Designing Multi-Modal Ecosystem Monitoring Technologies: A Network of Networks Approach

Christopher B. Anderson
Planet Labs PBC
San Francisco, CA 94107
christopher@planet.com

April 22, 2025

The patterns and processes discernible in natural ecosystems still remain the most appropriate standard available... What is needed are countless elegant solutions keyed to particular places and problems.

Jackson and Piper (1989)

1 Introduction

The central promise of ecosystem monitoring technologies — like bioacoustic, camera trap, citizen science, eDNA, and satellite data — is to reveal changes in the structure and composition of the Earth's ecological systems to facilitate timely and effective conservation action (Langhammer et al., 2024). The opportunities for new technologies to provide data-driven conservation insights have been clear for several decades (Hobbs and Mooney, 1990; Roughgarden et al., 1991), but interest accelerated rapidly about a decade ago as access to data, computing power, and new measurement tools flourished (Snaddon et al., 2013; Pimm et al., 2015; Marvin et al., 2016). Over ten years of research and development have further increased the stability and predictability of these systems, where the opportunities and limitations for each type of data are now fairly well characterized (Anderson, 2018; Jäckel et al., 2021; Fraisl et al., 2022; Tuia et al., 2022). Now, the fusion of multimodal observation systems, where data from multiple sources are combined to provide novel and emerging insights, is developing as a key research frontier (Daroya et al., 2024; Sastry et al., 2024; Zhu et al., 2024).

From a technology systems perspective, the emergence of multimodal analyses marks a key transition from multiple independent systems in developing states towards a mature, interconnected, and network-like state. The former is typically characterized by linear flows of information within a single system, such as automating wildlife detections from camera traps (Beery et al., 2019) or classifying habitats from satellite data (Alleaume et al., 2018). The latter is characterized by flows of information across multiple systems, such as dynamic species abundance maps derived from fusing citizen science and satellite data (Fink et al., 2023) or integrations of eDNA and camera trap data to monitor species site use patterns (Tetzlaff et al., 2024). As methods for fusing data across modalities emerge, so will the technical systems that manage the flows of data between systems and end-users.

Mult-modal ecosystem monitoring networks are desirable because, while individual technology systems specialize in measuring ecological patterns at distinct spatial and temporal scales, characterizing the complex dynamics inherent to ecosystems is best achieved through multiscale analysis (Chave, 2013). While multimodal ecosystem monitoring technology networks represent a meaningful end state, few are currently operational and there are even fewer blueprints to outline how these systems should be built. In anticipation of the upcoming opportunities to build and deploy such networks, this essay reviews some of the emergent properties of systems in general, as well as the very systems we seek to better monitor — ecosystems — to identify key principles to guide software architecture and technology design (Meadows, 2008). What technology design patterns will best facilitate timely multisource insights into ecological change?

2 Ecological Design

Ecosystem monitoring technologies are systems that produce a time series of digital data describing the structure and function of ecological systems, including the stocks and flows of organisms and resources through an environment. These technologies — called *Digital Earth Technologies* by Bakker (2024) — typically refer to a system with four primary components: hardware, software, data, and people. In the most simplistic representation, measurements are made by hardware, processed by software, stored as data, and those data are analyzed by users or by downstream software systems. These components can be broken into subsystems — like the power and communications systems onboard satellites — but it is useful to represent each technology as a network of components. The purpose and function of each system are often self-contained, such as classifying bird calls or detecting logging activity from acoustic signals. But multimodal monitoring systems will be more complex, managing information flows between systems, creating a network of networks. How will and how should such networks operate?

Ecological systems are networks of networks, for example, and their structures reveal some relevant architectural patterns for technology design. Ecology and architecture are didactic disciplines, revealing the complex, multiscale interactions between components in natural and built systems. While architecture is nearly always diagrammatic, prescriptive, and future facing — providing blueprints to build from — the origins of modern ecological research often involved inspection, interrogation, and disentanglement. This often takes the form of isolating and examining the roles of individual components in a system, such as removing a species from a trophic network to examine how the rest of the network responds. The goal of this type of research is often to develop a clear, relational understandings of the species interactions and the effect sizes of changes within complex, interdependent systems.

While a diagrammatic, predictive architecture of the Earth's ecosystems may not yet exist, there are at least a series of core foundational principles for how ecological systems work, which can inform how to build complex technology systems. This is not a strictly novel framing; such topics have been explored in-depth within software systems literature. Messerschmitt and Szyperski (2003) made the connections between software systems and ecosystems explicit in the early 2000s; Mens and Grosjean (2015) expanded on this theme, using trophic systems as a model to explore how resourcing developers drives innovation; Keil et al. (2018) articulated the emergent evolutionary and ecological dynamics at play among Linux distributions; Jacobides et al. (2018) extended the ecosystem analogy to describe interactions between markets and businesses. Metaphors and analogies are prominent in these works and will be here, too.

The metaphors of three additional ecological processes are extended here to the realm of technology development: succession dynamics, resilience, and stable states. What follows is not a discussion of implementation details but a high-level synthesis of key ecological processes that apply directly to technology systems, primarily to software and data. The goal is to articulate how natural design patterns can be used to build clear, relational technology systems that map the patterns of nature themselves — to develop a blueprint for biomimicry.

3 Succession Dynamics

Succession is a prominent process within ecological systems. In plant communities, it refers to a directional change in the composition and structure of a community over time, which begins when a disturbance — an event that removes or introduces part of a community — is followed by new establishment or by regrowth (Gurevitch, Jessica et al., 2006). Ecological communities frequently exist under dynamic and homeostatic conditions, shifting between directional turnover processes and self-repairing, self-sustaining processes (Fig. 1). Tensions between these processes typically result in increased community diversity, promoting stability within the system over time (Hatton et al., 2024).

