

Decentralized Infrastructure for Neuro(science)

Jonny L. Saunders
November 12, 2021

Contents

1	Introduction	4
2	The State of Things	6
2.1	The Costs of being Deinfrastructured	6
2.2	(Mis)incentives in Scientific Software	8
2.2.1	Incentivized Fragmentation	8
2.2.2	Domain-Specific Silos	9
2.2.3	“The Long Now” of Immediacy vs. Idealism	9
2.2.4	“Neatness” vs “Scruffiness”	10
2.2.5	Taped-on Interfaces: Open-Loop User Testing	12
2.2.6	Platforms, Industry Capture, and the Profit Motive	12
2.2.7	Protection of Institutional and Economic Power	13
2.3	The Ivies, Institutes, and “The Rest of Us”	14
2.3.1	Institutional Core Facilities	14
2.3.2	Centralized Institutes	15
2.3.3	Meso-scale collaborations	15
2.3.4	The rest of us...	16
3	A Draft of Decentralized Scientific Infrastructure	17
3.1	Design Principles	17
3.1.1	Integration, not Invention	18
3.1.2	Embrace Heterogeneity, Be Uncoercive	18
3.1.3	Empower People, not Systems	18
3.1.4	Infrastructure is Social	19
3.1.5	Usability Matters	19
3.2	Shared Data	20
3.2.1	Formats as Onramps	20
3.2.2	Peer-to-peer as a Backbone	21
3.2.3	Archives Need Communities	24
3.2.4	Linked Data or Surveillance Capitalism?	26
3.2.5	Federated Systems (of Language)	33
3.3	Shared Tools	38
3.3.1	Analytical Frameworks	39

3.3.2	Experimental Frameworks	43
3.3.3	Abstraction & Protocol Design	48
3.4	Shared Knowledge	50
3.4.1	The Wiki Way	52
3.4.2	The Wiki Way	53
3.4.3	Rebuilding Scientific Communication	54
3.4.4	Credit Assignment	55
4	Conclusion	56
4.1	Shared Governance	56
4.2	Tactics & Strategy	56
4.3	Contrasting visions for science	56
4.3.1	The worst platform capitalist world	56
4.3.2	What we could hope for	56
5	References	57

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 \[1\]](#)

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

Acknowledgements 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
- NWB & DANDI teams
- Kris Chauvin
- Phil Parker
- Chris Rogers
- Danny Mclanahan
- Petar
- 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!

1. Introduction

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

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

If the term infrastructure conjures images of highways and plumbing, then surely digital infrastructure would be flattered at the association. By analogy they illustrate many of its promises and challenges: when designed to, it can make practically impossible things trivial, allowing the development of cities by catching water where it lives and snaking it through tubes and tunnels sometimes directly into your kitchen. Its absence or failure is visible and impactful, as in the case of power outages. There is no guarantee that it “optimally” satisfies some set of needs for the benefit of the greatest number of people, as in the case of the commercial broadband duopolies. It exists not only as its technical reality, but also as an embodied and shared set of social practices, and so even when it does exist its form

is not inevitable or final; as in the case of bottled water producers competing with municipal tap water on a behavioral basis despite being dramatically less efficient and more costly. Finally it is not socially or ethically neutral, and the impact of failure to build or maintain it is not equally shared, as in the expression of institutional racism that was the Flint, Michigan water crisis [3].

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

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

I insist that what I will describe is *not utopian* but is eminently practical — the truly impractical choice is to do

nothing and continue to rest the practice of science on a pyramid scheme [5] of underpaid labor. With a bit of development to integrate and improve the tools, **everything I propose here already exists and is widely used.** A central principle of decentralized systems is embracing heterogeneity: harnessing the power of the diverse ways we do science instead of constraining them. Rather than a patronizing argument that everyone needs to fundamentally alter the way they do science, the systems that I describe are specifically designed to be easily incorporated into existing practices and adapted to variable needs. In this way I argue decentralized systems are *more practical* than the dream that one system will be capable of expanding to the scale of all science — and as will hopefully become clear, inarguably *more powerful* than a disconnected sea of centralized platforms and services.

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

The social and technical perspectives are both essential, but make some conflicting demands on the construction of the piece: Infrastructuring requires considering

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

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

2. The State of Things

2.1 The Costs of being Deinfrastructured

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

To give a sense of the scale of need for digital scientific infrastructure, as well as a general scope for the problems the proposed system is intended to address, I will

list some of the present costs. These lists are grouped into rough and overlapping categories, but make no pretense at completeness and have no particular order.

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

- A prodigious duplication and dead-weight loss of labor as each lab, and sometimes each person within each lab, will reinvent basic code, tools, and practices from scratch. Literally it is the inefficiency of

the **Harberger's triangle** in the supply and demand system for scientific infrastructure caused by inadequate supply. Labs with enough resources are forced to pay from other parts of their grants to hire professional programmers and engineers to build the infrastructure for their lab (and usually their lab or institute only), but most just operate on a purely amateur basis. Many PhD students will spend the first several years of their degree re-solving already-solved problems, chasing the tails of the wrong half-readable engineering whitepapers, in their 6th year finally discovering the technique that they actually needed all along. That's not an educational or training model, it's the effect of displacing the undone labor of unbuilt infrastructure on vulnerable graduate workers almost always paid poverty wages.

- At least the partial cause of the phenomenon where “every scientist needs to be a programmer now” as people who aren't particularly interested in being programmers — which is *fine* and *normal* — need to either suffer through code written by some other unlucky amateur or learn an entire additional discipline in order to do the work of the one they chose. Because there isn't more basic scientific programming infrastructure, everyone needs to be a programmer.
- A great deal of pain and alienation for early-career researchers (ECRs) not previously trained in programming before being thrown in the deep end. Learning data hygiene 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 pinpriciest 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 soci-**

ety:

- An insular system where the inaccessibility of all the “contextual” knowledge [7, 8] 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 knowledge 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 abscension 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 incom-

patible 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 infrastructure 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” [1]

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

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

2.2 (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.

2.2.1 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 publi-

cations, 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. [9]

For an example in Neuroscience, one can browse the papers that cite the DeepLabCut paper [10] 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 [11, 12].

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 [13].

2.2.2 Domain-Specific Silos

When funding exists for scientific infrastructure development, it typically comes in the form of side effects from, or administrative supplements to research grants. The NIH describes as much in their Strategic Plan for Data Science [14]:

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 prior-

itized innovation and hypothesis testing over user service, utility, access, or efficiency. In addition, although the need for open and efficient data sharing is clear, where to store and access datasets generated by individual laboratories—and how to make them compliant with FAIR principles—is not yet straightforward. Overall, it is critical that the data-resource ecosystem become seamlessly integrated such that different data types and information about different organisms or diseases can be used easily together rather than existing in separate data “silos” with only local utility.

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

2.2.3 “The Long Now” of Immediacy vs. Idealism

Digital infrastructure development takes place at multiple timescales simultaneously — from the momentary work of implementing it, through longer timescales of planning, organization, and documenting to the imagined indefinite future of its use — what Ribes and Finholt call “The Long Now. [15]” 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-trodden 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 [16].

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. [17]

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?* [15]

2.2.4 "Neatness" vs "Scruffiness"

Closely related to the tension between "Now" and "Later" is the tension between "Neatness" and "Scruffiness." Lindsay Poirier traces its reflection in the semantic web community as the way that differences in "thought styles" result in different "design logics" [18]. 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" pri-

oritized 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. [18]”

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.” [19] 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 unobvious relationship with beliefs about who should be able to exercise power and what ends a system should serve [20] .

VIEWS OF THE FUTURE	
The last force on us – us	
The standards elephant of yesterday – OSI.	
The standards elephant of today – its right here.	
As the Internet and its community grows, how do we manage the process of change and growth?	
<ul style="list-style-type: none"> • Open process – let all voices be heard. • Closed process – make progress. • Quick process – keep up with reality. • Slow process – leave time to think • Market driven process – the future is commercial. • Scaling driven process – the future is the Internet. 	
We reject: kings, presidents and voting.	
We believe in: rough consensus and running code.	

SLIDE 19

A slide from David Clark’s “Views of the Future”[21] 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!) [22]

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 downstream technologies that have found purchase across information sciences, library sciences, and other applied domains¹. 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

¹This 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 [23] and the profusion of knowledge graph-based medical platforms.

it, wonder why people aren't using it, funding and energy dissipates, rinse, repeat. The difficulty of scaling an exacting vision of how data should be formatted, the tools researchers should use for their experiments, and so on is that they require dramatic and sometimes total changes to the way people do science. The alternative is not between standardization and chaos, but a potential third way is designing infrastructures that allow the diversity of approaches, tools, and techniques to be expressed in a common framework or protocol along with the community infrastructure to allow the continual negotiation of their relationship.

2.2.5 Taped-on Interfaces: Open-Loop User Testing

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

I intend the notion of “interface” in an expansive way: In addition to the graphical user interface (GUI) exposed to the end-user, I am referring generally to all points of contact with users, developers, and other software. Interfaces are intrinsically social, and include the surrounding documentation and experience of use – part of using an API is being able to figure out how to use it! The typical form of scientific software is a black box: I implemented an algorithm of some kind, here is how to use it, but beneath the surface there be dragons.

Ideally, software would be designed with programming interfaces and documentation at multiple scales of complexity to enable clean entrypoints for developers with differing levels of skill and investment to contribute. Additionally, it would include interfaces for use and integration with other software. Without care given to either of these interfaces, the community of 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.

2.2.6 Platforms, Industry Capture, and the Profit Motive

Publicly funded science is an always-irresistable golden goose for private industry. The fragmented interests of scientists and the historically light touch of funding agencies on encroaching privatization means that if some company manages to capture and privatize a corner of scientific practice they are likely to keep it. Industry capture has been thoroughly criticized in the context of the journal system (eg. recently, [24]), 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. [25]). 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 pay-walled (See a more thorough treatment of platform capitalism in science in [4])

As isolated services, one can imagine the practice of science devolving along a similar path as the increasingly-fragmented streaming video market: to do my work I need to subscribe to a data storage service, a cloud computing service, a platform to host my experiments, etc. For larger software platforms, however, vertical integration of multiple complementary services makes their impact on infrastructure more insidious. Locking users into more and more services makes for more and more revenue, which encourages platforms to be as mutually incompatible as they can get away with [26]. 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 [27] that are sucking as much data up as they can to market tools like algorithmic ranking of scientific productivity [28] and making data sharing agreements with ICE [29]. Or see our use of AWS and the laundry list of human rights abuses by Amazon [30]. 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 [31].

Funding models and incentive structures in science are uniformly aligned towards the platformization of scientific infrastructure. Aside from the corporate double-speak 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 infrastruc-

ture 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 [14])

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 [32] 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 [33]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.

2.2.7 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

²Their 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” [34]

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.

2.3 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.

2.3.1 Institutional Core Facilities

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

PNI Systems Core lists **subprojects** echo a lot of the thoughts here, particularly around effort duplication³:

Creating an Optical Instrumentation Core will address

³Thanks a lot to the one-and-only stunning and brilliant Dr. Eartha Mae Guthman for suggesting looking at the BRAIN initiative grants as a way of getting insight on core facilities.

the 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.

While core facilities are an excellent way of expanding access, reducing redundancy, and standardizing tools within an institution, as commonly structured they can displace work spent on those efforts outside of the in-

stitution. 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 adapt⁵. I would consider this example a high-water mark, and the median openness of core infrastructure falls far below it. I was unable to find an example of a core facility that maintained publicly-accessible documentation on the construction and operation of its experimental infrastructure or the management of its facility.

