, 44(5):859–868, April 2019. ISSN 1740-634X. https://doi.org/10.1038/s41386-018-0303-6.

Bram Cohen. The BitTorrent Protocol Specification, February 2017.

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, pages 1–2, New York, NY, USA, July 2014. Association for Computing Machinery. ISBN 978-1-4503-2893-7. https://doi.org/10.1145/2616498.2616528.

Michican Civil Rights Commission. The Flint Water Crisis: Systemic Racism Through the Lens of Flint. Technical report, Michican Civil Rights Commission, February 2017.

The Biomedical Data Translator Consortium. The Biomedical Data Translator Program: Conception, Culture, and Community. *Clinical and Translational Science*, 12(2):91–94, 2019a. ISSN 1752-8062. https://doi.org/10.1111/cts.12592.

The Biomedical Data Translator Consortium. Toward A Universal Biomedical Data Translator. *Clinical and Translational Science*, 12(2):86–90, 2019b. ISSN 1752-8062. https://doi.org/10.1111/cts.12591.

csillag. Fuzzy Anchoring, April 2013.

Robert E. Cummings. "WhatWas a Wiki, and Why Do I Care? A Short and Usable History of Wikis" - Wildwiki. https://web.archive.org/web/20090921042301/http://www.wildwiki.net/mediawiki/index.php?title=%E2%80%9CWhat September 2009.

Marija Djokic-Petrovic, Vladimir Cvjetkovic, Jeremy Yang, Marko Zivanovic, and David J. Wild. PIBAS FedSPARQL: A web-based platform for integration and exploration of bioinformatics datasets. *Journal of Biomedical Semantics*, 8(1):42, September 2017. ISSN 2041-1480. https://doi.org/10.1186/s13326-017-0151-z.

Ian Dunham. *What.CD: A Legacy of Sharing*. PhD thesis, Rutgers University - School of Graduate Studies, 2018.

dwhly. bioRxiv Selects Hypothesis to Enable Annotation on Preprints, September 2017.

Anders Eklund, Thomas E. Nichols, and Hans Knutsson. Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates. *Proceedings of the National Academy of Sciences*, 113 (28):7900–7905, July 2016. ISSN 0027-8424, 1091-6490. https://doi.org/10.1073/pnas.1602413113.

Elsevier. 360° advertising solutions | Advertising | Advertisers. https://www.elsevier.com/advertising-reprints-supplements/advertising, a.

Elsevier. Drug design optimization. https://www.elsevier.com/solutions/professional-services/drug-design-optimization, b.

Elsevier. Topic Prominence in Science - Scival | Elsevier Solutions. https://www.elsevier.com/solutions/scival/features/topic-prominence-in-science, a.

Elsevier. Topic Prominence in Science - Scival | Elsevier Solutions. https://www.elsevier.com/solutions/scival/features/topic-prominence-in-science, b.

Benedikt Fecher, Sascha Friesike, Marcel Hebing, and Stephanie Linek. A reputation economy: How individual reward considerations trump systemic arguments for open access to data. *Palgrave Communications*, 3(1):1–10, June 2017. ISSN 2055-1045. https://doi.org/10.1057/palcomms.2017.51.

Stacia Fleisher. Other Transaction Award Policy Guide - Biomedical Data Translator Program. page 38, September 2019.

Jo Freeman. The Tyranny of Stucturelessness. https://www.jofreeman.com/joreen/tyranny.htm, 1970.

Prateek Goel, Adam J Johs, Manil Shrestha, and Rosina O Weber. Explanation Container in Case-Based Biomedical Question-Answering. page 10, September 2021.

Sten Grillner, Nancy Ip, Christof Koch, Walter Koroshetz, Hideyuki Okano, Miri Polachek, Mu-ming Poo, and Terrence J. Sejnowski. Worldwide initiatives to advance brain research. *Nature Neuroscience*, 19(9):1118–1122, September 2016. ISSN 1546-1726. https://doi.org/10.1038/nn.4371.

Thomas Grote and Philipp Berens. On the ethics of algorithmic decision-making in healthcare. *Journal of Medical Ethics*, 46(3):205–211, March 2020. ISSN 0306-6800, 1473-4257. https://doi.org/10.1136/medethics-2019-105586.

Jonathan Grudin. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM*, 37(1):92–105, January 1994. ISSN 0001-0782. https://doi.org/10.1145/175222.175230.

