MODULARITY THEORY AND INTERNET REGULATION

*Christopher S. Yoo**

Recent debates over Internet policy have focused on the network's architecture. For example, the central justification for the Federal Communications Commission's Open Internet proceeding is that the Internet's architecture was critical to its past success and must be preserved. Unfortunately, policymakers and commentators have failed to provide any theoretical basis for determining whether any network configuration is optimal or when changed circumstances might justify a change to the architecture. The result is a regulatory approach that is unnecessarily static and reflexively accepting of the status quo, while failing to provide a basis for distinguishing between architectural changes that are part of the network's natural evolution and those that are potentially anticompetitive.

This Article fills the void by providing an analytical framework for assessing network architecture based on modularity theory. It synthesizes modularity theory into five key concepts: near decomposition, interdependencies, abstraction/information hiding, requisite variety, and testing/integration. It then surfaces the tradeoffs inherent in the architectural decision by identifying the benefits and costs associated with modular architectures and analyzing the dynamics of architectural change. It concludes by using the framework to evaluate a series of recent policy applications, including the Telecommunications Act of 1996, network neutrality, the transition from IPv4 to IPv6, calls for opening the Apple iPhone's application programming interfaces ("API"s), and the evolution of future Internet architectures.

* John H. Chestnut Professor of Law, Professor of Communication, Professor of Computer & Information Science, and Founding Director of the Center for Technology, Innovation and Competition, University of Pennsylvania. I would like to thank Sandra Braman, Jonathan Cave, Louis Kaplow, David Schizer, Henry Smith, and attendees at the First Annual Conference on Law and Computer Science at the University of Pennsylvania; 40th Telecommunications Policy Research Conference; the Intellectual Property Scholars Conference; Tilburg Law and Economics Center; the Warren Center for Network & Data Sciences of the University of Pennsylvania; the Harvard Law School Law & Economics Seminar; and the Conference on "Regulating the Evolving Internet Ecosystem," cosponsored by the Federal Communications Commission, American Enterprise Institute, the University of Nebraska College of Law for their comments on earlier drafts. I would also like to thank the Oscar and Miriam Handler Foundation, the New York State Bar Foundation, and University of Pennsylvania Law School's Center for Technology, Innovation and Competition for their financial support.

An architectural approach based on modularity theory yields a number of important insights. It surfaces the inevitability of the tradeoff between generality and cost minimization and how generality can obstruct certain types of innovation. It underscores how parallel experimentation can accelerate innovation at the same time that the lack of coordination can stunt it. Moreover, like any design hierarchy, modular systems can resist technological change. Finally, it under-scores the contingent nature of network architecture and provides heuristics for identifying when architectural change can be beneficial.

TABLE OF CONTENTS

"[T]his Committee must keep in view a fundamental fact about the Internet: . . . the innovation and explosive growth of the Internet is directly linked to its particular architectural design. . . . If this Committee wants to preserve that growth and innovation, it should take steps to protect this fundamental design."

-Statement of Lawrence Lessig Before the Senate Committee on Commerce, Science & Transportation[1]

I. INTRODUCTION

Recent debates over Internet policy have proceeded from the premise that the Internet's success has stemmed in large part from its architecture and that this architecture must be preserved if that success is to continue. The current debate over network neutrality provides an apt illustration of these dynamics. The Open Internet Order that the D.C. Circuit struck down in January 2014 lauded how the network "architecture enables innovators to create and offer new applications and services without needing approval from any controlling entity, be it a network provider, equipment manufacturer, industry body, or government agency," which in turn permits "new technologies to be developed and distributed by a broad range of sources, not just by the companies that operate the network."[2] Although the details of the revised Open Internet Order currently pending before the Federal Communications Commission has not yet been released, most expect that the new order will follow the same line of reasoning.

Commentators generally agree that *modularity*, the partitioning of a system into subsystems that are structurally independent but work together, represents one of the key architectural principles around which

1. *Net Neutrality: Hearing Before the Sen. Comm. on Commerce, Sci. & Transp.*, 109th Cong. 54 (2006) [hereinafter *Net Neutrality Hearing*], *available at* http://www.gpo.gov/fdsys/pkg/CHRG-109shrg30115/pdf/CHRG-109shrg30115.pdf (statement of Prof. Lawrence Lessig).

2. Preserving the Open Internet Broadband Industry Practices, Report and Order, 25 FCC Rcd. 17905, 17910 ¶ 13 (2010) [hereinafter Open Internet Order].

the Internet was designed and to which it owes much of its success.[3] Computer scientists almost universally discuss modularity almost exclusively in laudatory terms and rightfully so.[4] The shift to modularity played a critical role in triggering the explosion in computer technology since the mid-1960s.[5] Indeed, as will be subsequently discussed at greater length, it represents the primary mechanism for making complex systems tractable.[6] Furthermore, the current modular architecture has proven incredibly resilient and robust over a period spanning several decades.[7]

That said, recognizing that modularity provides substantial benefits still leaves a host of questions unanswered. For example, is modularity always the best choice? If not, under what circumstances is modularity optimal and suboptimal?[8] The absence of such a theory overlooks the fact that "neither modular designs nor interdependent designs are *inherently* superior."[9] On the contrary, engineering principles recognize that no one architecture does everything well and thus, that every architecture necessarily involves tradeoffs. Assessments about the relative merits of particular modular schemes require evaluating the forces resting on each side of this balance in any particular context.

3. *See, e.g.*, *Net Neutrality Hearing*, supra note 1, at 10, 12–13 (statement of Vinton G. Cerf); YOCHAI BENKLER, THE WEALTH OF NETWORKS: HOW SOCIAL PRODUCTION TRANSFORMS MARKETS AND FREEDOM 100–03 (2006); LAWRENCE LESSIG, THE FUTURE OF IDEAS 92 (2001); BARBARA VAN SCHEWICK, INTERNET ARCHITECTURE AND INNOVATION 38–40 (2010); JONATHAN ZITTRAIN, THE FUTURE OF THE INTERNET—AND HOW TO STOP IT 31 (2008); Joseph Farrell & Philip J. Weiser, *Modularity, Vertical Integration, and Open Access Policies: Towards a Convergence of Antitrust and Regulation in the Internet Age*, 17 HARV. J.L. & TECH. 85, 90 (2003). Other major architectural principles include protocol layering and the end-to-end argument. For discussions of those concepts, see Christopher S. Yoo, *Protocol Layering and Internet Policy*, 161 U. PA. L. REV. 1707 (2013); and Christopher S. Yoo, *Would Mandating Network Neutrality Help or Hurt Broadband Competition?: A Comment on the End-to-End Debate*, 3 J. ON TELECOMM. & HIGH TECH. L. 23, 23 (2004). This Article addresses only constructed modular architectures in the digital realm. In so doing, it ignores the important literature studying on how modular systems spontaneously evolve in biological systems, which is important, but less relevant to modern information technology. *See, e.g.*, Scott F. Gilbert et al., *Resynthesizing Evolutionary and Developmental Biology*, 173 DEVELOPMENTAL BIOLOGY 357 (1996); George von Dassow & Ed Munro, *Modularity in Animal Development and Evolution: Elements of a Conceptual Framework for EvoDevo*, 285 J. EXPERIMENTAL ZOOLOGY 307 (1999).

4. *See, e.g.*, David Clark, *Forward to the First Edition*, *in* LARRY L. PETERSON & BRUCE S. DAVIE, COMPUTER NETWORKS: A SYSTEMS APPROACH ix, ix (4th ed. 2007) (observing that "[a]ll good computer scientists worship the god of modularity"); M.A. Padlipski, *A Perspective on the ARPANET Reference Model* 1, 7 (Request for Comments 871, Sept. 1982) [hereinafter RFC 871], *available at* http://tools.ietf.org/html/rfc871 (calling modularity a "buzzword" and noting that "'[e]verybody knew' modularity was a Good Thing").

5. CARLISS Y. BALDWIN & KIM B. CLARK, DESIGN RULES, VOL 1: THE POWER OF MODULARITY 221 (2000).

6. *Id.*

7. The protocols that form the essence of the Internet date from 1974. *See* Vinton G. Cerf & Robert E. Kahn, *A Protocol for Packet Network Intercommunication*, 22 IEEE TRANSACTIONS ON COMM. 637 (1974). They became the exclusive core routing protocol on January 1, 1983. Jon Postel, *NCP/TCP Transition Plan* (Network Working Group Request for Comments 801, Nov. 1981), *available at* http://tools.ietf.org/html/rfc801.

8. *See* Carliss Y. Baldwin & Kim B. Clark, *Managing in an Age of Modularity*, HARV. BUS. REV., Sept.–Oct. 1997, at 84, 86 ("If modularity brings so many advantages, why aren't all products [and processes] fully modular?").

9. BALDWIN & CLARK, *supra* note 5, at 258; *accord id.* at 257 (noting that "there is no economic 'right answer' that works in all cases" when choosing "between an interconnected and a modular approach").

Moreover, even when modularity is desirable, there are many possible ways to modularize a system. Any designer of a modular system must decide how many modules to create, which design elements should be assigned to each module, where the interfaces between modules should be located, and how the interfaces between the modules should be configured.[10] Identification of those determinants allows policymakers to create a dynamic theory of architectural design by asking an equally important question: how does one recognize the conditions under which a modular architecture should change?

Unfortunately, the considerations underlying these decisions remain poorly understood. As one commentator has noted, "the design tradeoffs inherent in abstracting from physical resources are rarely acknowledged in the computing literature."[11] Other commentators similarly observed that "there is little systematic research on how decision makers partition designs into modules and what . . . the risks [are] of partitioning incorrectly."[12] As a result, the literature tends to treat modular architectures as immutable artifacts without providing any heuristics to help determine when change might be beneficial.

This Article seeks to fill the void by exploring the theory underlying modularity and applying it to recent and current issues in Internet policy. Part II lays out the basic concepts that serve as the foundation for modularity theory. These include near decomposition, the role of interdependencies, abstractions/interfaces, hidden information, requisite variety, and testing/integration. Together these concepts underscore that the optimal structure of any complex system is determined by technological interdependencies and social considerations.

Part III describes modularity's benefits, including the way it makes complexity more manageable, increases the speed of innovation, facilitates outsourcing, and promotes flexibility. Part IV surfaces the tradeoff inherent in modularity by analyzing its costs, including the sunk costs of designing a modular architecture, reduction in cost minimization, restrictions on innovation, limitations on alternate institutional forms, lack of coordination, and excessive rigidity.

Part V examines the dynamics underlying changes in modular networks. This framework makes clear that modular architectures are not natural constructs. Instead, they depend on factors such as the nature of the interdependencies, technological heterogeneity, speed of technological change, and demand heterogeneity. As such, they should be expected to change over time.

10. Sendil K. Ethiraj & Daniel Levinthal, *Modularity and Innovation in Complex Systems*, 50 MGMT. SCI. 159, 162 (2004); *see also* Richard N. Langlois, *Modularity in Technology and Organization*, 49 J. ECON. BEHAVIOR & ORG. 19, 24 (2002) ("The real issue is normally not whether to be modular but *how* to be modular. Which modularization, which structure of encapsulation boundaries, will yield the best system decomposition?").

11. Jean-François Blanchette, *A Material History of Bits*, 62 J. AM. SOC'Y INFO. SCI. & TECH. 1042, 1047 (2011).

12. Ethiraj & Levinthal, *supra* note 10, at 160.

Part VI applies the framework to four major issues in telecommunications policy: access to unbundled network elements under the Telecommunications Act of 1996, network neutrality, the shift from Internet Protocol version 4 ("IPv4") to version 6 ("IPv6"), calls for platforms such as the Apple iPhone and social media sites such as Twitter and Facebook to provide open access to their application programming interfaces ("APIs"), and proposals for future Internet architectures. These examples show how the theory underlying modularity can provide some insights into how best to craft Internet policy.

II. MODULARITY THEORY BASICS

Any assessment of whether a particular modular regime is socially desirable necessarily depends on having some theoretical framework for understanding why modularity exists and what benefits it provides. This Part lays out the basic concepts that provide the foundation for that theory, including near decomposition, the role of interdependencies, abstractions and hidden information in interfaces, requisite variety, and testing and integration.

A. Near Decomposition as a Solution to Complexity

The special challenges associated with developing complex systems are well illustrated by an anecdote related by Frederick Brooks in his celebrated book, *The Mythical Man Month*, which documents IBM's efforts to design System/360, the first fully modular family of computer systems.[13] Consistent with the prevailing conventional wisdom that everyone working on a project should have access to all of the available information about the project, the managers of the System/360 team insisted that every programmer maintain a formal workbook documenting all of the other parts of the system. After only six months, the workbook was more than five feet thick and required one hundred and fifty pages of updates every day.[14] This led to Brooks' famous observation that adding more people sometimes slows down projects instead of speeding them up. This results from the communication cost associated with keeping more people informed possibly exceeding the potential advantages of more man-hours and a greater division of labor.[15]

The classic solution to this problem is implicit in Herbert Simon's landmark analysis of complex systems.[16] Simon suggested that one of the best ways to reduce complexity is to decompose the overall system into a series of subsystems in which interactions within subsystems are relatively frequent and stronger, while interactions between subsystems are rela-

13. FREDERICK P. BROOKS, THE MYTHICAL MAN-MONTH: ESSAYS ON SOFTWARE ENGINEERING (1975); Langlois, *supra* note 10, at 22 (discussing Brooks' book).
14. BROOKS, *supra* note 13, at 76–77.
15. *Id.* at 18–19.
16. Herbert A. Simon, *The Architecture of Complexity*, 106 PROC. AM. PHIL. SOC'Y 467 (1962).

tively rare and weaker.[17] This allows subsystems to operate largely independently of one another in the short run, but to integrate into a larger complex system over the long run.[18] Such a system would only be *nearly* decomposable rather than *completely* decomposable, as the fact that all of the subsystems were part of a larger system meant that some interconnections between subsystems would necessarily remain. That said, interactions among subsystems would remain relatively weak, but nonnegligible.[19]

Near decomposition of complex systems yields a number of advantages. As an initial matter, it makes complex systems easier to describe and comprehend.[20] By creating a larger number of intermediate forms that can constitute building blocks for the larger system, near decomposition permits experimentation to occur on a smaller scale instead of with the system as a whole.[21] The existence of stable intermediate forms also allows complex systems to evolve more rapidly.[22]

A related insight emerges from the field of software engineering, which has modeled complex tasks into a series of abstract sequential processes.[23] Subdividing a larger process into a series of smaller subprocesses greatly facilitated software engineers' ability to verify the accuracy of the system by allowing them to test portions of the overall system in isolation.[24] Similarly, design scholars have turned to near decomposition to deal with complexity of designs by organizing tasks into groups that are "interlinked, yet sufficiently free of one another to adjust independently in a feasible amount of time."[25]

B. Interdependencies as a Determinant of Module Boundaries

To say that complex systems are more easily analyzed and developed if broken down into smaller subcomponents does not provide much information about how such a decomposition should be implemented. Consistent with Simon's observation that interactions should be frequent and strong within subsystems and rare and weak across subsystems,[26] Carliss Baldwin and Kim Clark argue that the key to understanding the

17. *Id.* at 474, 477.
18. *Id.* at 474.
19. *Id.* Simon noted that to be partially decomposable, a complex system must be hierarchical. Systems that consist of a repeating structure, such as a linear polymer or a crystal such as a diamond, are too flat to permit partial decomposition. *Id.* at 469. To the extent that such systems are regarded as hierarchical, they represent the trivial case of a hierarchy with a span of one. *Id.* at 471.
20. *Id.* at 477.
21. *See id.* at 470–71.
22. *Id.* at 473.
23. For a classic early description, see Edsger W. Dijkstra, *The Structure of the "THE"-Multiprogramming System*, 11 COMM. ACM 341, 343 (1968).
24. *Id.* at 344.
25. CHRISTOPHER ALEXANDER, NOTES ON THE SYNTHESIS OF FORM 43 (1967); *see also id.* at 81–83, 116–19.
26. *See* Simon, *supra* note 16, at 477 and accompanying text (observing that in near decomposable systems, "[i]ntracomponent linkages are generally stronger than intercomponent linkages").

"points of natural division" between the elements of a modular architecture is *task interdependencies*.[27] System architects must group tasks that have strong interdependencies with each other inside the same protomodule, and then must "systematically . . . sever all dependencies known to exist across the protomodules."[28] This implies that a well-designed module is "a unit whose structural elements are powerfully connected among themselves and relatively weakly connected to elements in other units."[29] The boundaries between modules are located at the "thin crossing points" where interdependencies are the thinnest.[30] Design theorist Christopher Alexander similarly advocated including variables whose interactions are very rich within the same subset, thereby permitting the subsets to act relatively independently.[31] Organizational theorist James Thompson advocated grouping reciprocally interdependent tasks together and relying on key positions to serve as bridges to other groups.[32]

Encapsulating highly interdependent tasks within the same module also reduces the costs of communication and coordination by reducing the need for cross-boundary communication,[33] with cheaper cross-group communication leading to a larger number of modules.[34]

The initial modular design must be based on the architect's expertise.[35] Establishing these basic design rules has the effect of privileging certain parameters and "declaring certain parts of the design space to be out of bounds."[36] No matter how well system architects believe they un-

27. BALDWIN & CLARK, *supra* note 5, at 64.

28. *Id.* at 70.