2.3.2 Centralized Institutes

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

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 Data-Joint 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_description.cfm?aid=9444126&icde=0

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

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

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

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

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

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

2.3.3 Meso-scale collaborations

Given the diminishing returns to scale for centralized organizations, many have called for smaller, “meso-scale” collaborations and consortia that combine the efforts of multiple labs [38]. The most successful consortium of this kind has been the International Brain Laboratory [39, 7], a group of 22 labs spread across six countries. They have been able to realize the promise of

big team neuroscience, setting a new standard for performing reproducible experiments performed by many labs [40] and developing data management infrastructure to match [41] (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. [7]

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

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 [42] 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 post-docs, that time can conflict with work on their degrees or personal research which are still far more potent instruments of “remaining employed in science” than collaboration. In the case that only the most well-funded labs and institutions realize the benefits of big team science without explicit consideration given to scientific equity, mesoscale collaborations could have the unintended consequence of magnifying the skewed distribution of access to technical expertise and instrumentation.

2.3.4 The rest of us...

Outside of ivies with rich core facilities, institutes like the Allen, or nascent multi-lab consortia, the rest of us are largely on our own, piecing together what we can from proprietary and open source technology. The

world of open source scientific software has plenty of energy and lots of excellent work is always being done, though constrained by the circumstances of its development described briefly above. Anything else comes down to whatever we can afford with remaining grant money, scrape together from local knowledge, methods sections, begging, borrowing, and (hopefully not too much) stealing from neighboring labs.

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

3. A Draft of Decentralized Scientific Infrastructure

Where do we go from here?

The decentralized infrastructure I will describe here is similar to previous notions of “grass-roots” science articulated within systems neuroscience [38] but has broad and deep history in many domains of computing. My intention is to provide a more prescriptive scaffolding for its design and potential implementation as a way of painting a picture of what science could be like. This sketch is not intended to be final, but a starting point for further negotiation and refinement.

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

3.1 Design Principles

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

!! need to integrate [20]

```
{% include rdfa.html id="protocols" resource="protocols"
typeof="skos:Concept" contents="### Protocols, not
Platforms" %}
```

Much of the basic technology of the internet was developed as *protocols* that describe the basic attributes and operations of a process. `{% include rdfa2.html resource="protocols" property="skos:example" contents="A simple and common example is email over SMTP (Simple Mail Transfer Protocol)" %}` [43]. 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 receive emails to and from any other. The protocol basis of email is the reason why it is possible to send an email from a gmail account to a hotmail account (or any other hacky homebrew SMTP client) despite being wholly different pieces of software.

In contrast, *platforms* provide some service with a specific body of code usually without any pretense of generality. In contrast to email over SMTP, we have grown accustomed to not being able to send a message to someone using Telegram from WhatsApp, switching between multiple mutually incompatible apps that serve nearly identical purposes. Platforms, despite being *theoretically* more limited than associated protocols, are attractive for many reasons: they provide funding

and administrative agencies a single point of contracting and liability, they typically provide a much more polished user interface, and so on. These benefits are short-lived, however, as the inevitable toll of lock-in and shadowy business models is realized.

3.1.1 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” [44]. As a result, IP was developed as a ‘common language’ that could be implemented on any hardware, and upon which other, more complex tools could be built. This is also a cultural practice: when the system doesn’t meet some need, one should try to extend it rather than building a new, separate system — and if a new system is needed, it should be interoperable with those that exist.

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

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

3.1.2 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. In-

stead it should provide affordances that give a clear way for users to extend the system to fit their needs [45]. 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?” [46].

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

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

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

3.1.3 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 [44].

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

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

3.1.4 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 [47, 48].

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

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

3.1.5 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" [45] 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) [49]. *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 front-end 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 decen-

tralized 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 inertia.

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

3.2 Shared Data

3.2.1 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 [50], so a comprehensive treatment is impractical here and I will use the Neurodata Without Borders:N (NWB)[51] 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 [52] and making a modular, extensible format ecosystem.

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

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

to outweigh the ease of use from ad-hoc and historical formats.

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

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

3.2.2 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

grants⁶ 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 files⁷. 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 ex-filtrate, reformat, and rehost the data. The same is true of isolated, local, institutional-level servers and related database platforms, with the additional problem of skewed funding allocation making them unaffordable for many researchers.

Peer-to-peer (p2p) systems solve many of these problems, and I argue are the only type of technology capable of making a database system that can handle the scale of all scientific data. There is an enormous degree of variation between p2p systems⁸, but they share a set of architectural advantages. The essential quality of any p2p system is that rather than each participant in a network interacting only with a single server that hosts all the data, everyone hosts data and interacts directly with each other.

For the sake of concreteness, we can consider a (simplified) description of Bittorrent [54], 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 be-

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

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

⁸peer to peer systems are, maybe predictably, a whole academic subdiscipline. See [53] for reference.

gin 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 [55]. According to a database dump from 2013 [56], 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[57]. Despite more than 15 years of concerted effort by governments and intellectual property holders, the pirate bay is still alive and kicking [58]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 [59] – 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 [60]. 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” [61]. And by groups who disseminate !! return here talking about ddosevrets.

- The network is extremely **scalable** since there is no cost to connecting new peers and the users of a system expand the storage capacity of the system depending on their 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 “seed-boxes”[62] when they use the bittorrent protocol, and “web seeds”[63] 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

⁹knock on wood

other Archive users who have downloaded these Torrents already.” [64]

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 [65], 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 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-grated 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 [66, 67], 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 like 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.” [68]¹⁰ Dat [69], 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 [70], but for now sufficient to say p2p systems can handle version control.

Bittorrent swarms are vulnerable to data loss if all the peers seeding a file disconnect (though the tail is longer than typically assumed, see [71]), 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 [72]. 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 empiri-

¹⁰Git, 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.

cal question whether or not a barter system works or is necessary.

[Solid](#) is a project that almost exactly meets all these needs [73, 74, 75]. 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.

3.2.3 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 [76, 77], 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.” [77]. 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 [76], 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 [71]). 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.

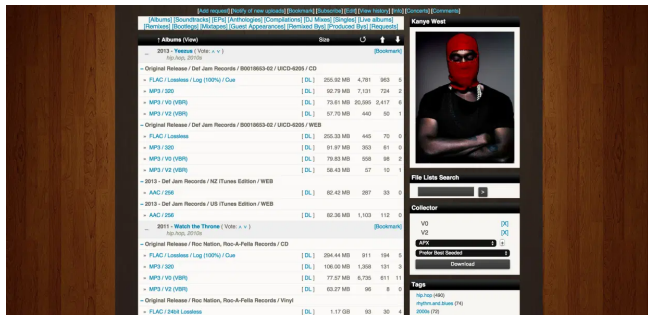
What.cd was a bittorrent tracker that was arguably the largest collection of music that has ever existed. At the time of its destruction in 2016, it was host to just over

¹¹for a detailed description of the site and community, see Ian Dunham's dissertation [78]

¹²Though 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.”

one million unique releases, and approximately 3.5 million torrents¹² [78]. 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 [79] [^lostartists]. 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:

“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” [80]



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 [81]). Invites were extremely rare, and the interview process was demanding to the point where [extensive guides](#) were written to prepare for them.

¹³Though 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 cassette 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?

The what.cd incentive system was based on a required ratio of data uploaded vs. data downloaded [82]. 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 [83].

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 [84, 85].

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 output¹³, 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 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 community*. 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 [86]. 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 [87]. 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 [88] or *AcademicTorrents* [89]. Even with our extensions of version control and some model of automatic mirroring of data across the network, we still have some work to do. To address these and several other remaining needs for scientific data infrastructure, we can take inspiration from *federated systems*.

3.2.4 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-general¹⁴. This design pattern results in a fragmented landscape of databases with limited interoperability. In a moment we'll consider *federated systems* as a way to resolve this dichotomy and continue developing the design of our p2p data infrastructure, but for now we need a better sense of the problem.

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



NIH's *Gene* tool included many specific tools for visualizing, cross-referencing, and aggregating genetic data. Shown is the “genomic regions, transcripts, and product” plot for

Mouse Cdh1, which gives useful, common summary descriptions of the gene, but is not useful for, say, visualizing reading proficiency data.

General-purpose databases like **figshare** and **zenodo**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.

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

¹⁴To 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)

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

terminology, allocate differing granularity and develop different groupings and hierarchies for the same phenomena. At mildest, differences in representational systems can be incompatible, but at their worst they can reflect and reinforce prejudices and become tools of intellectual and social power struggles. Every subdiscipline has conflicting *practical* needs, with infinite variation in privacy demands, different priorities between storage space, bandwidth, and computational power, and so on. In all cases the boundaries of our myopia are impossible to gauge: we might think we have arrived at a suitable schema for biology, chemistry, and physics... but what about the historians?

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

“Participants describe the individual sequence database systems as if they were shadows, poor representations of a widely-agreed-upon ideal. We find, however, that by looking across the landscape of databases, a different picture emerges. Instead, **each decision about the implementation of a particular database system plants a stake for a community boundary. The databases are not so much imperfect copies of an ideal as they are arguments about what the ideal Database should be.** [...] When the microbial ecology project adopted the database system from the traditional genomic “gene finders,” they expected the database to be a boundary object. They knew they would have to customize it to some extent, but thought it would be able to “travel across borders and maintain some sort of constant identity”. In the end, however, **the system was so tailored to a specific set of research questions that the collection of data, the set of tools, and even the social organization of the project had to be significantly changed.** New analysis tools were developed and old tools were discarded. Not only was the database ported to a different technology, the data itself was significantly restructured to fit the new tools and approaches. While the database development projects had begun by working together, in the end they were unable to collaborate. **The system that was supposed to tie these groups together could not be shielded from the controversies that formed the boundaries between the communities of practice.**” [90]

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 onto-

logical 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, possibilities, and the like [91]

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 sources, 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. [14]

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 [92]. OTA awards give the granting agency broad flexibility in to whom and for what money can be given, and consist of an initial competitive segment with possibility for indefinite noncompetitive extensions at the discretion of the agency [93].

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 [94]. 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](#) [95], 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.

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 lack of a universal language for expressing clinical and biomolecular observations presents a risk of isolation or marginalization of data that are relevant for answering a particular inquiry, but are never accessed because of a failure in translation.

Based on these observations, our final assertion is that automating the ability to reason across integrated data sources and providing users who pose inquiries with a dossier of translated answers coupled with full provenance and confidence in the results is critical if we wish to accelerate clinical and translational insights, drive new discoveries, facilitate serendipity, improve clinical-trial design, and ultimately improve clinical care. This final assertion represents the driving motivation for the Translator system. [94]

They articulate a very similar set of beliefs about the impossibility of a unified dataset or ontology¹⁶[94], 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 [96], and descriptions from contractors [97], 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 [96] — 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 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 in-

put 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 [97]) 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 [98]. 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 source¹⁷.

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 [94], 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 be-

gins to creep in.

That is remarkably different than what is articulated in 2019 [94] 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 [99]. Tools like these have been thoroughly problematized elsewhere, eg. [100, 101, 102, 103].

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

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

respect to TERT. [96]

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 [104]). Zidovudine, as a nucleoside reverse transcriptase inhibitor, does inhibit telomerase reverse transcriptase [105] , but can also cause anemia and lower red blood cell counts [106] — 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 [107] . 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 over-

lap, 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 [108] and even literally providing health care [109] , and the uploading of all scientific and medical data onto AWS with entirely unenforceable promises of data privacy [110] — 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 ClinicalKey, 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 Oncol-

ogy) 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 Smart-API, drafting an ontology, and making a knowledge graph, are all directly useful to reducing barriers to desiloing data and shared in the vision articulated here.

The problems here come in a few mutually 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 al-

gorithmic 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 web¹⁸ 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.

I think the problem with the initial and eventual goals of the translator 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 pharmaceuti-

cal companies, I think most people are interested primarily in the data of their colleagues and surrounding disciplines. Every infrastructural model is an act of balancing constraints, and prioritizing “all data” seems to imply “for some people.” Who is supposed to be able to upload data? change the ontology? inspect the machine learning model? Who is in charge of what? Who is a knowledge-graph query engine useful for?