Olivia Guest and Andrea E. Martin. How Computational Modeling Can Force Theory Building in Psychological Science. *Perspectives on Psychological Science*, page 1745691620970585, January 2021. ISSN 1745-6916. https://doi.org/10.1177/1745691620970585.

Melissa A Haendel. A Common Dialect for Infrastructure and Services in Translator. https://reporter.nih.gov/project-details/10330632, February 2021.

Ruth Hailu. NIH-funded project aims to build a 'Google' for biomedical data, July 2019.

Yaroslav O. Halchenko, Kyle Meyer, Benjamin Poldrack, Debanjum Singh Solanky, Adina S. Wagner, Jason Gors, Dave MacFarlane, Dorian Pustina, Vanessa Sochat, Satrajit S. Ghosh, Christian Mönch, Christopher J. Markiewicz, Laura Waite, Ilya Shlyakhter, Alejandro de la Vega, Soichi Hayashi, Christian Olaf Häusler, Jean-Baptiste Poline, Tobias Kadelka, Kusti Skytén, Dorota Jarecka, David Kennedy, Ted Strauss, Matt Cieslak, Peter Vavra, Horea-Ioan Ioanas, Robin Schneider, Mika Pflüger, James V. Haxby, Simon B. Eickhoff, and Michael Hanke. DataLad: Distributed system for joint management of code, data, and their relationship. *Journal of Open Source Software*, 6(63):3262, July 2021. ISSN 2475-9066. https://doi.org/10.21105/joss.03262.

Aaron Halfaker, R. Stuart Geiger, Jonathan T. Morgan, and John Riedl. The Rise and Decline of an Open Collaboration System: How Wikipedia's Reaction to Popularity Is Causing Its Decline. *American Behavioral Scientist*, 57(5):664–688, May 2013. ISSN 0002-7642, 1552-3381. https://doi.org/10.1177/0002764212469365.

Terry Hancock. OpenOffice.org is Dead, Long Live LibreOffice – or, The Freedom to Fork. http://freesoftwaremagazine.com/articles/openoffice_org_dead_long_live_libreoffice/, October 2010.

Michael Hanke, Franco Pestilli, Adina S. Wagner, Christopher J. Markiewicz, Jean-Baptiste Poline, and Yaroslav O. Halchenko. In defense of decentralized research data management. *Neuroforum*, 27(1):17–25, February 2021. ISSN 1868-856X. https://doi.org/10.1515/nf-2020-0037.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825): 357–362, September 2020. ISSN 1476-4687. https://doi.org/10.1038/s41586-020-2649-2.

Ali Hasnain, Qaiser Mehmood, Syeda Sana e Zainab, Muhammad Saleem, Claude Warren, Durre Zehra, Stefan Decker, and Dietrich Rebholz-Schuhmann. BioFed: Federated query processing over life sciences linked open data. *Journal of Biomedical Semantics*, 8 (1):13, March 2017. ISSN 2041-1480. https://doi.org/10.1186/s13326-017-0118-0.

James Heathers. The Real Scandal About Ivermectin. https://www.theatlantic.com/science/archive/2021/10/ivermectin-research-problems/620473/, October 2021.

heatherstaines. Preprint Services Gather to Explore an Annotated Future, February 2018.

Dennis Heimbigner and Dennis McLeod. A federated architecture for information management. *ACM Transactions on Information Systems*, 3(3):253–278, July 1985. ISSN 1046-8188. https://doi.org/10.1145/4229.4233.

Benjamin Mako Hill and Aaron Shaw. Wikipedia and the End of Open Collaboration? *Wikipedia @ 20*, page 12, 2019.

Michael A. Hiltzik. Taming the Wild, Wild Web. https://web.archive.org/web/20010801142640/https://www.latimes.com/business/la-072601netarch.story, July 2001.

Sameer Hinduja. Deindividuation and Internet Software Piracy. *CyberPsychology & Behavior*, 11(4):391–398, August 2008. ISSN 1094-9313. https://doi.org/10.1089/cpb.2007.0048.

Matthew Hodgson. Gitter now speaks Matrix! https://matrix.org/blog/2020/12/07/gitter-now-speaks-matrix, December 2020a.

Matthew Hodgson. Welcoming Gitter to Matrix! https://matrix.org/blog/2020/09/30/welcoming-gitter-to-matrix, September 2020b.

John Hoffman and DeHackEd. HTTP-Based Seeding Specification. http://www.bittornado.com/docs/webseed-spec.txt.