29. *Id.* at 63; *accord* MEILIR PAGE-JONES, THE PRACTICAL GUIDE TO STRUCTURED SYSTEMS DESIGN 101–03 (1980) (arguing that the measure of good modular design is the extent to which it minimizes interdependence between modules to make them as independent as possible); Sendil K. Ethiraj et al., *The Dual Role of Modularity: Innovation and Imitation*, 54 MGMT. SCI. 939, 939 (2008) ("It is generally accepted that a modular design is based on a principle of encapsulating interdependencies within self-contained units called modules and minimizing reciprocal interdependencies between modules."); Eric von Hippel, *Task Partitioning: An Innovation Process Variable*, 19 RES. POL'Y 407, 409, 411–12 (1990) (arguing that innovation projects should be partitioned to minimize problem-solving interdependencies, which occur when changes in the information relating to one task require problem-solving in tasks on the other side of the partition); *see also* Melissa A. Schilling, *Toward a General Modular Systems Theory and Its Application to Interfirm Product Modularity*, 25 ACAD. MGMT. REV. 312, 316 (2000) (noting that the scope of modularity is determined by a system's "synergistic specificity," which arises when "components of the system . . . require such extensive interaction . . . that any change in a component requires extensive compensating changes in other components of the system, or else functionality is lost.").

30. Carliss Y. Baldwin & Kim B. Clark, *Modularity in the Design of Complex Engineering Systems*, *in* COMPLEX ENGINEERED SYSTEMS: SCIENCE MEETS TECHNOLOGY 175, 199 (Dan Braha et al. eds., 2006); *accord* von Hippel, *supra* note 29, at 409. A related, but distinct, approach focuses not on current interdependencies, but rather on future technological opportunities, by identifying "difficult design decisions or design decisions which are likely to change" and making sure that those decisions are locked within a single module. D.L. Parnas, *On the Criteria To Be Used in Decomposing Modules*, 15 COMM. ACM 1053, 1058 (1972).

31. ALEXANDER, *supra* note 25, at 43, 64–65, 121–24.

32. JAMES D. THOMPSON, ORGANIZATIONS IN ACTION 58–60 (1967).

33. von Hippel, *supra* note 29, at 409–10.

34. *See* Scott Schaefer, *Product Design Partitions with Complementary Components*, 38 J. ECON. BEHAV. & ORG. 311, 320 (1999).

35. *See* BALDWIN & CLARK, *supra* note 5, at 68.

36. *Id.* at 68–69.

derstand the systems they are designing, they cannot fully understand the way the various components interact with one another until they gain experience with the system by experimenting with different solutions.[37] If done badly, the proposed modularization can cause the system to perform suboptimally[38] or may require a redesign of the architecture.[39] In the worst case scenario, the interdependencies cannot be solved without redesigning the basic architecture, in which case the modularization will fail.[40]

This means that modular systems are never designed in a vacuum. Redesigns of current modular systems are framed by the existing modularization. Even with respect to new modularizations, they must respond and adjust to the initial design proposed by the architect, which, because of the lack of perfect prescience, typically includes some interdependencies that the architect did not foresee.[41] The recursive and restricted nature of this adjustment process means that modularization emerges more through a process of "improvisation, bricolage, and drift" than from "planification and control."[42]

C. Interface Design and the Role of Abstraction and Hidden Information

Dividing a production process into blocks of interdependent processes is only one step in modularizing a system. In addition, the architects must structure the interactions between the modules.[43] To do this, each module must be reduced to an *abstraction*,[44] which is a simplification

37. *See id.* at 254 ("Given such a high degree of complexity, it simply is not possible for designers to know enough about the system to eliminate all uncertainty. Thus each new design is fundamentally an experiment. Its outcome may be guessed, but it cannot be known ahead of time."); Ethiraj & Levinthal, *supra* note 10, at 172 ("Designers engage in acts of creation, but unlike a divine creator, they lack omniscience. Choices of modules are guesses about appropriate decompositions—decompositions that even in reality are only partial (i.e., nearly decomposable).").

38. *See* Langlois, *supra* note 10, at 26 (noting that "freezing the design rules too early may result in an inferior modularization").

39. BALDWIN & CLARK, *supra* note 5, at 70 ("Imposing a design rule when one is ignorant of the true underlying interdependencies can lead to design failure. . . . The best course of action, if possible, is to rescind the original design rule, which was based on insufficient knowledge of the critical interdependencies.").

40. *Id.* at 86 ("In a newly modularized design, when the modules are first brought together, problematic interactions usually appear and workarounds to those problems must be devised. In the worst cases, the workaround will feed back into the module designs, and the modularization itself will fail.").

41. *See* Schaefer, *supra* note 34, at 325–26 (showing that the problem devising a truly optimal modular design partition is NP complete in that its complexity increases exponentially with the size of the problem and concluding "it would seem unlikely that a firm could ever hope to uncover an optimal modular design partition for a complex product").

42. Blanchette, *supra* note 11, at 1055; *accord* Ethiraj & Levinthal, *supra* note 10, at 160, 162, 172.

43. BALDWIN & CLARK, *supra* note 5, at 77.

44. For a seminal statement about the benefits of abstractions, see Dijkstra, *supra* note 23, at 343. *See also* HERBERT A. SIMON, THE SCIENCES OF THE ARTIFICIAL 17–20, 100–01 (1969). For an earlier statement advocating that programs be built around generalized software modules, see W.C. McGee, *Generalization: Key to Successful Electronic Data Processing*, 6 J. ACM 1 (1959).

of reality "that suppresses details of elements that do not affect how [modules] use, are used by, relate to, or interact with other elements."[45]

Once a system architect predetermines how the modules will interact with one another, the architect must ensure that the programmers working on each module refrain from introducing new interdependencies that fall outside of the design. Individual programmers who have access to all of the information about the entire system will be tempted to use that information to improve the performance of the module on which they are working. The problem, according to David Parnas' seminal work, is that doing so can "disastrously increase the connectivity of the system" by introducing connections between modules that violate the goal of keeping the intermodule interdependencies to a minimum.[46]

Parnas identified an ingenious way to ensure that the architect retains control over the overall structure. Instead of making all of the design information available to everyone, Parnas argued that architects could maintain the architecture's integrity by maintaining strict control over the distribution of information across the system.[47] Specifically, architects could carefully design the interfaces connecting modules so that they include only the information associated with the interdependencies that other modules were permitted to take into account and to withhold information about the module's inner working from the other modules.[48] Information included in the interface is part of the *visible information* shared between modules; design parameters not included in the interface are part of the *hidden information*.[49]

Hiding information about the highly interdependent tasks that are supposed to be encapsulated within modules makes those tasks opaque to other modules and thus prevents them from creating interconnections with those tasks that fall outside the architectural design.[50] This permits other modules to treat every other module as a "black box,"[51] which greatly reduces communication costs.[52] After the computer science community criticized Parnas' proposal as "radical" and unworkable,[53] Parnas took it upon himself to validate the concept by undertaking a real-world implementation of it.[54]

45. LEN BASS ET AL., SOFTWARE ARCHITECTURE IN PRACTICE 21 (2d ed. 2003).
46. David L. Parnas, *Information Distribution Aspects of Design Methodology*, 1 INFO. PROCESSING 71: PROC. IFIP CONG. 71, at 339, 342 (1972).
47. *Id.*
48. *Id.* at 344; *accord* Parnas, *supra* note 30, at 1056 ("The second decomposition was made using 'information hiding' as a criterion. . . . Every module in the second decomposition is characterized by its knowledge of a design decision which it hides from all others. Its interface or definition was chosen to reveal as little as possible about its inner workings." (citation omitted)).
49. BALDWIN & CLARK, *supra* note 5, at 73–75.
50. Langlois, *supra* note 10, at 22.
51. BALDWIN & CLARK, *supra* note 5, at 91.
52. Langlois, *supra* note 10, at 22–23.
53. BROOKS, *supra* note 13, at 78.
54. D.L. Parnas et al., *The Modular Structure of Complex Systems*, PROC. 7TH INT'L CONF. ON SOFTWARE ENG'G 408 (1984).

Interfaces thus establish the fundamental assumptions that the various modules in a system make about one another and predetermine the manner in which the modules will interact.[55] As such, decisions about what information to include in module interfaces thus represent critical architectural choices that must be made deliberately. As one noted modularity theorist pointed out, "Managers should understand that component interfaces are not minor technical details to be left to the engineering staff. Rather, interfaces will determine the range of strategic flexibilities the company will have to configure its products and adapt its processes in the future."[56] In other words, interface design determines the functionality of the system. Moreover, the points of interactions cannot be dealt with in an ad hoc manner. Instead, they must be maintained throughout the architecture in a consistent and systematic way.[57]

Limiting the number of interdependencies that pass between modules and the number of states associated with those interdependencies of which other modules have to take account simplifies the complexity of the system. For example, interdependencies sometimes form a circuit that loops back onto itself, such as occurs when task *A* depends on the design of task *B*, which depends on the design of task *C*, which recursively depends on the design of task *A*. When this occurs, the only way to determine the effect of the interdependencies is to cycle through a series of iterations until the design converges on a solution.[58] If the design architecture does not limit the number of interdependencies, system designers must test every possible combination of interdependencies, which can lead to a "combinatorial explosion of system variants" if left uncabined.[59] Modularity reduces the number of cycles required by narrowing and predetermining the universe of potentially relevant interdependencies and states.[60]

D. Requisite Variety as a Demand-Side Consideration

While insightful, the interdependency-driven approach to modularity is subject to an important limitation. In focusing on the technological characteristics of production processes, the modularity literature focuses exclusively on the supply side. This approach ignores the impact of changes on the demand side, such as changes in geographic dispersion and the desire for variety. The addition of demand-side considerations provides another potential impetus for architectural change.

55. Parnas, *supra* note 30, at 339.
56. Ron Sanchez, *Fitting Together a Modular Approach*, 81 MFG. ENG'R 216, 217 (2002); *see also* Ethiraj & Levinthal, *supra* note 10, at 172 (noting that in addition to innovation, "modularity has other important implications for strategy, organizational coordination, incentives, and so on").
57. BALDWIN & CLARK, *supra* note 5, at 73.
58. *Id.* at 51–52, 68–70; *see also* Ethiraj & Levinthal, *supra* note 10, at 160 (describing the trial-and-error process in which changes in one module of the Itanium chip would cause ripple effects that disrupted the work on other modules).
59. BALDWIN & CLARK, *supra* note 5, at 272–75.
60. Dijkstra, *supra* note 23, at 344.

In this regard, the literature on general purpose technologies ("GPT"s) is instructive. GPTs are defined as technologies that are (1) widely used, (2) capable of ongoing technical improvement, and (3) enabling innovation in applications sectors.[61] In one sense, the persistence of GPTs represents something of a puzzle. Adam Smith's famous admonition that "the division of labor is limited by the extent of the market" suggests that as the demand for a product increases, its inputs should be produced by increasingly specialized vertically disintegrated firms.[62]

Timothy Bresnahan and Alfonso Gambardella provide an answer to this puzzle.[63] When the growth of the market consists not of an increase in the output of a single good, but rather the proliferation of distinct application sectors supported by the GPT, the effect of market growth is to entrench the GPT rather than to prompt its vertical disintegration.[64] At the same time, Bresnahan and Gambardella recognize that "the broad applicability of a general specialty is not free" and that as the technological needs become increasingly differentiated, the superior matching benefits from localizing technology eventually dominate the benefits from generalization.[65]

In other words, increases in heterogeneity in demand increases the benefits of relying on custom inputs tailored to a particular application's specific needs, which in turn makes GPTs relatively less attractive. At some point, demand becomes so heterogeneous that abandonment of the GPT becomes optimal.

Increases in the volatility of the external environment can also create pressures for the architecture to change, as demonstrated by W. Ross Ashby's landmark Law of Requisite Variety, which studied ways that cybernetic systems comprised of multiple independent components can restore equilibrium when confronted with disturbances.[66] Ashby's model takes the form of a two-step game. In the first step, an external disturbance occurs. In the second step, a regulator chooses from the set of responses that it has designed to respond to each disturbance. Ashby illustrated his model with the matrix depicted in Figure 1.[67]

61. Timothy Bresnahan, *General Purpose Technologies*, *in* 2 HANDBOOK OF THE ECONOMICS OF INNOVATION 761, 764 (Bronwyn Hall & Nathan Rosenberg eds., 2010).

62. ADAM SMITH, AN INQUIRY INTO THE NATURE AND CAUSES OF THE WEALTH OF NATIONS 21 (1776); *see also* George J. Stigler, *The Division of Labor Is Limited by the Extent of the Market*, 59 J. POL. ECON. 185 (1951).

63. *See generally* Timothy Bresnahan & Alfonso Gambardella, *The Division of Inventive Labor and the Extent of the Market*, *in* GENERAL PURPOSE TECHNOLOGIES AND ECONOMIC GROWTH 253 (Elhanan Helpman ed., 1998).

64. *Id.* at 255, 261–62, 272–73.

65. *Id.* at 255, 269–70, 273.

66. W. ROSS ASHBY, AN INTRODUCTION TO CYBERNETICS 206–13 (1956).

67. *Id.* at 202.

FIGURE 1

		Response		
		α	β	γ
Disturbance	1	*b*	*a*	*c*
	2	*a*	*c*	*b*
	3	*c*	*b*	*a*

Ashby added the further constraint that no column contain a repeated outcome. If so, the regulator need not change its move in response to different disturbances, which made the game "too easy to be of interest."[68] In other words, any rows with duplicate outcomes can simply be collapsed into a single row and treated as the same disturbance.

Suppose that the regulator's goal is to force good outcome *a* and avoid bad outcomes *b* and *c*. The number of responses at its disposal gives it the ability to do so. If disturbance 1 occurs, the regulator chooses response β. If disturbance 2 occurs, the regulator chooses response α. If disturbance 3 occurs, the regulator chooses response γ. Indeed, this table gives the regulator complete control over all outcomes, as it can also force outcomes *b* or *c* to occur regardless of the nature of the disturbance.

Ashby next explored the relationship between the number of disturbances, the number of responses at the regulator's disposal, and the number of possible outcomes. If there are only three types of disturbances and the regulator has three responses at its disposal, the regulator can design its responses to ensure that the game always results in a single good outcome, depicted in Figure 2 by the letter *k*.

68. *Id.* at 204.

FIGURE 2

	Response		
Disturbance	α	β	γ
1	*k*	-	-
2	-	*k*	-
3	-	-	*k*
4	*l*	-	-
5	-	*l*	-
6	-	-	*l*
7	*m*	-	-
8	-	*m*	-
9	-	-	*m*

Suppose, however, that a fourth type of disturbance emerges. The assumption that any particular outcome cannot appear more than once in any particular column dictates that the existence of this fourth type of disturbance necessarily means that the regulator can no longer restrict the game to a single outcome. Instead, it must tolerate at least two possible outcomes, *k* and *l*, and can do so for up to six disturbances. The emergence of a seventh type of disturbance would require the regulator to permit a third outcome, *m*.[69]

Stated more generally, if the regulator has *n* possible responses, it can ensure a single outcome so long as the number of disturbances is less than or equal to *n*, after which point the number of possible outcomes must increase to two. The number of disturbances can increase up to 2*n* before the number of possible outcomes increases again to three. The number of outcomes will increase for every additional *n* disturbances. In short, the total variety of outcomes cannot be fewer than the number of disturbances divided by the number of responses.[70] This implies that as the number of potential disturbances increases, the regulator either must be able to tolerate a wider range of outcomes or must increase the number of possible responses.

69. *See id.* at 205–06.
70. *Id.* at 206.

This insight leads directly to the Law of Requisite Variety. Simply stated, increasing the variety of responses is the only way to force down the variety of outcomes arising from the variety in disturbances.[71] In other words, only variety (in responses) can destroy variety (in disturbances).[72] Or in the words of another theorist, "to be efficaciously adaptive, the internal complexity of a system must match the external complexity it confronts."[73]

The Law of Requisite Variety has direct implications for the design of a modular architecture. Two noted modularity theorists have created a model in which certain changes in the environment give rise to a certain type of modularization. Most importantly for our purposes, fragmentation of the external environment (through increased "geographic dispersion, specialized market niches, and varied demands on the system") requires that the system adopt new responses to register the additional sources of variety.[74]

Furthermore, because the number of distinct disturbances in the environment determines the number of responses that must be built into the architecture, a regulator must ascertain which disturbances are in fact distinct. While it is tempting to treat every single disturbance as unique, increasing the number of responses is expensive.[75] On the other hand, grouping multiple disturbances together risks oversimplification, which in turn would leave the system unable to respond to important contingencies.[76] The designer of a modular system must determine which disturbances must be taken into account and which can be safely ignored. Such an exercise cannot be conducted in the abstract and must reflect the motivations and expectations around which the system was built[77] as well as the context surrounding it.[78]

The demand-side literature thus augments the focus on interdependency by adding a second potential source of architectural change. Unlike interdependencies, which, because they reflect the nature of the production function, are necessarily supply-side considerations, factors such as geographic dispersion, market differentiation, and the desire for variety are quintessential demand-side considerations. In this manner, the addition of demand-side factors is consistent with the core economic insight of neoclassical economics that rejected value as determined solely

71. *Id.*

72. *Id.* at 207.