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

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

3.2.5 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 databases¹⁹ were proposed in the early 1980’s [111] and have been developed and refined in the decades since as an alternative to either centralization or non-integration [112, 113, 114]. Their

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

¹⁹though 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.

application to the dispersion of scientific data in local filesystems is not new [115, 116, 117], but their implementation is more challenging than imposing order with a centralized database or punting the question into the unknowable maw of machine learning.

!! 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 [118].

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 [115].

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 [119]:

- 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 accommodate 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 [120]. 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 ORCID²⁰. Someone would then be able to use their @namespace 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 [121]). Let us also assume that there is no categorical difference between @usernames used by individual researchers, institutions, 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.

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.

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

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

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

There are many strategies for making my ontology extension available to others in a federated network. We could use a distributed hash table, or **DHT**, like bit-torrent, which distributes references to information across a network of peers (eg. [122]). 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)** [123], 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** [124], so is already capable of representing linked data, and the related ActivityStreams vocabulary [125] 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 configura-

tion for a federating group that automatically Accepts 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 rule-set, 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 property²¹, 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.

```
<#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-6²². We might not have followed the exact rules, and we only made a few links rather than a single authoritative 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!

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 [126] as has been proposed for biology many times before [127, 116, 117]. 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 [118]. 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

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

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

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.

3.3 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 ex-

perimental 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 [129], and develop **one of them** so I will be brief at risk of repeating myself or appearing self-serving.

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 [128, 118] 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 - [88] 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 talking 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!

!! 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 [130]

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.

3.3.1 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` [131] and `scipy` [132], through tools that implement a class of analysis like `DeepLabCut` [10] and `scikit-learn` [133], to tools for a specific technique like `MoSeq` [134] and `DeepSqueak` [135]. 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 tool-builders, 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](#) [136], which recently expanded its facility for modularity with its recent [Elements](#) project [137]. 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](#) [138]) 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 [139]). 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 over-complicate the pseudocode example, but say we successfully specify a 3 dimensional (width x height x channels) array with 3 channels as input, and 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/fas
  @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 some-

one 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.pkl')

    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:

```
<#project-name>
  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

3.3.2 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.

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.

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 [129], 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 exist-

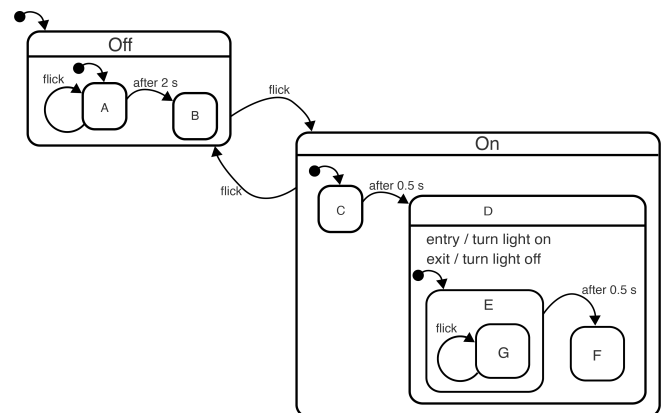
ing infrastructure in a lab.

- 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 researchers 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 experimen-

tal 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 \[140\]](#) 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 [141]. 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 [129] 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 Tasks 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 Tasks 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 inexact replicated experimental apparatuses.

In contrast, workflows in Bonsai [\[142, 143\]](#), 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-2>
  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 vocab-

ularies 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](#) [144] 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.

3.3.3 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 system. 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 [145, 146, 147, 148], 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 [149]

!! 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 reparameterized 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 analysis 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.

3.4 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 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 [27]. In the next breath, they describe how “in databases & tools and electronic reference, representing over a third of divisional²³ revenue, we continued to drive good growth through content development and enhanced machine learning [ML] and natural language processing [NLP] based functionality.”

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 clin-

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

icians 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. [150]

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 [151, 152] .

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?” [153] . 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.” [154]

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 an-

other mechanism that reentrenches scientific inequity for profit. Knowledge disorganization, coupled with a little surveillance capitalism that monitors the activity of colleagues and rivals [24] , 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 [155] .

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 already 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 arcany 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 [156], and brave and highly-skilled investigators like [Elisabeth Bik](#) have found thousands of papers with evidence of purposeful manipulation [157, 158]. !! 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 preparation 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](#) [159], [Open Neuroscience](#), and many others to index technical knowledge and tools. Finally we use sites like [PubPeer](#) and ResearchGate for comment and criticism.

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 wholly 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** - 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.

!! There are many tunnels of internet history that have traced different parts of these axes, but particularly relevant here are the overlapping histories of early wikis, wikipedia, the semantic web, and activitypub. 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.

3.4.1 The Wiki Way

x

current status - our current systems are largely journals and conferences, but as evidenced by the unfortunate but widescale adoption of twitter, scientific communication naturally spans dispersed forms of communication at different scales of formality, length, public engagement, etc. - when we go to organize ourselves, why is the best we can do google docs and slack?

contextual knowledge needed - our limited systems of communication also render large sections of needed scientific communication without venue. The existing tools that *do* give some means of sharing technical knowledge are distinctly charity-driven, and don't confer the same type of credit incentive that publications do.

!! important of ease of leaving <http://meatballwiki.org/wiki/RightToLeave>

!! we've been tracing a distinction between the ability to express fluidly the contents of our reality with developing platforms that sift through it in an automated way, something that was an explicit cultural division throughout the semantic web project [160], which Peter Norvig (director of search at Google at the time) primarily attributes to user incompetence [161]. On trust, TBL says "Berners-Lee agreed with Norvig that deception on the Internet is a problem, but he argued that part of the Semantic Web is about identifying the originator of information, and identifying why the information can be trusted, not just the content of the information itself."

!! more techniques of communitiy growth <http://meatballwiki.org/wiki/RewardReputation>

!! wikis work! but they can break when people get too much power! <http://www.aaronsw.com/weblog/whorunswikipedia>

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

scientific infrastructure.

!! importantly, should also have means of ingest for existing tools and elements – easy to import existing papers and citation trees, plugins for existing data sharing systems.

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

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

!! should trace the development of AP and the difficulty of doing these things as a way to explaining the ecosystem and the different parts that are needed in it: <https://www.w3.org/TR/social-web-protocols/>

3.4.2 The Wiki Way

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. [-1, 329, 162]

!! wiki cultural history stuff!! differences in wiki philosophy, deletists vs not. !! meatball stuff on community maintenace, conflict resolution,

!! also hints at the problems, difficulties with wiki culture

It's not too late to turn things around. Specs could be moved back into the wiki until they're nearly done. Editors, instead of being gatekeepers, could be helpful moderators. A clear process for making controversial decisions could be decided on. And the validator could fol-

low consensus instead of leading it. But do the people running the show want this?

Standards bodies tread a fine line between organizations for the public good and shelters for protecting collusion that would be otherwise illegal under antitrust law. For the dominant vendors involved, the goal is to give the illusion of openness while giving themselves full control to enforce their will behind the scenes.

The IETF is a good example of this. Often lauded by the public as a model of openness and and and freedom, the reality is that working group chairs, appointed by a self-elected ruling board, get away with declaring whatever they want (usually an inferior and difficult to implement alternative) as “rough consensus”, routinely ignoring comments from the public and objections from working group members. One working group (in charge of DNS extensions) went so far as to censor mail from working group members. The dictators running the IETF, when informed, didn’t seem to mind.

Is the same sort of thing at work in the Pie/Echo/Atom Project? It appears so at first glance: Sam running the show from behind the scenes, putting friends in charge of the specs (although that isn’t what actually happened). The lack of a dispute-resolution process only makes things worse: when there’s no clear guide on how to make decisions or contributions, it’s far from obvious how to challenge a decision Sam has made. [163]

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 [164]

!! give the example of the autopilot wiki

!! contextual knowledge stuff in this section, theory wiki stuff in next section

Two essential features coordinate this information to better serve our organizational decision-making, learning, and memory. The first is our constellation of Working Groups that maintain and distribute local, specialized knowledge to other groups across the network. [...] A second, more emergent property is the subgroup of IBL researchers who have become experts, liaisons, and interpreters of knowledge across the network. These members each manage a domain of explicit records (e.g., written protocols) and tacit information (e.g., colloquialisms, decision histories) that are quickly and informally disseminated to address real-time needs and problems. A remarkable nimbleness is afforded by this system of rapid responders deployed across our web of Working Groups. However, this kind of internalized knowledge can be vulnerable to drop-out when people leave the collaboration, and can be complex to archive. An ongoing

challenge for our collaboration is how to archive both our explicit and tacit processes held in both people and places. This is not only to document our own history but as part of a roadmap for future science teams, whose dynamics are still not fully understood. [7]

[165]

!! Read and cite! [166]

!! [167]

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

!! lots of scientific wikis - [https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Molecular_B](https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Molecular_Biology)

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

!! interlex

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

!! Semantic combination of databases in science are also not new [168, 127] . We need both though! semantic federated databases!

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

3.4.3 Rebuilding Scientific Communication

!! skohub! <https://skohub.io/>

!! take stock of our communication technology, we publish pdfs in journals, have science twitter, and then a bunch of private slacks and smalltime stuff??? Science is fundamentally a communicative process, liter-

ally every part fo the system that I have described has been built around the ability to express the structure of things, the order of things, how it relates to other things and *that's communication baby*. The system we've imagined so far takes us so far from forums and the ultra-dominant feed -> shallow thread-based communication that we're used to though. This is a system where we can have continuous dialogue about linked topics, be able to branch and see the reflections and subtle variations on ideas in the same place that we have our data, analysis, and tools.

!! theory wiki example from presentation

!! discovery of papers for scientists as well as general public, being able to trace history.

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

!! stop sweating about computational accuracy and completeness. the only danger is a system that makes appeal to perfection and promises accuracy like those sold in golden foil by the platform capitalists. if we are conceptualizing this appropriately as a *system of communication* where particular results are intended to be *interpreted in context* then we would treat computational errors and semantic inaccuracies like we do with *language*: like a *joke*.

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>

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 comlumn 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>

!! caution about slipping into techno-utopianism even here, we need the UI and tooling here to be simple to not only use but also build on. yes that does mean yet another framework! but this one is the most mythical yet, because I don't really know what it would look like! but bad UI has killed lots of projects, eg. IPFS (though it's not dead just slow!) <https://macwright.com/2019/06/08/ipfs-again.html> <https://blog.bluzelle.com/ipfs-is-not-what-you-think-it-is-e0aa8dc69b>

3.4.4 Credit Assignment

the work of maintaining the system can't be invisible, read & cite [166, 1]

!! 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.

4. Conclusion

!! summary of the system design

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

4.1 Shared Governance

!! the broad and uncertain future here is how this system will be governed and how it will be operated. 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_\(software\)](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>

4.2 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.

4.3 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).

4.3.1 The worst platform capitalist world

!! ahh huh you know what it is

4.3.2 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.