James Howison and Julia Bullard. Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature. *Journal of the Association for Information Science and Technology*, 67(9):2137–2155, 2016. ISSN 2330-1643. https://doi.org/10.1002/asi.23538.

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, pages 459–470, New York, NY, USA, February 2013. Association for Computing Machinery. ISBN 978-1-4503-1331-5. https://doi.org/10.1145/2441776.2441828.

Kyle R. Hukezalie, Naresh R. Thumati, Hélène C. F. Côté, and Judy M. Y. Wong. In Vitro and Ex Vivo Inhibition of Human Telomerase by Anti-HIV Nucleoside Reverse Transcriptase Inhibitors (NRTIs) but Not by Non-NRTIs. *PLoS ONE*, 7(11):e47505, November 2012. ISSN 1932-6203. https://doi.org/10.1371/journal.pone.0047505.

Richard Hull. Managing semantic heterogeneity in databases: A theoretical prospective. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '97, pages 51–61, New York, NY, USA, May 1997. Association for Computing Machinery. ISBN 978-0-89791-910-4. https://doi.org/10.1145/263661.263668.

Iain. Freebase is dead, long live Freebase, May 2019.

H. J. Jackson. *Marginalia: Readers Writing in Books*. Yale University Press, January 2001. ISBN 978-0-300-09720-7.

Adele Lu Jia, Xiaowei Chen, Xiaowen Chu, Johan A. Pouwelse, and Dick H. J. Epema. How to Survive and Thrive in a Private BitTorrent Community. In Davide Frey, Michel Raynal, Saswati Sarkar, Rudrapatna K. Shyamasundar, and Prasun Sinha, editors, *Distributed Computing and Networking*, Lecture Notes in Computer Science, pages 270–284, Berlin, Heidelberg, 2013. Springer. ISBN 978-3-642-35668-1. https://doi.org/10.1007/978-3-642-35668-1_19.

Brewster Kahle. Over 1,000,000 Torrents of Downloadable Books, Music, and Movies, August 2012.

Gary A Kane, Gonçalo Lopes, Jonny L Saunders, Alexander Mathis, and Mackenzie W Mathis. Real-time, low-latency closed-loop feedback using markerless posture tracking. *eLife*, 9:e61909, December 2020. ISSN 2050-084X. https://doi.org/10.7554/eLife.61909.

Ian A. Kash, John K. Lai, Haoqi Zhang, and Aviv Zohar. Economics of BitTorrent communities. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 221–230, New York, NY, USA, April 2012. Association for Computing Machinery. ISBN 978-1-4503-1229-5. https://doi.org/10.1145/2187836.2187867.

Vipul Kashyap and Amit Sheth. Semantic and schematic similarities between database objects:a context-based approach. *The VLDB Journal*, 5(4):276–304, December 1996. ISSN 0949-877X. https://doi.org/10.1007/s007780050029.

Eddie Kim. After 15 Years, the Pirate Bay Still Can't Be Killed. https://melmagazine.com/en-us/story/after-15-years-the-pirate-bay-still-cant-be-killed, May 2019.

Naomi Klein. Were the DC and Seattle Protests Unfocused?, July 2001.

Christof Koch and Allan Jones. Big Science, Team Science, and Open Science for Neuroscience. *Neuron*, 92(3):612–616, November 2016. ISSN 0896-6273. https://doi.org/10.1016/j.neuron.2016.10.019.

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)*, pages 1–10, Delft, Netherlands, August 2010. IEEE. ISBN 978-1-4244-7140-9. https://doi.org/10.1109/P2P.2010.5569963.

Sudhir Kumar and Joel Dudley. Bioinformatics software for biologists in the genomics era. *Bioinformatics*, 23(14):1713–1717, July 2007. ISSN 1367-4803. https://doi.org/10.1093/bioinformatics/btm239.

The International Brain Laboratory, Valeria Aguillon-Rodriguez, Dora E. Angelaki, Hannah M. Bayer, Niccolò Bonacchi, Matteo Carandini, Fanny Cazettes, Gaelle A. Chapuis, Anne K. Churchland, Yang Dan, Eric E. J. Dewitt, Mayo Faulkner, Hamish Forrest, Laura M. Haetzel, Michael Hausser, Sonja B. Hofer, Fei Hu, Anup Khanal, Christopher S. Krasniak, Inês Laranjeira, Zachary F. Mainen, Guido T. Meijer, Nathaniel J. Miska, Thomas D. Mrsic-Flogel, Masayoshi Murakami, Jean-Paul Noel, Alejandro Pan-Vazquez, Cyrille Rossant, Joshua I. Sanders, Karolina Z. Socha, Rebecca Terry, Anne E. Urai, Hernando M. Vergara, Miles J. Wells, Christian J. Wilson, Ilana B. Witten, Lauren E. Wool, and Anthony Zador. Standardized and reproducible measurement of decision-making in mice. *bioRxiv*, page 2020.01.17.909838, October 2020a. https://doi.org/10.1101/2020.01.17.909838.