73. Max Boisot & Bill McKelvey, *Complexity and Organization-Environment Relations: Revisiting Ashby's Law of Requisite Variety*, *in* THE SAGE HANDBOOK OF COMPLEXITY AND MANAGEMENT 279, 279 (Peter Allen et al. eds., 2011).

74. J. Douglas Orton & Karl E. Weick, *Loosely Coupled Systems: A Reconceptualization*, 15 ACAD. MGMT. REV. 203, 207–208, 217 fig.1 (1990).

75. Boisot & McKelvey, *supra* note 73, at 283.

76. *Id.* at 279.

77. *Id.* at 283.

78. Jeff Goldstein, *Requisite Variety and the Difference That Makes a Difference: An Introduction to W. Ross Ashby's "Variety, Constraint and Law of Requisite Variety"*, 13 EMERGENCE: COMPLEXITY & ORG. 190, 192–97 (2011).

by supply-side factors and replaced it with a vision in which value is determined by the interaction of both supply and demand.[79]

The addition of demand-side considerations suggests that the debate over whether innovation is driven by consumer-driven "demand pull" or technologically determined "supply push" is based on something of a false dichotomy. Interestingly, these two frameworks can have very different implications for increased complexity. While an increase in interdependencies can lead to an increase in module size in order to encapsulate more of them within a single module, an increase in the complexity of the external environment implies a decrease in module size in order to allow the system to better deal with the underlying variety.

It is tempting to conclude that increases in demand heterogeneity can be met simply by increasing the number of modules and simplifying the interfaces connecting them.[80] Dividing a system into a larger number of modules increases the number of possible product configurations combinatorially.[81] Greater configurability has little value when demand is homogeneous, but increases in value as consumer preferences become more diverse.[82] Variety also provides the most benefit when technologies are changing rapidly and the level of competition is intense,[83] and provides fewer advantages when the technological and business environment is relatively stable.[84]

As I shall later discuss in greater detail, this perspective ignores the fact that modularity promotes certain types of new product configurations while inhibiting others.[85] Modularity makes feasible innovations that involve new combinations or the improvement of existing modules. At the same time, predefining the relationship between certain clusters of tasks and the information passing between them forecloses innovations that would recombine tasks in a fundamentally different way.[86] The impact of modularity on product configuration is thus ambiguous. Increasing the number of modules thus only facilitates recombinations that "are compatible with the overall system architecture."[87] At the same

79. *See generally* ALFRED MARSHALL, PRINCIPLES OF ECONOMICS (London, MacMillan 1890).

80. Orton & Weick, *supra* note 74, at 210 (noting that "registering [sources of variety] improves when elements become more numerous and the constraints among them weaken").

81. Schilling, *supra* note 29, at 323.

82. *Id.* at 317, 324–25.

83. Melissa A. Schilling & H. Kevin Steensma, *The Use of Modular Organizational Forms: An Industry-Level Analysis*, 44 ACAD. MGMT. J. 1149, 1162 (2001).

84. Langlois, *supra* note 10, at 23–24.

85. *See infra* Part IV.C.

86. *See* Rebecca M. Henderson & Kim B. Clark, *Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms*, 35 ADMIN. SCI. Q. 9, 11–13 (1990) (distinguishing between changes in knowledge with respect to individual components and changes in architectural knowledge "about the ways in which the components are integrated and linked together into a coherent whole").

87. Schilling, *supra* note 29, at 315.

time, it preempts recombinations that would reorganize the hierarchy in a different manner.[88]

E. *Testing and Integration*

Another essential element of any modular design is a means for testing and system integration.[89] Because any initial modular design is necessarily incomplete, the architects of any modular system must provide some way to ensure that the system is functioning properly and that each module is functioning in accordance with the overall design.[90] Indeed, such tests constitute a necessary precondition to distributed production systems in which different modules are produced by different firms. In the words of Baldwin and Clark, "The testable, verifiable dimensions of the module are the foundation that supports arm's length-contracts and market transactions."[91] Indeed, "without tests, there is no way to know what is being bought and sold."[92]

The need to test each possible state one module can occupy against every possible combination of states that other modules can occupy represents what Baldwin and Clark have called the "Achilles' heel of modular designs."[93] Without modularity, this problem would have been far worse. Dijkstra's seminal analysis concluded that under integrated, non-modular architectures, "the number of 'relevant states' would have exploded to such a height that exhaustive testing would have been an illusion."[94] The failure to limit the number of ways different components can interact with one another leads to a "combinatorial explosion of system variants."[95]

If the only available tests operate at the system level, architects cannot escape the combinatorial explosion of states that must be tested against one another.[96] Thus, when system-level tests are the only option, designers inevitably tend towards unmodularized, integrated designs and low levels of experimentation.[97] The exponential nature of the combinatorial explosion means that this problem cannot be overcome simply by making system-level testing more rapid and less expensive.[98]

The situation is quite different when tests can be conducted on individual modules. Restricting the number of interdependencies and the number of states associated with those interdependencies that other

88. Kim B. Clark, *The Interaction of Design Hierarchies and Market Concepts in Technological Evolution*, 14 RES. POL'Y 235, 249 (1985).
89. BALDWIN & CLARK, *supra* note 5, at 72, 77, 246.
90. *Id.* at 70–72, 77.
91. *Id.* at 380
92. *Id*.
93. *Id.* at 272.
94. Dijkstra, *supra* note 23, at 344.
95. BALDWIN & CLARK, *supra* note 5, at 273–75.
96. *See supra* note 95 and accompanying text.
97. BALDWIN & CLARK, *supra* note 5, at 275.
98. *Id.* at 276.

parts of the system must take into account dramatically reduces the number of combinations that must be tested.[99] Module-level testing permits the evaluation of individual modules against the module metrics inherent in the design rather than the entire universe of possible combinations.[100] Later studies have confirmed this insight, showing that the cost of testing grows sublinearly with the number of physical elements for modular architectures, whereas the cost of testing grows exponentially for integrated architectures.[101]

The presence of robust module-level testing is thus a necessary precondition for any modular system in which different firms produce different components. Indeed, the ability to verify that each module is working as envisioned by the architecture is a necessary condition to third-party, distributed production of modular components.[102] Over time, new design rules are created to address the out-of-module interdependencies revealed by the testing. Eventually, the design rules may become so complete that testing and system integration can be conducted by the module designers and end users or may even disappear.[103] At this point, the architects will have modularized themselves out of a job.[104]

III. THE BENEFITS OF MODULARITY

It is widely recognized that modular architectures provide a number of benefits. Dividing the overall system into a series of subsystems makes complexity more manageable. It accelerates innovation by allowing experiments on different parts of the system to proceed in parallel. It facilitates the division of labor across different work groups and firms. It increases value by allowing the postponement of key technical decisions. Lastly, it promotes flexibility by allowing individual modules to be changed without disturbing the other parts of the complex system.

A. Making Complexity More Manageable

Breaking down complex systems into subparts helps make them more manageable. As an initial matter, partitioning a large system into smaller modules divides the larger task into subtasks "small enough to be fully comprehend[ed] by a single individual."[105] In this manner, modularity allows architects to work on individual modules in isolation without having to grapple with the architecture as a whole.[106] Conversely, an ar-

99. *Id.* at 277–79.
100. *Id.* at 277–78.
101. Christoph H. Loch et al., *Parallel and Sequential Testing of Design Alternatives*, 45 MGMT. SCI. 663, 673 (2001).
102. BALDWIN & CLARK, *supra* note 5, at 380.
103. *Id.* at 77, 250, 268.
104. *Id.* at 268–69.
105. Blanchette, *supra* note 11, at 1046.
106. Parnas, *supra* note 30, at 1054, 1056.

chitect can understand the overall system without having to understand the details of every module.[107]

Moreover, the fact that the design architecture predetermines the relationships between modules makes the elements and functions of complex systems easier to describe and understand.[108] One needs to know only the design rules defining the interactions with adjacent modules.[109] The details of the interactions contained within modules can be ignored without losing any relevant information.[110] Moreover, reducing the total number of interdependencies and organizing them in systematic, predictable ways yields particularly strong benefits in reducing the cost of testing programs.

B. *Enabling the Division of Labor Across Work Groups and Firms*

Modularity also has major organizational implications. As Parnas noted in his seminal analysis on information hiding, modularity allows "separate groups [to] work on each module with little need for communication."[111] Richard Langlois and Paul Robertson similarly note how modularity facilitates the division of labor by enabling autonomous innovation that requires little coordination among modules.[112] Intermodule coordination is enforced not by managerial authority, but rather by the implicit relationships embedded in the module interfaces without the need for hierarchy of managerial authority.[113] It is the information structures embedded in the interfaces that bind the parts of the modular architecture together.[114]

The same independence that permits modules to be developed by separate work groups also facilitates their development by different firms.[115] Because the coordination is embedded in the information structure of the architecture, firms can jointly organize tasks without being subject to common managerial authority.[116] Moreover, the existence of

107. *See* Parnas et al., *supra* note 54, at 410.

108. *See* Simon, *supra* note 16, at 477.

109. *See* BALDWIN & CLARK, *supra* note 5, at 91 ("The design rules are the visible information . . . .").

110. *Id.*

111. Parnas, *supra* note 30, at 1054.

112. Richard N. Langlois & Paul L. Robertson, *Networks and Innovation in a Modular System: Lessons from the Microcomputer and Stereo Component Industries*, 21 RES. POL'Y 297, 302 (1992).

113. Ron Sanchez & Joseph T. Mahoney, *Modularity, Flexibility, and Knowledge Management in Product and Organization Design*, 17 STRATEGIC MGMT. J. 63, 64–66, 73 (1996).

114. *Id.* at 66.

115. *See* Baldwin & Clark, *supra* note 8, at 84–85; Langlois, *supra* note 10; Sanchez & Mahoney, *supra* note 113; Schilling & Steensma, *supra* note 83; Karl Ulrich, *The Role of Product Architecture in the Manufacturing Firm*, 24 RES. POL'Y 419, 435 (1995); *cf.* Ethiraj & Levinthal, *supra* note 10, at 172 (noting that firms may not have sufficient internal variety to permit sufficient experimentation). For more cautious appraisals, see Stefano Brusoni, *The Limits to Specialization: Problem Solving and Coordination in "Modular Networks"*, 26 ORG. STUD 1885 (2005); Stefano Brusoni & Andrea Prencipe, *Unpacking the Black Box of Modularity: Technologies, Products and Organizations*, 10 INDUS. & CORP. CHANGE 179 (2001); Glenn Hoetker, *Do Modular Products Lead to Modular Organizations?*, 27 STRATEGIC MGMT. J. 501 (2006).

116. Sanchez & Mahoney, *supra* note 113, at 64, 66–68.

modules separated by predefined interfaces not only allows development of different modules by different firms; it also creates more entry points for new firms, which in turn can promote greater competition.[117]

The interdependency-oriented theory of modularity is closely related to, but analytically distinct from, Ronald Coase's theory of the firm.[118] Coase argued that firms decide whether to perform certain functions internally or to contract them out by comparing the cost of negotiating a contract to outsource the function with the cost of monitoring and overseeing the performance of the function internally.[119] Modularity theory adopts a similar logic, but focuses on the costs of coordination and testing, with highly interdependent tasks being performed within a single module (and thus within a single firm) and highly independent tasks being performed by different modules (and thus potentially by different firms intermediated by a transaction).[120] The key difference is that transaction costs are potentially technologically defeasible and tend to decline over time. The complexities of testing multiple states are more fundamental and unlikely to dissipate with changes in technology.

From this perspective, the relevant primitive units of analysis are not transactions, but rather tasks and their interdependencies.[121] Indeed, it is the underlying task structure that determines the firm's boundaries, rather than considerations such as bounded rationality, uncertainty, vulnerability to opportunism, and information asymmetry that motivate transaction cost economics.[122] Indeed, particularly dense areas of interdependencies can create "transaction-free zones" that are best encapsulated within corporations.[123] This in turn creates what some have called a "modularity theory of the firm" in which "[f]irms arise as islands of non-modularity in a sea of modularity."[124]

Modularity theory thus represents a return to a more technological vision of the theory of the firm, in which the scope of a firm is determined by the interconnections between steps in the production process.[125] This approach responds to the criticism that transaction cost economics

117. Langlois & Robertson, *supra* note 112, at 301.

118. *See* Christopher S. Yoo, *Beyond Coase: Emerging Technologies and Property Theory*, 160 U. PA. L. REV. 2189, 2205 (2012).

119. R.H. Coase, *The Nature of the Firm*, 4 ECONOMICA 386, 392 (1937).

120. Langlois, *supra* note 10, at 32–34.

121. Carliss Y. Baldwin, *Where Do Transactions Come From? Modularity, Transactions, and the Boundaries of Firms*, 17 INDUS. & CORP. CHANGE 155, 156, 162–63 (2007).

122. *Id.* at 163.

123. *Id.* at 181–83.

124. Langlois, *supra* note 10, at 34.

125. Perhaps the prototypical example is vertical integration in the steel industry to take advantage of thermal efficiencies, in which the integration of refining ore into ingot and then recasting ingot into forms obviated the need to reheat the molten metal. *See, e.g.*, F.M. SCHERER & DAVID ROSS, INDUSTRIAL MARKET STRUCTURE AND ECONOMIC PERFORMANCE 94 (3d ed. 1980); George J. Stigler, *The Extent and Bases of Monopoly*, 32 AM. ECON. REV. 1, 22 (1942).

"focuses on the conditions of exchange, to the neglect of the conditions of production."[126]

C. Promoting Flexibility

In addition to making complexity more manageable, decomposing the larger system into subsystems that interact with one another in predefined ways allows designers working on each subsystem to focus exclusively on the aspect of the problem to which they have been assigned and ignore all other aspects. As Simon pointed out, "Decomposing [a complex structure] into semi-independent components corresponding to its many functional parts" permits "[t]he design of each component [to] be carried out with some degree of independence of the design of the others, since each will affect the others largely through its function and independently of the details of the mechanisms that accomplish the function."[127]

The limitation of intermodule interdependencies also facilitates innovation by making it possible to modify individual modules without worrying unduly about the impact of those changes on the other parts of the system. Because tasks related to information hidden inside a module necessarily cannot affect other modules, those tasks can be modified without affecting other modules or having to inform them about those modifications.[128] As long as the interfaces remain stable and the number of interactions remain carefully constrained, software designers can change individual modules with a manageable amount of testing.[129] This should be true with respect both to changes to individual modules and to replacements of entire modules with new modules.[130]

The relative isolation of individual modules makes it easier to incorporate better solutions without having to make changes to the entire system.[131] At the same time, the intramodule flexibility provided by modularity accommodates uncertainty by making it easier to incorporate subsequent improvements into the system.[132] Modularity facilitates experimentation because any disturbances that may arise from any changes can be localized and analyzed within a single module,[133] which buffers the overall system against ripple effects associated with subsequent changes.[134] The greater flexibility allows the network to evolve in response to new technological developments.[135]

126. Michael G. Jacobides & Sidney G. Winter, *The Co-Evolution of Capabilities and Transaction Costs: Explaining the Institutional Structure of Production*, 26 STRATEGIC MGMT. J. 395, 398 (2005).
127. SIMON, *supra* note 44, at 73.
128. BALDWIN & CLARK, *supra* note 5, at 73; Parnas, *supra* note 30, at 1054.
129. Parnas et al., *supra* note 54, at 409.
130. Ethiraj & Levinthal, *supra* note 10, at 164.
131. BALDWIN & CLARK, *supra* note 5, at 91.
132. *Id*.
133. *See id.* at 142; Sanchez & Mahoney, *supra* note 113, at 64–65.
134. Orton & Weick, *supra* note 74, at 214.
135. BALDWIN & CLARK, *supra* note 5, at 216.