5. References

References

- [1] Geoffrey C. Bowker et al. “Toward Information Infrastructure Studies: Ways of Knowing in a Networked Environment”. In: *International Handbook of Internet Research*. Ed. by Jeremy Hunsinger, Lisbeth Klastrup, and Matthew Allen. Dordrecht: Springer Netherlands, 2010, pp. 97–117. ISBN: 978-1-4020-9789-8. DOI: [10.1007/978-1-4020-9789-8_5](https://doi.org/10.1007/978-1-4020-9789-8_5). URL: https://doi.org/10.1007/978-1-4020-9789-8_5 (visited on 03/07/2021) (cit. on pp. 3, 8, 55).
- [2] David Tilson, Kalle Lyytinen, and Carsten Sørensen. “Digital Infrastructures: The Missing IS Research Agenda”. In: *Information Systems Research* 21.4 (Dec. 2010), pp. 748–759. ISSN: 1047-7047, 1526-5536. DOI: [10.1287/isre.1100.0318](https://doi.org/10.1287/isre.1100.0318). URL: <http://pubsonline.informs.org/doi/abs/10.1287/isre.1100.0318> (visited on 05/02/2021) (cit. on p. 3).
- [3] Michican Civil Rights Commission. *The Flint Water Crisis: Systemic Racism Through the Lens of Flint*. Tech. rep. Michican Civil Rights Commission, Feb. 2017. URL: https://web.archive.org/web/20210518020755/https://www.michigan.gov/documents/mdcr/VFlintCrisisRep-F-Edited3-13-17_554317_7.pdf (visited on 05/18/2021) (cit. on p. 5).
- [4] Philip Mirowski. “The Future(s) of Open Science”. In: *Social Studies of Science* 48.2 (Apr. 2018), pp. 171–203. ISSN: 0306-3127. DOI: [10.1177/0306312718772086](https://doi.org/10.1177/0306312718772086). URL: <https://doi.org/10.1177/0306312718772086> (visited on 03/27/2021) (cit. on pp. 5, 12).
- [5] Charleeze Ponzi. *Is Science a Pyramid Scheme? The Correlation between an Author’s Position in the Academic Hierarchy and Her Scientific Output per Year*. Jan. 2020. DOI: [10.31235/osf.io/c3xg5](https://doi.org/10.31235/osf.io/c3xg5). URL: <https://osf.io/preprints/socarxiv/c3xg5/> (visited on 09/16/2021) (cit. on p. 6).
- [6] Matthew J. Bietz, Toni Ferro, and Charlotte P. Lee. “Sustaining the Development of Cyberinfrastructure: An Organization Adapting to Change”. In: *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*. CSCW ’12. New York, NY, USA: Association for Computing Machinery, Feb. 2012, pp. 901–910. ISBN: 978-1-4503-1086-4. DOI: [10.1145/2145204.2145339](https://doi.org/10.1145/2145204.2145339). URL: <https://doi.org/10.1145/2145204.2145339> (visited on 09/21/2021) (cit. on p. 6).
- [7] Lauren E. Wool and The International Brain Laboratory. “Knowledge across Networks: How to Build a Global Neuroscience Collaboration”. In: (July 2020). DOI: [10.1016/j.conb.2020.10.020](https://doi.org/10.1016/j.conb.2020.10.020). URL: <https://psyarxiv.com/f4uaj/> (visited on 03/07/2021) (cit. on pp. 8, 15, 16, 54).
- [8] STEPHEN R. BARLEY and BETH A. BECHKY. “In the Backrooms of Science: The Work of Technicians in Science Labs”. In: *Work and Occupations* 21.1 (Feb. 1994), pp. 85–126. ISSN: 0730-8884. DOI: [10.1177/0730888494021001004](https://doi.org/10.1177/0730888494021001004). URL: <https://doi.org/10.1177/0730888494021001004> (visited on 03/15/2021) (cit. on p. 8).
- [9] James Howison and James D. Herbsleb. “Incentives and Integration in Scientific Software Production”. In: *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*. CSCW ’13. New York, NY, USA: Association for Computing Machinery, Feb. 2013, pp. 459–470. ISBN: 978-1-4503-1331-5. DOI: [10.1145/2441776.2441828](https://doi.org/10.1145/2441776.2441828). URL: <https://doi.org/10.1145/2441776.2441828> (visited on 09/21/2021) (cit. on p. 9).
- [10] Alexander Mathis et al. “DeepLabCut: Markerless Pose Estimation of User-Defined Body Parts with Deep Learning”. In: *Nature Neuroscience* 21.9 (Sept. 2018), pp. 1281–1289. ISSN: 1546-1726. DOI: [10.1038/s41593-018-0209-y](https://doi.org/10.1038/s41593-018-0209-y). URL: <https://www.nature.com/articles/s41593-018-0209-y> (visited on 09/22/2021) (cit. on pp. 9, 39).

- [11] Serghei Mangul et al. “Improving the Usability and Archival Stability of Bioinformatics Software”. In: *Genome Biology* 20.1 (Feb. 2019), p. 47. ISSN: 1474-760X. DOI: [10.1186/s13059-019-1649-8](https://doi.org/10.1186/s13059-019-1649-8). URL: <https://doi.org/10.1186/s13059-019-1649-8> (visited on 09/22/2021) (cit. on p. 9).
- [12] Sudhir Kumar and Joel Dudley. “Bioinformatics Software for Biologists in the Genomics Era”. In: *Bioinformatics* 23.14 (July 2007), pp. 1713–1717. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/btm239](https://doi.org/10.1093/bioinformatics/btm239). URL: <https://doi.org/10.1093/bioinformatics/btm239> (visited on 09/22/2021) (cit. on p. 9).
- [13] James Howison and Julia Bullard. “Software in the Scientific Literature: Problems with Seeing, Finding, and Using Software Mentioned in the Biology Literature”. In: *Journal of the Association for Information Science and Technology* 67.9 (2016), pp. 2137–2155. ISSN: 2330-1643. DOI: [10.1002/asi.23538](https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23538). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23538> (visited on 09/21/2021) (cit. on p. 9).
- [14] *NIH Strategic Plan for Data Science*. Tech. rep. National Institutes of Health, June 2018. URL: https://web.archive.org/web/20210907014444/https://datascience.nih.gov/sites/default/files/NIH_Strategic_Plan_for_Data_Science_Final_508.pdf (visited on 09/22/2021) (cit. on pp. 9, 13, 28).
- [15] David Ribes and Thomas Finholt. “The Long Now of Technology Infrastructure: Articulating Tensions in Development”. In: *Journal of the Association for Information Systems* 10.5 (May 2009), pp. 375–398. ISSN: 15369323. DOI: [10.17705/1jais.00199](https://aisel.aisnet.org/jais/vol10/iss5/5/). URL: <https://aisel.aisnet.org/jais/vol10/iss5/5/> (visited on 05/02/2021) (cit. on pp. 9, 10).
- [16] Stephen Altschul et al. “The Anatomy of Successful Computational Biology Software”. In: *Nature Biotechnology* 31.10 (Oct. 2013), pp. 894–897. ISSN: 1546-1696. DOI: [10.1038/nbt.2721](https://www.nature.com/articles/nbt.2721). URL: <https://www.nature.com/articles/nbt.2721> (visited on 09/22/2021) (cit. on p. 10).
- [17] Sean B. Palmer. *Ditching the Semantic Web?* Mar. 2008. URL: <http://inamidst.com/whits/2008/ditching> (visited on 09/24/2021) (cit. on p. 10).
- [18] Lindsay Poirier. “A Turn for the Scruffy: An Ethnographic Study of Semantic Web Architecture”. In: *Proceedings of the 2017 ACM on Web Science Conference*. WebSci ’17. New York, NY, USA: Association for Computing Machinery, June 2017, pp. 359–367. ISBN: 978-1-4503-4896-6. DOI: [10.1145/3091478.3091505](https://doi.org/10.1145/3091478.3091505). URL: <https://doi.org/10.1145/3091478.3091505> (visited on 07/01/2021) (cit. on pp. 10, 11).
- [19] Michael A. Hiltzik. *Taming the Wild, Wild Web*. July 2001. URL: <https://web.archive.org/web/20010801142640/https://www.latimes.com/business/la-072601netarch.story> (visited on 11/05/2021) (cit. on p. 11).
- [20] Rebekah Larsen. “The Political Nature of TCP/IP”. In: (2012), p. 56 (cit. on pp. 11, 17).
- [21] David Clark. “A Cloudy Crystal Ball - Visions of the Future”. In: *Proceedings of the Twenty-Fourth Internet Engineering Task Force*. July 1992, pp. 539–543. URL: <https://www.ietf.org/proceedings/24.pdf> (visited on 11/05/2021) (cit. on p. 11).
- [22] Aaron Swartz. “Aaron Swartz’s A Programmable Web: An Unfinished Work”. In: *Synthesis Lectures on the Semantic Web: Theory and Technology* 3.2 (Feb. 2013), pp. 1–64. ISSN: 2160-4711, 2160-472X. DOI: [10.2200/S00481ED1V01Y201302WBE005](http://www.morganclaypool.com/doi/abs/10.2200/S00481ED1V01Y201302WBE005). URL: <http://www.morganclaypool.com/doi/abs/10.2200/S00481ED1V01Y201302WBE005> (visited on 06/30/2021) (cit. on p. 11).
- [23] Iain. *Freebase Is Dead, Long Live Freebase*. May 2019. URL: <https://medium.com/@iainsproat/freebase-is-dead-long-live-freebase-6c1daff44d19> (visited on 09/17/2021) (cit. on p. 11).
- [24] Björn Brembs et al. “Replacing Academic Journals”. In: (Sept. 2021). DOI: [10.5281/zenodo.5526635](https://zenodo.org/record/5526635). URL: <https://zenodo.org/record/5526635> (visited on 09/24/2021) (cit. on pp. 12, 51).
- [25] *Elsevier and Seven Bridges Receive NIH Data Commons Grant for Biomedical Data Analysis*. Nov. 2017. URL: <https://www.elsevier.com/about/press-releases/archive/science-and-technology/elsevier-and-seven-bridges-receive-nih-data-commons-grant-for-biomedical-data-analysis> (visited on 10/06/2021) (cit. on p. 12).