The International Brain Laboratory, Niccolò Bonacchi, Gaelle Chapuis, Anne Churchland, Kenneth D. Harris, Max Hunter, Cyrille Rossant, Maho Sasaki, Shan Shen, Nicholas A. Steinmetz, Edgar Y. Walker, Olivier Winter, and Miles Wells. Data architecture for a large-scale neuroscience collaboration. *bioRxiv*, page 827873, February 2020b. https://doi.org/10.1101/827873.

Morgan G. I. Langille and Jonathan A. Eisen. BioTorrents: A File Sharing Service for Scientific Data. *PLoS ONE*, 5(4), April 2010. ISSN 1932-6203. https://doi.org/10.1371/journal.pone.0010071.

Rebekah Larsen. The Political Nature of TCP/IP. page 56, 2012.

Ed S. Lein, Michael J. Hawrylycz, Nancy Ao, Mikael Ayres, Amy Bensinger, Amy Bernard, Andrew F. Boe, Mark S. Boguski, Kevin S. Brockway, Emi J. Byrnes, Lin Chen, Li Chen, Tsuey-Ming Chen, Mei Chi Chin, Jimmy Chong, Brian E. Crook, Aneta Czaplinska, Chinh N. Dang, Suvro Datta, Nick R. Dee, Aimee L. Desaki, Tsega Desta, Ellen Diep, Tim A. Dolbeare, Matthew J. Donelan, Hong-Wei Dong, Jennifer G. Dougherty, Ben J. Duncan, Amanda J. Ebbert, Gregor Eichele, Lili K. Estin, Casey Faber, Benjamin A. Facer, Rick Fields, Shanna R. Fischer, Tim P. Fliss, Cliff Frensley, Sabrina N. Gates, Katie J. Glattfelder, Kevin R. Halverson, Matthew R. Hart, John G. Hohmann, Maureen P. Howell, Darren P. Jeung, Rebecca A. Johnson, Patrick T. Karr, Reena Kawal, Jolene M. Kidney, Rachel H. Knapik, Chihchau L. Kuan, James H. Lake, Annabel R. Laramee, Kirk D. Larsen, Christopher Lau, Tracy A. Lemon, Agnes J. Liang, Ying Liu, Lon T. Luong, Jesse Michaels, Judith J. Morgan, Rebecca J. Morgan, Marty T. Mortrud, Nerick F. Mosqueda, Lydia L. Ng, Randy Ng, Geralyn J. Orta, Caroline C. Overly, Tu H. Pak, Sheana E. Parry, Sayan D. Pathak, Owen C. Pearson, Ralph B.

Puchalski, Zackery L. Riley, Hannah R. Rockett, Stephen A. Rowland, Joshua J. Royall, Marcos J. Ruiz, Nadia R. Sarno, Katherine Schaffnit, Nadiya V. Shapovalova, Taz Sivisay, Clifford R. Slaughterbeck, Simon C. Smith, Kimberly A. Smith, Bryan I. Smith, Andy J. Sodt, Nick N. Stewart, Kenda-Ruth Stumpf, Susan M. Sunkin, Madhavi Sutram, Angelene Tam, Carey D. Teemer, Christina Thaller, Carol L. Thompson, Lee R. Varnam, Axel Visel, Ray M. Whitlock, Paul E. Wohnoutka, Crissa K. Wolkey, Victoria Y. Wong, Matthew Wood, Murat B. Yaylaoglu, Rob C. Young, Brian L. Youngstrom, Xu Feng Yuan, Bin Zhang, Theresa A. Zwingman, and Allan R. Jones. Genome-wide atlas of gene expression in the adult mouse brain. *Nature*, 445(7124):168–176, January 2007. ISSN 1476-4687. https://doi.org/10.1038/nature05453.

Rachel Lerman. Amazon built its own health-care service for employees. Now it's selling it to other companies. *Washington Post*, March 2021. ISSN 0190-8286.