D. Facilitating Parallel Experimentation

Modularity not only permits certain technological decisions to be postponed. It also allows experiments with different technical solutions to be conducted simultaneously in parallel.[136] This not only reduces the time needed to complete the design of the complex system.[137] Isolating each module from the effects of other modules means that "its value is no longer hostage to that of distant elements of the system" and that "[a]s a result, each module becomes an independent unit of selection and a point of potential value creation within the overall system."[138]

One of the primary virtues in segmenting tasks in this manner is that the relative independence between modules allows them to be developed simultaneously in parallel.[139] Simultaneous experimentation with various alternative approaches increases the rate of learning by trial and error.[140] Although parallel testing proceeds more rapidly than serial testing, it does forego the benefits of learning between tests, which can increase the number of tests performed. Running multiple concurrent experiments through parallel testing nonetheless remains favored when learning from test to test is minimal and when testing is slow, inexpensive, or imperfect.[141]

The existence of multiple dimensions along which a design can be improved technologically unlocks the value identified by real option theory. When potential technological improvements arise for systems that consist of a single, interconnected design, the architect only has a single decision: whether to adopt the improvement or not.[142]

Modular designs create many more options. At a minimum, they create as many options as there are modules.[143] In addition, the architect can combine improvements and nonimprovements to different modules in different ways. This in turn creates a myriad of possible trajectories for the system.[144]

Consider, for example, a two-module system. Instead of a binary choice between incorporation of the new technology into the interconnected system, modularity raises the possibility of partial adoption, that is, adoption with respect to module A, but not B, or vice versa. In this sense, dividing an interconnected system into two modules increases the number of options from two to four. By extension, a three-module design

136. *Id.* at 91, 238.

137. *See id.* at 91; *accord* Parnas, *supra* note 30, at 1054 (noting that under modularity, "development time should be shortened because separate groups would work on each module with little need for communication"); *see also* Simon, *supra* note 16, at 473 (observing how the existence of stable intermediate forms can cause complex systems to evolve more rapidly).

138. BALDWIN & CLARK, *supra* note 5, at 237.

139. *See id.* at 91; Ulrich, *supra* note 115, at 435; *see also* Ethiraj & Levinthal, *supra* note 10, at 160 (noting that "modularity offers the advantage of parallelism").

140. Langlois & Robertson, *supra* note 112, at 301.

141. Loch et al., *supra* note 101, at 664, 674.

142. BALDWIN & CLARK, *supra* note 5, at 236, 238, 252.

143. *Id.* at 236.

144. *Id.* at 238.

yields eight options. In other words, as the number of modules (*n*) increases linearly, the number of options increases exponentially (2^n). This approach assumes that the modularizations are symmetric, but the analysis generalizes to more complex assumptions.

FIGURE 3

Adopt for system	yes	1
	no	2

		Adopt for module *B*	
		yes	no
Adopt for module *A*	yes	1	3
	no	2	4

		Adopt for Module *C*			
		yes		no	
		Adopt for module *B*		Adopt for module *B*	
		yes	no	yes	no
Adopt for module *A*	yes	1	3	5	7
	no	2	4	6	8

Modularity not only increases the number of options; it increases their value. Note first that if an architect had to accept both positive and negative values, the expected value of any experiment would be zero. But an architect can simply fail to incorporate the value of an experiment into the design. Thus, all positive experiments have a positive impact on value, while all negative experiments have no effect at all. The net result of cutting off the left-hand tail of the distribution means that the average experiment would have positive value. A simple calculation of the area under the positive half of the curve indicates that the expected value is 0.3989 times the standard deviation, σ.[145]

Baldwin and Clark's model also assumes that the variance increases as the system becomes more complex, with variance increasing linearly with the number of modules *n* (i.e., variance $\sigma_n^2 = \sigma^2 n$). Because the standard deviation is by definition the square root of the variance, the standard deviation equals $\sigma\sqrt{n}$.[146] This has the effect of widening the distribution of possible outcomes, increasing both the number of high value and low value possibilities. In the case of the IBM System/360, dividing the system into twenty-five modules increased the value of the system by five times.

The value is increased still further if multiple experiments are run with respect to each of the modules. In effect, each of these experiments is a random draw on the same distribution of expected value for each module, with only the one with the highest value being incorporated into the design. The existence of these multiple draws increases the likelihood that the best experiment will be significantly higher than the expected result.

145. *Id.* at 256.
146. *See id.* at 255.

In essence, enabling multiple experiments taps into Robert Merton's key insight into the value of options, which is that a portfolio of options is more valuable than an option on a portfolio.[147] Indeed, experimentation increased the value of the IBM System/360 by an additional five times, which means that combining modularity with experimentation increased the value of the IBM System/360 by twenty-five times.[148] Whether this would be profitable depends on whether the increase in value exceeds the fixed costs of modularization and the cost of experimentation.

E. Enabling Permissionless Innovation

Modularity has the additional advantage of making innovation permissionless.[149] Encapsulating certain parameters within modules permits innovators to innovate with respect to those hidden parameters without disturbing the rest of the system.[150] Control is maintained by the design rules embedded in the architecture rather than by any firm or actor. Any person can connect as long as they comply with the design rules.[151] In short, the architecture is set up so that hidden-module designers—designers of components that are completely encapsulated within a module—"do not have to have detailed knowledge about the whole system under development. Hidden-module designers have only to master the design rules."[152]

This permits module designers to act independently of the system architects as well as one another.[153] This in turn enables decentralized decisionmaking without causing the system to lose coordination.[154] Modularity also means that those who wish to experiment at the module level do not need to obtain permission from the system designers before doing so.[155] The would-be module experimenters need only make sure that their module revisions comply with the design rules of the architecture.[156] The result is to disperse and decentralize the process of innovation. It also has the effect of moving the options from the center of the system of the modules.[157] This in turn shifts the value gained from those experiments from the center of the system to the modules.[158]

147. *Id.* at 259.
148. *Id.* at 207–211.
149. *Id.* at 14, 223, 252, 264, 268, 336–37, 347–48.
150. *Id.* at 348.
151. *Id.* at 268, 306.
152. *Id.* at 348.
153. *Id.* at 348.
154. *Id.* at 268.
155. *Id.* at 14 ("No architect had to give permission for these changes to take place; the possibilities were inherent in the modularity of the design itself."); *accord id.* at 223, 268.
156. *Id.* at 268, 306.
157. *Id.* at 268.
158. *Id.*

IV. MODULARITY'S LATENT COSTS

Although modularity yields substantial benefits, it should not be regarded as a panacea. On the contrary, an early IETF document authored by MIT computer scientist and then-chief protocol architect David Clark observed that network engineers are often unpleasantly surprised to find that "modularity is one of the chief villains in attempting to obtain good performance."[159] This statement underscores the importance of understanding the implicit tradeoffs underlying every modularity decision.

A. The Sunk Costs of Modularization

The production of a modular system takes place in three distinct phases. During the first stage, the architects formulate the design rules that determine the scope of the modules and what information to make visible in the interfaces through which each module will interact with the others. During the second stage, smaller groups engage in independent parallel activity to develop the modules. During the third stage, the various modules are tested and integrated into a single system.[160]

One of the most straightforward costs of modularity results from the need to construct and disseminate a modular design up front. Such efforts can be extremely time consuming and demanding.[161] The presence of the initial design phase makes "modular systems . . . much more difficult to design than comparable interconnected systems."[162] The architects must know a great deal about the relevant products and processes and the interdependencies in the relevant tasks and must incorporate their insights into design rules specified long in advance.[163] The cost of developing these design principles represents a sunk-cost investment that must be recouped.[164] Moreover, the inevitable uncertainty about the design necessarily means that some chance always remains that the architecture is badly done, in which case it will underperform.[165]

B. Restrictions on Cost Minimization

Beyond the sunk costs of creating the architecture, modularity embodies a more fundamental tradeoff. The generality of modular systems makes them flexible, but increases their costs. Consider, for example, the problem of packing as many objects into a closet as possible.[166] The most

159. David D. Clark, *Modularity and Efficiency in Protocol Implementation* 1 (Request for Comments 817, July 1982) [hereinafter RFC 817], *available at* http://tools.ietf.org/html/rfc817.

160. BALDWIN & CLARK, *supra* note 5, at 72.

161. *Id.* at 86, 247; Baldwin & Clark, *supra* note 30, at 199; Langlois, *supra* note 10, at 23.

162. Baldwin & Clark, *supra* note 8, at 86.

163. *Id.*

164. BALDWIN & CLARK, *supra* note 5, at 247; *accord* Langlois, *supra* note 10, at 23 (noting that despite modularity's benefits, "there is no free lunch" in that "[a] well-decomposed modular system must pay a kind of fixed cost that an integrated system need not pay").

165. BALDWIN & CLARK, *supra* note 5, at 254, 257; Ethiraj & Levinthal, *supra* note 10, at 172.

166. The example is taken from Blanchette, *supra* note 11, at 1047.

efficient way to do so would be to place each object in the closet one at a time, carefully taking into account the idiosyncrasies of each object's shape. The problem is that the interdependencies between the objects mean that any attempt to retrieve or shift the objects after they have been packed in will threaten to perturb the entire arrangement. A classic solution to this problem would be to pack all of the objects into storage bins before placing them in the closet. The bins would make the objects easier to shift, but would store the items less efficiently on account of the empty space in each of the bins and any gaps left between the bins when packing them into the closet.

The same cost/flexibility tradeoff arises with respect to modularity.[167] As Baldwin and Clark note, the benefits of modularity must be traded off against the fact that "designers will lose the ability to explore some parts of the space of designs—in effect, the architects will restrict the search, declaring some parts of the design space to be out of bounds."[168] The problem, long recognized by computer scientists, is that generality exacts a cost. For example, McGee's seminal paper on generalization is based on the premise that handtailored programs, while expensive to create and modify, are usually more cost effective (which computer scientists describe using the word, "efficient").[169] Philip Agre notes that "all abstractions . . . entail[] a decrease in efficiency," providing one example "located at the extreme in the space of trade-offs between freedom of abstraction and efficiency of implementation" which "offer[ed] a maximum of freedom with a minimum of efficiency" as well as another example that presented "the trade-off between efficiency of implementation and freedom of abstraction in its most radical form."[170] Another commentator has similarly noted that "the more a language's constructs abstract away from the underlying physical machine, the less efficient the resulting code tends to be."[171]

The tradeoff between flexibility and generality provides insights into when the benefits of modularity exceed the costs. As noted earlier, modularity yields the most benefits when demand is heterogeneous, technology is changing rapidly, and competition is intense.[172] These considerations affect not just whether to modularize, but also how to modularize. For example, increasing the number of modules increases testing expense, as tiny modules can cause a combinatorial explosion of testing permutations.[173] Sendil Ethiraj and Daniel Levinthal develop an interesting model integrating these considerations, which trades off the destabilizing effects from having too many modules against the slower rate of

167. *See* KARL T. ULRICH & STEVEN D. EPPINGER, PRODUCT DESIGN AND DEVELOPMENT 202 (5th ed. 2011); Blanchette, *supra* note 11, at 1048.
168. BALDWIN & CLARK, *supra* note 5, at 68–69.
169. McGee, *supra* note 44, at 2.
170. PHILIP AGRE, COMPUTATION AND HUMAN EXPERIENCE 68, 73, 76 (1997).
171. Blanchette, *supra* note 11, at 1047.
172. *See supra* Part II.D.
173. *See* BALDWIN & CLARK, *supra* note 5, at 272, 275; Loch et al., *supra* note 101, at 673.

innovation from having too few modules. They show that creating more modules unlocks parallelism and avoids premature fixation on inferior designs, but leads to more intermodule interdependencies and requires more testing.[174] Although they tested multiple scenarios,[175] their general conclusion is that having too many modules poses greater risks than having too few.[176]

C. Restriction of Certain Types of Innovation

Decisions about how to implement modularity may affect applications differently.[177] To use an example from software design, the process for releasing memory that is no longer needed can require all programs to stop temporarily, a burden that will rest particularly heavily on real-time applications.[178]

Decisions about what information to hide and what information to make visible can also have a differential impact on applications. The set of visible information necessarily has a direct impact on the functionality of the system. Because it naturally limits the type of information that can pass between modules, any functionality that depends on any other information must be handled within one particular module. Consequently, modularization represents a precommitment to the idea that certain functions must be performed by certain modules. At the same time, it reflects a judgment of the type of functions that require coordination across modules.

The advent of USB ports to connect printers to personal computers provides a useful example. The interface between these two devices is designed to include the minimum amount of information that must pass between these two modules. Many aspects about printers are extremely dependent on whether laser printing, inkjet printing, or some other technology is used, the volume of the printer, and other key design decisions. Information hiding allows these interdependencies to be encapsulated within the printer so that the personal computers connected to those printers do not need to know anything about how any particular printer solves those functions. As long as the personal computer presents its data in accordance with the appropriate format, the printer should function properly. In addition, as long as printers are prepared to process the visible information presented by personal computers through the interface, they remain free to redesign any technical aspects that solely involve the hidden information. Note, however, that in limiting the information that

174. Ethiraj & Levinthal, *supra* note 10, at 160.

175. Ethiraj and Levinthal considered two types of innovation: internal (which they call local search) and borrowing modules from other companies (which they call recombination). They also compared the results when one assumed that the architects have the design perfect with the results when they do not. *Id.* at 160–61, 164–65.

176. *Id.* at 170–72.

177. *See* Blanchette, *supra* note 11, at 1047 (noting that "the trade-offs implied by modularity will not affect all applications equally, or even the same application under all circumstances").

178. *Id.*

can pass between personal computers and printers, the design of the interface inevitably imposes limits on the ways these two devices can interact.

Thus, although flexibility is generally regarded as one of the advantages of modular systems, closer inspection reveals that modular structures facilitate only certain types of innovation while impeding others. Specifically, modular systems are very good at promoting improvements and replacements of individual modules, which require little coordination with other modules. They are less accommodating to systemic innovations that reorganize the ways that modules interact with one another.[179]

The literature on *design hierarchies* suggests a similar conclusion.[180] Design hierarchies exist when innovations are not standalone technological developments, but rather are part of a web of interdependent technological processes. Not all of the components of a design hierarchy are equally important. Some have a higher degree of connection to components than others.[181]

Commentators have recognized that design hierarchies represent something of a mixed blessing from the standpoint of innovation. On the one hand, they facilitate innovations that are consistent with the hierarchy. On the other hand, they discourage innovations that are inconsistent with the hierarchy.[182] Thus, any innovation that restructures the way a design hierarchy operates must overcome the challenge of coordinating the behavior of multiple autonomous actors, each of which is pursuing its own agenda.[183] This is particularly true for components of the design hierarchy that are particularly tightly connected with other components.[184] Because changes to core components necessarily require extensive changes to peripheral components, even changes that ultimately prove successful in the long run are likely to be accompanied in the short run by poorer system performance and widespread institutional resistance.[185]

179. Langlois, *supra* note 10, at 25–26 (distinguishing between modular innovation, which takes place within modules, and architectural innovation, which changes the relationship between modules, and observing how an architecture can get stuck in an inferior modularization); Langlois & Robertson, *supra* note 140, at 302 (citing David J. Teece, *Profiting from Technological Innovation: Implications for Integration, Collaboration, Licensing and Public Policy*, 15 RES. POL'Y 285, 288 (1986)).

180. For an earlier discussion, see Christopher S. Yoo, *Product Life Cycle Theory and the Maturation of the Internet*, 104 NW. U. L. REV. 641, 655–56 (2010).

181. Clark, *supra* note 88, at 243; Johann Peter Murmann & Koen Frenken, *Toward a Systematic Framework for Research on Dominant Designs, Technological Innovations, and Industrial Change*, 35 RES. POL'Y 925, 940–42 (2006); Michael L. Tushman & Johann Peter Murmann, *Dominant Designs, Technology Cycles, and Organizational Outcomes*, 20 RES. ORG. BEHAV. 231, 249–51 (1998); Michael L. Tushman & Lori Rosenkopf, *Organizational Determinants of Technological Change: Toward a Sociology of Technological Evolution*, 14 RES. ORG. BEHAV. 311, 334 (1992).

182. Clark, *supra* note 88, at 249.

183. Henry W. Chesbrough & David J. Teece, *When Is Virtual Virtuous? Organizing for Innovation*, 74 HARV. BUS. REV., Jan.–Feb. 1996, at 65, 67–68; Jacobides & Winter, *supra* note 126, at 404.

184. Murmann & Frenken, *supra* note 181, at 942–43.

185. *Id.* at 942.

Empirical studies have confirmed that changes to core components are harder to displace.[186]

Characterizing innovation as residing solely on the promodularity side of the tradeoff, as sometimes occurs in the literature, adopts an unnecessarily narrow view.[187] In terms of innovation, modularity is more properly regarded as something of a two-edged sword that simultaneously promotes innovations that are consistent with the architecture and obstructs innovations that require a different configuration of tasks.

D. Limitations on Alternative Institutional Forms

As discussed above, some scholars believe that "products design organizations"[188] and that modular product designs necessarily lead to modular organizational designs.[189] To the extent this is true, the fact that technological interdependencies determine the natural boundaries between modules prevents the segregation of certain tasks into different firms. The technological determinants of modularity thus restricts the extent to which designers can select module boundaries to promote competition and other economic considerations, as some have suggested.[190]

In addition, the decentralized, mix-and-match world of modularity exists in considerable tension with the insights of the New Institutional Economics.

Consider the landmark article by David Teece exploring why the entrepreneurs who come up with innovative ideas often do not end up being the person who benefits the most from them.[191] Teece pointed out that most innovations are not products by themselves. Instead, they must be combined with other complementary assets before they can form a marketable product.[192] When that is the case, economic success depends as much on the ability to bargain effectively with owners of complementary assets as it does on having ironclad patent protection over the innovation.[193]

If the owners of the complementary assets are in a stronger bargaining position, innovators may want to receive ex ante assurances before undertaking any irreversible investments that would be rendered value-

186. *E.g.*, Alan MacCormack et al., *The Impact of Component Modularity on Design Evolution: Evidence from the Software Industry* (Harv. Bus. Sch., Working Paper No. 08-038, 2007), *available at* http://www.hbs.edu/research/pdf/08-038.pdf.