- [26] Ian MacInnes. “Compatibility Standards and Monopoly Incentives: The Impact of Service-Based Software Licensing”. In: *International Journal of Services and Standards* 1.3 (Jan. 2005), pp. 255–270. ISSN: 1740-8849. DOI: [10.1504/IJSS.2005.005799](https://doi.org/10.1504/IJSS.2005.005799). URL: <https://www.inderscienceonline.com/doi/abs/10.1504/IJSS.2005.005799> (visited on 09/24/2021) (cit. on p. 12).
- [27] “RELX Annual Report 2020”. In: (2020), p. 196. URL: <https://www.relx.com/~media/Files/R/RELX-Group/documents/reports/annual-reports/2020-annual-report.pdf> (cit. on pp. 13, 50).
- [28] Björn Brembs. *Algorithmic Employment Decisions in Academia?* Sept. 2021. URL: <http://bjoern.brembs.net/2021/09/algorithmic-employment-decisions-in-academia/> (visited on 11/05/2021) (cit. on p. 13).
- [29] Sam Biddle. *LexisNexis to Provide Giant Database of Personal Information to ICE*. Apr. 2021. URL: <https://theintercept.com/2021/04/02/ice-database-surveillance-lexisnexis/> (visited on 09/24/2021) (cit. on p. 13).
- [30] “Criticism of Amazon”. In: *Wikipedia* (Sept. 2021). URL: https://en.wikipedia.org/w/index.php?title=Criticism_of_Amazon&oldid=1043543093 (visited on 09/25/2021) (cit. on p. 13).
- [31] Elsevier. *Topic Prominence in Science - Scival | Elsevier Solutions*. URL: <https://www.elsevier.com/solutions/scival/features/topic-prominence-in-science> (visited on 11/05/2021) (cit. on p. 13).
- [32] Stephen Buranyi. “Is the Staggeringly Profitable Business of Scientific Publishing Bad for Science?” In: *The Guardian* (June 2017). ISSN: 0261-3077. URL: <https://www.theguardian.com/science/2017/jun/27/profitable-business-scientific-publishing-bad-for-science> (visited on 09/25/2021) (cit. on p. 13).
- [33] R. Todd Reilly. *NIH STRIDES Initiative*. Jan. 2021. URL: https://web.archive.org/web/20211006011408/https://ncihub.org/resources/2422/download/21.01.08_NIH_STRIDES_Presentation.pptx (visited on 10/06/2021) (cit. on p. 13).
- [34] *STRIDES Initiative Success Story: University of Michigan TOPMed | Data Science at NIH*. Oct. 2020. URL: <https://web.archive.org/web/20210324024612/https://datascience.nih.gov/strides-initiative-success-story-university-michigan-topmed> (visited on 10/06/2021) (cit. on p. 13).
- [35] Ed S. Lein et al. “Genome-Wide Atlas of Gene Expression in the Adult Mouse Brain”. In: *Nature* 445.7124 (Jan. 2007), pp. 168–176. ISSN: 1476-4687. DOI: [10.1038/nature05453](https://doi.org/10.1038/nature05453). URL: <https://www.nature.com/articles/nature05453> (visited on 03/15/2021) (cit. on p. 15).
- [36] Sten Grillner et al. “Worldwide Initiatives to Advance Brain Research”. In: *Nature Neuroscience* 19.9 (Sept. 2016), pp. 1118–1122. ISSN: 1546-1726. DOI: [10.1038/nn.4371](https://doi.org/10.1038/nn.4371). URL: <https://www.nature.com/articles/nn.4371> (visited on 03/15/2021) (cit. on p. 15).
- [37] Christof Koch and Allan Jones. “Big Science, Team Science, and Open Science for Neuroscience”. In: *Neuron* 92.3 (Nov. 2016), pp. 612–616. ISSN: 0896-6273. DOI: [10.1016/j.neuron.2016.10.019](https://doi.org/10.1016/j.neuron.2016.10.019). URL: <https://www.sciencedirect.com/science/article/pii/S0896627316307206> (visited on 03/15/2021) (cit. on p. 15).
- [38] Zachary F. Mainen, Michael Häusser, and Alexandre Pouget. “A Better Way to Crack the Brain”. In: *Nature News* 539.7628 (Nov. 2016), p. 159. DOI: [10.1038/539159a](https://doi.org/10.1038/539159a). URL: <http://www.nature.com/news/a-better-way-to-crack-the-brain-1.20935> (visited on 03/09/2021) (cit. on pp. 15, 17).
- [39] Larry F. Abbott et al. “An International Laboratory for Systems and Computational Neuroscience”. In: *Neuron* 96.6 (Dec. 2017), pp. 1213–1218. ISSN: 0896-6273. DOI: [10.1016/j.neuron.2017.12.013](https://doi.org/10.1016/j.neuron.2017.12.013). URL: <https://www.sciencedirect.com/science/article/pii/S0896627317311364> (visited on 03/15/2021) (cit. on p. 15).
- [40] The International Brain Laboratory et al. “Standardized and Reproducible Measurement of Decision-Making in Mice”. In: *bioRxiv* (Oct. 2020), p. 2020.01.17.909838. DOI: [10.1101/2020.01.17.909838](https://doi.org/10.1101/2020.01.17.909838). URL: <https://www.biorxiv.org/content/10.1101/2020.01.17.909838v5> (visited on 03/15/2021) (cit. on p. 16).

- [41] The International Brain Laboratory et al. “Data Architecture for a Large-Scale Neuroscience Collaboration”. In: *bioRxiv* (Feb. 2020), p. 827873. DOI: [10.1101/827873](https://doi.org/10.1101/827873). URL: <https://www.biorxiv.org/content/10.1101/827873v2> (visited on 03/15/2021) (cit. on p. 16).
- [42] Gonalo Lopes et al. “Bonsai: An Event-Based Framework for Processing and Controlling Data Streams”. In: *Frontiers in Neuroinformatics* 9 (2015). ISSN: 1662-5196. DOI: [10.3389/fninf.2015.00007](https://doi.org/10.3389/fninf.2015.00007). URL: <https://www.frontiersin.org/articles/10.3389/fninf.2015.00007/full> (visited on 03/15/2021) (cit. on p. 16).
- [43] *Rfc5321 - Simple Mail Transfer Protocol*. URL: <https://datatracker.ietf.org/doc/html/rfc5321#section-3> (visited on 09/27/2021) (cit. on p. 17).
- [44] D. Clark. “The Design Philosophy of the DARPA Internet Protocols”. In: *Symposium Proceedings on Communications Architectures and Protocols*. SIGCOMM ’88. New York, NY, USA: Association for Computing Machinery, Aug. 1988, pp. 106–114. ISBN: 978-0-89791-279-2. DOI: [10.1145/52324.52336](https://doi.org/10.1145/52324.52336). URL: <https://doi.org/10.1145/52324.52336> (visited on 03/15/2021) (cit. on pp. 18, 19).
- [45] Brian E. Carpenter. *RFC 1958 - Architectural Principles of the Internet*. June 1996. URL: <https://tools.ietf.org/html/rfc1958> (visited on 03/15/2021) (cit. on pp. 18, 19).
- [46] Tim Berners-Lee. *Principles of Design*. 1998. URL: <https://www.w3.org/DesignIssues/Principles.html#Decentrali> (visited on 03/15/2021) (cit. on p. 18).
- [47] Jonathan Grudin. “Groupware and Social Dynamics: Eight Challenges for Developers”. In: *Communications of the ACM* 37.1 (Jan. 1994), pp. 92–105. ISSN: 0001-0782. DOI: [10.1145/175222.175230](https://doi.org/10.1145/175222.175230). URL: <https://doi.org/10.1145/175222.175230> (visited on 10/11/2021) (cit. on p. 19).
- [48] Dave Randall et al. “Distributed Ontology Building as Practical Work”. In: *International Journal of Human-Computer Studies* 69.4 (Apr. 2011), pp. 220–233. ISSN: 1071-5819. DOI: [10.1016/j.ijhcs.2010.12.011](https://doi.org/10.1016/j.ijhcs.2010.12.011). URL: <https://www.sciencedirect.com/science/article/pii/S1071581911000024> (visited on 10/11/2021) (cit. on p. 19).
- [49] John Markoff. “Tomorrow, the World Wide Web!;Microsoft, the PC King, Wants to Reign Over the Internet”. In: *The New York Times* (July 1996). ISSN: 0362-4331. URL: <https://www.nytimes.com/1996/07/16/business/tomorrow-world-wide-web-microsoft-pc-king-wants-reign-over-internet.html> (visited on 10/12/2021) (cit. on p. 19).
- [50] Archive Team. *Scientific Data Formats - Just Solve the File Format Problem*. URL: http://justsolve.archiveteam.org/wiki/Scientific_Data_formats (visited on 09/27/2021) (cit. on p. 20).
- [51] Oliver R  bel et al. “NWB:N 2.0: An Accessible Data Standard for Neurophysiology”. In: *bioRxiv* (Jan. 2019), p. 523035. DOI: [10.1101/523035](https://doi.org/10.1101/523035). URL: <https://www.biorxiv.org/content/10.1101/523035v1> (visited on 03/15/2021) (cit. on p. 20).
- [52] Oliver R  bel et al. *The Neurodata Without Borders Ecosystem for Neurophysiological Data Science*. Mar. 2021. DOI: [10.1101/2021.03.13.435173](https://doi.org/10.1101/2021.03.13.435173). URL: <https://www.biorxiv.org/content/10.1101/2021.03.13.435173v1> (visited on 09/27/2021) (cit. on p. 20).
- [53] Xuemin Shen et al. *Handbook of Peer-to-Peer Networking*. Springer Science & Business Media, Mar. 2010. ISBN: 978-0-387-09751-0 (cit. on p. 21).
- [54] Bram Cohen. *The BitTorrent Protocol Specification*. Feb. 2017. URL: https://www.bittorrent.org/beps/bep_0003.html (visited on 09/28/2021) (cit. on p. 21).
- [55] Janko Roettgers. *The Pirate Bay: Distributing the World’s Entertainment for \$3,000 a Month*. July 2009. URL: <https://gigaom.com/2009/07/19/the-pirate-bay-distributing-the-worlds-entertainment-for-3000-a-month/> (visited on 09/28/2021) (cit. on p. 22).
- [56] *The Pirate Bay - Archiveteam*. Sept. 2020. URL: https://wiki.archiveteam.org/index.php?title=The_Pirate_Bay&oldid=45467 (visited on 09/28/2021) (cit. on p. 22).

- [57] Jeffrey Spies. *Data Integrity for Librarians, Archivists, and Criminals: What We Can Steal from Bitcoin, BitTorrent, and Usenet*. Mar. 2017. URL: <https://www.cni.org/topics/digital-curation/data-integrity-for-librarians-archivists-and-criminals-what-we-can-steal-from-bitcoin-bittorrent-and-usenet> (visited on 10/01/2021) (cit. on p. 22).
- [58] Eddie Kim. *After 15 Years, the Pirate Bay Still Can't Be Killed*. May 2019. URL: <https://melmagazine.com/en-us/story/after-15-years-the-pirate-bay-still-cant-be-killed> (visited on 09/28/2021) (cit. on p. 22).
- [59] Ernesto Van der Sar. *The Open Bay: Now Anyone Can Run A Pirate Bay 'Copy'*. Dec. 2014. URL: <https://torrentfreak.com/open-bay-now-everyone-can-run-pirate-bay-copy-141219/> (visited on 09/28/2021) (cit. on p. 22).
- [60] Ernesto Van der Sar. *What.Cd Is Dead, But The Torrent Hydra Lives On*. Dec. 2016. URL: <https://torrentfreak.com/what-cd-is-dead-but-the-torrent-hydra-lives-on-161202/> (visited on 03/18/2021) (cit. on p. 22).
- [61] Jason Scott. *Geocities Torrent Update*. Dec. 2010. URL: <http://ascii.textfiles.com/archives/2894> (visited on 09/30/2021) (cit. on p. 22).
- [62] Dario Rossi et al. "Peeking through the BitTorrent Seedbox Hosting Ecosystem". In: *Traffic Monitoring and Analysis*. Ed. by Alberto Dainotti, Anirban Mahanti, and Steve Uhlig. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2014, pp. 115–126. ISBN: 978-3-642-54999-1. DOI: 10.1007/978-3-642-54999-1_10 (cit. on p. 22).
- [63] John Hoffman and DeHackEd. *HTTP-Based Seeding Specification*. URL: <http://www.bittornado.com/docs/webseed-spec.txt> (visited on 09/30/2021) (cit. on p. 22).
- [64] Brewster Kahle. *Over 1,000,000 Torrents of Downloadable Books, Music, and Movies*. Aug. 2012. URL: <http://blog.archive.org/2012/08/07/over-1000000-torrents-of-downloadable-books-music-and-movies/> (visited on 09/30/2021) (cit. on p. 23).
- [65] G. Kreitz and F. Niemela. "Spotify – Large Scale, Low Latency, P2P Music-on-Demand Streaming". In: *2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P)*. Delft, Netherlands: IEEE, Aug. 2010, pp. 1–10. ISBN: 978-1-4244-7140-9. DOI: 10.1109/P2P.2010.5569963. URL: <http://ieeexplore.ieee.org/document/5569963/> (visited on 10/01/2021) (cit. on p. 23).
- [66] Andrey Andreev et al. "Biologists Need Modern Data Infrastructure on Campus". In: *arXiv:2108.07631 [q-bio]* (Aug. 2021). arXiv: 2108.07631 [q-bio]. URL: <http://arxiv.org/abs/2108.07631> (visited on 09/24/2021) (cit. on p. 23).
- [67] Adam S. Charles et al. "Toward Community-Driven Big Open Brain Science: Open Big Data and Tools for Structure, Function, and Genetics". In: *Annual Review of Neuroscience* 43 (July 2020), pp. 441–464. ISSN: 1545-4126. DOI: 10.1146/annurev-neuro-100119-110036 (cit. on p. 23).
- [68] Juan Benet. "IPFS - Content Addressed, Versioned, P2P File System". In: *arXiv:1407.3561 [cs]* (July 2014). arXiv: 1407.3561 [cs]. URL: <http://arxiv.org/abs/1407.3561> (visited on 03/20/2021) (cit. on p. 23).
- [69] Maxwell Ogden. *Dat - Distributed Dataset Synchronization And Versioning*. Preprint. Open Science Framework, Jan. 2017. DOI: 10.31219/osf.io/nsv2c. URL: <https://osf.io/nsv2c> (visited on 10/01/2021) (cit. on p. 23).
- [70] Constantinos Patsakis and Fran Casino. "Hydras and IPFS: A Decentralised Playground for Malware". In: *International Journal of Information Security* 18.6 (Dec. 2019), pp. 787–799. ISSN: 1615-5262, 1615-5270. DOI: 10.1007/s10207-019-00443-0. arXiv: 1905.11880. URL: <http://arxiv.org/abs/1905.11880> (visited on 10/01/2021) (cit. on p. 23).
- [71] C. Zhang et al. "Unraveling the BitTorrent Ecosystem". In: *IEEE Transactions on Parallel and Distributed Systems* 22.7 (July 2011), pp. 1164–1177. ISSN: 1558-2183. DOI: 10.1109/TPDS.2010.123 (cit. on pp. 23, 24).