Bo Leuf and Ward Cunningham. *The Wiki Way : Quick Collaboration on the Web*. Boston : Addison-Wesley, 2001. ISBN 978-0-201-71499-9.

Witold Litwin, Leo Mark, and Nick Roussopoulos. Interoperability of multiple autonomous databases. *ACM Computing Surveys*, 22(3): 267–293, September 1990. ISSN 0360-0300. https://doi.org/10.1145/96602.96608.

Z. Liu, P. Dhungel, D. Wu, C. Zhang, and K. W. Ross. Understanding and Improving Ratio Incentives in Private Communities. In *2010 IEEE 30th International Conference on Distributed Computing Systems*, pages 610–621, June 2010. https://doi.org/10.1109/ICDCS.2010.90.

Gonçalo Lopes and Patricia Monteiro. New Open-Source Tools: Using Bonsai for Behavioral Tracking and Closed-Loop Experiments. *Frontiers in Behavioral Neuroscience*, 15:53, 2021. ISSN 1662-5153. https://doi.org/10.3389/fnbeh.2021.647640.

Gonçalo Lopes, Niccolò Bonacchi, João Frazão, Joana P. Neto, Bassam V. Atallah, Sofia Soares, Luís Moreira, Sara Matias, Pavel M. Itskov, Patrícia A. Correia, Roberto E. Medina, Lorenza Calcaterra, Elena Dreosti, Joseph J. Paton, and Adam R. Kampff. Bonsai: An event-based framework for processing and controlling data streams. *Frontiers in Neuroinformatics*, 9, 2015a. ISSN 1662-5196. https://doi.org/10.3389/fninf.2015.00007.

Gonçalo Lopes, Niccolò Bonacchi, João Frazão, Joana P. Neto, Bassam V. Atallah, Sofia Soares, Luís Moreira, Sara Matias, Pavel M.

Itskov, Patrícia A. Correia, Roberto E. Medina, Lorenza Calcaterra, Elena Dreosti, Joseph J. Paton, and Adam R. Kampff. Bonsai: An event-based framework for processing and controlling data streams. *Frontiers in Neuroinformatics*, 9:7, 2015b. ISSN 1662-5196. https://doi.org/10.3389/fninf.2015.00007.

Ian MacInnes. Compatibility standards and monopoly incentives: The impact of service-based software licensing. *International Journal of Services and Standards*, 1(3):255–270, January 2005. ISSN 1740-8849. https://doi.org/10.1504/IJSS.2005.005799.

Zachary F. Mainen, Michael Häusser, and Alexandre Pouget. A better way to crack the brain. *Nature News*, 539(7628):159, November 2016. https://doi.org/10.1038/539159a.

Serghei Mangul, Lana S. Martin, Eleazar Eskin, and Ran Blekhman. Improving the usability and archival stability of bioinformatics software. *Genome Biology*, 20(1):47, February 2019. ISSN 1474-760X. https://doi.org/10.1186/s13059-019-1649-8.

John Markoff. Tomorrow, the World Wide Web!;Microsoft, the PC King, Wants to Reign Over the Internet. *The New York Times*, July 1996. ISSN 0362-4331.

Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21(9):1281–1289, September 2018. ISSN 1546-1726. https://doi.org/10.1038/s41593-018-0209-y.

Paul E. Meehl. Theoretical risks and tabular asterisks: Sir Karl, Sir Ronald, and the slow progress of soft psychology. *Journal of Consulting and Clinical Psychology*, 46(4):806–834, 1978. ISSN 1939-2117(Electronic),0022-006X(Print). https://doi.org/10.1037/0022-006X.46.4.806.

M Meulpolder, L D'Acunto, M Capota, M Wojciechowski, J A Pouwelse, D H J Epema, and H J Sips. Public and private BitTorrent communities: A measurement study. page 5.

Greg Miller. A Scientist's Nightmare: Software Problem Leads to Five Retractions. *Science*, 314(5807):1856–1857, December 2006. ISSN 0036-8075, 1095-9203. https://doi.org/10.1126/science.314.5807.1856.

Philip Mirowski. The future(s) of open science. *Social Studies of Science*, 48(2):171–203, April 2018. ISSN 0306-3127. https://doi.org/10.1177/0306312718772086.

nateangell. Announcing TRiP: Transparent Review in Preprints, Powered by Hypothesis Annotation, September 2019.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, October 2019. https://doi.org/10.1126/science.aax2342.

Maxwell Ogden. Dat - Distributed Dataset Synchronization And Versioning. Preprint, Open Science Framework, January 2017.

