EMERGENT STANDARDS: ENABLING COLLABORATION ACROSS INSTITUTIONS

JED SUNDWALL

WHITE PAPER #003

OCTOBER 2025

THE INSTITUTIONAL ARCHITECTURE LAB

WHAT IS TIAL

The Institutional Architecture Lab was formed in 2023 by Sir Geoff Mulgan, Jessica Seddon and Juha Leppänen in an effort to help the institutional design community coalesce, learn together, and grow. Each of us has been involved in various stages of creating new organisations and other institutions. Like many other people, we have witnessed first-hand the absence of a formal community along the way — or a place where we can learn from past experience. We are aware that there is a lot of great work happening around the world, but nowhere to recognise it.

AUTHOR

Jed Sundwall
Executive Director, Radiant Earth

Layout and cover: Kirmo Kivelä
Figure: David Szakaly (davidope)

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The COVID-19 pandemic revealed a striking lack of global data coordination among public institutions. While national governments and the World Health Organization struggled to collect and share timely information, a small team at Johns Hopkins University built a public dashboard that became the world's most trusted source of COVID-19 data. Their work showed that planetary-scale data infrastructure can emerge quickly when practitioners act, even without formal authority. It also exposed a deeper truth: we cannot build global coordination on data without shared standards, yet it is unclear who gets to define what those standards might be.

This paper examines how the World Wide Web has created the conditions for useful data standards to emerge in the absence of any clear authority. It begins from the premise that **standards are** not just technical artefacts but **the "connective tissue" that enables collaboration across institutions**. They extend language itself, allowing people and systems to describe the world in compatible terms. Using the history of the web, the paper traces how small, loosely organised groups have repeatedly developed standards that now underpin global information exchange.

Four examples illustrate this pattern:

— **RSS** (Really Simple Syndication) allowed information to flow across websites and became the backbone of today's content distribution systems.

— **Markdown** simplified web publishing by turning plain text into structured documents.

— **GTFS** (General Transit Feed Specification) provided a common format for transit data that allowed agencies around the world to reach riders through applications like Google Maps.

— **STAC** (SpatioTemporal Asset Catalogs) did the same for satellite imagery, creating a shared structure for vast Earth observation datasets.

Each arose from practitioners solving specific problems. They succeeded because they **were open**, **simple**, and **immediately useful**.

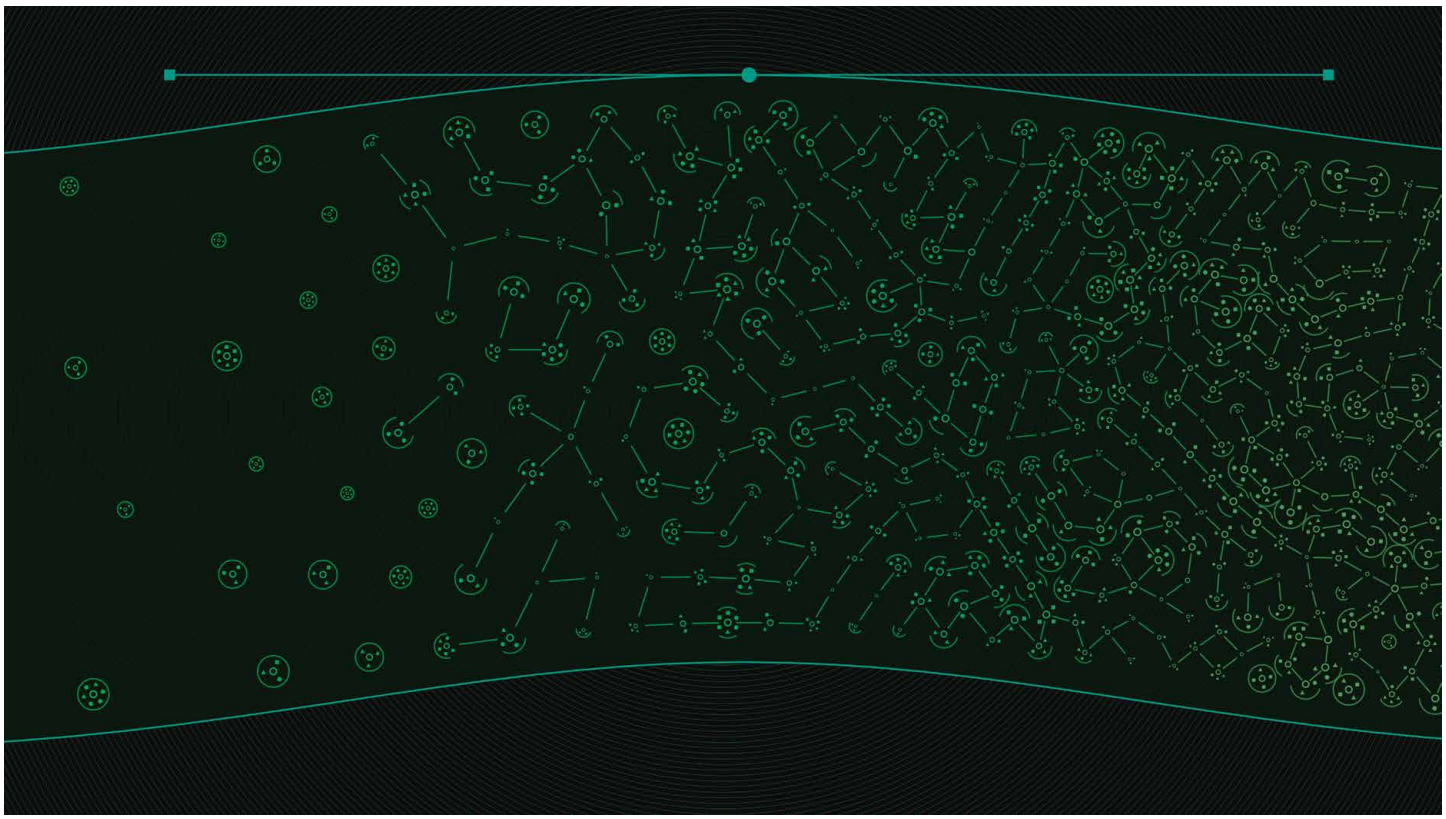
From these cases, several principles emerge:

1. Successful standards evolve from small, functioning systems rather than grand designs. They remain lightweight and adaptable. Their creators are practitioners who understand real constraints.

2. Once these standards prove useful, their adoption by credible institutions transforms them into durable infrastructure. RSS gained momentum when the New York Times adopted it; GTFS scaled globally once Google Maps embraced it; STAC became a fixture of Earth data management when NASA and Microsoft implemented it.

3. Institutions seeking to coordinate around shared data – whether in health, climate, or other global challenges – should learn from these examples. Instead of designing top-down systems, they should create conditions in which small groups can build minimal, working standards that others can adopt. This involves funding collaborative sprints, supporting open formats, and adopting procurement policies that favour interoperability over proprietary control.

Figure. Data standards emerge from simplicity and usefulness until they become durable infrastructure.



DATA STANDARDS: THE SOCIAL FOUNDATION OF INSTITUTIONAL COLLABORATION

During the early days of COVID-19, it became clear that established public health institutions were not equipped to gather or share global public health data. While nation states and the World Health Organization scrambled, a small group at The Johns Hopkins University (JHU) Center for Systems Science and Engineering created a pandemic dashboard that received over 1 billion visits per day. The JHU dashboard quickly became an authoritative source for global COVID-19 data used worldwide.

About six months after the launch of the JHU dashboard, its creators published a short letter in *The Lancet* titled "A need for open public data standards and sharing in light of COVID-19."[1] In it, they call for the creation of "a standardised reporting system for systematically collecting, visualising, and sharing high-quality data on emerging infectious and notifiable diseases in real-time is established." The letter explains the challenges of creating such a system, which include "the ambiguity of parameter definitions," "discrepancies in reporting across sources," "inconsistencies in reporting medium," "adaptability to changes in reporting over time," and "privacy concerns." They propose that the system be "hosted by an apolitical organisation, sufficiently resourced and empowered, and supported by a global community of public health experts."

While it is hard to disagree with the sentiment of the letter, it describes an impossible task. Research on government intelligence during COVID-19 supports this[2]: even basic data like death counts were defined and recorded differently across countries, creating exactly the kind of coordination failures the JHU team identified. A study of 16 countries found "very varied ability of countries to share information and data between tiers of government," with tensions evident even within individual nations. In federal systems, intelligence from different regions – such as Swiss cantons and German Länder – was not standardized and used different technical interfaces.

As the researchers noted, a fundamental question is "Who should be empowered to set standards, particularly in federal systems which are supposed to have decentralized power?" If coordination proved difficult between national and local governments within the same country, coordinating across governments around the world becomes far more difficult.

> *"We can't create carbon markets, track biodiversity loss, monitor deforestation, or respond effectively to climate change without broad agreement on how to share data."*

The need for data standards demonstrated by COVID-19 is a perennial concern that affects coordination on global challenges. We can't create carbon markets, track biodiversity loss, monitor deforestation, or respond effectively to climate change without broad agreement on how to share data.

Four years since the JHU team published their letter, the creation of such a system seems less plausible than ever. As of 2022, the US provided almost one third of all global health assistance[3], but much of that funding is now at risk. JHU, along with many US universities and public health experts, are contending with substantial federal funding cuts[4]. Global alliances are more brittle than they were pre-pandemic, and public health is increasingly subject to political pressure.

Despite this, the JHU dashboard proves that sharing useful data at planetary scale is possible. A small group at one university quickly created and maintained a dashboard used worldwide. The dashboard became authoritative because it was practically useful, not because it had the imprimatur of any government. It showed that motivated practitioners can create planetary-scale data infrastructure, even if only temporarily. As nation states become less willing to fund or collaborate on global challenges, our task is to create conditions that help practitioners share data quickly and build shared understanding.

This requires understanding the social foundations of institutional architecture rather than dreaming up well-funded apolitical organizations that might solve our problems. All human organization relies on social foundations such as standardized weights and measures, shared methods of timekeeping, and language itself. Data standards extend language to describe our world more precisely, enabling data-driven collaboration across institutions. More concretely, they are like standard screw drives[5] that enable the interoperability of construction tools. Anyone seeking to create new data institutions needs to understand the origins of data standards.

The Internet underpins the JHU dashboard, and this paper examines how useful data standards have repeatedly emerged on the world wide web without a central authority. The web's short history reveals that loosely coordinated actors can develop shared languages for exchanging data that are more agile and responsive to real-world needs than top-down approaches that consistently fail. This paper explains how and offers practical recommendations for cultivating conditions in which such standards can emerge more deliberately.

LANGUAGES OF THE WEB: HOW THE WORLD WIDE WEB INCUBATES DATA STANDARDS

On August 6, 1991, Tim Berners-Lee sent a message titled “WorldWideWeb: Summary” to a group of researchers working on a concept called “hypertext.”[6] He described the web as a project “to make an easy but powerful global information system.” Berners-Lee's message is short, but it describes a few foundational concepts that combined to make the web such an explosive innovation.