- [72] Ian Clarke et al. “Freenet: A Distributed Anonymous Information Storage and Retrieval System”. In: *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability Berkeley, CA, USA, July 25–26, 2000 Proceedings*. Ed. by Hannes Federrath. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2001, pp. 46–66. ISBN: 978-3-540-44702-3. DOI: [10.1007/3-540-44702-4_4](https://doi.org/10.1007/3-540-44702-4_4). URL: https://doi.org/10.1007/3-540-44702-4_4 (visited on 03/18/2021) (cit. on p. 23).
- [73] Sarven Capadisli et al. *Solid Protocol*. Dec. 2020. URL: <https://solidproject.org/TR/protocol> (visited on 10/24/2021) (cit. on p. 24).
- [74] Andrei Vlad Sambra et al. “Solid: A Platform for Decentralized Social Applications Based on Linked Data”. In: *MIT CSAIL & Qatar Computing Research Institute, Tech. Rep.* (2016), p. 16 (cit. on p. 24).
- [75] *Solid - P2P Foundation*. URL: <https://wiki.p2pfoundation.net/Solid> (visited on 10/24/2021) (cit. on p. 24).
- [76] Jonathan Robert Basamanowicz. “Release Groups and Digital Copyright Piracy”. Thesis. Arts & Social Sciences: School of Criminology, May 2011. DOI: [10/etd6644_JBasamanowicz.pdf](https://doi.org/10.1007/978-3-642-35668-1_19). URL: <https://summit.sfu.ca/item/11710> (visited on 10/01/2021) (cit. on p. 24).
- [77] Sameer Hinduja. “Deindividuation and Internet Software Piracy”. In: *CyberPsychology & Behavior* 11.4 (Aug. 2008), pp. 391–398. ISSN: 1094-9313. DOI: [10.1089/cpb.2007.0048](https://doi.org/10.1089/cpb.2007.0048). URL: <https://www.liebertpub.com/doi/abs/10.1089/cpb.2007.0048> (visited on 10/01/2021) (cit. on p. 24).
- [78] Ian Dunham. “What.CD: A Legacy of Sharing”. PhD thesis. Rutgers University - School of Graduate Studies, 2018. DOI: [10.7282/T3V128F3](https://doi.org/10.7282/T3V128F3). URL: <https://rucore.libraries.rutgers.edu/rutgers-lib/58981/> (visited on 03/16/2021) (cit. on pp. 24, 25).
- [79] Jody Rosen. “The Day the Music Burned”. In: *The New York Times* (June 2019). ISSN: 0362-4331. URL: <https://www.nytimes.com/2019/06/11/magazine/universal-fire-master-recordings.html> (visited on 03/18/2021) (cit. on p. 25).
- [80] Nikhil Sonnad. *A Eulogy for What.Cd, the Greatest Music Collection in the History of the World—until It Vanished*. Nov. 2016. URL: <https://qz.com/840661/what-cd-is-gone-a-eulogy-for-the-greatest-music-collection-in-the-world/> (visited on 03/16/2021) (cit. on p. 25).
- [81] M Meulpolder et al. “Public and Private BitTorrent Communities: A Measurement Study”. In: (), p. 5 (cit. on p. 25).
- [82] Adele Lu Jia et al. “How to Survive and Thrive in a Private BitTorrent Community”. In: *Distributed Computing and Networking*. Ed. by Davide Frey et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013, pp. 270–284. ISBN: 978-3-642-35668-1. DOI: [10.1007/978-3-642-35668-1_19](https://doi.org/10.1007/978-3-642-35668-1_19) (cit. on p. 25).
- [83] Z. Liu et al. “Understanding and Improving Ratio Incentives in Private Communities”. In: *2010 IEEE 30th International Conference on Distributed Computing Systems*. June 2010, pp. 610–621. DOI: [10.1109/ICDCS.2010.90](https://doi.org/10.1109/ICDCS.2010.90) (cit. on p. 25).
- [84] Ian A. Kash et al. “Economics of BitTorrent Communities”. In: *Proceedings of the 21st International Conference on World Wide Web*. WWW ’12. New York, NY, USA: Association for Computing Machinery, Apr. 2012, pp. 221–230. ISBN: 978-1-4503-1229-5. DOI: [10.1145/2187836.2187867](https://doi.org/10.1145/2187836.2187867). URL: <https://doi.org/10.1145/2187836.2187867> (visited on 03/28/2021) (cit. on p. 25).
- [85] 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)*. July 2011, pp. 1–6. DOI: [10.1109/ICCCN.2011.6005944](https://doi.org/10.1109/ICCCN.2011.6005944) (cit. on p. 25).
- [86] Benedikt Fecher et al. “A Reputation Economy: How Individual Reward Considerations Trump Systemic Arguments for Open Access to Data”. In: *Palgrave Communications* 3.1 (June 2017), pp. 1–10. ISSN: 2055-1045. DOI: [10.1057/palcomms.2017.51](https://doi.org/10.1057/palcomms.2017.51). URL: <https://www.nature.com/articles/palcomms201751> (visited on 10/01/2021) (cit. on p. 26).

- [87] Jordan Bross. “Community, Collaboration and Contribution: Evaluating a BitTorrent Tracker as a Digital Library.” M.S. in Library Science. UNC Chapel Hill, Dec. 2013. URL: <https://doi.org/10.17615/g1cw-kw06> (cit. on p. 26).
- [88] Morgan G. I. Langille and Jonathan A. Eisen. “BioTorrents: A File Sharing Service for Scientific Data”. In: *PLoS ONE* 5.4 (Apr. 2010). ISSN: 1932-6203. DOI: [10.1371/journal.pone.0010071](https://doi.org/10.1371/journal.pone.0010071). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2854681/> (visited on 04/01/2021) (cit. on pp. 26, 38).
- [89] Joseph Paul Cohen and Henry Z. Lo. “Academic Torrents: A Community-Maintained Distributed Repository”. In: *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment*. XSEDE '14. New York, NY, USA: Association for Computing Machinery, July 2014, pp. 1–2. ISBN: 978-1-4503-2893-7. DOI: [10.1145/2616498.2616528](https://doi.org/10.1145/2616498.2616528). URL: <https://doi.org/10.1145/2616498.2616528> (visited on 10/04/2021) (cit. on p. 26).
- [90] Matthew J. Bietz and Charlotte P. Lee. “Collaboration in Metagenomics: Sequence Databases and the Organization of Scientific Work”. In: *ECSCW 2009*. Ed. by Ina Wagner et al. London: Springer, 2009, pp. 243–262. ISBN: 978-1-84882-854-4. DOI: [10.1007/978-1-84882-854-4_15](https://doi.org/10.1007/978-1-84882-854-4_15) (cit. on p. 28).
- [91] Werner Ceusters and Barry Smith. “Foundations for a Realist Ontology of Mental Disease”. In: *Journal of Biomedical Semantics* 1.1 (Dec. 2010), p. 10. ISSN: 2041-1480. DOI: [10.1186/2041-1480-1-10](https://doi.org/10.1186/2041-1480-1-10). URL: <https://doi.org/10.1186/2041-1480-1-10> (visited on 10/07/2021) (cit. on p. 28).
- [92] The Biomedical Data Translator Consortium. “The Biomedical Data Translator Program: Conception, Culture, and Community”. In: *Clinical and Translational Science* 12.2 (2019), pp. 91–94. ISSN: 1752-8062. DOI: [10.1111/cts.12592](https://doi.org/10.1111/cts.12592). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cts.12592> (visited on 10/05/2021) (cit. on p. 29).
- [93] Stacia Fleisher. “Other Transaction Award Policy Guide - Biomedical Data Translator Program”. In: (Sept. 2019), p. 38 (cit. on p. 29).
- [94] The Biomedical Data Translator Consortium. “Toward A Universal Biomedical Data Translator”. In: *Clinical and Translational Science* 12.2 (2019), pp. 86–90. ISSN: 1752-8062. DOI: [10.1111/cts.12591](https://doi.org/10.1111/cts.12591). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cts.12591> (visited on 10/05/2021) (cit. on pp. 29, 30).
- [95] Richard Bruskiewich et al. *Bioblink/Biolink-Model*: 2.2.5. Zenodo. Sept. 2021. URL: <https://zenodo.org/record/5520104> (visited on 10/05/2021) (cit. on p. 29).
- [96] Prateek Goel et al. “Explanation Container in Case-Based Biomedical Question-Answering”. In: Sept. 2021, p. 10. URL: https://web.archive.org/web/*/https://gaia.fdi.ucm.es/events/xcb/papers/ICCBR_2021_paper_100.pdf (cit. on pp. 29, 31).
- [97] *ROBOKOP - CoVar*. Oct. 2021. URL: <https://web.archive.org/web/20211006030919/https://covar.com/case-study/robokop/> (visited on 10/06/2021) (cit. on pp. 29, 30).
- [98] A Ram et al. “Transphobia, Encoded: An Examination of Trans-Specific Terminology in SNOMED CT and ICD-10-CM”. In: *Journal of the American Medical Informatics Association* ocab200 (Sept. 2021). ISSN: 1527-974X. DOI: [10.1093/jamia/ocab200](https://doi.org/10.1093/jamia/ocab200). URL: <https://doi.org/10.1093/jamia/ocab200> (visited on 10/07/2021) (cit. on p. 30).
- [99] Ruth Hailu. *NIH-Funded Project Aims to Build a 'Google' for Biomedical Data*. July 2019. URL: <https://www.statnews.com/2019/07/31/nih-funded-project-aims-to-build-a-google-for-biomedical-data/> (visited on 10/06/2021) (cit. on p. 30).
- [100] Thomas Grote and Philipp Berens. “On the Ethics of Algorithmic Decision-Making in Healthcare”. In: *Journal of Medical Ethics* 46.3 (Mar. 2020), pp. 205–211. ISSN: 0306-6800, 1473-4257. DOI: [10.1136/medethics-2019-105586](https://doi.org/10.1136/medethics-2019-105586). URL: <https://jme.bmj.com/content/46/3/205> (visited on 10/07/2021) (cit. on p. 30).