Marius Pachitariu, Nicholas Steinmetz, Shabnam Kadir, Matteo Carandini, and Harris Kenneth D. Kilosort: Realtime spike-sorting for extracellular electrophysiology with hundreds of channels. Article; https://web.archive.org/web/20211015215729/https://www.biorxiv.org/content/10.1101/061481v1, Cold Spring Harbor Laboratory, June 2016.

Sean B. Palmer. Ditching the Semantic Web? http://inamidst.com/whits/2008/ditching, March 2008.

Trishan Panch, Heather Mattie, and Rifat Atun. Artificial intelligence and algorithmic bias: Implications for health systems. *Journal of Global Health*, 9(2):020318, November 2019a. ISSN 2047-2978. https://doi.org/10.7189/jogh.09.020318.

Trishan Panch, Heather Mattie, and Leo Anthony Celi. The "inconvenient truth" about AI in healthcare. *npj Digital Medicine*, 2(1): 1–3, August 2019b. ISSN 2398-6352. https://doi.org/10.1038/s41746-019-0155-4.

Constantinos Patsakis and Fran Casino. Hydras and IPFS: A Decentralised Playground for Malware. *International Journal of Information Security*, 18(6):787–799, December 2019. ISSN 1615-5262, 1615-5270. https://doi.org/10.1007/s10207-019-00443-0.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research*, 12(null):2825–2830, November 2011. ISSN 1532-4435.

Giuseppe Pirrò, Domenico Talia, and Paolo Trunfio. A DHT-based semantic overlay network for service discovery. *Future Generation Computer Systems*, 28(4):689–707, April 2012. ISSN 0167-739X. https://doi.org/10.1016/j.future.2011.11.007.

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, pages 359–367, New York, NY, USA, June 2017. Association for Computing Machinery. ISBN 978-1-4503-4896-6. https://doi.org/10.1145/3091478.3091505.

Charleeze Ponzi. Is science a pyramid scheme? The correlation between an author's position in the academic hierarchy and her scientific output per year, January 2020.

Corey Quinn. You Can't Trust Amazon When It Feels Threatened. https://www.lastweekinaws.com/blog/you-cant-trust-amazon-when-it-feels-threatened/, March 2021.

A Ram, Clair A Kronk, Jacob R Eleazer, Joseph L Goulet, Cynthia A Brandt, and Karen H Wang. Transphobia, encoded: An examination of trans-specific terminology in SNOMED CT and ICD-10-CM. *Journal of the American Medical Informatics Association*, (ocab200), September 2021. ISSN 1527-974X. https://doi.org/10.1093/jamia/ocab200.

Dave Randall, Rob Procter, Yuwei Lin, Meik Poschen, Wes Sharrock, and Robert Stevens. Distributed ontology building as practical work. *International Journal of Human-Computer Studies*, 69(4):220–233, April 2011. ISSN 1071-5819. https://doi.org/10.1016/j.ijhcs.2010.12.011.

R. Todd Reilly. NIH STRIDES Initiative, January 2021.

David Ribes and Thomas Finholt. The Long Now of Technology Infrastructure: Articulating Tensions in Development. *Journal of the Association for Information Systems*, 10(5):375–398, May 2009. ISSN 15369323. https://doi.org/10.17705/1jais.00199.

Janko Roettgers. The Pirate Bay: Distributing the World's Entertainment for $3,000 a Month. https://gigaom.com/2009/07/19/the-pirate-bay-distributing-the-worlds-entertainment-for-3000-a-month/, July 2009.

Jody Rosen. The Day the Music Burned. *The New York Times*, June 2019. ISSN 0362-4331.

Dario Rossi, Guilhem Pujol, Xiao Wang, and Fabien Mathieu. Peeking through the BitTorrent Seedbox Hosting Ecosystem. In Alberto Dainotti, Anirban Mahanti, and Steve Uhlig, editors, *Traffic Monitoring and Analysis*, Lecture Notes in Computer Science, pages 115–126, Berlin, Heidelberg, 2014. Springer. ISBN 978-3-642-54999-1. https://doi.org/10.1007/978-3-642-54999-1_10.