187. *See, e.g.*, JOHN PALFREY & URS GASSER, INTEROP: THE PROMISE AND PERILS OF HIGHLY INTERCONNECTED SYSTEMS 75–88 (2012) (describing interoperability as promoting innovation while having negative effects on privacy and security); ZITTRAIN, *supra* note 3, at 40–43 (characterizing modularity as promoting generativity, but being vulnerable to security risks).

188. Sanchez & Mahoney, *supra* note 113, at 64.

189. *See supra* note 115 and accompanying text.

190. *See, e.g.*, David D. Clark et al., *Tussle in Cyberspace: Defining Tomorrow's Internet*, 13 IEEE/ACM TRANSACTIONS ON NETWORKING 462, 468–70 (2005).

191. David J. Teece, *Profiting from Technological Innovation: Implications for Integration, Collaboration, Licensing and Public Policy*, 15 RES. POL'Y 285, 285 (1986).

192. *Id.* at 288.

193. *Id.* at 291–92.

less if they are unable to reach agreement with the owner of the complementary asset.[194] Under such circumstances, the classic solution is for the innovator and the owner to eliminate the risk that the other side may act opportunistically to claim a greater proportion of the surplus, either by merging with one another or by entering into a long-term contract.[195] Although such an arrangement would deviate from the mix-and-match world associated with modularity, it would promote entry by reducing the risk faced by innovating firms.[196]

A similar insight emerges from the seminal article by Timothy Bresnahan and Manuel Trajtenberg on GPTs.[197] GPTs create positive vertical externalities for applications sectors that build products based on the GPT.[198] The GPT creator, however, will invest based on a comparison of its private cost of the investment and its private return. The positive externality causes the GPT creator's private benefit to understate the social benefit of further investments, which results in too little investment in GPTs. At the same time, allowing GPT creators to appropriate more of the surplus risks causing too little investment in the applications sector.[199] The public goods aspect of GPTs thus creates a horizontal externality, as different applications-sector players attempt to induce their peers to bear more of the cost of financing improvements in the GPT. The obvious solution to both problems is to permit GPT creators and applications sector developers to merge or enter into some type of cooperative agreement that would allow the GPT creator to realize more the benefits of its investment.[200] Preventing GPTs from coordinating with applications will result in "'too little, too late' innovation" in both sectors.[201]

Both of these lines of research suggest permitting firms in modular spaces to experiment with alternative institutional forms could yield significant benefits. Imposition of a modular structure may prevent these benefits from being realized.

E. Lack of Coordination and Systemic Drift

The decentralized nature of decisionmaking in modular industries can lead to welfare loss. As noted earlier, for distributed production of modules to succeed, each module needs to be able to verify that the other modules are performing as expected. Indeed, Baldwin and Clark regard such testing standards to be an integral part of any modular design.[202] In the context of the Internet, the necessary verification tools

194. *Id.* at 294.
195. *See id.* at 290, 293–95.
196. *See id.* at 302.
197. Timothy F. Bresnahan & M. Trajtenberg, *General Purpose Technologies: "Engines of Growth?"*, 65 J. ECONOMETRICS 83, 85 (1995).
198. *Id.* at 94.
199. *Id.*
200. *Id.* at 96, 99.
201. *Id.* at 103.
202. *See supra* Part I.C.

simply do not exist. As discussed below, the visible information in the interfaces does not provide enough information for actors to verify whether the other modules are behaving as expected. The absence of any way to verify the conduct of the other modules effectively leaves actors in the position of having to rely on the honor system or some form of informal sanctions.[203]

Furthermore, local optimizing behavior may not maximize the performance of the overall system. Complex systems have the general property that "[i]f each subsystem, regarded separately, is made to operate with maximum efficiency, the system as a whole will not operate with utmost efficiency."[204] Consequently, because modularity focuses each module (and the owner of each module) on optimizing its individual interests, the net result is unlikely to optimize the performance of the system as a whole.[205]

Moreover, given that all companies are drawing off a common pool of surplus that they jointly create, they have the incentive to want others to do more.[206] As was the case with GPTs, this can lead to suboptimal investment in the entire modular ecosystem, as each actor hopes that the others will bear a greater proportion of the cost.[207]

One solution would be to have some actor organize all of the activities of a modular system around the optimal outcome. The problem is that in a decentralized system, individual actors have no incentive to bear the costs of promoting what is best for the system as a whole.[208] Indeed, many would regard centralized control as improper, and any actors who would be disadvantaged by such intervention would vigorously oppose it.

The result is that many modular systems will suffer from a lack of coordination. Without such leadership, modular systems often undergo long periods of systemic drift.

* * *

The foregoing analysis makes clear that modularity is not always the optimal architecture. Whether modularity is optimal is contingent on a number of factors. Moreover, any modular scheme necessarily requires the system to give up a degree of functionality. Modularity can also inter-

203. Charles L. Jackson, *Wireless Efficiency Versus Net Neutrality*, 63 FED. COMM. L.J. 445, 448–51 (2011).

204. LARS SKYTTNER, GENERAL SYSTEMS THEORY: IDEAS AND APPLICATIONS 93 (2001).

205. *See* Chesbrough & Teece, *supra* note 183, at 66; Jon Crowcroft et al., *Is Layering Harmful?*, 6 IEEE NETWORK MAG., 20, 23–24 (1992); John D. Day & Hubert Zimmerman, *The OSI Reference Model*, 71 PROC. IEEE 1334, 1335–36 (1983); Randy Bush & David Meyer, *Some Internet Architectural Guidelines and Philosophy* 7–8 (Network Working Group Request for Comments 3439, Dec. 2002) [hereinafter RFC 3439], *available at* http://tools.ietf.org/html/rfc3439.

206. Chesbrough & Teece, *supra* note 183, at 67–69.

207. *See supra* notes 197–200 and accompanying text.

208. *See* Robert Braden et al., Developing a Next-Generation Internet Architecture 16 (Univ. of S. Cal. Info. Sci. Inst. White Paper, 2000) (introducing the NewArch project funded by the Defense Advanced Research Projects Agency), *available at* http://www.isi.edu/newarch/DOCUMENTS/WhitePaper.pdf ("There is no commercial provider who believes that they [sic] hold the responsibility for the Internet architecture.").

fere with alternative institutional arrangements that firms can use to address unequal bargaining power or to internalize positive externalities.

The foregoing analysis also underscores that modularity is not a static construct. Indeed, modularity theorists expect the design to change as the architects gain a better understanding of the underlying interdependencies.[209] Technological and economic changes can create pressure for high-tech industries to evolve toward a fundamentally different architecture.[210] Examples include the desktop PC's absorption of functions that used to be provided by standalone peripheral devices—such as hard disks, modems, and WiFi cards[211]—and the advent of last-mile broadband networks—such as DSL and cable modem systems—both of which undercut the rationale for a standalone regional ISP.[212] Changes in technology and end-user demand for network services, however, can cause the nature and relative importance of particular interdependencies to change over time, which in turn creates pressure on the current modular architecture to change. Any such change not only involves transition costs,[213] but also any change in the location of transactions inherent in any such remodularization inevitably alters the structure of industries.[214] In fact, the difficulty in analyzing how modularity changes in industries in the midst of such a period of ferment is what led Baldwin and Clark to divide their work into two volumes.[215]

Attempts to evolve away from an existing modular architecture may thus represent nothing more than the natural response to changes in the economic and technological environment.[216] Indeed, experimentation in

209. von Hippel, *supra* note 29, at 408, 412; Langlois, *supra* note 10, at 25.

210. *See* Baldwin, *supra* note 121, at 180 (noting that "there is no process of technological determinism at work driving the task network toward ever-higher levels of modularity" and that changes in strategies, knowledge, and technologies can cause "task networks to become more integral (i.e., less modular) over time"); Jacobides & Winter, *supra* note 126, at 405 (noting how the emergence of new productive structures and new knowledge bases can cause the pattern of increasing specialization and vertical disintegration to reverse).

211. *See* Transamerica Computer Co. v. IBM Corp., 698 F.2d 1377, 1382–83 (9th Cir. 1983) (discussing the increasing integration between the CPU and certain peripherals); Cal. Computer Prods., Inc. v. IBM Corp., 613 F.2d 727, 743–44 (9th Cir. 1979) (discussing how the IBM new computer integrated the disk drive control function with the CPU); ILC Peripherals Leasing Corp. v. IBM Corp., 448 F. Supp. 228, 231–32 (N.D. Cal. 1978) (describing how IBM abandoned removable disk drives in favor of non-removable drives with the head-disk assembly integrated into the computer itself that provided greater storage capacity), *aff'd sub nom.* Memorex Corp. v. IBM Corp., 636 F.2d 1188 (9th Cir. 1980); Telex Corp. v. IBM Corp., 367 F. Supp. 258, 341–342 (N.D. Okla. 1973) (describing how IBM integrated its memories and control units into its CPUs), *rev'd on other grounds*, 510 F.2d 894 (10th Cir. 1975).

212. *See* Christopher S. Yoo, *Would Mandating Network Neutrality Help or Hurt Broadband Competition? A Comment on the End-to-End Debate*, 3 J. ON TELECOMM. & HIGH TECH. L. 23, 33–34 (2004).

213. *See* Langlois, *supra* note 10, at 25–26.

214. Baldwin, *supra* note 121, at 187.

215. CARLISS Y. BALDWIN & KIM B. CLARK, ROADMAP FOR DESIGN RULES 8 (2006), *available at* http://www.people.hbs.edu/cbaldwin/DR2/BaldwinDRRoadmap.pdf.

216. *See* Langlois, *supra* note 10, at 25–26.

new standards and competition between standards are properly regarded as a sign of innovative health.[217]

V. MODULARITY THEORY AND ARCHITECTURAL CHANGE

The foregoing catalog of the pluses and minuses of modularity provide a basis for assessing the value of any particular modular system at any particular time. That said, this analytical framework views modular architectures as static and fails to explain how modular architectures emerge and, once having done so, when and how they evolve. The fact that the shape of any modular architecture is determined by technological interdependencies on the supply side and by the heterogeneity of demand on the demand side indicates that the optimal architecture should change over time.

The inevitability of architectural change underscores the need for a theory of remodularization[218] as well as heuristics to help recognize when such change is necessary.[219] Fortunately, Baldwin and Clark offer a theory built around six core modular operators that represent the entire set of possible transformations and which provide significant insights into how and when modular architectures can and should change.

A. Modular Operators

Baldwin and Clark identified six operators that can transform a modular architecture: splitting, substitution, augmentation, exclusion, inversion, and porting.[220] The first two (splitting and substitution) can occur in nonmodular systems, while the other four can occur only in modular systems.[221] Moreover, five of the operators represent relatively minor changes to the architecture. Only inversion requires a fundamental reordering of the relevant tasks.

1. Splitting and Substitution

Consider *splitting*, which is the division of one module into two.[222] As noted earlier, splitting is not unique to modular architectures; indeed, it is how an interconnected architecture (which essentially consists of one module) becomes modular in the first place.[223] These modules can in turn

217. *See* Shane Greenstein, *Glimmers and Signs of Innovative Health in the Commercial Internet*, 8 J. ON TELECOMM. & HIGH TECH. L. 25, 42–55 (2010) (discussing how economic experimentation and vigorous standards competition are signs of innovative health).

218. Ethiraj & Levinthal, *supra* note 10, at 172 (noting that "we need to model the evolution of modularity itself").

219. Stefano Brusoni & Andrea Prencipe, A Dialectical Model of Organizational Loose Coupling: Modularity, Systems Integration, and Innovation 5 (June 27, 2005) (unpublished manuscript), *available at* http://www.druid.dk/uploads/tx_picturedb/ds2005-1543.pdf.

220. BALDWIN & CLARK, *supra* note 5, at 123, 132–42, 228.

221. *Id.* at 123, 136.

222. *Id.* at 123.

223. *Id.*

split again. *Substitution* is the replacement of one module design for another.[224] Baldwin and Clark see splitting and substitution as part of a cycle, which begins with splitting into two modules, followed by substitution of new technologies for particular modules, followed again by splitting as the benefits from further substitution begin to abate.[225]

Splitting, however, imposes only a relatively small change to a modular architecture. Because it by definition only affects the internal design structure of what had originally been one module in the system, it does not disturb the relationship with other modules or any of the information visible to other modules.

Substitution similarly does not disrupt the basic architecture at all. Indeed, the whole point of modularity is to swap out existing modules without affecting the internal workings of the other modules. Although substitution does improve the system, because it operates on only the hidden information, it does so in a way that does not disturb the basic architecture of the overall system.

2. *Exclusion and Augmentation*

Two other operators, exclusion and augmentation, together form part of a strategy that makes sense when users do not need all of the modules that are available or when designing the full set of possible modules is too costly at the time of launch.[226] When that is the case, the designer can reduce development costs by designing a narrow architecture (exclusion), and then expanding it after it is validated by customers (augmentation).[227] This permits the designer to conserve resources by deferring investments in additional functionality until the system has been validated by the marketplace. If the system is successful, the designers can commit additional resources to developing other parts of the system with greater impunity.[228]

The designers of the initial, minimal system must design it with future augmentation in mind. The key is design rules with clean interfaces that provide a large "shadow set" of potential new modules that can be created through augmentation.[229] If so, those who wish to augment the architecture need only understand how to fit the new module within the shadow design rules.[230]

Designing an architecture to support exclusion and augmentation in effect requires the virtual design of a more complete architecture that organizes the relationship between current and future models. The difference is primarily one of timing. In the case of complete architecture, all

224. *Id.* at 123, 281.
225. *Id.* at 281, 323.
226. *Id.* at 136, 308, 310.
227. *Id.* at 308.
228. *Id.* at 136–37, 313–14.
229. *Id.* at 321.
230. *Id.* at 306.

of the costs are committed up front. In the case of an architecture narrowed by exclusion in anticipation of future expansion through augmentation, many of these costs can be postponed. In this sense, the process of creating a complete architecture and a minimal architecture susceptible to later augmentation both require conceptualizing and enforcing a coherent architecture up front. If the initial minimal architecture is designed properly, subsequent augmentation does not cause any disruption to the design. In fact, the possibility of augmentation opens the door to coinvestment by other firms.[231] All these third parties need to do is look to the same set of visible information envisioned by the architects.[232]

3. Porting and Inversion

The final two modular operators effect more substantial architectural changes. Porting occurs when a module is made available to another modular system, such as when a printer made for one operating system is adapted to work with another. The process requires creating a shell around the module with a translator capable of conforming to the design rules of both systems.[233] If done properly, porting has no impact on the old architecture. Indeed, architects and designers of modules of the other system may not know that porting has taken place.[234]

The most interesting modular operator for purposes of this Article is inversion. Inversion "tak[es] previously hidden information and mov[es] it up the design hierarchy so that it is visible to a group of modules."[235] For example, printing subroutines used to be buried inside programs, with each program including its own unique implementation tailored to its program that was part of that program's hidden information and thus locked inside it. Over time, the fact that many programs use printing made it logical to move printing subroutines to a higher level of the design hierarchy to a place where it is visible to all other programs.[236] "[T]hrough inversion, what was hidden becomes visible."[237]

What began as a module inside another module is treated as a common solution and imposed on other programs.[238] It occurs when the value of the new architecture exceeds the sum of the cost of designing the new architecture, the cost of redesigning the hidden modules that previously relied on the new inverted module that used to be locked inside a

231. *Id.* at 321–22.
232. *Id.* at 322.
233. *Id.* at 140.
234. *Id.* at 140. Porting can cause the shell to become a new source of visible information in its own right apart from the system. *Id.* at 141. As such, it can create a new apex within the design hierarchy. *Id.* at 339.
235. *Id.* at 138 (internal quotation marks omitted).
236. *Id.*
237. *Id.* at 139.
238. *Id.* at 326.

module, and the lost net option value that would have accrued if experiments under the old architecture had been permitted to continue.[239]

By its nature, inversion represents a significant change that requires fundamental changes in the task structure.[240] As such, they are often highly controversial.[241] They take away designers' prerogatives over previously hidden information that was previously locked within a module, and thus subject to experimentation, and freezes it into the visible information that is part of the new design.[242] Indeed, inversion terminates all further experiments with respect to the previously hidden information that inversion makes visible.[243] Moreover, they require that adjacent modules now interact with a different set of visible information. They also require designers to adapt to a new set of rules and to develop a new set of skills.[244]

That said, inversions occupy a natural place in the life cycle of a modular regime. The creation of a new modular architecture will prompt a rash of experimentation. The returns yielded by these experiments will diminish over time. Eventually, the benefits from further experimentation no longer justify the cost, at which point pressure will mount for a new set of design rules through inversion.[245]

B. Obstacles to Significant Architectural Change

The conventional wisdom is that modern technologies are undergoing constant and rapid change, sometimes reflected in the term "Internet time."[246] Closer inspection reveals the phrase only to be half true. While modularity permits rapid innovation with respect to individual modules, changes in the architecture that structure the interactions among those modules occur glacially slowly.[247] Indeed, this stability is critical to the success of modular systems, since it provides those who are testing modular innovations with the fixed reference points around which to base their products as well as the expectation of being able to capitalize on any improvements that prove successful. And architectures that have been established are notoriously hard to change.[248]

This means that modular architectures are effective at promoting innovations that are consistent with the existing stack, such as substitution and to a lesser extent splitting, exclusion, and augmentation. At the

239. *Id.* at 327–28.
240. *Id.* at 127.
241. *Id.* at 327.
242. *Id.*
243. *Id.* at 326.
244. *Id.* at 327.
245. *Id.* at 323.
246. *See, e.g.*, MICHAEL A. CUSUMANO & DAVID B. YOFFIE, COMPETING ON INTERNET TIME: LESSONS FROM NETSCAPE AND ITS BATTLE WITH MICROSOFT (1998).
247. Blanchette, *supra* note 11, at 1054.
248. BALDWIN & CLARK, *supra* note 5, at 89.

same time, they tend to hinder innovations that require a reorganization of the tasks performed by the stack through an operator such as inversion.