3.1 Balancing feedback loops in ecosystems

Balancing feedback loops are equilibrating or goal-seeking structures in systems, and are both sources of stability and sources of resistance to change (Meadows, 2008). These feedback loops are driven by many processes in ecological systems, such as coevolution, competition, disturbance, and resource availability. These processes have several parallels in technology systems, like research partnerships, institutional competition, the release or deprecation of an analysis platform, and access to compute power. To design a stable ecosystem monitoring network, we can draw inspiration from the properties of stable natural systems. The following is an adapted summary of the emergent properties of forests and other ecosystems (Bradley, 1994).

- Forests and ecosystems are an ever-changing continuum of living and nonliving things and processes embedded in time, not suspended from time.
- Forests are mixtures of living and nonliving things and processes that are self-organized, self-repairing, and self-sustaining. They are dynamic, yet relatively stable.
- Diversity is a fundamental property of forests and other ecosystems. It emerges for many "reasons" at system levels, from biogeochemical cycling to dispersal mechanics to species interactions to geomorphic turnover.
- The world and its forests are coevolving and interrelated systems of things and processes that meet many "ends" and functions.
- Forests are complex landscapes whose patterns reflect crucial underlying structure and process.

3.2 Dynamic succession in software

In software systems, a new system is established by the first lines of code. Software continues to grow over time as new functions are written, new features are added, and the system evolves toward production or fails and goes locally extinct (Bloch, Michael et al., 2012). Balancing loops within software systems include unit and integration testing, which increase stability by mitigating regression risks and ensuring continuity over time (i.e. the software continues to operate as expected). Automated testing workflows

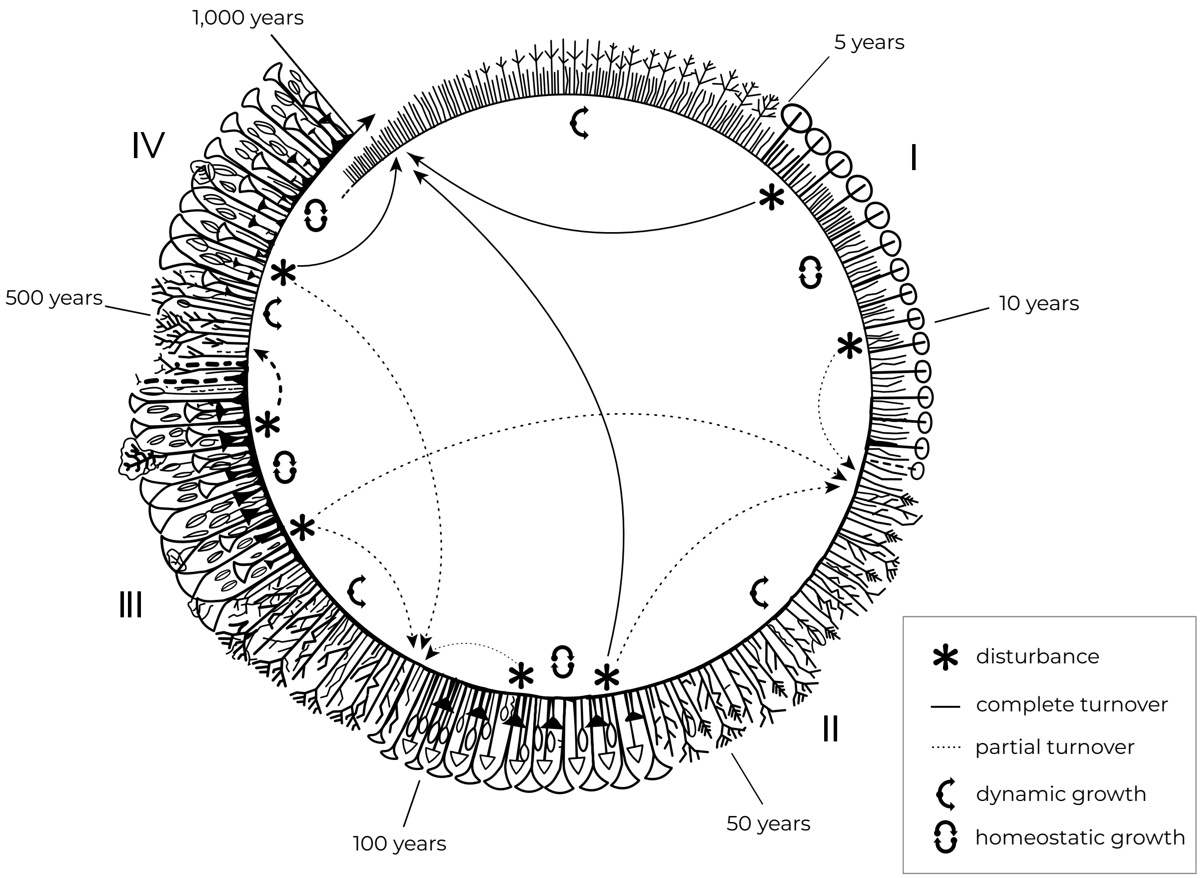

Figure 1: Radial succession diagram illustrating growth and turnover within a forest over time (Hallé et al., 1978). The icons correspond to different vegetation structural archetypes, showing how forests change in response to competition, disturbance, and time. The successive phases (I, II, III, IV) include dynamic growing stages and homeostatic steady stages (diverging and circular arrows). Disturbances effects are heterogeneous, resulting in either complete or partial community turnover (solid or dashed lines), which could drive regressions to previous states or advance to new, more complex states. Time is represented on a log scale.

underpin continuous integration, continuous deployment, and continuous delivery, which all shorten the feedback loops between developers and users of a system, and improve predictability (Shahin et al., 2017). Software versioning and deterministic dependency management increase stability by ensuring reproducibility, particularly in new runtime environments. However, a software system's dependency tree can act as both a source of stability and a source of resistance to change. Stability can be achieved by pinning dependencies to specific versions, but can also introduce resistance to change by not integrating features from new releases.