He explains that the web “consists of documents, and links” and that a “browser” can be used to load and view documents shared on the web. If a user had a mouse (not assumed in 1991), they could click links within documents to go to other documents. Anyone capable of connecting a computer to the Internet could publish documents on the WWW if they use a standard “markup language,” a way to encode text so computers can format it and include links. His intention was to create a system that would require little effort from his peers to “allow small contributions.” His message concludes with links to a demonstration and documentation for readers to learn how to contribute.

A technical way of describing this vision is that Berners-Lee proposed a “client-server” architecture. In this architecture, anyone with a computer (i.e. a server) could publish content. This content could be requested and read by anyone else with a browser (i.e. a client) on the same network. For this to work, the server needs to understand the requests from the client, and the client needs to understand the server's response. That is, they both needed to speak the same language, the “markup language” Berners-Lee referred to.

The first foundational concept worth noting is that Berners-Lee and his employer (CERN, the European Organization for Nuclear Research) claimed no ownership of the web. His message was an open invitation to collaborators. Anyone was, and still is, free to copy and use the software to run a web server. In contrast to other proprietary and more rigid communication systems[7], the web enabled anyone to find ways to communicate using markup language. As more researchers and individuals gained access to computers, all were free to experiment with publishing on the web.

The second foundational concept is that the web wasn't designed around any particular kind of information. While the first application of the web was to enable researchers to share academic documents and data with each other, Berners-Lee realized it could be used for much more. He said that it “should provide a basis for extension into multimedia, and allow those who share application standards to make full use of them across the web.” Thirty-four years later, he was right. The web is now used to share zettabytes of data[8], from simple text documents and memes to massive corpora of genomic and climate data.

> *"A complex system that works is invariably found to have evolved from a simple system that worked. A complex system designed from scratch never works and cannot be patched up to make it work."*

The key phrase in Berners-Lee's message is "those who share application standards." The web is a simple system that functions only if data providers and data receivers agree on standards of how data should be structured. As Berners-Lee puts it, "The WWW model gets over the frustrating incompatibilities of data format between suppliers and readers by allowing negotiation of format between a smart browser and a smart server." This architecture motivated data publishers to follow standards that would allow them to reach anyone with a browser. In other words, the web became a mechanism that helped people quickly find shared languages to describe data.

As more people, computers, and sensors are added to the web, new languages emerge. Anyone who has used social media has experienced this directly. The act of "liking" content online is a new form of communication with humans, algorithms, and networks made possible by the web. The ubiquity of emojis is another example of a new language made possible by humans on networked computers (it's hard for me to remember going through life without the expressive power afforded by the upside-down smile emoji).

When the creators of the JHU COVID-19 dashboard called for a "standardized reporting system," they were right to identify "ambiguity of parameter definitions," "discrepancies in reporting across sources," and "inconsistencies in reporting medium" as challenges. However, these are precisely the kinds of linguistic challenges the web can solve over time if used properly. The first step to creating shared data infrastructure is getting collaborators to agree on how to describe and exchange data. Once this is done, functional systems can emerge. This rule of system design is encapsulated in Gall's Law[9]:

A complex system that works is invariably found to have evolved from a simple system that worked. A complex system designed from scratch never works and cannot be patched up to make it work. You have to start over with a working simple system.

The complex systems built upon the web today corroborate Gall's Law. Nearly every data dashboard, app, social media site, and crypto scheme on the Internet today is built upon the basis of the flexible client-server architecture of the web. Failure to understand this dynamic leads to wasting time and money creating (or waiting for someone to create) top-down systems that are frequently myopic, inflexible, and expensive to maintain.

To understand how languages emerge on the web, we briefly review four data standards that have gained traction. The following sections refer to a combination of standards, protocols, patterns, file formats, schemas, naming conventions, and languages, but amidst the jargon are stories of people working together (and often arguing) to organize data in ways that are maximally flexible and accessible. Finding structures others can adopt worldwide is hard, but possible.

RSS: SYNDICATED DATA AS SHARED INFRASTRUCTURE

Really Simple Syndication (RSS) emerged in the late 1990s to address a problem created by a rapidly growing WWW – users were drowning in content. With websites, blogs, and news sites multiplying rapidly and being updated frequently, there was no efficient way for users to keep up. Visiting each site manually to check for new content was time-consuming and impractical, particularly before broadband connections. Content creators faced a similar problem: they had no simple way to notify their audience when new articles or posts were published. To solve this, various attempts were mounted throughout the late 1990s to enable the "syndication" of content on the web, notifying users when their favorite websites were updated.

RSS solved this problem by creating a new kind of document called an *RSS feed*. Data publishers could use these feeds to notify their audience when new content was available. An RSS feed requires that content be organized into discrete items along with their time and date of publication (such as a blog post or news story). Users could subscribe to RSS feeds to receive updates from multiple websites in a single location. This let users stay informed without the hassle of manually checking websites. It also allowed publishers to reach an audience of subscribers rather than intermittent visitors wandering through the web. RSS became the backbone of web syndication, laying the groundwork for modern content distribution systems.