One part of this slow rate of architectural change stems from coordination problems. The literature on network economic effects has long recognized how the presence of an installed base can delay the adoption of a new technology even when that technology is superior.[249] The problems are likely to be more severe in a modular architecture, since changes to the architecture would require the coordination with the modules both above and below the modules being inverted.[250] As a result, some scholars have raised the concern that modular architectures may be unduly resistant to this type of change.[251] This resistance may lead firms to abandon modular architectures in favor of more vertically integrated market structures.[252]

The literature on innovation has long recognized how each architecture creates its own technological paradigm that identifies the problems worth solving and the solutions that are the most promising.[253] As noted earlier, one key aspect of modularity is that it allows organizations to focus their attention on individual modules and to disregard the system as a whole.[254] Ironically, the same quality that reduces complexity makes organizations less attentive to potential architectural changes. In addition, a modular architecture establishes a technical agenda for a product's development that directs research along innovation avenues consistent with the hierarchy.[255] Each modular architecture thus creates its own evolutionary trajectory.[256] These paradigms, moreover, become ingrained in the communication channels and information filters that organizations use to manage information, which tends to further reinforce the status quo.[257] Admittedly, these problems are worse for nonmodular systems,[258] but they exist for modular systems as well.

These considerations suggest that casting modularity as a tradeoff between long-term evolvability and short-run inefficiency paints an in-

249. *See, e.g.*, Joseph Farrell & Garth Saloner, *Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation*, 76 AM. ECON. REV. 940, 942 (1986).

250. Blanchette, *supra* note 11, at 1054.

251. Chesbrough & Teece, *supra* note 183, at 66, 68; Jacobides & Winter, *supra* note 126, at 404; Langlois, *supra* note 10, at 26.

252. Langlois & Robertson, *supra* note 140, at 302.

253. Yoo, *supra* note 180, at 656.

254. *See supra* Part III.C.

255. *See* Giovanni Dosi, *Technological Paradigms and Technological Trajectories*, 11 RES. POL'Y 147, 152 (1982); *see also* Devendra Sahal, *Technological Guideposts and Innovation Avenues*, 14 RES. POL'Y 61, 71, 78–79 (1985) (describing how "specific information avenues" lead to particular products).

256. BALDWIN & CLARK, *supra* note 5, at 328.

257. Philip Anderson & Michael L. Tushman, *Technological Discontinuities and Dominant Designs: A Cyclical Model of Technological Change*, 35 ADMIN. SCI. Q. 604, 618 (1990); Stefano Brusoni & Andrea Prencipe, *Patterns of Modularization: The Dynamics of Product Architecture in Complex Systems*, 8 EUR. MGMT. REV. 67, 67–68 (2011); Henderson & Clark, *supra* note 86, at 15–19.

258. *See* BALDWIN & CLARK, *supra* note 5, at 57–59.

complete picture.[259] Instead, the differences between modular and architectural innovation suggest that there are evolvability considerations on both sides of the equation.

Moreover, the nature of the two types of innovation is also likely to be different. While innovation with respect to individual modules is likely to be very rapid, it is also likely to be incremental and consider only possibilities permitted by the existing architecture. The best it can achieve is to reflect the local maximum permitted by the existing architecture.[260]

Architectural innovation has the opposite characteristics: while very slow, the freedom to consider all possibilities, including those inconsistent with the current architecture, makes it more likely to make a sharply discontinuous jump to a completely different local optimum.[261] If the value of this other optimum is sufficiently large, architectural innovation can yield benefits that more than compensate for the speed advantage enjoyed by modular innovation.[262] Indeed, in contrast to the conventional wisdom, which holds that rapid technological change favors modularity, the greater potential of nonmodular architectures to take advantage of major technological developments suggests volatility actually favors the latter.[263]

The modularity literature also helps provide intuitions about whether change is likely to occur too quickly or too slowly. Several considerations suggest the latter. For example, the sunk costs associated with remodularization create the risk of path dependence that causes the system to become locked into an inferior modularization.[264] That said, as noted earlier, the value of modularity depends on having a significant degree of stability. Moreover, architectural change is very expensive.[265] Circumstances thus may exist when the magnitude of the benefits do not justify the cost.[266]

To say that the architecture should change only rarely is not to say that it should never do so. The fact that the optimal modular architecture is largely a function of technical considerations and demand characteristics implies that optimal architecture will be dynamic in the long run[267]

259. VAN SCHEWICK, *supra* note 3, at 169.

260. Stefano Brusoni et al., *The Value and Costs of Modularity: A Problem-Solving Perspective*, 4 EUR. MGMT. REV. 121, 122–123, 130 (2007).

261. *Id.* at 123, 128, 130.

262. *Id.* at 128.

263. *Id.*

264. Langlois, *supra* note 10, at 26; Schaefer, *supra* note 34, at 325.

265. BALDWIN & CLARK, *supra* note 5, at 89; von Hippel, *supra* note 29, at 409; Langlois, *supra* note 10, at 23.

266. Timothy F. Bresnahan, *New Modes of Competition: Implications for the Future Structure of the Computer Industry*, *in* COMPETITION, INNOVATION AND THE MICROSOFT MONOPOLY: ANTITRUST IN THE DIGITAL MARKETPLACE 155, 161 (Jeffrey A. Eisenach & Thomas M. Lenard eds., 1999).

267. *See* BALDWIN & CLARK, *supra* note 5, at 156, 180.

and that modular architectures may need to undergo remodularization from time to time.[268]

Accordingly, modularity theory provides a theory of architectural change, along with reasons to suspect that such change may be too long in coming. The power of this analytical framework will be explored through the case studies presented in the next Part.

VI. POLICY APPLICATIONS

Modularity theory provides a number of insights into many recent and current policy issues. These include unbundling under the Telecommunications Act of 1996, network neutrality, calls for opening Application Programming Interfaces ("APIs") for platforms such as the iPhone and Twitter, and proposals for clean-slate redesigns of the network architecture.

A. Unbundling of Local Telephone Networks

Starting first with an example from the traditional telephone network, perhaps the key regulatory innovation of the past several decades is the requirement that incumbent providers offer unbundled access to all of the elements of their networks. The FCC articulated the seminal unbundling requirement as part of its *Computer Inquiries*,[269] which arose when local telephone companies began to move beyond offering traditional voice communications (which the FCC called *basic services*) to offer voicemail, caller ID, Internet access, and other new advanced services that combined the transport of communications with computer processing, storage, or interaction with stored information (which the FCC called *enhanced services*).[270]

The second *Computer Inquiry* attempted to prevent large local telephone companies from favoring their own enhanced service offerings by requiring them to provide enhanced services through a separate subsidiary and to offer all enhanced service providers the same access to their transport infrastructure.[271] The FCC became concerned that mandating such a high degree of vertical disintegration was preventing new services

268. Langlois, *supra* note 10, at 25.

269. For an overview of the *Computer Inquiries*, see Daniel F. Spulber & Christopher S. Yoo, *Access to Networks: Economic and Constitutional Connections*, 88 CORNELL L. REV. 885, 1005–09 (2003).

270. 47 C.F.R. § 64.702(a) (2015).

271. Amendment of Section 64.702 of the Commission's Rules and Regulation's (Second Computer Inquiry), 77 F.C.C.2d 384, 217–39 ¶¶ 201–260 (1980), *aff'd sub nom.* Computer & Commc'ns Indus. Ass'n v. FCC, 693 F.2d 198 (D.C. Cir. 1982). This recapitulated the structural separation requirement imposed by the first *Computer Inquiry*, which focused on the pervious distinction between *communications services* and *data processing services*, which ultimately proved to be untenable. Regulatory and Policy Problems Presented by Interdependence of Computer and Communication Services and Facilities, Final Decision and Order, 28 F.C.C.2d 267, 270–74 ¶¶ 11–22 (1971), *aff'd sub nom.* GTE Serv. Corp. v. FCC, 474 F.2d 724 (2d Cir. 1973).

from appearing.[272] As a result, the *Third Computer Inquiry* created a regime where local telephone companies could avoid the separate subsidiary requirement if they provided unbundled access to their networks on an element-by-element basis.[273]

The Telecommunications Act of 1996 extended unbundling to every incumbent local telephone company.[274] Requiring incumbents to provide access to all of their network elements on an unbundled basis was the centerpiece of Congress' landmark effort to stimulate competition in local telephone service. The unbundling requirement imposed by the 1996 Act did include one key limitation. It required the FCC to determine whether access to proprietary network elements was "necessary" and whether the failure to provide access to such elements would "impair" the requesting carrier's ability to provide service.[275] The statute also required incumbents to provide unbundled access at any technically feasible point.[276]

The hope was that new entrants would use unbundling to combine their own facilities with facilities leased from the incumbents (those elements that still were natural monopolies) to provide service. In the words of Nicholas Economides, "[t]he basic logic behind the Telecommunications Act was to break the network into components and let everyone compete in every part."[277] In so doing, policymakers followed the same approach that they adopted when competition became possible with respect to telephone handsets (known as customer premises equipment or CPE), long distance services, and new information services that combined transmission with data processing: isolate those portions that continued to exhibit natural monopoly characteristics and mandate open access to those portions.[278]

Unbundling is now widely regarded as a failure.[279] As of today, meaningful facilities-based competition for wireline telephone services

272. Amendment of Sections 64.702 of Commission's Rules and Regulations (Third Computer Inquiry), Report and Order, 104 F.C.C.2d 958, 1002–12 ¶¶ 78–99 (1986).

273. The FCC called this regime Open Network Architecture (ONA). *Id.* at 1064–66 ¶¶ 214–17. Until it was fully implemented, the FCC mandated a more limited, transitional regime known as Comparably Efficient Interconnection (CEI). *Id.* at 1035–43 ¶¶ 147–166.

274. 47 U.S.C. § 251(c)(3) (2012).

275. *Id.* § 251(d)(2)(A)–(B).

276. 47 U.S.C. § 251(c)(3).

277. Nicholas Economides, *Vertical Leverage and the Sacrifice Principle: Why the Supreme Court Got* Trinko *Wrong*, 61 N.Y.U. ANN. SURV. AM. L. 379, 390 (2005).

278. Christopher S. Yoo, *Deregulation vs. Reregulation of Telecommunications: A Clash of Regulatory Paradigms*, 36 J. CORP. L. 847, 850 (2011).

279. *See, e.g.*, Susan P. Crawford, *Network Rules*, LAW & CONTEMP. PROBS., Spring 2007, at 51, 87 n.165 ("Mandated unbundling under the Act is widely viewed to have been a failure."); Richard A. Epstein, *Takings, Commons, and Associations: Why the Telecommunications Act of 1996 Misfired*, 22 YALE J. ON REG. 315, 315–16 (2005) ("There is widespread agreement today on all sides of the telecommunications wars that something is deeply flawed with the design or implementation (or both) of the Telecommunications Act of 1996."); Mark A. Lemley & Philip J. Weiser, *Should Property or Liability Rules Govern Information?*, 85 TEX. L. REV. 783, 811 (2007) ("[T]he unbundling regime of the 1996 Act represents, on almost all accounts, a policy failure."); Kevin Werbach, *Only Connect*, 22 BERKELEY TECH. L.J. 1233, 1237 (2007) ("The current legal framework, embodied in the Telecommunications Act of 1996 (1996 Act), is widely regarded as a colossal failure.").

has yet to emerge.[280] Although other reasons may exist for this outcome, modularity theory provides an answer that has not been appreciated.

Consider the statutory requirement that incumbents provide unbundled access at any technically feasible point.[281] This requirement presumes that it is possible to create a modular element at any location in the network. Modularity theory suggests why this approach was problematic. Modular interfaces cannot be established anywhere; they can be created only at thin crossing points where the number of interdependencies is relatively low.

Modularity theory also provides insights into problems with adopting an exclusively economically oriented approach for determining which elements would be modularized. After three failed attempts,[282] the FCC was finally able to implement unbundling in a manner that withstood judicial review.[283] Following the D.C. Circuit's instructions that the rules be "linked (in some degree) to natural monopoly,"[284] the agency's rules adopted a decidedly economic approach. A proprietary network element was "necessary" only if lack of access would preclude the requesting carrier from providing service.[285] With respect to nonproprietary network elements, requesting carriers would be "impaired" if lack of access represented a barrier to entry likely to make it uneconomic for a reasonably efficient competitor to enter.[286] In both cases, regulators had to take into account whether the requesting carrier could build the requested network elements itself or contract for the same services with another provider.[287] The prices for unbundled access would be based on forward-looking economic cost.[288]

The FCC's approach focuses on a network element's cost characteristics without taking into account the interdependencies between that element and the other portions of the network. If the element is tightly integrated with other components, any attempt to mandate access would be expected to fail regardless of whether it is a natural monopoly or is unavailable through other means.

280. *See* Robert W. Crandall & Leonard Waverman, *The Failure of Competitive Entry into Fixed-Line Telecommunications: Who Is at Fault?*, 2 J. COMPETITION L. & ECON. 113, 114–25 (2006). Interestingly, competition for local telephone services has arisen intermodally, with wireless telephony now serving as an effective substitute for traditional wireline services. Interestingly, regulators were initially reluctant to regard cellular telephones as substitutes for wireline service. Daniel F. Spulber & Christopher S. Yoo, *Toward a Unified Theory of Access to Local Telephone Networks*, 61 FED. COMM. L.J. 43, 86–87 (2008).

281. 47 U.S.C. § 251(c)(3).

282. *See* AT&T Corp. v. Iowa Utils. Bd., 525 U.S. 366, 387–92 (1999); U.S. Telecom Ass'n v. FCC, 359 F.3d 554, 571–73 (D.C. Cir. 2004); U.S. Telecomm. Ass'n v. FCC (*USTA I*), 290 F.3d 415, 422–28 (D.C. Cir. 2002).

283. *See* Covad Commc'ns Co. v. FCC, 450 F.3d 528, 538–43 (D.C. Cir. 2006).

284. *USTA 1*, 290 F.3d at 427.

285. 47 C.F.R. § 51.317(a)(1) (2015).

286. *Id.* § 51.317(b).

287. *Id.* § 51.317(a)(1) & (b).

288. *Id.* § 51.505.

Such interdependencies are more likely to arise under unbundling than under other forms of access because, as the Supreme Court noted in *Trinko*, unbundling involves network elements "deep within the bowels" of a local telephone network that can be made available only if "[n]ew systems [are] designed and implemented simply to make that access possible."[289]

Thus, unlike mandated access to CPE and long distance, which involved a market boundary that was "simple, easy to monitor and require[d] little information," the higher level of interdependencies associated with unbundling made it less likely that it would prove successful.[290] In short, unbundling attempted to introduce transactions into what is technologically a transaction-free zone.[291]

Modularity theory also sheds new light on some of the disputes that arose during the debates over unbundling. For example, the local telephone companies asked the FCC to interpret the unbundling requirement imposed by the third *Computer Inquiry* so as not to mandate "fundamental unbundling" on an element-by-element basis and instead only mandate access to larger combinations known as basic service elements[292] only to see that effort overturned on judicial review.[293] To the extent that those larger combinations reflected interdependencies, modularity theory suggests that requiring that bundles of network elements be leased as a unit would have been well advised.

Similarly, when implementing the 1996 Act, the FCC issued a rule that was eventually upheld by the Supreme Court preventing local telephone companies from separating network elements that had already been combined together before leasing them to competitors.[294] Again, modularity theory underscores that certain elements form integrated units that should be provided together.