Oliver Rübel, Andrew Tritt, Benjamin Dichter, Thomas Braun, Nicholas Cain, Nathan Clack, Thomas J. Davidson, Max Dougherty, Jean-Christophe Fillion-Robin, Nile Graddis, Michael Grauer, Justin T. Kiggins, Lawrence Niu, Doruk Ozturk, William Schroeder, Ivan Soltesz, Friedrich T. Sommer, Karel Svoboda, Ng Lydia, Loren M. Frank, and Kristofer Bouchard. NWB:N 2.0: An Accessible Data Standard for Neurophysiology. *bioRxiv*, page 523035, January 2019. https://doi.org/10.1101/523035.

Oliver Rübel, Andrew Tritt, Ryan Ly, Benjamin K. Dichter, Satrajit Ghosh, Lawrence Niu, Ivan Soltesz, Karel Svoboda, Loren Frank, and Kristofer E. Bouchard. The Neurodata Without Borders ecosystem for neurophysiological data science, March 2021.

Andrei Vlad Sambra, Essam Mansour, Sandro Hawke, Maged Zereba, Nicola Greco, Abdurrahman Ghanem, Dmitri Zagidulin, Ashraf Aboulnaga, and Tim Berners-Lee. Solid: A Platform for Decentralized Social Applications Based on Linked Data. *MIT CSAIL & Qatar Computing Research Institute, Tech. Rep.*, page 16, 2016.

Jonny L. Saunders and Michael Wehr. Autopilot: Automating behavioral experiments with lots of Raspberry Pis. *bioRxiv*, page 807693, October 2019. https://doi.org/10.1101/807693.

Jodi Schneider, Alexandre Passant, and John G. Breslin. Understanding and improving Wikipedia article discussion spaces: 2011 ACM Symposium. pages 808–813, 2011. https://doi.org/10.1145/1982185.1982358.

Jason Scott. Geocities Torrent Update, December 2010.

Helen Shen. Meet this super-spotter of duplicated images in science papers. *Nature*, 581(7807):132–136, May 2020. https://doi.org/10.1038/d41586-020-01363-z.

Xuemin Shen, Heather Yu, John Buford, and Mursalin Akon. *Handbook of Peer-to-Peer Networking*. Springer Science & Business Media, March 2010. ISBN 978-0-387-09751-0.

Amit P. Sheth and James A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, September 1990. ISSN 0360-0300. https://doi.org/10.1145/96602.96604.

Ana Claudia Sima, Tarcisio Mendes de Farias, Erich Zbinden, Maria Anisimova, Manuel Gil, Heinz Stockinger, Kurt Stockinger, Marc Robinson-Rechavi, and Christophe Dessimoz. Enabling semantic queries across federated bioinformatics databases. *Database*, 2019

(baz106), January 2019. ISSN 1758-0463. https://doi.org/10.1093/database/baz106.

James M Snell and Evan Prodromou. Activity Streams 2.0. https://www.w3.org/TR/activitystreams-core/, May 2017.

David A. W. Soergel. Rampant software errors may undermine scientific results. *F1000Research*, 3, July 2015. ISSN 2046-1402. https://doi.org/10.12688/f1000research.5930.2.

Nikhil Sonnad. A eulogy for What.cd, the greatest music collection in the history of the world—until it vanished. https://qz.com/840661/what-cd-is-gone-a-eulogy-for-the-greatest-music-collection-in-the-world/, November 2016.

Jeffrey Spies. Data Integrity for Librarians, Archivists, and Criminals: What We Can Steal from Bitcoin, BitTorrent, and Usenet, March 2017a.

Jeffrey Spies. A Workflow-Centric Approach to Increasing Reproducibility and Data Integrity. August 2017b.

Manu Sporny, Dave Longley, Gregg Kellogg, Markus Lanthaler, Pierre-Antoine Champin, and Niklas Lindström. JSON-LD 1.1 - A JSON-based Serialization for Linked Data. https://www.w3.org/TR/json-ld/, July 2020.

Springer Nature. Branded Content. https://partnerships.nature.com/product/branded-content-native-advertising/.

Swartz. Making More Wikipedias (Aaron Swartz's Raw Thought). http://www.aaronsw.com/weblog/morewikipedias, September 2006a.

Aaron Swartz. Who Writes Wikipedia? (Aaron Swartz's Raw Thought). http://www.aaronsw.com/weblog/whowriteswikipedia, September 2006b.

Aaron Swartz. Aaron Swartz's A Programmable Web: An Unfinished Work. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 3(2):1–64, February 2013. ISSN 2160-4711, 2160-472X. https://doi.org/10.2200/S00481ED1V01Y201302WBE005.

Archive Team. Scientific Data formats - Just Solve the File Format Problem. http://justsolve.archiveteam.org/wiki/Scientific_Data_formats.