- [101] Ziad Obermeyer et al. “Dissecting Racial Bias in an Algorithm Used to Manage the Health of Populations”. In: *Science* 366.6464 (Oct. 2019), pp. 447–453. DOI: [10.1126/science.aax2342](https://doi.org/10.1126/science.aax2342). URL: <https://www.science.org/doi/full/10.1126/science.aax2342> (visited on 10/07/2021) (cit. on p. 30).
- [102] Trishan Panch, Heather Mattie, and Rifat Atun. “Artificial Intelligence and Algorithmic Bias: Implications for Health Systems”. In: *Journal of Global Health* 9.2 (Nov. 2019), p. 020318. ISSN: 2047-2978. DOI: [10.7189/jogh.09.020318](https://doi.org/10.7189/jogh.09.020318). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6875681/> (visited on 10/07/2021) (cit. on p. 30).
- [103] Trishan Panch, Heather Mattie, and Leo Anthony Celi. “The “Inconvenient Truth” about AI in Healthcare”. In: *npj Digital Medicine* 2.1 (Aug. 2019), pp. 1–3. ISSN: 2398-6352. DOI: [10.1038/s41746-019-0155-4](https://doi.org/10.1038/s41746-019-0155-4). URL: <https://www.nature.com/articles/s41746-019-0155-4> (visited on 10/07/2021) (cit. on p. 30).
- [104] Melissa A Haendel. *A Common Dialect for Infrastructure and Services in Translator*. Feb. 2021. URL: <https://reporter.nih.gov/project-details/10330632> (visited on 10/07/2021) (cit. on p. 31).
- [105] Kyle R. Hukezalie et al. “In Vitro and Ex Vivo Inhibition of Human Telomerase by Anti-HIV Nucleoside Reverse Transcriptase Inhibitors (NRTIs) but Not by Non-NRTIs”. In: *PLoS ONE* 7.11 (Nov. 2012), e47505. ISSN: 1932-6203. DOI: [10.1371/journal.pone.0047505](https://doi.org/10.1371/journal.pone.0047505). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3499584/> (visited on 10/06/2021) (cit. on p. 31).
- [106] Zidovudine - Patient | NIH. URL: <https://clinicalinfo.hiv.gov/en/drugs/zidovudine/patient> (visited on 10/06/2021) (cit. on p. 31).
- [107] RePORT & RePORTER “Biomedical Data Translator”. Oct. 2021. URL: https://reporter.nih.gov/search/kDJ97zGUFaIBiltUmyd_Q/projects?sort_field=FiscalYear&sort_order=desc (visited on 10/05/2021) (cit. on p. 31).
- [108] AWS Announces AWS Healthcare Accelerator for Startups in the Public Sector. June 2021. URL: <https://aws.amazon.com/blogs/publicsector/aws-announces-healthcare-accelerator-program-startups-public-sector/> (visited on 10/06/2021) (cit. on p. 31).
- [109] Rachel Lerman. “Amazon Built Its Own Health-Care Service for Employees. Now It’s Selling It to Other Companies.” In: *Washington Post* (Mar. 2021). ISSN: 0190-8286. URL: <https://www.washingtonpost.com/technology/2021/03/17/amazon-healthcare-service-care-expansion/> (visited on 10/06/2021) (cit. on p. 31).
- [110] Corey Quinn. *You Can’t Trust Amazon When It Feels Threatened*. Mar. 2021. URL: <https://www.lastweekinaws.com/blog/you-cant-trust-amazon-when-it-feels-threatened/> (visited on 10/13/2021) (cit. on p. 31).
- [111] Dennis Heimbigner and Dennis McLeod. “A Federated Architecture for Information Management”. In: *ACM Transactions on Information Systems* 3.3 (July 1985), pp. 253–278. ISSN: 1046-8188. DOI: [10.1145/4229.4233](https://doi.org/10.1145/4229.4233). URL: <https://doi.org/10.1145/4229.4233> (visited on 03/25/2021) (cit. on p. 33).
- [112] Witold Litwin, Leo Mark, and Nick Roussopoulos. “Interoperability of Multiple Autonomous Databases”. In: *ACM Computing Surveys* 22.3 (Sept. 1990), pp. 267–293. ISSN: 0360-0300. DOI: [10.1145/96602.96608](https://doi.org/10.1145/96602.96608). URL: <https://doi.org/10.1145/96602.96608> (visited on 03/25/2021) (cit. on p. 33).
- [113] Vipul Kashyap and Amit Sheth. “Semantic and Schematic Similarities between Database Objects: A Context-Based Approach”. In: *The VLDB Journal* 5.4 (Dec. 1996), pp. 276–304. ISSN: 0949-877X. DOI: [10.1007/s007780050029](https://doi.org/10.1007/s007780050029). URL: <https://doi.org/10.1007/s007780050029> (visited on 03/25/2021) (cit. on p. 33).
- [114] 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. New York, NY, USA: Association for Computing Machinery, May 1997, pp. 51–61. ISBN: 978-0-89791-910-4. DOI: [10.1145/263661.263668](https://doi.org/10.1145/263661.263668). URL: <https://doi.org/10.1145/263661.263668> (visited on 03/24/2021) (cit. on p. 33).

- [115] Susanne Busse et al. “Federated Information Systems: Concepts, Terminology and Architectures”. In: (1999), p. 40 (cit. on p. 34).
- [116] Marija Djokic-Petrovic et al. “PIBAS FedSPARQL: A Web-Based Platform for Integration and Exploration of Bioinformatics Datasets”. In: *Journal of Biomedical Semantics* 8.1 (Sept. 2017), p. 42. ISSN: 2041-1480. DOI: [10.1186/s13326-017-0151-z](https://doi.org/10.1186/s13326-017-0151-z). URL: <https://doi.org/10.1186/s13326-017-0151-z> (visited on 05/07/2021) (cit. on pp. 34, 37).
- [117] Ali Hasnain et al. “BioFed: Federated Query Processing over Life Sciences Linked Open Data”. In: *Journal of Biomedical Semantics* 8.1 (Mar. 2017), p. 13. ISSN: 2041-1480. DOI: [10.1186/s13326-017-0118-0](https://doi.org/10.1186/s13326-017-0118-0). URL: <https://doi.org/10.1186/s13326-017-0118-0> (visited on 05/07/2021) (cit. on pp. 34, 37).
- [118] Michael Hanke et al. “In Defense of Decentralized Research Data Management”. In: *Neuroforum* 27.1 (Feb. 2021), pp. 17–25. ISSN: 1868-856X. DOI: [10.1515/nf-2020-0037](https://www.degruyter.com/document/doi/10.1515/nf-2020-0037/html). URL: <https://www.degruyter.com/document/doi/10.1515/nf-2020-0037/html> (visited on 10/13/2021) (cit. on pp. 34, 37, 38).
- [119] Amit P. Sheth and James A. Larson. “Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases”. In: *ACM Computing Surveys* 22.3 (Sept. 1990), pp. 183–236. ISSN: 0360-0300. DOI: [10.1145/96602.96604](https://doi.org/10.1145/96602.96604). URL: <https://doi.org/10.1145/96602.96604> (visited on 03/25/2021) (cit. on p. 34).
- [120] Angela Bonifati et al. “Distributed Databases and Peer-to-Peer Databases: Past and Present”. In: *ACM SIGMOD Record* 37.1 (Mar. 2008), pp. 5–11. ISSN: 0163-5808. DOI: [10.1145/1374780.1374781](https://doi.org/10.1145/1374780.1374781). URL: <https://doi.org/10.1145/1374780.1374781> (visited on 10/11/2021) (cit. on p. 34).
- [121] *Meatball Wiki: PersonalCategories*. URL: <http://meatballwiki.org/wiki/PersonalCategories> (visited on 10/22/2021) (cit. on p. 35).
- [122] Giuseppe Pirrò, Domenico Talia, and Paolo Trunfio. “A DHT-Based Semantic Overlay Network for Service Discovery”. In: *Future Generation Computer Systems* 28.4 (Apr. 2012), pp. 689–707. ISSN: 0167-739X. DOI: [10.1016/j.future.2011.11.007](https://www.sciencedirect.com/science/article/pii/S0167739X11002305). URL: <https://www.sciencedirect.com/science/article/pii/S0167739X11002305> (visited on 10/11/2021) (cit. on p. 35).
- [123] Christopher Webber et al. *ActivityPub*. W3C Recommendation. W3C, Jan. 2018. URL: <https://www.w3.org/TR/2018/REC-activitypub-20180123/> (cit. on p. 35).
- [124] Manu Sporny et al. *JSON-LD 1.1 - A JSON-Based Serialization for Linked Data*. July 2020. URL: <https://www.w3.org/TR/json-ld/> (visited on 10/12/2021) (cit. on p. 35).
- [125] James M Snell and Evan Prodromou. *Activity Streams 2.0*. May 2017. URL: <https://www.w3.org/TR/activitystreams-core/> (visited on 10/12/2021) (cit. on p. 35).
- [126] *SPARQL 1.1 Federated Query*. Mar. 2013. URL: <https://www.w3.org/TR/sparql11-federated-query/> (visited on 10/13/2021) (cit. on p. 37).
- [127] Ana Claudia Sima et al. “Enabling Semantic Queries across Federated Bioinformatics Databases”. In: *Database* 2019.baz106 (Jan. 2019). ISSN: 1758-0463. DOI: [10.1093/database/baz106](https://doi.org/10.1093/database/baz106). URL: <https://doi.org/10.1093/database/baz106> (visited on 03/30/2021) (cit. on pp. 37, 54).
- [128] Yaroslav O. Halchenko et al. “DataLad: Distributed System for Joint Management of Code, Data, and Their Relationship”. In: *Journal of Open Source Software* 6.63 (July 2021), p. 3262. ISSN: 2475-9066. DOI: [10.21105/joss.03262](https://joss.theoj.org/papers/10.21105/joss.03262). URL: <https://joss.theoj.org/papers/10.21105/joss.03262> (visited on 10/02/2021) (cit. on p. 38).
- [129] Jonny L. Saunders and Michael Wehr. “Autopilot: Automating Behavioral Experiments with Lots of Raspberry Pis”. In: *bioRxiv* (Oct. 2019), p. 807693. DOI: [10.1101/807693](https://www.biorxiv.org/content/10.1101/807693v1). URL: <https://www.biorxiv.org/content/10.1101/807693v1> (visited on 03/12/2021) (cit. on pp. 38, 43, 45).