These processes — writing code, writing tests, automating deployment, integrating with other systems — occur over time as developers contribute code, representing a system dynamically moving from initialization to operation. Iterative and exploratory software development is particularly prominent within scientific programming, where growing code organically, refactoring regularly, and avoiding premature optimization are cited as best practices Balaban et al. (2021). Version control systems like Git facilitate these patterns by providing an ongoing record of changes within a system. This enables developers to understand the processes that created the current state — the succession dynamics — and allows users to revert back to previous states if necessary. These tools are critical because code is regularly refactored to improve efficiency and clarity, which are self-repairing and self-organizing processes. The axiom that ”rewriting is the essence of writing” applies equally to writing software (Zinsser, 2001).

Change is common in both ecological and software systems, and stable systems expect and promote balancing feedback loops to manage equilibration. This does not mean, however, that all ecological and all software systems inevitably and linearly move towards stability. The frequency and intensity of disturbances within these systems can cause major disruptions, such as a high-intensity wildfire or the loss of a software team. Not all systems are resilient to such changes. Designing for resilience should be a primary goal for ecosystem monitoring networks, which will necessarily manage complex interdependencies between systems. What does resilience mean from the perspective of ecological design?

4 Designing for Resilience

> *Resilience implies the capacity of systems to withstand external disturbances and internal malfunctions. Resilient systems absorb shock gracefully and forgive human error. Resilience does not imply a static condition; it implies flexibility that allows a system to survive unexpected stress. Resilient design does not achieve the greatest possible efficiency all the time, but achieves a deeper efficiency by avoiding failures that jeopardize the operation and maintenance of the system.*
>
> *Orr (1992)*

Resilience in ecology was first defined as the amount of disturbance an ecosystem could withstand without changing its self-organizing processes and structures (i.e. the ecosystem's stable state; Holling, C.S., 1973). But since ecological communities grow and respond to changes over time, experiencing different degrees and types of stability, the concept of alternative stable states emerged to describe how communities can maintain and change between multiple self-organizing states over time (Beisner et al., 2003). Ecological resilience in the context of alternative stable states then refers to the limits of a stability domain and is defined by the magnitude of disturbance that a system can absorb before changing stable states (Gunderson, 2000).

Alternative stable states have parallels in software systems. Semantic versioning, for example, minimizes instability by clearly describing and tracking the release history of software packages, from bug fixes to minor releases to major releases (e.g., from v1.0.1 to v1.1.0 to v2.0.0; Preston-Werner 2013). Bug fixes and minor releases upgrade functionality in an internally consistent, backwards-compatible fashion; a self-repairing process that maintains a stable state. Major releases, which introduce incompatible changes to the system architecture, mark a transition to a new stable state. The primary functions of the software remain intact, but the underlying processes and structures that produce them have been reorganized. In the context of succession dynamics, minor releases represent homeostatic changes, while major releases represent dynamic directional changes (Fig. 1).

Developing multimodal ecosystem monitoring networks will require transitioning technology systems between stable states; from a series of self-contained technologies to a network of interconnected ones. An effective network should itself be resilient to disruptive disturbances, such as the loss of one mode of measurement, in the same way that a single technology system should be resilient to disturbances, like the loss of one instrument. A resilient network is likely to emerge from a series of underlying technology systems that are themselves resilient.

What makes a technology system resilient? Resilient systems exhibit certain qualities, and these qualities have been found to generalize across disciplines (Lovins, 2003; Orr, 2004; Meadows, 2008). These include:

- Simplicity and repairability
- Diversity and redundancy of components
- Modular, dispersed structure
- Multiple short interconnections between components
- Loose coupling of components in a hierarchy
- Decentralized control
- Rapid feedback mechanisms

Networks of ecosystem monitoring technologies will be built, maintained, and used by a geographically diverse community of users. Resilience here implies building small, resource-efficient, locally adaptable, culturally suitable, and technologically elegant solutions, where the failure of one component does not jeopardize much else (Orr, 1992). No technology system is going to immediately demonstrate these properties: they emerge over time as a result of succession. Developers and system architects both implicitly or explicitly manage the processes of development that move software systems between stable states, and the mechanisms of this turnover are iterative design, refactoring, and review. Once the underlying design goals for a system align with the goals of resilience, they should slowly transition from developing to mature system states.

5 Mature Stable States

The development of multimodal ecosystem monitoring networks represents a transition from a series of independent, developing systems towards a mature, network-like state (Table 1). And while the patterns of succession and resilience are likely to be shared at both the subsystem and network levels, managing flows of information between systems may require new configurations that are not present within any subsystem. How do we expect the transitions observed between developing and mature ecological systems to apply to technology systems?