Reading an RSS feed requires a special kind of browser called an RSS reader that displays feeds of new content rather than simple web pages. In the early 2000s, RSS readers were built into web browsers and email applications, but they have become less common in favor of algorithmic feeds created by social media platforms. Despite this, RSS remains relevant as the technology that powers blog subscriptions, services like Substack and Ghost, and, notably, all podcast subscriptions. A podcast app is a special kind of RSS reader that allows users to subscribe to RSS feeds published by podcast producers – whenever a podcast app is opened, it checks feeds for updates and displays new podcast episodes. Podcast apps rely on RSS, enabling podcasters to invite listeners to subscribe "wherever [they] get [their] podcasts"[10].

WHO CREATED RSS?

The history of RSS is well documented[11], but there is no clear answer to who actually invented it.

RSS was initially developed by Netscape in 1999 as "Rich Site Summary" to syndicate content to the My.Netscape.com web portal. Netscape ultimately abandoned the idea, but enough people found the concept compelling that they continued to promote it, creating tools to both produce and read RSS feeds. Dave Winer is credited with evangelizing the concept among bloggers and other publishers. After disagreements about implementation details, Winer proposed a simplified version of RSS (now renamed to Really Simple Syndication) which became the de facto standard for web syndication around November 2002, when the New York Times started offering RSS feeds of its content.

An abbreviated history of RSS is this: an assortment of programmers from different companies argued about web syndication for about eight years, and at some point it seemed that RSS was good-enough.

WHY DID RSS TAKE OFF?

RSS files are written using Extensible Markup Language (XML), which was a widely used and understood method of structuring information at the time RSS was developed. By using existing methods and keeping the standard lightweight, RSS was accessible to both developers and users. Developers didn't need specialized technical knowledge or resources to implement it, and users could easily subscribe to feeds without diving into the technical details. This balance between simplicity and functionality helped RSS spread rapidly and became the foundation for content syndication on the web.

Also important is the fact that many businesses built on the early web benefited from the structure provided by RSS. Ramanathan V. Guha, one of the early contributors to RSS at Netscape, explained to us that “RSS brought together Microsoft, Yahoo, and a whole lot of others to create this standard. RSS allowed search engineers to get structured data onto their pages, which increased click-through rates on those searches. That, in turn, incentivized websites and platforms to get behind RSS.” In other words, RSS proved itself as a technology that could help consumers find the content they wanted. This helped publishers reach more consumers.

Guha continued: “The magic happens when you combine a need that a lot of people share with the technology that addresses that need. If you want to get tens of thousands of people behind your standard, you have to have an incentive. Someone has to justify the engineering work.” Netscape's interest in aggregating data into its own web portal funded the foundation of RSS and Internet users' desire for faster access to information fueled adoption of it.

Enlisting a standards or policy-making organization to do the heavy lifting might sound like a good strategy, but it's one that usually fails, Guha says. “They often come with extremely complicated requirements of their own, which can throttle progress and stifle adoption.” Indeed, around the same time RSS emerged, another more comprehensive standard for web syndication called Information and Content Exchange (ICE) was created with help from Microsoft, Adobe, Sun, CNET, National Semiconductor, Tribune Media Services, Ziff Davis and Reuters. Adoption of ICE was hampered by its complexity and the simpler RSS standard prevailed.

In the end, RSS had no clear corporate owner or backer, which gave it a distinct advantage because no one could use it as a proprietary tool. Its evangelists were able to champion RSS as a shared solution without locking others out. RSS emerged out of an alignment between a widespread need, a good-enough solution, and just enough sponsorship and adoption by influential individuals and organizations.

MARKDOWN: MINIMAL SYNTAX, BROAD ADOPTION

While RSS emerged to solve the problem of content syndication, another challenge of the early web was the complexity of creating content.

Tim Berners-Lee's first message about the web stated that adding content to the web was “as simple as writing a few SGML files which point to your existing data.” SGML stands for Standard Generalized Markup Language, and it spawned Hypertext Markup Language (HTML) which is the standard used to display most documents in web browsers today. Markup languages define how computers understand how to format or structure a document that is sent over a network. While markup languages like HTML and XML were suitable for early adopters of the web, most people are not likely to have the time, interest, or need to learn them. The complexity of creating web content made it prohibitive for many people to contribute.

Markdown is a markup language created by John Gruber in 2004 to make it easier for more people to create well-structured documents for the web. Gruber explains on his website that Markdown is intended to allow web writers to “write using an easy-to-read, easy-to-write plain text format.”[12] Markdown simply specifies how a writer should structure a plain text document so that it can be converted into Hypertext Markup Language (HTML, which is a type of SGML) that can be displayed as a web page.

While HTML is designed to accommodate a wide array of layouts and customizations for interactive websites, Markdown focuses on the narrow subset of HTML's capabilities that are useful for writing clear documents. To emphasize text in HTML, a writer needs to enclose the text within HTML emphasis tags like `<em>this text should be emphasized</em>`. In Markdown, the same emphasis can be achieved by enclosing text within asterisks: `*this text should be emphasized*`.

This simplicity is by design. Markdown provides a simple way for humans to format text that computers can understand.

HOW DID MARKDOWN BECOME A STANDARD?

Today, Markdown syntax is used to format text by many applications on the web including GitHub, Discourse, Microsoft Teams, and Google Docs. The text produced by ChatGPT is formatted using Markdown. In June 2024, a writer at Wired speculated that the code used to convert Markdown formatted text into HTML might be the most ubiquitous code on the web[13].

Despite its wide adoption and practical value, Markdown is not governed as a formal standard by anyone. The only official version is 1.0.1, published by Gruber on his website in December 2004.

Markdown's ubiquity is rooted in its ability to ensure that content written in one format can be displayed consistently across platforms and publishing tools. By using plain text and established conventions, Markdown made web writing more approachable.