B. The Internet Protocol

For the past several years, debates over Internet policy have been dominated by two big issues: network neutrality and the transition from IPv4 to IPv6. The Internet is designed around modular architecture embodied in the layered suite of protocols known as the Transmission Control Protocol/Internet Protocol ("TCP/IP").[295] In many ways, the Internet

289. Verizon Commc'ns Inc. v. Law Offices of Curtis V. Trinko, LLP, 540 U.S. 398, 410 (2004).

290. Gerald R. Faulhaber, *Policy-Induced Competition: The Telecommunications Experiments*, 15 INFO. ECON. & POL'Y 73, 77–78, 83, 86, 91 (2003).

291. *See supra* note 123 and accompanying text.

292. Filing and Review of Open Network Architecture Plans, Memorandum Opinion and Order, 4 FCC Rcd. 1, 13–14 ¶¶ 5–8, 41–42, 69 (1988).

293. California v. FCC, 39 F.3d 919, 925–30 (9th Cir. 1994).

294. Implementation of the Local Competition Provisions in the Telecommunications Act of 1996, First Report and Order, 11 FCC Rcd. 15499, 15647 ¶ 293 (1996) (codified at 47 C.F.R. § 51.315(b)), *aff'd sub nom.* AT&T Co. v. Iowa Utils. Bd., 525 U.S. 366, 393–95 (1999).

295. *See, e.g.*, Crowcroft et al., *supra* note 205, at 23 ("Layering is about modularizing the functions performed on data during its transfer from one machine to another, so that complexity of the transfer can be managed."); Douglas C. Sicker & Lisa Blumensaadt, *Misunderstanding the Layered*

represents a classic example of how to use modularity to simplify a complex system. Its central focus was to allow a heterogeneous group of hosts to interconnect through an equally heterogeneous set of transmission technologies.[296] For example, each of the first four ARPANET sites (UCLA, UC Santa Barbara, SRI, and the University of Utah) each used a different type of computer as its host (CDS Σ-7, IBM 360/75, XDS 940, DEC PDP 10), which operated fundamentally incompatible design principles.[297] In addition, the original experiment that validated the Internet as a principle successfully integrated a wireline telephone network, a satellite-based network, and a mobile wireless network to send a single transmission.[298]

The lynchpin of the TCP/IP suite is the network-layer protocol known as the Internet Protocol ("IP").[299] IP is a module that mediates both the diversity of applications and computers being run by end users (known as hosts) on the one hand and the diversity of transmission technologies on the other. Because of this, the Internet's architecture is often portrayed as an hourglass, with a wide variety of host-based protocols at the top of the hourglass and a wide variety of networking and transmission technologies at the bottom.[300] IP is "[t]he hourglass' narrow waist," which "represents a minimal and carefully chosen set of global capabilities that allows both higher-level applications and lower-level communication technologies to coexist, share capabilities, and evolve rapidly."[301] In other words, IP represents a modular interface that reflects careful decisions about which interdependencies are allowed to be reflected in interactions between hosts and networks. It is the limits on the visible information built into this interface that allows applications to ignore the details of the underlying transmission technology and vice versa.

TCP/IP yields all of the advantages associated with modularity. It reduces the complexity of the system by rendering subsystems as independent as possible from one another. It promotes flexibility and allows for simultaneous parallel experiments by a diversity of organizations. It enables third-party development of new applications without permission

Model(s), 4 J. ON TELECOMM. & HIGH TECH. L. 299, 305 (2006) ("Each layer provides a well-defined set of services to the layers above it and depends on lower layers for its own operation, thus creating modularity.").

296. JANET ABBATE, INVENTING THE INTERNET 51–52, 131–32 (1999).

297. C. Stephen Carr et al., *HOST-HOST Communication Protocol in the ARPA Network*, 36 AFIPS CONF. PROC. 589, 589 (1970).

298. Packet Radio System Development: Quarterly Management Report No. 15, at 5–7 (Stanford Research Institute Project 2325-NS, Nov. 12, 1976).

299. Barry M. Leiner et al., *The DARPA Internet Protocol Suite*, IEEE COMM. MAG., Mar. 1985, at 29, 31; s*ee also* ANDREW S. TANENBAUM, COMPUTER NETWORKS 432 (4th ed. 2003) (calling IP "[t]he glue that holds the whole Internet together"); Brian E. Carpenter, *Architectural Principles of the Internet* 2 (Network Working Group Request for Comments 1958, June 1996), *available at* http://tools.ietf.org/html/rfc1958 (noting that "[t]he key to global connectivity is the inter-networking layer").

300. *See, e.g.*, COMPUTER SCI. & TELECOMM. BD., NAT'L RESEARCH COUNCIL, THE INTERNET'S COMING OF AGE 36–38, 126–29 (2001); Steve Deering, *Watching the Waist of the Protocol Hourglass*, PROC. 51ST INTERNET ENG'G TASK FORCE 2 (2001), http://www.ietf.org/proceedings/51/slides/plenary-1/index.html.

301. PETERSON & DAVIE, *supra* note 4, at 30.

from any other actors. However, even the architects of the TCP/IP protocol recognized that it is "a mixed blessing."[302] In fact, it is subject to a number of the drawbacks associated with all modular architectures.

1. *Restrictions on Functionality in IPv4*

Like all modular architectures, TCP/IP employs information hiding to limit and structure the interactions between modules. In so doing, it essentially embodies a precommitment about which types of information are important enough to pass between the modules. As a result, the limited nature of the interfaces inevitably "leads to a lack of understanding as to what one layer wishes to obtain from another."[303]

The details of why this is so are best understood by examining the information contained in the IP header, which constitutes the critical interface in this particular modularization. The IP version 4 header consists of a fixed part consisting of five 32-bit words and an optional part that can consist of up to ten additional 32-bit words.

FIGURE 4: THE IPV4 HEADER

◄——32 bits——►

Version	IHL	Type of Service	▨	Total Length	
Identification				DF/MF	Fragment Offset
Time to Live		Protocol		Header Checksum	
Source Address					
Destination Address					
Options (if any)					

Source: Info. Sci. Inst., *Internet Protocol: DARPA Internet Program Protocol Specification* 11 (Request for Comments 791, 1981), *available at* http://tools.ietf.org/pdf/rfc791.

In the first word, *Version* specifies whether the packet is an IPv4 packet or some other version. Because the length of the header varies, *IHL* stands for Internet Header Length and indicates when the header

302. RFC 817, *supra* note 159, at 24.

303. *Id.* at 12; *accord* RFC 3439, *supra* note 205, at 7–8 (noting that layering "hide[s] vital information that lower layers may need to optimize their performance" and requires "that the optimization of each layer . . . be done separately," which can "conflict with efficient implementation of data manipulation functions"); Crowcroft et al., *supra* note 205, at 23 ("[T]he flip side to modularization and data-hiding is that tuning the efficiency of the data path for transfer of data becomes difficult, as important details such as buffer sizes are hidden from each layer. Vertical partitioning emphasizes the discontinuities in the data path, which then obstruct the application from receiving the quality of service it requires.").

ends and the data begins, which is necessary because the inclusion of options at the end of the header can cause the size of the header to vary. *Type of Service* allows senders to assign different levels of propriety to particularly packets. *Total Length* indicates the size of the entire packet, including both the data and the header. The second word consists of fields—*Identification*, *DF* (for Don't Fragment), *MF* (for More Fragments), and *Fragmentation Offset*—that are designed to allow IP to divide a large packet into smaller fragments. The third word consists of three fields. *Time to Live* is a counter that is decremented each hop the packet traverses until the counter reaches zero in order to ensure that packets without a proper destination address do not wander around the Internet forever. *Protocol* specifies whether the packet is associated with TCP, UDP, or some other transport protocol. *Header Checksum* verifies the accuracy of the header. The fourth and fifth words consist of the *Source Address* and the *Destination Address*. The *Options* fields are rarely used except for debugging purposes.[304]

The visible information contained in the IP header inevitably constrains the type of information that can pass between the hosts connected to the edge of the network and the network itself. Because the information in the IP header is basically limited to the type of service, transport protocol, source address, and destination address, it cannot support functionality that depends on any other information. This has particular import with respect to three functions: congestion management, reliability, and security.

a. Congestion Management

Congestion management represents one of the central issues in the network neutrality debate. Indeed, the FCC listed congestion management as an example of "reasonable network management" exempted from the nondiscrimination requirements established by its Open Internet Order.[305] The problem is that the IP header does not include a field through which hosts and the network can exchange information about congestion. In other words, congestion is one of the interdependencies that is part of the hidden information of the architecture and thus is opaque to all of the modules.

Congestion occurs when multiple hosts attempt to use the network at the same time. Individual hosts, however, generally only possess information about their activities and typically lack information about the behavior of other hosts. The nodes in the core of the network are in a far better position to see the flows being generated by multiple users.[306] The fact that congestion management requires information that is available

304. TANENBAUM, *supra* note 299, at 433–36.

305. Open Internet Order, *supra* note 2, at 17951–52 ¶¶ 81–82, 17955–56 ¶¶ 91–92.

306. Sally Floyd & Van Jacobson, *Random Early Detection Gateways for Congestion Avoidance*, 1 IEEE/ACM TRANSACTIONS ON NETWORKING 397, 397 (1993).

only in the core similarly favors a core-based solution. At the same time, when congestion arises, the proper response is for hosts to cut their sending rates in half.[307]

Ideally, the IP header would contain a bit that networks could use to signal to hosts that the network is congested. Indeed, Raj Jain proposed just such a solution during the Internet's early days.[308] A network-based solution would have required upgrading the network's core routers, which would have taken a long time and would have been prohibitively expensive.[309] Instead, Van Jacobson and Mike Karels devised a solution based on the host-based approach to reliability embedded in TCP/IP.[310] Under this approach, receiving hosts are supposed to acknowledge every packet they receive.[311] If the sending host does not receive an acknowledgement within the expected amount of time, it resends the packet.[312]

Jacobson and Karels realized that networks typically drop packets for only two reasons: either the packet became corrupted or the packet encountered a congested buffer that was full.[313] Because wireline networks rarely corrupt packets, hosts could take the failure to receive an acknowledgement as a de facto signal that the network was congested and as an indication that they needed to reduce traffic.[314] Unlike a network-based solution, Jacobson and Karels' solution only required changing a few lines of code in every host.[315]

Jacobson and Karels never intended for their solution to be a permanent one.[316] As an initial matter, because it depended on an inference from the lack of an acknowledgement, it worked only for traffic based on transport protocols that use acknowledgments, such as TCP.[317] Modern applications such as streaming video and VoIP have placed increasing emphasis on protocols that do not use acknowledgements, such as UDP.[318]

307. Van Jacobson, *Congestion Avoidance and Control*, ACM SIGCOMM COMPUTER COMM. REV., Aug 1988, at 314, 318.

308. K.K. Ramakrishnan & Raj Jain, *Congestion Avoidance in Computer Networks with a Connectionless Network Layer: Concepts, Goals and Methodology,* PROC. COMPUTER NETWORKING 1, 6–7 (1988), *available at* http://www.cse.wustl.edu/~jain/papers/cr1.htm.

309. RICHARD BENNETT, DESIGNED FOR CHANGE: END-TO-END ARGUMENTS, INTERNET INNOVATION, AND THE NET NEUTRALITY DEBATE 16 (2009), *available at* http://itif.org/files/2009-designed-for-change.pdf (noting that the network adopted Jacobson's solution because changing a few lines of code was cheaper than upgrading hundreds of routers).

310. Jacobson, *supra* note 307, at 314.

311. Info. Sci. Inst., *Transmission Control Protocol: DARPA Internet Program Protocol Specification* 4, 10 (Request for Comments 793, Sept. 1981), *available at* http://tools.ietf.org/html/rfc793.

312. *Id.*

313. Jacobson, *supra* note 307, at 319.

314. *Id.*

315. *Id.* at 314–15, 321.

316. *See id.* at 322.

317. Mark Handley, *Why the Internet Only Just Works*, 24 BT TECH. J. 119, 120 (2006).

318. JAMES F. KUROSE & KEITH W. ROSS, COMPUTER NETWORKING: A TOP-DOWN APPROACH 213 (5th ed. 2010) (calling UDP's lack of congestion control "controversial" because UDP's failure to reduce its rate in response to packet loss can cause UDP traffic to "crowd[] out . . . TCP sessions"); *id.* at 293 ("From the perspective of TCP, the multimedia applications running over UDP are not being

Just as problematic is the growing importance of wireless broadband technologies. Unlike wireline networks, wireless networks often drop packets for reasons other than congestion, such as when atmospheric conditions or reflections create a dead spot that limits the amount of bandwidth available or when a bad handoff between cell sites causes a transmission to become dropped.[319] Thus, the advent of wireless broadband further undercuts the inference-based solution used to work around the absence of a field in the IP header through which networks can notify hosts directly about the presence of congestion. A later modification known as Explicit Congestion Notification ("ECN") has made some information about congestion visible in the IP header,[320] and a current initiative known as Congestion Exposure ("ConEx") is underway to insert additional information into the IP header.[321] Neither is as yet mandatory in Internet implementations, and neither is widely deployed.

b. Reliability

Changes in the ways end users are using the Internet, both in terms of technologies and in terms of applications, can cause the optimal configuration of the layered stack to change. Currently, responsibility for guaranteeing reliability rests with the hosts. While this approach made sense when networks were based around wireline telephone technologies, it makes less sense when wireless technologies are involved. Unlike wireline networks, which rarely drop packets for reasons other than congestion, wireless networks suffer from much higher loss rates caused by the sensitivity of spectrum-based transmission to local conditions.[322]

For this reason, early wireless networks like the San Francisco Bay Packet Radio Network ("PRNET") employed a network-based reliability system known as forward-error correction.[323] This difference between wireline and wireless technologies also explains why modern wireless broadband networks are increasingly deploying network-based reliability systems, such as Automatic Repeat reQuest ("ARQ"), that detect transmission errors and retransmit the missing data without waiting for the host-based retransmission timer to expire and without consuming the additional network resources needed to retrieve the packet all the way

fair—they do not cooperate with the other connections nor adjust their transmission rates appropriately. Because TCP congestion control will decrease its transmission rate in the face of increasing congestion (loss), while UDP sources need not, it is possible for UDP sources to crowd out TCP traffic.").

319. Christopher S. Yoo, *The Changing Patterns of Internet Usage*, 63 FED. COMM. L.J. 67, 78–79 (2010).

320. *See* K. Ramakrishnan, The Addition of Explicit Congestion Notification (ECN) to IP (Network Working Group Request for Comments 3168, Sept. 2001), *available at* http://tools.ietf.org/pdf/rfc3168.

321. IETF, Congestion Exposure (ConEx): Description of Working Group, http://datatracker.ietf.org/wg/conex/charter/ (last visited Sept. 20, 2015).

322. Yoo, *supra* note 319, at 77–80.

323. Robert E. Kahn et al., *Advances in Packet Radio Technology*, 66 PROC. IEEE 1468, 1488–95 (1978).

from the host.[324] Other techniques that allow routers in the core to participate in the transport layer exist as well.[325]

These experiments with shifting responsibility for reliability from the hosts operating at the edge of the network into the network itself are part of the burgeoning literature on cross-layer design in wireless networks.[326] This shift represents a clear deviation from the allocation of functions embodied in the TCP/IP reference model. That said, the pragmatic approach of network engineering counsels against basing objections to incorporating cross-layer design into the network on rigid adherence to a fundamentalist principle.[327] Instead, it envisions that the allocation of functions will likely change over time with shifts in technology and the underlying demand for network services.

c. Security

Another good example of the limits of the information contained in the IP header is network security. The current architecture does not permit verifiable information about the identity of particular end users to pass through the protocol stack.[328] Despite the growing need for security, the network has been slow to adapt to this new reality. The shifting importance of these concerns underscores the importance of not regarding any particular layered architecture as if it were a natural construct. Moreover, it underscores the potential dangers of using regulation to enshrine any particular architecture into law.

2. *Increase in Variety*

Another force providing impetus toward remodularization is the fundamental change in the external environment. The number of end users connecting to the Internet has exploded and become more heterogeneous in terms of geography, interests, and capabilities.[329] They are using a wider variety of applications, many of which place new and more intensive demands on the network.[330] End users are connecting through an ever broadening array of devices connected to an increasingly heterogene-

324. KUROSE & ROSS, *supra* note 318, at 219–27; TANENBAUM, *supra* note 299, at 208–11.

325. *See* TANENBAUM, *supra* note 299, at 553–55 (exploring indirect TCP and the inclusion of snooping agents as possible solutions to the problem).

326. For surveys of this literature, see Fotis Foukalas et al., *Cross-Layer Design Proposals for Wireless Mobile Networks: A Survey and Taxonomy*, 10 IEEE COMM. SURVS. & TUTORIALS 70 (2008); Sanjay Shakkottai et al., *Cross-Layer Design for Wireless Networks*, 41 IEEE COMM. MAG. 74 (2003); Vineet Srivastava & Mehul Motani, *Cross-Layer Design: A Survey and the Road Ahead*, 43 IEEE COMM. 112 (2005). For a more skeptical assessment, see Vikas Kawadia & P.R. Kumar, *A Cautionary Perspective on Cross-Layer Design*, 12 IEEE WIRELESS COMM. MAG. 3 (2005).