The early stages of technology development are characterized by a small number of pioneering organizations or researchers — or species, in this metaphor — with more participants emerging as the technology becomes better understood. Specialization typically follows as more organizations adopt the technology and as new applications are tailored to address specific use cases (Jacobides et al., 2018). For example, general advances in computer vision (Krizhevsky et al., 2012; He et al., 2016) have been fundamental to advancing automatic species identification from camera traps, cell phone pictures, and satellite data alike, with highly specialized innovations proliferating in

Ecosystem Property	Developing Stages	Mature Stages
Species diversity	Low	High
Niche partitioning	General	Specific
Coexistence dynamics	Competition	Specialization
Life cycles	Short, simple	Long, complex
Growth strategy	Fast, uncontrolled	Feedback control
Food chain	Linear	Web-like
Nutrient conservation	Wasteful	Efficient
Stability	Low	High

Table 1: Emergent properties of ecosystems at developing and mature stages. Developing stages are represented in Fig. 1 by phases I and II, and mature stages are represented by phases III and IV. Adapted from (Benyus, 1997; Allenby and Cooper, 1994).

each of these fields (Ueda, 2020; Kattenborn et al., 2021; Hernandez et al., 2024; Gillespie et al., 2024).

The technological innovation systems literature has identified that the initial phase of technology development is characterized by poorly defined products with high uncertainty, followed by a period of rapid growth where standards and value chains form and adoption takes off (Taylor and Taylor, 2012; Markard, 2020). Adoption eventually saturates, growth slows down, competition increases, and there is a shakeout where many participants leave the industry, leading to stabilization.

Many ecosystem monitoring technologies are still in this period of rapid, highly uncertain development. Geospatial foundation models are a good example of a poorly defined but rapidly developing technology. Foundation models seek to over-generalize and solve every problem everywhere better than solutions tailored to a particular place or context (Mai et al., 2023). Based on the premise that one model could emerge to provide the foundation on which all further analyses will stand, modeling approaches have proliferated without design standards or a clear articulation of how to create alignment across models (Rolf et al., 2024).

Despite the imprecise premise — it is unlikely that one model generalizes across all geospatial modalities — attempting to solve the problem at all has identified key opportunities. Efforts to harmonize data across modalities — from bioacoustic, camera trap, citizen science, and satellite data — may be converging around a data format that could facilitate the flow of information across systems: one-dimensional numerical arrays, otherwise known as embeddings. Transforming data into a compatible analytical structure would create a simple, efficient mechanism for fusing data between modalities, improving predictive power, and applying transfer learning between contexts (Ma et al., 2024). Fusing data from multiple models could also improve resilience by reducing dependencies on any one system, and could help transition the flow of information from a linear workflow to a more networked, web-like configuration.

6 Ten Principles for Designing Multi Modal Ecosystem Monitoring Networks

Ecosystem monitoring technologies have developed to the point where automated, multiscale, multimodal monitoring networks are conceptually and technically feasible (Sparrow et al., 2020; Besson et al., 2022; Pollock et al., 2025). Such networks are likely to emerge as system-scale shifts from developing to mature technology states (Table 1). Automating information flows from multiple systems will be critical here, and the key technology design challenge is to create coherent information feeds that allow us to respond to and mitigate change, and to evaluate progress towards conservation targets.

The emergent properties of ecosystems themselves might illuminate the principles for how networks can evolve from rapidly growing, highly uncertain products to stable, specialized, and interconnected components within larger systems. The lessons of succession dynamics, resilience, and alternative stable states in ecology that can guide the development of the next generation of ecosystem monitoring networks. How can new technology systems be built to mirror the processes and patterns of the ecological systems they monitor? How should these principles be translated from metaphor to mechanics?

Decentralize by design Monitoring technologies often manage a directional flow of information, passing data from one or more sources through a model to generate predictions, a process typically managed by one organization. Emerging monitoring networks will instead resemble a web-like system with multiple connections between components, managed by multiple organizations. This will likely start by linking linear systems, like merging two or three modalities together to produce an ensemble prediction, then grow towards multi-node networks. The exact pathway towards mature, stable-state systems with short feedback loops — where models improve as more data are collected and circulated through a system — is not yet clear, but it will certainly be charted by experiment-driven prototyping and open collaboration across multiple teams.

Focus on data interoperability over software or platform interoperability One comparable network-of-networks technology system is the Internet of Things, which emerged as the result of alignment around consistent network communications and data protocols (e.g. a well-defined network topology; Karimi and Atkinson, 2013; Rahman and Asyhari, 2019). This framework for decentralized, network-driven communications may provide a useful blueprint for monitoring systems, and a network of ecosystem monitoring technologies might be well framed as an Internet of Nature (Galle et al., 2019).

Data is the key resource that will flow between systems and should be the focus for standardization because it can be described in platform-independent formats. Since these systems will be managed by different organizations with different technology stacks, it will likely be faster and more sustainable to agree on data standards than to convince organizations to standardize their programming languages, software packages, or cloud service providers. This will also simplify the onboarding process for new teams, increasing the overall diversity and specialization of participants in the network.

Align on core data and metadata standards Establishing efficient flows of information between systems will require alignment on how data are provided and described across modalities. This is a delicate balance. Specify too many standards and the system becomes too rigid; too few and duplicate effort will be spent integrating disparate datasets.

At the outset, it will be worth aligning on how to represent the core dimensions of ecological data: how to describe location, time, taxonomy, and measurement units. Several relevant open standards already exist, such as the SpatioTemporal Asset Catalog specification to describe space and time (Zhao et al., 2021), Darwin Core and Humboldt Core for standardizing species observations (Wieczorek et al., 2012; Guralnick et al., 2018), and the Essential Biodiversity Variables to categorize biodiversity metrics (Pereira et al., 2013).

Alignment does not necessarily require that existing data must be processed or reprocessed to fit a certain standard, but refers to alignment on how data should be loaded and analyzed (e.g., to ensure straightforward joins or intersections across datasets). Software can play a key role in managing compatibility. Middleware that translates and standardizes data, such as the Taxonomic Name Resolution Service (Boyle et al., 2013), can be used to facilitate consistent analyses between teams.