This document was drafted in Markdown.

GTFS: THE LINGUA FRANCA OF TRANSIT AGENCIES WORLDWIDE

The stories of RSS and Markdown demonstrate how early web developers coordinated to solve issues faced by everyone creating human-readable content on the web. However, the web's potential extends beyond sharing documents and blog posts. The *General Transit Feed Specification* (GTFS) shows how the web became a tool to improve public transit.

If you've ever used Google or Apple Maps to plan a trip using public transportation, you've benefited from GTFS. GTFS began in 2005 at Portland, Oregon's TriMet transit agency, where employees Tim and Bibiana McHugh (yes, they're married) recognized that transit data should be easier to access on mobile phones. They started to publish TriMet's transit data openly for Google Maps and others to repurpose. Google, eager to enhance its Google Maps platform, collaborated with TriMet to create a standardized format that other transit agencies could adopt. GTFS emerged as the answer.

As RSS uses XML files and Markdown uses plain text files, GTFS uses CSV files, one of the simplest and most universally understood data formats. One goal of GTFS was to make transit data accessible and usable by agencies of all sizes, not just those with large technical teams, and using CSVs put GTFS within reach of anyone who knew how to create a spreadsheet.

By sticking to the basics – routes, schedules, and stops – GTFS provided enough structure to standardize transit data without overwhelming agencies. This approach made it easy for even small agencies to share their data, while giving developers the tools they needed to build useful transit applications.

For TriMet and other transit authorities, GTFS helped them reach more riders. For Google and other map applications, the addition of transit data made their service more useful, attracting more loyal users. People worldwide now benefit from public transportation information in a consistent format.

HOW DID TRIMET AND GOOGLE SET THE STANDARD?

No one asked TriMet and Google to create GTFS as a global standard. While GTFS has become the de facto standard for transit authorities to share data about their systems, larger government bodies have been hard at work developing their own standards such as the Transit Communications Interface Profiles (TCIP) standard proposed by the American Public Transportation Association and the European Union's Transmodel and derived specifications. Despite these organizations' authority, the complexity of adopting their approaches has inhibited adoption, similar to how ICE failed to gain traction compared to RSS.

Anyone who has analyzed transit data standards will tell you that GTFS has fewer features than some other specifications, but the simplicity of GTFS is what allowed smaller agencies with limited resources to adopt it quickly. “A lot of players see value in a small dataset that focuses on riders,” says Drew Dara-Abrams, Co-Founder and Principal at Interline, which helps organizations understand and improve transporta-

tion networks. He continues, “Keep it as small and simple as you can. TCIP had the blessing of the Federal Transit Agency and they created massive amounts of documents, but nothing came of it.”

“You had the federal government mandating TCIP, while Google was making a useful application available for free that used GTFS data,” says Aaron Antrim, Co-founder of MobilityData, a non-profit organization that stewards GTFS. “Developers ran with GTFS, and the whole thing snowballed. Now, every major platform – Google Maps, Apple Maps, Bing – uses GTFS.”

Dara-Abrams also believes that the simplicity of GTFS allowed it to benefit from what he calls benign neglect. As he and other software developers excited about GTFS would ask public agencies to create and share datasets, he noticed that it was clear no one was tasked to work on GTFS as their full-time job. Instead, he says, “it was more like a hobbyist topic, where agencies could submit feeds to Google. The community was committed to a minimum viable specification, and the upside of benign neglect was that no one was making it any heavier than it needed to be. No one said we needed globally unique IDs at every stop, which would have raised too many questions at the beginning.” The constraints of a minimum viable spec gave the GTFS community freedom to innovate while growing quickly.

GTFS's simplicity is part of its success, but it wouldn't have happened without Google. Similar to how Netscape's early support lent credibility to RSS, Google's adoption of GTFS was instrumental to its success. Google Maps was the most popular mobile mapping application by far in 2005 (Apple Maps wasn't released until 2012) and transit authorities all over the world had strong incentives to figure out how to follow TriMet's lead and open up their data in a standard that Google could use.

Before being named the *General Transit Feed Specification*, it was called the *Google Transit Feed Specification* which discouraged some organizations from adopting the standard, assuming that it was a proprietary format controlled by Google. Today, it is the de facto transit data exchange standard, used by over 10,000 transit agencies around the world to power many applications and tools for transit riders, managers, and planners.

STAC: SHARED STRUCTURE FOR DIVERSE EARTH DATA

While GTFS shows how the web can improve everyday transportation, our final example demonstrates how the web enables sharing of massive scientific datasets. *The SpatioTemporal Asset Catalogs* (STAC) specification provides a common language to describe geospatial information and showcases how people have collaborated to use the web to share enormous volumes of satellite imagery.

Competition in the space industry has lowered the costs of developing sensors and launching them into orbit, leading to petabytes of diverse satellite data about our planet. Satellite operators have produced data in inconsistent formats with varying metadata structures, making it hard for developers, scientists, and analysts to search and analyze data across multiple sources. STAC arose to address the complexity of

finding and using this ever-increasing array of satellite imagery and other Earth observation data products.

STAC was created in 2017 by developers from the geospatial community, including organizations like Radiant Earth (where I serve as Executive Director) and Azavea. The goal was to create a standardized, open format that allowed geospatial data to be easily discovered and accessed. By using widely adopted web technologies like JSON and cloud storage, STAC enabled developers to build tools that could work across datasets from different providers using commoditized cloud services.