- [130] Jeffrey Spies. “A Workflow-Centric Approach to Increasing Reproducibility and Data Integrity”. In: (Aug. 2017). URL: <https://scholarworks.iu.edu/dspace/handle/2022/21729> (visited on 10/01/2021) (cit. on p. 39).
- [131] Charles R. Harris et al. “Array Programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. ISSN: 1476-4687. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). URL: <https://www.nature.com/articles/s41586-020-2649-2> (visited on 10/13/2021) (cit. on p. 39).
- [132] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17.3 (Mar. 2020), pp. 261–272. ISSN: 1548-7105. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2). URL: <https://www.nature.com/articles/s41592-019-0686-2> (visited on 10/13/2021) (cit. on p. 39).
- [133] Fabian Pedregosa et al. “Scikit-Learn: Machine Learning in Python”. In: *The Journal of Machine Learning Research* 12.null (Nov. 2011), pp. 2825–2830. ISSN: 1532-4435 (cit. on p. 39).
- [134] Alexander B. Wiltschko et al. “Revealing the Structure of Pharmacobehavioral Space through Motion Sequencing”. In: *Nature Neuroscience* 23.11 (Nov. 2020), pp. 1433–1443. ISSN: 1546-1726. DOI: [10.1038/s41593-020-00706-3](https://doi.org/10.1038/s41593-020-00706-3). URL: <https://www.nature.com/articles/s41593-020-00706-3> (visited on 10/13/2021) (cit. on p. 39).
- [135] Kevin R. Coffey, Russell G. Marx, and John F. Neumaier. “DeepSqueak: A Deep Learning-Based System for Detection and Analysis of Ultrasonic Vocalizations”. In: *Neuropsychopharmacology* 44.5 (Apr. 2019), pp. 859–868. ISSN: 1740-634X. DOI: [10.1038/s41386-018-0303-6](https://doi.org/10.1038/s41386-018-0303-6). URL: <https://www.nature.com/articles/s41386-018-0303-6> (visited on 10/13/2021) (cit. on p. 39).
- [136] Dimitri Yatsenko, Edgar Y. Walker, and Andreas S. Tolias. “DataJoint: A Simpler Relational Data Model”. In: *arXiv:1807.11104 [cs]* (July 2018). arXiv: [1807.11104 \[cs\]](https://arxiv.org/abs/1807.11104). URL: <http://arxiv.org/abs/1807.11104> (visited on 10/14/2021) (cit. on p. 40).
- [137] Dimitri Yatsenko et al. “DataJoint Elements: Data Workflows for Neurophysiology”. In: *bioRxiv* (Mar. 2021), p. 2021.03.30.437358. DOI: [10.1101/2021.03.30.437358](https://doi.org/10.1101/2021.03.30.437358). URL: <https://www.biorxiv.org/content/10.1101/2021.03.30.437358v1> (visited on 04/30/2021) (cit. on p. 40).
- [138] Marius Pachitariu et al. *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. Chap. New Results, p. 061481. DOI: [10.1101/061481](https://doi.org/10.1101/061481). URL: <https://www.biorxiv.org/content/10.1101/061481v1> (visited on 10/15/2021) (cit. on p. 40).
- [139] W. M. P. van der Aalst and A. H. M. ter Hofstede. “YAWL: Yet Another Workflow Language”. In: *Information Systems* 30.4 (June 2005), pp. 245–275. ISSN: 0306-4379. DOI: [10.1016/j.is.2004.02.002](https://doi.org/10.1016/j.is.2004.02.002). URL: <https://www.sciencedirect.com/science/article/pii/S0306437904000304> (visited on 10/16/2021) (cit. on p. 41).
- [140] Michael Wulf. *BEADL XML Documentation V 0.1*. July 2020. URL: <http://archive.org/details/beadl-xml-documentation-v-0.1> (visited on 11/04/2021) (cit. on p. 44).
- [141] NWB Behavioral Task WG. *NWB Behavioral Task WG*. Apr. 2020. URL: <http://archive.org/details/nwb-behavioral-task-wg> (visited on 11/05/2021) (cit. on p. 44).
- [142] Gonalo Lopes et al. “Bonsai: An Event-Based Framework for Processing and Controlling Data Streams”. In: *Frontiers in Neuroinformatics* 9 (2015), p. 7. ISSN: 1662-5196. DOI: [10.3389/fninf.2015.00007](https://doi.org/10.3389/fninf.2015.00007). URL: <https://www.frontiersin.org/article/10.3389/fninf.2015.00007> (visited on 11/09/2021) (cit. on p. 46).
- [143] Gonalo Lopes and Patricia Monteiro. “New Open-Source Tools: Using Bonsai for Behavioral Tracking and Closed-Loop Experiments”. In: *Frontiers in Behavioral Neuroscience* 15 (2021), p. 53. ISSN: 1662-5153. DOI: [10.3389/fnbeh.2021.647640](https://doi.org/10.3389/fnbeh.2021.647640). URL: <https://www.frontiersin.org/article/10.3389/fnbeh.2021.647640> (visited on 11/09/2021) (cit. on p. 46).

- [144] Francesc Alté and Mercedes Fernández-Alonso. “PyTables: Processing And Analyzing Extremely Large Amounts Of Data In Python”. In: *PyCon2003*. Apr. 2003, p. 9 (cit. on p. 47).
- [145] Greg Miller. “A Scientist’s Nightmare: Software Problem Leads to Five Retractions”. In: *Science* 314.5807 (Dec. 2006), pp. 1856–1857. ISSN: 0036-8075, 1095-9203. DOI: [10.1126/science.314.5807.1856](https://doi.org/10.1126/science.314.5807.1856). URL: <https://science.sciencemag.org/content/314/5807/1856> (visited on 04/07/2021) (cit. on p. 49).
- [146] David A. W. Soergel. “Rampant Software Errors May Undermine Scientific Results”. In: *F1000Research* 3 (July 2015). ISSN: 2046-1402. DOI: [10.12688/f1000research.5930.2](https://doi.org/10.12688/f1000research.5930.2). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4629271/> (visited on 04/07/2021) (cit. on p. 49).
- [147] Anders Eklund, Thomas E. Nichols, and Hans Knutsson. “Cluster Failure: Why fMRI Inferences for Spatial Extent Have Inflated False-Positive Rates”. In: *Proceedings of the National Academy of Sciences* 113.28 (July 2016), pp. 7900–7905. ISSN: 0027-8424, 1091-6490. DOI: [10.1073/pnas.1602413113](https://doi.org/10.1073/pnas.1602413113). URL: <https://www.pnas.org/content/113/28/7900> (visited on 04/07/2021) (cit. on p. 49).
- [148] Jayanti Bhandari Neupane et al. “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”. In: *Organic Letters* 21.20 (Oct. 2019), pp. 8449–8453. ISSN: 1523-7060. DOI: [10.1021/acs.orglett.9b03216](https://doi.org/10.1021/acs.orglett.9b03216). URL: <https://doi.org/10.1021/acs.orglett.9b03216> (visited on 04/07/2021) (cit. on p. 49).
- [149] Gary A Kane et al. “Real-Time, Low-Latency Closed-Loop Feedback Using Markerless Posture Tracking”. In: *eLife* 9 (Dec. 2020). Ed. by Gordon J Berman et al., e61909. ISSN: 2050-084X. DOI: [10.7554/eLife.61909](https://doi.org/10.7554/eLife.61909). URL: <https://doi.org/10.7554/eLife.61909> (visited on 07/30/2021) (cit. on p. 49).
- [150] RELX Annual Report 2019. 2019. URL: <https://www.relx.com/~media/Files/R/RELX-Group/documents/reports/annual-reports/2019-annual-report.pdf> (visited on 11/03/2021) (cit. on p. 51).
- [151] Springer Nature. *Branded Content*. URL: <https://partnerships.nature.com/product/branded-content-native-advertising/> (visited on 11/11/2021) (cit. on p. 51).
- [152] Elsevier. *360° Advertising Solutions | Advertising | Advertisers*. URL: <https://www.elsevier.com/advertising-reprints-supplements/advertising> (visited on 11/11/2021) (cit. on p. 51).
- [153] Elsevier. *Drug Design Optimization*. URL: <https://www.elsevier.com/solutions/professional-services/drug-design-optimization> (visited on 11/11/2021) (cit. on p. 51).
- [154] Elsevier. *Topic Prominence in Science - Scival | Elsevier Solutions*. URL: <https://www.elsevier.com/solutions/scival/features/topic-prominence-in-science> (visited on 11/11/2021) (cit. on p. 51).
- [155] Johan S. G. Chu and James A. Evans. “Slowed Canonical Progress in Large Fields of Science”. In: *Proceedings of the National Academy of Sciences* 118.41 (Oct. 2021). ISSN: 0027-8424, 1091-6490. DOI: [10.1073/pnas.2021636118](https://doi.org/10.1073/pnas.2021636118). URL: <https://www.pnas.org/content/118/41/e2021636118> (visited on 10/24/2021) (cit. on p. 51).
- [156] James Heathers. *The Real Scandal About Ivermectin*. Oct. 2021. URL: <https://www.theatlantic.com/science/archive/2021/10/ivermectin-research-problems/620473/> (visited on 10/24/2021) (cit. on p. 52).
- [157] Helen Shen. “Meet This Super-Spotter of Duplicated Images in Science Papers”. In: *Nature* 581.7807 (May 2020), pp. 132–136. DOI: [10.1038/d41586-020-01363-z](https://doi.org/10.1038/d41586-020-01363-z). URL: <https://www.nature.com/articles/d41586-020-01363-z> (visited on 10/24/2021) (cit. on p. 52).
- [158] Elisabeth M. Bik, Arturo Casadevall, and Ferric C. Fang. “The Prevalence of Inappropriate Image Duplication in Biomedical Research Publications”. In: *mBio* 7.3 (June 2016), e00809–16. DOI: [10.1128/mBio.00809-16](https://doi.org/10.1128/mBio.00809-16). URL: <https://journals.asm.org/doi/10.1128/mBio.00809-16> (visited on 10/24/2021) (cit. on p. 52).
- [159] Samantha R. White et al. “The Future Is Open: Open-Source Tools for Behavioral Neuroscience Research”. In: *eNeuro* 6.4 (Aug. 2019), ENEURO.0223–19.2019. ISSN: 2373-2822. DOI: [10.1523/ENEURO.0223-19.2019](https://doi.org/10.1523/ENEURO.0223-19.2019). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6712209/> (visited on 11/11/2021) (cit. on p. 52).

- [160] Aaron Swartz. *The Techniques of Mass Collaboration: A Third Way Out (Aaron Swartz's Raw Thought)*. July 2006. URL: <http://www.aaronsw.com/weblog/masscollab2> (visited on 10/22/2021) (cit. on p. 53).
- [161] Candace Lombardi. *Google Exec Challenges Berners-Lee* | CNET News.Com. Jan. 2007. URL: https://web.archive.org/web/20070105030625/http://news.com.com/Google+exec+challenges+Berners-Lee/2100-1025_3-6095705.html (visited on 10/22/2021) (cit. on p. 53).
- [162] Bo Leuf and Ward Cunningham. *The Wiki Way : Quick Collaboration on the Web*. Boston : Addison-Wesley, 2001. ISBN: 978-0-201-71499-9. URL: http://archive.org/details/isbn_9780201714999 (visited on 10/22/2021) (cit. on p. 53).
- [163] Aaron Swartz. *Secrets of Standards*. Aug. 2003. URL: <http://www.aaronsw.com/weblog/001027> (visited on 10/20/2021) (cit. on p. 54).
- [164] 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*. Tweet. Oct. 2021. URL: https://twitter.com/beka_valentine/status/1454522998594043906 (visited on 11/09/2021) (cit. on p. 54).
- [165] Maged Kamel Boulos. "Semantic Wikis: A Comprehensible Introduction with Examples from the Health Sciences". In: *Journal of Emerging Technologies in Web Intelligence* 1 (Aug. 2009). DOI: [10.4304/jetwi.1.1.94-96](https://doi.org/10.4304/jetwi.1.1.94-96) (cit. on p. 54).
- [166] Tadeu Classe et al. "A Distributed Infrastructure to Support Scientific Experiments". In: *Journal of Grid Computing* 15.4 (Dec. 2017), pp. 475–500. ISSN: 1572-9184. DOI: [10.1007/s10723-017-9401-7](https://doi.org/10.1007/s10723-017-9401-7). URL: <https://doi.org/10.1007/s10723-017-9401-7> (visited on 03/09/2021) (cit. on pp. 54, 55).
- [167] Benjamin M. Good, Joseph T. Tennis, and Mark D. Wilkinson. "Social Tagging in the Life Sciences: Characterizing a New Metadata Resource for Bioinformatics". In: *BMC Bioinformatics* 10.1 (Sept. 2009), p. 313. ISSN: 1471-2105. DOI: [10.1186/1471-2105-10-313](https://doi.org/10.1186/1471-2105-10-313). URL: <https://doi.org/10.1186/1471-2105-10-313> (visited on 04/08/2021) (cit. on p. 54).
- [168] Kei-Hoi Cheung et al. "Semantic Web Approach to Database Integration in the Life Sciences". In: *Semantic Web*. Ed. by Christopher J. O. Baker and Kei-Hoi Cheung. Boston, MA: Springer US, 2007, pp. 11–30. ISBN: 978-0-387-48436-5. DOI: [10.1007/978-0-387-48438-9_2](https://doi.org/10.1007/978-0-387-48438-9_2). URL: http://link.springer.com/10.1007/978-0-387-48438-9_2 (visited on 03/30/2021) (cit. on p. 54).