Prioritize cloud-native and API-driven access patterns Cloud-native workflows can shift data access patterns from analyzing stocks of data to flows of data; from analyzing local copies of data to web-based analyses. Application programming interfaces (APIs) are a core component of cloud-native systems that allow software applications to communicate and exchange data, which are often hosted as web services. APIs provide an abstraction layer that permits interoperability across systems, handling key regulatory functions such as authentication, authorization, rate limiting, and data standards enforcement.

APIs can establish communication contracts between systems, regulating flows of information, as well as signaling and diagnosing which parts of the system are down, which are critical feedback loops for developers. Organizing access around APIs allows data providers to curate and manage their data production systems internally and only expose data via web services, which regulate and instruct users how data can be accessed. API-driven designs will become increasingly valuable as access patterns evolve from analyzing stocks of data (e.g., downloading static training datasets) to processing flows of data (e.g., near-real-time inference).

Use clear versioning strategies for software, data, and APIs to ensure stability Technology systems grow and change over time via succession dynamics, transitioning between alternative stable states as model accuracy and processing efficiency improve over time. However, changes in one system can have major effects on downstream systems, and not all changes can be immediately integrated across the full network, since different teams update their systems at different rates. This poses risks to reproducibility, which is critical for scientific analysis.

How can we ensure that changes in one system do not introduce instability across the whole system? Establishing clear and consistent versioning standards is one way to mitigate this risk. By establishing clear versioning information across the system — for datasets, for models, APIs, and software packages — users can expect deterministic behavior based on version numbers and can pin their dependencies to specific deployments or datasets. Version information can also document data provenance (i.e. the history of a dataset's origin and transformations), providing data consumers with traceable information on the raw data used in a derivative product.

Create staging and production environments for testing releases across systems Establishing both staging and production environments, also referred to as "next" and "live" environments, improves software resilience by testing new changes in an isolated environment before moving to production. Staging environments can help developers test stable-state behaviors to avoid breaking changes to downstream systems and make it easy to revert changes, improving repairability. In the context of multimodal monitoring networks, staging environments will be the primary location to test new dataset releases.

Staged integration testing creates a rapid feedback mechanism for data consumers to characterize how their models respond to new data in an isolated environment. It should also establish cross-team feedback loops, as data consumers can communicate issues to data providers. Although this approach will create several short and efficient feedback loops, it is also likely to create longer, more complex release cycles as developers collaborate to balance changes, improvements, and stability.

Test against consistent, independent benchmarks to understand performance shifts over time Automated software testing is recommended as a best practice because it reduces regression risks (i.e., introducing breaking changes). This principle can be extended beyond software and applied to automated model performance testing, which could be considered continuous evaluation. Ecosystem monitoring technologies often involve predictive modeling, and models can be retrained as new observations or new predictive features become available, which then change the model predictions. This typically improves overall accuracy, though not always uniformly. An updated camera trap species classification model may improve the predictive accuracy for certain taxa but decrease for others, for example.

Since data consumers are concerned about the consistency of the predictions they use, data producers should evaluate model performance using consistent independent benchmark datasets so consumers can clearly understand how new versions of data are likely to affect downstream systems. Automated continuous evaluation workflows will shorten feedback loops for internal developers, increase transparency between teams, and provide some protection against regressions during integration.

Promote open, asynchronous communications within and between systems A modular, dispersed structure with short connections between components will eventually describe a mature technology system, but will better first describe the working relationships between the researchers and developers in organizations that work to build these systems. At this stage in the maturity of ecosystem monitoring technologies, people are still the most critical components of systems, navigating the evolving research landscape and making complex scientific and technical choices.

Linking complex systems will require many forms of communication regarding strategic alignment, knowledge sharing, and technical feedback. Teams that contribute to ecosystem monitoring networks should provide multiple avenues for receiving input and contributions from other teams, like discussion forums and software issue trackers. Asynchronous communication should be prioritized to ensure accessibility to a large audience, including teams who may not contribute to the network until later. Open, asynchronous communication will be critical for establishing shared understanding between teams, improving system resilience by creating knowledge redundancy across teams.

Encourage and plan for growing participation Dynamic, resilient systems should include many diverse components: multiple data providers, multiple modeling teams, and multiple end users. This diversity of components is not likely to be present initially, but will eventually emerge as succession dynamics evolve. As observed in the technological innovation systems literature, the early phases of development are characterized by a small but growing number of teams building high uncertainty products before adoption saturates, growth slows, and participants shake out, leading to stabilization (Markard, 2020). Contributors should be expected to enter and exit the network over time, which should not compromise the overall stability of an effectively decentralized system.

Advance the technology to advance the science Hierarchical systems evolve from the bottom up, and a key systems tenet is that the purpose of the upper layers of the hierarchy is to serve the purposes of the lower layers (Meadows, 2008). This principle certainly applies to technology systems, where the network of components should principally manage the efficient flow of information between systems. But it also reinforces that the fundamental purpose of ecosystem monitoring technologies is to provide timely and high-quality data on how ecological communities are changing.

Although mature technology systems exhibit self-repairing and self-organizing properties, which repair and organize the technology system itself, we should always remember that the operations of these systems should function principally in service of advancing our understanding of how the planet's ecosystems are changing.

7 Conclusion

The organizing principles of ecosystems reveal the design patterns that ecosystem monitoring technologies can adopt for quantifying changes to ecological communities. Monitoring technologies should be designed around the principles of short linkages and rapid feedback loops between components, which applies to all levels of the technology stack and to the communications between people.