> *"By focusing on the essentials and leaving room for customization through extensions, STAC struck the right balance between flexibility and simplicity, which made it easier for organizations to adopt and implement."*

STAC's structure consists of three core components. A *STAC Item* represents a single image or data file. A STAC Catalog is an organized list of links that helps people browse STAC items. A *STAC Collection contains* metadata about a group of items, such as who published them and when they were collected. STAC also supports "extensions" that add additional metadata to any component. This modularity lets users extend the standard when necessary without overcomplicating the base specification. By focusing on the essentials and leaving room for customization through extensions, STAC struck the right balance between flexibility and simplicity, which made it easier for organizations to adopt and implement.

Matt Hanson, Director of Aerospace at Element84, was at the first STAC sprint (a small meeting of like-minded software developers) in Boulder, Colorado. "Chris [Holmes, lead developer of STAC] was always focused on getting to the minimum number of fields," Hanson says. "In a lot of standards, everyone just dumps their ideas in. But STAC was a conscious effort from the beginning: minimum fields, two or three at most. We talked a lot about what was the absolute minimum required in order to describe something spatial."

Many organizations quickly adopted STAC. Just 5 years after the STAC sprint in 2017, STAC had been adopted by organizations including NASA, the U.S. Geological Survey, the European Space Agency, Geoscience Australia, Planet, Maxar, AWS, and Microsoft. A 2022 attempt by Planet to count the number of public resources made available with STAC metadata found over 550 million data files. Planet and Maxar maintain private STAC catalogs that contain billions of items.

HOW LANGUAGES (AND STANDARDS) ARE FORMED: PATTERNS OF EMERGENCE AND INSTITUTIONAL ADOPTION

Having examined these examples of successful web standards, we can explore the common patterns that contributed to their success. Web standards may appear "institutionless" because they are often created by ad hoc and ephemeral groups of practitioners who coalesce around specific problems, develop solutions, and then dissolve once their work is complete. However, if a web standard gains traction, it becomes connective tissue that enables other institutions to collaborate more effectively across organizational boundaries.

USEFULNESS

In the final chapter of *Seeing Like a State*, Jim C. Scott made the following observation about how human languages evolve:

Influence over the direction of a language is never equally distributed, but innovation comes from far and wide, and if others find a particular innovation useful or apposite, they will adopt it as part of their language. In language [...] seldom is the name of an innovator remembered, and this, too, helps to make the result a joint, mutual product.

A human language is a technology, and technologies endure to the extent that people find them useful. While empires have imposed languages on populations throughout history, the *direction* of language is subject to many forces rather than being determined by any singular power. One thing shared by all of the technologies described in this paper is that none of them are proprietary, which has allowed them to be influenced by innovators "far and wide." They gain usefulness because anyone can adopt them without being beholden to their creator. Like spoken language, they have proven useful to share information, and they will endure as long as they remain useful.

This characteristic of shared standards is frustrating to entities seeking to build businesses around proprietary standards, but it's impossible to estimate the economic value of language itself.

SIMPLICITY

Like human languages, web standards will endure as long as they are useful. Another common characteristic of successful web standards like RSS, Markdown, GTFS, and STAC is that they are designed to be simple. By encouraging simple data formats, they allow participation from collaborators with varying levels of technical capacity.

Their simplicity also makes them more interoperable, creating the basis for more complex implementations. Remember Gall's Law:

A complex system that works is invariably found to have evolved from a simple system that worked. A complex system designed from scratch never works and cannot be patched up to make it work. You have to start over with a working simple system.

In 2023, a group of researchers published a paper titled "On the roles of function and selection in evolving systems" in the Proceedings of the National Academy of Sciences. They propose the "law of increasing functional information" and suggest that "all evolving systems – including but not limited to life – are composed of diverse components that can combine into configurational states that are then selected for or against based on function."

If we want to develop systems that help us navigate complex phenomena like pandemics or climate change, we need to know how to design complex systems. They are not built as complete systems, but rather emerge over time as simple systems interact. Well-designed foundational systems enable the creation of simpler abstractions. This makes complex technologies accessible to broader audiences. Markdown is an example of an abstraction that helps more people create HTML, and podcast apps are abstractions that help more people benefit from RSS.

PROFESSIONAL CREDIBILITY

The web, RSS, Markdown, GTFS, and STAC were all created by small, self-organized groups of peers who had a clear understanding of the problems they wanted to solve and a practical need to do so. These groups are temporary and task-focused – the RSS working group included developers from different companies who argued for years before settling on a solution, while the STAC community began with a small sprint in Boulder, Colorado, then dispersed after creating the initial specification.

Anyone interested in funding the development of standards will benefit from finding and listening to a few key professionals who are likely to implement standards or benefit from their implementation. Allow them to lead the effort as their practical knowledge of the issues at hand will be the best guide to success. This process might be messy and it may include a number of false starts, but it is likely less expensive than hiring outside experts or trying to corral an overly broad group of stakeholders.

INSTITUTIONAL CREDIBILITY

While the approaches described above are nonproprietary, they have benefited from the credibility that comes from adoption by legitimate[14] institutions.

The web first gained traction with the credibility of CERN and early adopters at established universities around the world. RSS got its initial momentum from Netscape and became mainstream when it was adopted by the New York Times. Markdown became credible sometime around the point when it seemed every online coding tool supported it. GTFS would not exist without the pull of Google Maps and its many users who

demanded their transit agencies to adopt it. STAC benefited from early adoption by the U.S. Geological Survey and Microsoft.