David Tilson, Kalle Lyytinen, and Carsten Sørensen. Digital Infrastructures: The Missing IS Research Agenda. *Information Systems Research*, 21(4):748–759, December 2010. ISSN 1047-7047, 1526-5536. https://doi.org/10.1287/isre.1100.0318.

Nathaniel Tkacz. The Spanish Fork: Wikipedia's ad-fuelled mutiny. *Wired UK*, January 2011. ISSN 1357-0978.

Nathaniel Tkacz. *Wikipedia and the Politics of Openness*. University of Chicago Press, December 2014. ISBN 978-0-226-19244-4. https://doi.org/10.7208/9780226192444.

beka valentine. C2wiki is an exercise in dialogical methods. of laying bare the fact that knowledge and ideas are not some truth delivered from On High, but rather a social process, a conversation, a dialectic, between various views and interests, October 2021.

W. M. P. van der Aalst and A. H. M. ter Hofstede. YAWL: Yet another workflow language. *Information Systems*, 30(4):245–275, June 2005. ISSN 0306-4379. https://doi.org/10.1016/j.is.2004.02.002.

Ernesto Van der Sar. The Open Bay: Now Anyone Can Run A Pirate Bay 'Copy', December 2014.

Ernesto Van der Sar. What.cd is Dead, But The Torrent Hydra Lives on, December 2016.

Iris van Rooij and Giosuè Baggio. Theory Before the Test: How to Build High-Verisimilitude Explanatory Theories in Psychological Science. *Perspectives on Psychological Science*, page 1745691620970604, January 2021. ISSN 1745-6916. https://doi.org/10.1177/1745691620970604.

Fernanda B. Viegas, Martin Wattenberg, Jesse Kriss, and Frank van Ham. Talk Before You Type: Coordination in Wikipedia. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 78–78, Waikoloa, HI, January 2007. IEEE. ISBN 978-0-7695-2755-0. https://doi.org/10.1109/HICSS.2007.511.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: Fundamental

algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. ISSN 1548-7105. https://doi.org/10.1038/s41592-019-0686-2.

Christopher Webber, Jessica Tallon, Erin Shepherd, Amy Guy, and Evan Prodromou. ActivityPub. W3C recommendation, W3C, January 2018.

NWB Behavioral Task WG. NWB Behavioral Task WG, April 2020.

Samantha R. White, Linda M. Amarante, Alexxai V. Kravitz, and Mark Laubach. The Future Is Open: Open-Source Tools for Behavioral Neuroscience Research. *eNeuro*, 6(4):ENEURO.0223–19.2019, August 2019. ISSN 2373-2822. https://doi.org/10.1523/ENEURO.0223-19.2019.

Alexander B. Wiltschko, Tatsuya Tsukahara, Ayman Zeine, Rockwell Anyoha, Winthrop F. Gillis, Jeffrey E. Markowitz, Ralph E. Peterson, Jesse Katon, Matthew J. Johnson, and Sandeep Robert Datta. Revealing the structure of pharmacobehavioral space through motion sequencing. *Nature Neuroscience*, 23(11):1433–1443, November 2020. ISSN 1546-1726. https://doi.org/10.1038/s41593-020-00706-3.

Lauren E. Wool and The International Brain Laboratory. Knowledge across networks: How to build a global neuroscience collaboration. July 2020. https://doi.org/10.1016/j.conb.2020.10.020.

Michael Wulf. *BEADL XML Documentation V 0.1*. July 2020.

Dimitri Yatsenko, Edgar Y. Walker, and Andreas S. Tolias. DataJoint: A Simpler Relational Data Model. *arXiv:1807.11104 [cs]*, July 2018.

Dimitri Yatsenko, Thinh Nguyen, Shan Shen, Kabilar Gunalan, Christopher A. Turner, Raphael Guzman, Maho Sasaki, Daniel Sitonic, Jacob Reimer, Edgar Y. Walker, and Andreas S. Tolias. DataJoint Elements: Data Workflows for Neurophysiology. *bioRxiv*, page 2021.03.30.437358, March 2021. https://doi.org/10.1101/2021.03.30.437358.

C. Zhang, P. Dhungel, D. Wu, and K. W. Ross. Unraveling the BitTorrent Ecosystem. *IEEE Transactions on Parallel and Distributed Systems*, 22(7):1164–1177, July 2011. ISSN 1558-2183. https://doi.org/10.1109/TPDS.2010.123.