The goals of ecosystem monitoring technologies at large and the goals of modern ecological research clearly align around rapidly testing and understanding how the components of ecological systems respond to change. If we align our technology systems with the principles of ecological resilience, our monitoring networks may be able to keep up with the pace at which the world's ecosystems are changing — and help us change with them.

Acknowledgments Thank you to Maxwell Joseph and Amy Rosenthal for insightful input and feedback. The findings and views described herein do not necessarily reflect those of Planet Labs PBC.

References

S. Alleaume, P. Dusseux, V. Thierion, L. Commagnac, S. Laventure, M. Lang, J.-B. Féret, L. Hubert-Moy, and S. Luque. A generic remote sensing approach to derive operational essential biodiversity variables (EBVs) for conservation planning. *Methods in ecology and evolution / British Ecological Society*, 9(8):1822–1836, Aug. 2018. doi:10.1111/2041-210x.13033. Publisher: Wiley.

B. R. Allenby and W. E. Cooper. Understanding industrial ecology from a biological systems perspective. *Environmental Quality Management*, 3(3):343–354, Mar. 1994. doi:10.1002/tqem.3310030310.

C. B. Anderson. Biodiversity monitoring, earth observations and the ecology of scale. *Ecology Letters*, 21(10): 1572–1585, Oct. 2018. doi:10.1111/ele.13106.

K. Bakker. *Gaia's Web: How Digital Environmentalism Can Combat Climate Change, Restore Biodiversity, Cultivate Empathy, and Regenerate the Earth*. MIT Press, Apr. 2024.

G. Balaban, I. Grytten, K. D. Rand, L. Scheffer, and G. K. Sandve. Ten simple rules for quick and dirty scientific programming. *PLOS Computational Biology*, 17(3):e1008549, Mar. 2021. doi:10.1371/journal.pcbi.1008549.

S. Beery, D. Morris, and S. Yang. Efficient Pipeline for Camera Trap Image Review, 2019. Version Number: 1.

B. Beisner, D. Haydon, and K. Cuddington. Alternative stable states in ecology. *Frontiers in Ecology and the Environment*, 1(7):376–382, Sept. 2003. doi:10.1890/1540-9295(2003)001[0376:ASSIE]2.0.CO;2.

J. M. Benyus. *Biomimicry: Innovation Inspired by Nature*. Harper Collins, June 1997.

M. Besson, J. Alison, K. Bjerge, T. E. Gorochowski, T. T. Høye, T. Jucker, H. M. R. Mann, and C. F. Clements. Towards the fully automated monitoring of ecological communities. *Ecology Letters*, 25(12):2753–2775, Dec. 2022. doi:10.1111/ele.14123.

Bloch, Michael, Blumberg, Sven, and Laartz, Jurgen. Delivering large-scale IT projects on time, on budget, and on value. 2012.

B. Boyle, N. Hopkins, Z. Lu, J. A. Raygoza Garay, D. Mozzherin, T. Rees, N. Matasci, M. L. Narro, W. H. Piel, S. J. Mckay, S. Lowry, C. Freeland, R. K. Peet, and B. J. Enquist. The taxonomic name resolution service: an online tool for automated standardization of plant names. *BMC Bioinformatics*, 14(1):16, Dec. 2013. doi:10.1186/1471-2105-14-16.

D. Bradley. Maser's "The Redesigned Forest" reconsidered, Apr. 1994.

J. Chave. The problem of pattern and scale in ecology: what have we learned in 20 years? *Ecology Letters*, 16 Suppl 1:4–16, May 2013. doi:10.1111/ele.12048. Publisher: Wiley.

R. Daroya, E. Cole, O. Mac Aodha, G. Van Horn, and S. Maji. WildSAT: Learning Satellite Image Representations from Wildlife Observations, 2024. Version Number: 1.

D. Fink, A. Johnston, M. Strimas-Mackey, T. Auer, W. M. Hochachka, S. Ligocki, L. Oldham Jaromczyk, O. Robinson, C. Wood, S. Kelling, and A. D. Rodewald. A Double machine learning trend model for citizen science data. *Methods in Ecology and Evolution*, 14(9): 2435–2448, Sept. 2023. doi:10.1111/2041-210X.14186.

D. Fraisl, G. Hager, B. Bedessem, M. Gold, P.-Y. Hsing, F. Danielsen, C. B. Hitchcock, J. M. Hulbert, J. Piera, H. Spiers, M. Thiel, and M. Haklay. Citizen science in environmental and ecological sciences. *Nature Reviews Methods Primers*, 2(1):64, Aug. 2022. doi:10.1038/s43586-022-00144-4.

N. J. Galle, S. A. Nitoslawski, and F. Pilla. The Internet of Nature: How taking nature online can shape urban ecosystems. *The Anthropocene Review*, 6(3):279–287, Dec. 2019. doi:10.1177/2053019619877103.

L. E. Gillespie, M. Ruffley, and M. Exposito-Alonso. Deep learning models map rapid plant species changes from citizen science and remote sensing data. *Proceedings of the National Academy of Sciences*, 121(37):e2318296121, Sept. 2024. doi:10.1073/pnas.2318296121.

L. H. Gunderson. Ecological Resilience—In Theory and Application. *Annual Review of Ecology and Systematics*, 31(1):425–439, Nov. 2000. doi:10.1146/annurev.ecolsys.31.1.425.

R. Guralnick, R. Walls, and W. Jetz. Humboldt Core – toward a standardized capture of biological inventories for biodiversity monitoring, modeling and assessment. *Ecography*, 41(5):713–725, May 2018. doi:10.1111/ecog.02942.