This pattern reveals how standards created by temporary communities become institutional infrastructure when adopted by established organizations. When the New York Times adopted RSS, it validated a standard created by a loose network of developers. When Google Maps embraced GTFS, it legitimized a specification created by a small team at TriMet. These adoptions transformed ad hoc technical solutions into infrastructure that enables coordination among thousands of organizations worldwide.

> *"The most successful approach may be to identify and support the ephemeral communities of practitioners who are already working on solutions, rather than trying to create top-down standards that may lack the practical knowledge necessary for widespread adoption."*

Institutions that seek to enable global collaboration need to judge carefully when to lead the generation of new data standards and when to back efforts started by others. The most successful approach may be to identify and support the ephemeral communities of practitioners who are already working on solutions, rather than trying to create top-down standards that may lack the practical knowledge necessary for widespread adoption.

THE WEB OF THE FUTURE

The patterns we've identified in successful web standards – their nonproprietary nature, simplicity, and ability to gain credibility through adoption – provide a framework for addressing current challenges facing the web.

On March 16, 2025, Tim Berners-Lee published an op-ed titled “Let's knock down social media's walled gardens” in the *Financial Times*[15]. He notes concerns about how the web has evolved and its unrealized promise to “build new systems that are better for society and for individuals.”

An obvious area of concern on the web is the power of social media companies that use the web's rapid client-server dynamic to capture people's attention with the singular goal of selling advertising. In the final chapter of his book *Analogia*, George Dyson succinctly explains how these business models are leading us into a global epistemological crisis:

Automata, advertising, and natural selection are an explosive mix. [...] Internet advertising drives a global high-gain amplifier connecting the reward sought by computers (more machine cycles and instructions) to the reward sought by humans (more of the stimulation now returned with every click). We set loose an evolutionary system that rewards machines that learn to control both how we feel and what we think.

In his op-ed, Berners-Lee explains that the web got to its current state “bit by bit, as a result of the natural application of markets and monopolies.” Institutions should align around nonproprietary standards and new business models that use the web to support the ongoing production of accurate data.

NEW NORMS AROUND INSTITUTIONAL TECHNOLOGY ADOPTION

This is not a call to boycott all tech companies, but for institutions to develop a more sophisticated understanding of technology business models and how those models can exert hegemonic control over data. Institutions should adopt a form of regime theory[16] in which they adopt norms around using open formats and commodity technologies and discourage the usage of proprietary services that funnel data into walled gardens.

Institutions should also prioritize supporting the practitioner communities who create shared infrastructure. This means funding sprints and collaborative spaces where technical experts can work together on shared problems, and creating procurement policies that favor open standards over proprietary solutions.

NEW BUSINESS MODELS

Another area in need of innovation is the development of new business models to support shared infrastructure required to produce or share data. As greater volumes of data are produced, the costs of storing, indexing, and distributing data increase. Many cloud-based software-as-a-service companies have built high-margin businesses managing infrastructure in exchange for subscription fees[17].

These businesses have made complex services more accessible to consumers, but, like many web-based businesses, they are often designed to lock their customers into their service and can lead to monopolies. Investors' expectations of high margins and rapid growth also make these businesses unlikely to provide services to poor communities or to address global challenges created by negative externalities.

Fortunately, the competitive market for technology services is making it less expensive to create and share data. It is time to create a new breed of data service providers that can do so without creating walled gardens. Invest in Open Infrastructure[18] provides guidance for scientific institutions to identify durable and open services that support research, and a growing group of research institutions are adopting the Principles of Open Scholarly Infrastructure to avoid making research beholden to investors' priorities[19]. Berners-Lee has founded Inrupt to do this for individuals' data[20]. Radiant Earth is creating a data publishing utility called Source Cooperative that encourages sharing scientific data in open standards on the web.

AGENTS OF THE WEB

Large language models (LLMs) that can write and manipulate text in human-like ways are a significant application of the web. LLMs like ChatGPT, Llama, and Claude were trained on massive corpora of text scraped from the web[21], and a key capability of these models is their ability to learn languages quickly. As noted previously, new languages will emerge as more people, computers, and sensors are added to the web. LLMs can learn those languages and provide tools to help us further understand our world. In 2024, Anthropic introduced an open standard called the Model Context Protocol (MCP) to accelerate the integration of LLMs with other systems.

The insight to take from this is that we are the ones who develop languages to describe the things that matter to us. We create and deploy the sensors that will train the models of the future. We humans still bear the responsibility of figuring out what we pay attention to and what we prioritize. We bear huge opportunity costs if we focus on the wrong things or let the capital markets set our priorities.

CONCLUSION: CREATING CONDITIONS TO BUILD TOGETHER

Simple, nonproprietary standards enable robust collaboration and understanding. From RSS to STAC, these standards emerged from small groups of practitioners who understood real-world needs. They succeeded because they were useful, simple, and gained credibility through adoption. But their impact extends far beyond their immediate technical purpose. **Standards function as foundational social infrastructure that shapes how new institutions and entire sectors develop.** RSS created the architectural foundation for modern social media, podcast ecosystems, and content management systems. GTFS enabled a global ecosystem of mobility applications that fundamentally changed how people navigate cities. Standards create the invisible scaffolding upon which larger developments build.

While these standards emerged organically, their success required deliberate architectural choices. The creators of RSS, Markdown, GTFS, and STAC architected solutions with careful attention to simplicity, extensibility, and adoption barriers. We cannot mandate standards into existence, but we can design conditions that make successful standards more likely to emerge and thrive. The JHU COVID-19 dashboard succeeded because it followed this pattern: a small, motivated group of practitioners created a practical solution that became infrastructure for the entire world.