Gurevitch, Jessica, Scheiner, Samuel M., and Fox, Gordon A. *The Ecology of Plants*. Sinauer Associates, Inc, 2nd edition, 2006.

F. Hallé, R. A. A. Oldeman, P. B. Tomlinson, and P. B. Tomlinson. *Tropical trees and forests: an architectural analysis*. Springer, Berlin Heidelberg, 1978.

I. A. Hatton, O. Mazzarisi, A. Altieri, and M. Smerlak. Diversity begets stability: Sublinear growth and competitive coexistence across ecosystems. *Science*, 383(6688): eadg8488, Mar. 2024. doi:10.1126/science.adg8488.

K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. pages 770–778, 2016.

A. Hernandez, Z. Miao, L. Vargas, S. Beery, R. Dodhia, P. Arbelaez, and J. M. L. Ferres. Pytorch-Wildlife: A Collaborative Deep Learning Framework for Conservation, 2024. Version Number: 4.

R. J. Hobbs and H. A. Mooney. *Remote Sensing of Biosphere Functioning*. Springer Science & Business Media, 1990.

Holling, C.S. Resilience and Stability of Ecological Systems. *Annual Review of Ecology and Systematics*, 4: 1–23, 1973.

W. Jackson and J. Piper. The Necessary Marriage Between Ecology and Agriculture. *Ecology*, 70(6):1591–1593, Dec. 1989. doi:10.2307/1938090.

M. G. Jacobides, C. Cennamo, and A. Gawer. Towards a theory of ecosystems. *Strategic Management Journal*, 39(8):2255–2276, Aug. 2018. doi:10.1002/smj.2904.

D. Jäckel, K. G. Mortega, U. Sturm, U. Brockmeyer, O. Khorramshahi, and S. L. Voigt-Heucke. Opportunities and limitations: A comparative analysis of citizen science and expert recordings for bioacoustic research. *PLOS ONE*, 16(6):e0253763, June 2021. doi:10.1371/journal.pone.0253763.

K. Karimi and G. Atkinson. What the Internet of Things (IoT) Needs to Become a Reality. Technical report, June 2013.

T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS journal of photogrammetry and remote sensing: official publication of the International Society for Photogrammetry and Remote Sensing*, 173: 24–49, Mar. 2021. doi:10.1016/j.isprsjprs.2020.12.010.

P. Keil, A. A. M. MacDonald, K. S. Ramirez, J. M. Bennett, G. E. García-Peña, B. Yguel, B. Bourgeois, and C. Meyer. Macroecological and macroevolutionary patterns emerge in the universe of GNU/Linux operating systems. *Ecography*, 41(11):1788–1800, Nov. 2018. doi:10.1111/ecog.03424.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

P. F. Langhammer, J. W. Bull, J. E. Bicknell, J. L. Oakley, M. H. Brown, M. W. Bruford, S. H. M. Butchart, J. A. Carr, D. Church, R. Cooney, S. Cutajar, W. Foden, M. N. Foster, C. Gascon, J. Geldmann, P. Genovesi, M. Hoffmann, J. Howard-McCombe, T. Lewis, N. B. W. Macfarlane, Z. E. Melvin, R. S. Merizalde, M. G. Morehouse, S. Pagad, B. Polidoro, W. Sechrest, G. Segelbacher, K. G. Smith, J. Steadman, K. Strongin, J. Williams, S. Woodley, and T. M. Brooks. The positive impact of conservation action. *Science*, 384(6694): 453–458, Apr. 2024. doi:10.1126/science.adj6598.

A. B. Lovins. *Small is Profitable: The Hidden Economic Benefits of Making Electrical Resources the Right Size*. Routledge, London, 2003. doi:10.4324/9781315073804.

Y. Ma, S. Chen, S. Ermon, and D. B. Lobell. Transfer learning in environmental remote sensing. *Remote Sensing of Environment*, 301:113924, Feb. 2024. doi:10.1016/j.rse.2023.113924.

G. Mai, W. Huang, J. Sun, S. Song, D. Mishra, N. Liu, S. Gao, T. Liu, G. Cong, Y. Hu, C. Cundy, Z. Li, R. Zhu, and N. Lao. On the Opportunities and Challenges of Foundation Models for Geospatial Artificial Intelligence, 2023. Version Number: 1.

J. Markard. The life cycle of technological innovation systems. *Technological Forecasting and Social Change*, 153:119407, Apr. 2020. doi:10.1016/j.techfore.2018.07.045.

D. C. Marvin, L. P. Koh, A. J. Lynam, S. Wich, A. B. Davies, R. Krishnamurthy, E. Stokes, R. Starkey, and G. P. Asner. Integrating technologies for scalable ecology and conservation. *Global Ecology and Conservation*, 7:262–275, July 2016. doi:10.1016/j.gecco.2016.07.002.

D. H. Meadows. *Thinking in Systems: A Primer*. Chelsea Green Publishing, 2008. Google-Books-ID: CpbLAgAAQBAJ.

T. Mens and P. Grosjean. The Ecology of Software Ecosystems. *Computer*, 48(10):85–87, Oct. 2015. doi:10.1109/MC.2015.298.

D. G. Messerschmitt and C. Szyperski. *Software Ecosystem: Understanding an Indispensable Technology and Industry*. MIT Press, Cambridge, MA, USA, Aug. 2003.

D. W. Orr. *Ecological Literacy: Education and the Transition to a Postmodern World*. SUNY Press, Jan. 1992. Google-Books-ID: aiRBTwqDvZ0C.