The value of these standards lies not just in technical interoperability but in creating shared understanding across diverse communities. When transit agencies worldwide use GTFS, they're participating in a common language that enables coordinated action. When scientists use STAC, they're deliberately building collective knowledge that transcends institutional boundaries. This shared understanding becomes the foundation for addressing challenges that no single organization can solve alone.

Rather than waiting for permanent bureaucracies to create comprehensive systems, we can move more quickly to support communities of practitioners working to address specific coordination challenges. The web already runs on open standards, but it's time for institutions to be more deliberate about creating new open standards and supporting the ones we already have. This often means supporting ad hoc communities of experts rather than trying to create permanent bureaucracies, and recognizing that the connective tissue enabling global collaboration is often created by temporary groups focused on practical, easily adopted solutions.

We've never had more access to information about our world – and never more tools to describe and interpret reality. But with that power comes the need to invest in collaboration. Only then can we ensure that new knowledge leads to a shared understanding of the world we're creating.

ACKNOWLEDGEMENTS

Jessica Seddon, Ramanathan V. Guha, Matt Hanson, George Dyson, Howard Butler, Aaron Antrim, Drew Dara-Abrams, Phillip Müller, Drew Breunig, Gina Trapani, Kaitlin Thaney, and Brook Brannon.

ABOUT THE AUTHOR

Jed Sundwall is a technologist based in Seattle, known for using the Internet to strengthen human cooperation on global challenges. He is the Executive Director of Radiant Earth, where he leads efforts to make geospatial and scientific data more accessible and usable. He also hosts *Techs on Texts*, a podcast exploring the relationship between literature and technology, and serves as a Technical Fellow at the Taylor Geospatial Engine and Technical Advisor to the Flickr Foundation. Previously, he built the AWS Open Data Program and launched the Amazon Sustainability Data Initiative, served as a technical advisor to the Landsat Advisory Group, PLACE, and the Center for Open Data Enterprise, and taught *Planetary-Scale Data Institutions* at Yale's Jackson School of Global Affairs. He has lived in Germany, Brazil, and Venezuela, speaks Spanish and some Portuguese, and is happiest swimming in natural bodies of water with his family.

ENDNOTES

1 Gardner, L., Ratcliff, J., Dong, E., & Katz, A. (2020). A need for open public data standards and sharing in light of COVID-19. The Lancet Infectious Diseases, 20(9), 1021-1022. https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30635-6/fulltext

2 Mulgan, G., Marsh, O., Henggeler, A., & Chen, K.-L. (2022). The Pandemic and the Global Challenge of Intelligence. International Public Policy Observatory. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4304021

3 Kaiser Family Foundation, "The U.S. Government and Global Health," 2024, https://www.kff.org/health-policy-101-the-u-s-government-and-global-health/?entry=table-of-contents-how-does-the-u-s-compare-to-other-donors-of-international-health-assistance

4 Johns Hopkins University, "Our Bond at a Moment of Challenge," March 4, 2025, https://president.jhu.edu/messages/2025/03/04/our-bond-at-a-moment-of-challenge/

5 Wikipedia, "List of screw drives," accessed August 2025, https://en.wikipedia.org/wiki/List_of_screw_drives

6 Tim Berners-Lee, "WorldWideWeb: Summary," August 6, 1991, https://www.w3.org/People/Berners-Lee/1991/08/art-6487.txt

7 Wikipedia, "Gopher (protocol)," accessed January 2025, https://en.wikipedia.org/wiki/Gopher_(protocol)

8 Wikipedia, "Zettabyte Era," accessed January 2025, https://en.wikipedia.org/wiki/Zettabyte_Era

9 Wikipedia, "John Gall (author)," accessed January 2025, https://en.wikipedia.org/wiki/John_Gall_(author)

10 Anil Dash, "Wherever you get podcasts," February 6, 2024, https://www.anildash.com/2024/02/06/wherever-you-get-podcasts/

11 Harvard University, "RSS Version History," accessed January 2025, https://cyber.harvard.edu/rss/rssVersionHistory.html

12 John Gruber, "Markdown," 2004, https://daringfireball.net/projects/markdown/

13 Wired, "The Eternal Truth of Markdown," June 2024, https://www.wired.com/story/the-eternal-truth-of-markdown/

14 TIAL, "Legitimacy Working Hypotheses," January 31, 2025, https://tial.org/wp-content/uploads/2025/01/Legitimacy-Workingh-Hypotheses-TIAL-BI-v31012025.pdf

15 Tim Berners-Lee, "Let's knock down social media's walled gardens," Financial Times, March 16, 2025, https://www.ft.com/content/79d2d19a-08df-48fc-9a6f-a9dbef58f642

16 Wikipedia, "Regime theory," accessed January 2025, https://en.wikipedia.org/wiki/Regime_theory

17 Stripe, "SaaS Gross Margin Explained: What It Is and Why It Is Important," 2024, https://stripe.com/resources/more/saas-gross-margin-explained-what-it-is-and-why-it-is-important

18 Invest in Open Infrastructure, "About," accessed January 2025, https://investinopen.org

19 Bilder G, Lin J, Neylon C (2020), The Principles of Open Scholarly Infrastructure, retrieved January 2025, https://doi.org/10.24343/C34W2H

20 Inrupt, "Solid Platform," accessed January 2025, https://www.inrupt.com/

21 Common Crawl, "Web Crawl Data," accessed January 2025, https://commoncrawl.org