D. W. Orr. *The Nature of Design: Ecology, Culture, and Human Intention*. Oxford University Press, Oxford, New York, Oct. 2004.

H. M. Pereira, S. Ferrier, M. Walters, G. N. Geller, R. H. G. Jongman, R. J. Scholes, M. W. Bruford, N. Brummitt, S. H. M. Butchart, A. C. Cardoso, N. C. Coops, E. Dulloo, D. P. Faith, J. Freyhof, R. D. Gregory, C. Heip, R. Höft, G. Hurtt, W. Jetz, D. S. Karp, M. A. McGeoch, D. Obura, Y. Onoda, N. Pettorelli, B. Reyers, R. Sayre, J. P. W. Scharlemann, S. N. Stuart, E. Turak, M. Walpole, and M. Wegmann. Essential Biodiversity Variables. *Science*, 339(6117):277–278, Jan. 2013. doi:10.1126/science.1229931.

S. L. Pimm, S. Alibhai, R. Bergl, A. Dehgan, C. Giri, Z. Jewell, L. Joppa, R. Kays, and S. Loarie. Emerging Technologies to Conserve Biodiversity. *Trends in Ecology & Evolution*, 30(11):685–696, Nov. 2015. doi:10.1016/j.tree.2015.08.008.

L. J. Pollock, J. Kitzes, S. Beery, K. M. Gaynor, M. A. Jarzyna, O. Mac Aodha, B. Meyer, D. Rolnick, G. W. Taylor, D. Tuia, and T. Berger-Wolf. Harnessing artificial intelligence to fill global shortfalls in biodiversity knowledge. *Nature Reviews Biodiversity*, Feb. 2025. doi:10.1038/s44358-025-00022-3.

T. Preston-Werner. Semantic Versioning 2.0.0, 2013.

M. A. Rahman and A. T. Asyhari. The Emergence of Internet of Things (IoT): Connecting Anything, Anywhere. *Computers*, 8(2):40, May 2019. doi:10.3390/computers8020040.

E. Rolf, K. Klemmer, C. Robinson, and H. Kerner. Mission Critical – Satellite Data is a Distinct Modality in Machine Learning, 2024. Version Number: 1.

J. Roughgarden, S. W. Running, and P. A. Matson. What Does Remote Sensing Do For Ecology? *Ecology*, 72(6): 1918–1922, Dec. 1991. doi:10.2307/1941546.

S. Sastry, S. Khanal, A. Dhakal, A. Ahmad, and N. Jacobs. TaxaBind: A Unified Embedding Space for Ecological Applications, 2024. Version Number: 1.

M. Shahin, M. Ali Babar, and L. Zhu. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access*, 5:3909–3943, 2017. doi:10.1109/ACCESS.2017.2685629.

J. Snaddon, G. Petrokofsky, P. Jepson, and K. J. Willis. Biodiversity technologies: tools as change agents. *Biology Letters*, 9(1):20121029, Feb. 2013. doi:10.1098/rsbl.2012.1029.

B. D. Sparrow, W. Edwards, S. E. Munroe, G. M. Wardle, G. R. Guerin, J. Bastin, B. Morris, R. Christensen, S. Phinn, and A. J. Lowe. Effective ecosystem monitoring requires a multi-scaled approach. *Biological Reviews*, 95(6):1706–1719, Dec. 2020. doi:10.1111/brv.12636.

M. Taylor and A. Taylor. The technology life cycle: Conceptualization and managerial implications. *International Journal of Production Economics*, 140(1):541–553, Nov. 2012. doi:10.1016/j.ijpe.2012.07.006.

S. J. Tetzlaff, A. D. Katz, P. J. Wolff, and M. E. Kleitch. Comparison of soil edna to camera traps for assessing mammal and bird community composition and site use. *Ecology and Evolution*, 14(7):e70022, July 2024. doi:10.1002/ece3.70022.

D. Tuia, B. Kellenberger, S. Beery, B. R. Costelloe, S. Zuffi, B. Risse, A. Mathis, M. W. Mathis, F. Van Langevelde, T. Burghardt, R. Kays, H. Klinck, M. Wikelski, I. D. Couzin, G. Van Horn, M. C. Crofoot, C. V. Stewart, and T. Berger-Wolf. Perspectives in machine learning for wildlife conservation. *Nature Communications*, 13(1): 792, Feb. 2022. doi:10.1038/s41467-022-27980-y.

K. Ueda. An Overview of Computer Vision in iNaturalist. *Biodiversity Information Science and Standards*, 4: e59133, Oct. 2020. doi:10.3897/biss.4.59133.

J. Wieczorek, D. Bloom, R. Guralnick, S. Blum, M. Döring, R. Giovanni, T. Robertson, and D. Vieglais. Darwin Core: An Evolving Community-Developed Biodiversity Data Standard. *PLoS ONE*, 7(1):e29715, Jan. 2012. doi:10.1371/journal.pone.0029715.

Y. Zhao, X. Yang, and R. R. Vatsavai. A Scalable System for Searching Large-scale Multi-sensor Remote Sensing Image Collections. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3780–3783, Orlando, FL, USA, Dec. 2021. IEEE. doi:10.1109/BigData52589.2021.9671679.

X. X. Zhu, Z. Xiong, Y. Wang, A. J. Stewart, K. Heidler, Y. Wang, Z. Yuan, T. Dujardin, Q. Xu, and Y. Shi. On the Foundations of Earth and Climate Foundation Models, 2024. Version Number: 1.

W. K. Zinsser. *On writing well : the classic guide to writing nonfiction*. Quill/A Harper Collins Books, 2001.