327. James Kempf & Rob Austein, *The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture* 8, 10 (Network Working Group, Request for Comments 3724, 2004), *available at* http://tools.ietf.org/html/rfc3724.

328. *See supra* note 202–03 and accompanying text.

329. CHRISTOPHER S. YOO, THE DYNAMIC INTERNET: HOW TECHNOLOGY, USERS, AND BUSINESSES ARE TRANSFORMING THE NETWORK 13–18 (2012).

330. *Id.* at 19–36.

ous set of transmission technologies.[331] The business environment has become more complex as well, both in terms of pricing and topology.[332]

The Law of Requisite Variety dictates that the increase in the diversity of the external environment will require the architecture to evolve in response, either by increasing the number of modules or by redesigning the interfaces to reduce the number of intermodule interactions. Either would require a remodularization of the Internet stack.

3. *Lack of Coordination*

Network engineers have also long recognized that the decentralized nature of congestion management can lead to persistent problems.[333] Because individual end users can maximize their private benefits by continuing to send traffic into the network while everyone else slows down, they have substantial private incentives to deviate from the social optimum.[334] The IP header does not provide any way for other actors to verify whether or not a host is responding appropriately to the presence of congestion. As noted earlier, the absence of a network-based means for ensuring that hosts reduce their transmission rates when the network is congested means that congestion management currently depends on what amounts to the honor system despite the fact that each end user has the incentive to deviate from the optimal behavior.[335] Although the Internet community may once have represented the type of close-knit community that could prevent such deviations from occurring, the rapid expansion of the Internet has undercut its ability to rely on social norms to protect against this type of behavior.[336]

The decisions surrounding what information to place in the IP header thus limits the network's ability to address certain types of problems. As in any modular architecture, the particular decisions about which interdependencies to acknowledge has a direct impact on the system's functionality. Imposing rules that forbid any deviations from the current architecture would risk foreclosing solutions to problems that continue to grow in importance.

331. *Id.* at 37–54.

332. *Id.* at 55–69.

333. Bob Braden et al., *Recommendations on Queue Management and Congestion Avoidance in the Internet* 9–10 (Network Working Group Request for Comments 2309, Apr. 1998) [hereinafter RFC 2309], *available at* http://tools.ietf.org/html/rfc2309.

334. Dimitri Papadimitriou et al., *Open Research Issues in Internet Congestion Control* 31 (Internet Research Task Force Request for Comments 6077, 2011), *available at* http://tools.ietf.org/html/rfc6077 ("In the current Internet architecture, congestion control depends on parties acting against their own interests. It is not in a receiver's interest to honestly return feedback about congestion on the path, effectively requesting a slower transfer. It is not in the sender's interest to reduce its rate in response to congestion if it can rely on others to do so."); RFC 2309, *supra* note 333, at 9–10, 11 (noting that TCP implementations "can grab an unfair share of the network bandwidth" by aggressively refusing to back off, which would logically lead to "a spiral of increasingly aggressive TCP implementations, leading back to the point where there is effectively no congestion avoidance and the Internet is chronically congested").

335. *See supra* note 203 and accompanying text.

336. YOO, *supra* note 329, at 82–99.

4. *Network Neutrality*

The decade-long debate over network neutrality appears to be approaching a major turning point. At its core, network neutrality proponents oppose discrimination (providing some traffic more favorable treatment than others) and paid prioritization.

Providing a comprehensive survey of the network neutrality debate exceeds the scope of this Article. For our purposes, it suffices to note that the ability to assign a higher priority to some packets has been a functionality contained in the visible information in the IPv4 since its inception. Prioritization would amount to nothing more than taking advantage of the functionality built into the architecture. In the words of David Clark, "[t]he Internet is not neutral and has not been neutral for a long time."[337]

Equally importantly, many features of IP-enabled networks depend on some degree of prioritization. Proprietary voice over Internet Protocol ("VoIP") services, such as Comcast Digital Voice, depend on some form of prioritization to ensure call quality, as does voice over LTE ("VoLTE"). Some video services, such as AT&T's U-verse network, ensure quality by prioritizing traffic associated with a single application (video) from a single provider (AT&T).[338] Finally, zero-rating systems such as T-Mobile's Music Freedom and Binge On allow users to stream music without having that usage count against their data caps.[339] Programs such as Facebook Zero, Twitter Zero, and Wikipedia Zero are using similar programs to help deploy Internet access in the developing world.[340] They do so by making one application cheaper than others in a manner completely consistent with the existing architecture.

337. David Clark, Written Statement to the En Banc Public Hearing on Broadband Network Management Practices (Feb. 25, 2007), *available at* https://transition.fcc.gov/broadband_network_management/022508/clark.pdf; *see also* Robert W. Hahn & Robert E. Litan, *Portioning Bit by Bit: The Myth of Network Neutrality and the Threat to Internet Innovation*, MILKEN INST. REV. 28, 31–33 (2007); Jonathan E. Nuechterlein, *Antitrust Oversight of an Antitrust Dispute: An Institutional Perspective on the Net Neutrality Debate*, 7 J. ON TELECOMM. & HIGH TECH. L. 19, 36–37 (2009); Christian Sandvig, *Net Neutrality Is the New Common Carriage*, 9 INFO 136, 137–42 (2007); Douglas A. Hass, Comment, *The Never-Was-Neutral Net and Why Informed End Users Can End the Net Neutrality Debates*, 22 BERKELEY TECH. L.J. 1565, 1576–77 (2007); Kai Zhu, Note, *Bringing Neutrality to Network Neutrality*, 22 BERKELEY TECH. L.J. 615, 634–36 (2007); Michael Grebb, *Neutral Net? Who Are You Kidding?*, WIRED (May 31, 2006), http://archive.wired.com/techbiz/it/news/2006/05/71012?currentPage=all; Andrea Renda, *I Own the Pipe, You Call the Tune: The Net Neutrality Debate and Its (Ir)relevance for Europe* 9–11 (2008), *available at* http://www.ceps.eu/system/files/1755.pdf; Craig McTaggart, Was the Internet Ever Neutral? (2006) (unpublished manuscript), *available at* http://ssrn.com/abstract=2117601.

338. Yoo, *supra* note 319, at 75–76.

339. T-MOBILE, http://www.t-mobile.com/offer/free-music-streaming.html (last visited Sept. 22, 2015).

340. *See generally* Diana Carew, Policy Memo, *Zero-Rating: Kick-Starting Internet Ecosystems in Developing Countries*, PROGRESSIVE POLICY INSTITUTE (2005), *available at* http://www.progressivepolicy.org/wp-content/uploads/2015/03/2015.03-Carew_Zero-Rating_Kick-Starting-Internet-Ecosystems-in-Developing-Countries.pdf.

5. *The Transition to IPv6*

Perhaps the biggest change to the modular architecture of the Internet is the transition to IPv6. IPv4 addresses consist of 32 bits, which supports 2^{32} or roughly 4.3 billion addresses. Although the original Internet architects thought that was more than enough to satisfy the expected demand,[341] they did not anticipate that every consumer would have multiple devices connected to the network, let alone be seeking connectivity for numerous appliances in their home. As a result, the Internet began running out of new addresses in 2011.[342] Consequently, the engineering community developed IPv6, which contains 2^{128} addresses[343] or enough to assign 47 trillion trillion addresses to every person on the planet.[344]

In addition to increasing the size of the address space, deployment of a new network protocol provided an occasion to consider whether to modify other aspects of the network architecture. For example, goals included providing for better support for security, transmission of real-time data, and mobility.[345] The resulting IPv6 header is depicted in Figure 5.

Version specifies whether the packet is an IPv4 or IPv6 packet. *Traffic Class* permits sending nodes and routers to assign different levels of priority to particular traffic similar to the Type of Service field in the IPv4 header. *Flow Label* provides a basis for associating different packets together for quality of service or real-time service. *Payload Length* describes the length of the packet following the header (including options). *Next Header* specifies whether the information immediately following the IPv6 header is one of the IPv6 extension headers that replaced the options in IPv4 or the transport protocol header that is the beginning of the payload. *Hop Limit* is a counter that is decremented by each time packet traverses a node until the counter reaches zero similar to the Time to Live field in the IPv4 header. *Source Address* and *Destination Address* specify the origin and termination point of the traffic.

341. Laurie J. Flynn, *Drumming Up More Addresses on the Internet*, N.Y. TIMES, Feb. 14, 2011, http://nyti.ms/1V9kfVO; Paul McNamara, *Why IPv6? Vint Cerf Keeps Blaming Himself*, NETWORK WORLD (Oct. 22, 2010, 9:49 AM), http://www.networkworld.com/article/2227543/software/why-ipv6--vint-cerf-keeps-blaming-himself.html.

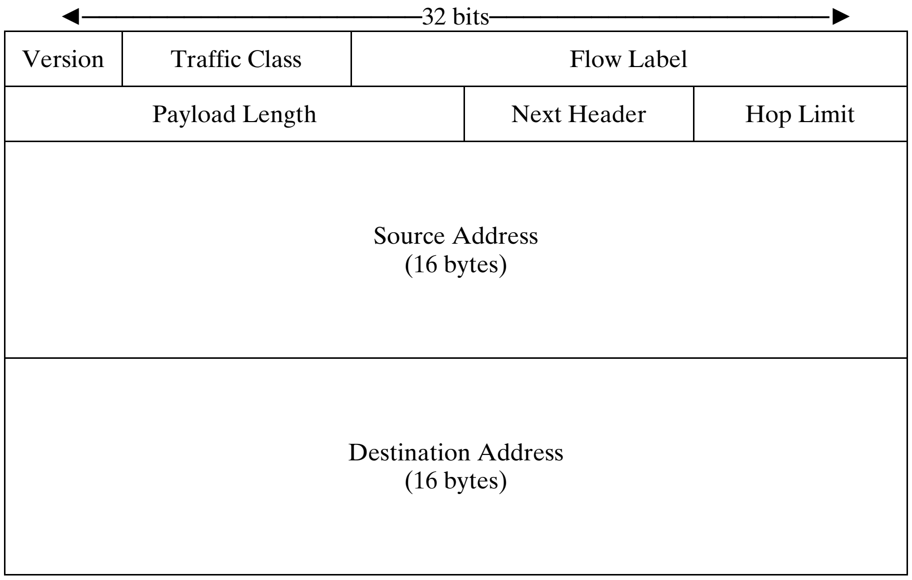
342. The Internet Assigned Numbers Authority ("IANA") allocates blocks of Internet addresses to five regional Internet registries ("RIRs"), which in turn allocate them to those who request them. IANA allocated its last block of addresses on February 3, 2011. The RIRs for the Asia/Pacific, Europe, and Latin American regions have already exhausted their allocations, with the RIRs for North America and Africa projected to be depleted in 2015 and 2019. *IPv4 Address Report*, POTAROO (Sept. 1, 2014), http://www.potaroo.net/tools/ipv4/index.html.

343. This is 340 undecillion or more specifically 340,282,366,920,938,463,463,374,607,431,768,211,456 addresses.

344. *See Internet Protocol Version 6: IPv6 for Consumers*, Fed. Comm. Comm'n (July 16, 2014), https://www.fcc.gov/guides/internet-protocol-version-6-ipv6-consumers (last visited Sept. 20, 2015).

345. TANENBAUM, *supra* note 299, at 465.

FIGURE 5: THE IPV6 HEADER



Source: Stephen E. Deering & Robert M. Hinden, *Internet Protocol, Version 6 (IPv6) Specification* 4 (Network Working Group Request for Comments 2460, 1998), *available at* http://tools.ietf.org/pdf/rfc2460.

As noted earlier, the information visible in the interface reflects the architecture's precommitment regarding the interdependencies that other modules are allowed and not allowed to take into account. Consequently, the visible information retained, removed, or added to the header as part of this remodularization is quite telling.

The retention of some fields is unsurprising. For example, the retention of fields specifying the source address, destination address, specifying the size of the packet, and time to live are unremarkable. The deletion of fields involving fragmentation and options reflect the shift of certain information outside of the primary header in order to speed up routing by including fewer fields, standardizing the header length and shifting responsibility for fragmentation and packet integrity to the hosts operating at the edge of the network.[346]

Other decisions with respect to IPv6 header have more significant implications for the functionality of the network. Two examples of these decisions are the inclusion of fields to facilitate prioritization and the omission of fields to support security and mobility.

346. *Id.* at 466–71.

a. Prioritization

One of the most striking aspects of the IPv6 header is the inclusion of fields to facilitate prioritization of traffic. As noted above, the IPv4 header has always included a Type of Service field to support providing different levels of quality of service.[347] Although the original specification defined the field to focus solely on three types of quality of service: delay, reliability, and throughput,[348] it was subsequently redefined to permit more flexible use of this field through a protocol known as DiffServ.[349] The IPv6 Traffic Class field was explicitly designed to ensure that the experiments in differentiated services built on the IPv4 Type of Service field would continue.[350] The retention of this field reflects a continued commitment to support this functionality.

What is even more striking is the addition of a new field to support even more sophisticated forms of quality of service. As noted earlier, the original architecture envisioned that the Internet would route each packet through the network independently.[351] The result is that packets associated with the same flow could follow different paths, which could cause them to arrive with inconsistent spacing or even out of order.

To solve this problem, the engineering community developed MultiProtocol Label Switching ("MPLS"), which assigns labels to packets associated with the same flow.[352] This permits all packets bearing the same label to be routed along the same path.[353] In addition, labels permit the implementation of a broad range of network policies, such as route selection, prioritization, load balancing, and other forms of traffic engineering that can provide for better security and quality of service.[354] The MPLS header in which the label was embedded typically operates just below the network layer.[355]

IPv6 borrows this aspect from MPLS by including a field for a Flow Label in the header of the network layer. It did so explicitly to enable "special handling by the IPv6 routers, such as non-default quality of ser-

347. Info. Sci. Inst., *Internet Protocol: DARPA Internet Program Protocol Specification* 2 (Request for Comments 791, Sept. 1981), *available at* http://tools.ietf.org/html/rfc791.

348. *Id.* at 12–13.

349. Steven Blake et al., *An Architecture for Differentiated Services* 10 (Network Working Group Request for Comments 2475, Dec. 1998), *available at* http://tools.ietf.org/html/rfc2475.

350. Steven E. Deering & Robert M. Hinden, *Internet Protocol, Version 6 (IPv6) Specification* 25 (Network Working Group Request for Comments 2460, 1998) [hereinafter RFC 2460], *available at* http://tools.ietf.org/html/rfc2460.

351. *See generally* Barry M. Leiner et al., *Brief History of the Internet*, Internet Society, http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet (last visited Sept. 20, 2015).

352. Eric C. Rosen et al., *Multiprotocol Label Switching Architecture* 35 (Network Working Group Request for Comments 3031, Jan. 2001), *available at* http://tools.ietf.org/html/rfc3031.

353. *Id.* at 4.

354. William Stallings, *MPLS*, INTERNET PROTOCOL J., Sept. 2001, at 2, 2–3.

355. The MPLS header can either encapsulate an IP packet, in which case it operates much like a data-link layer packet, or it can encapsulate transport layer packets, in which case it operates like the network layer. This is why it is sometimes said to operate at layer 2.5.

vice or 'real-time' service."[356] The inclusion of labels in the IPv6 specification represents a renewed commitment to supporting prioritization of traffic. Even more importantly, the inclusion of the Flow Label in the spanning layer visible to all other network actors means that such traffic management represents part of the core functionality of the architectural design.

b. Security

Another area of controversy with respect to the design of IPv6 was security. Traditionally, security in IPv4 was implemented by the hosts operating at the edges of the network rather than by the network itself.[357] The IETF developed a security protocol for IPv6 that was retrofitted into IPv4 under the name IP Security ("IPsec"). IPsec is a fairly complex suite of protocols.[358] Reduced to its basics, IPsec adds an extension header to the IP header with an authentication code to allow others to verify the packet's integrity and encrypts the original IP packet (along with padding needed to make the relevant encryption work) to ensure confidentiality. When it operates in transport mode, it only encrypts the payload, whereas when it operates in tunnel mode, it places the encrypted packet in another IP packet.[359] It then uses a designated number in the *Protocol* field in IPv4 or the *Next Header* field in IPv6 to let the receiving host or router know that the communication is being made using IPsec.[360] Thus, IPsec requires all receiving routers and hosts to recognize this designation. In other words, it requires that this designation be part of the visible information that all computers connected to the network are prepared to recognize.

The development of IPv6 involved a wide discussion of how much security-related information to place in the visible information.[361] In the end, the change was fairly modest, consisting only of adding to the IP header the information necessary to authenticate the communication and to decrypt the payload. The actual authentication was left to the hosts. Interestingly even though the initial specifications of IPv6 made this security regime mandatory,[362] the specification was revised in 2011 to make it optional.[363] The removal of this information from the mandatory aspects of the interface represents a major change in architectural commitments. The fact that IPv6 opted not to include this much information

356. RFC 2460, *supra* note 350, at 25.

357. TANENBAUM, *supra* note 299, at 473.

358. For an accessible discussion, see KUROSE & ROSS, *supra* note 318, at 734–40.

359. *Id.* at 737–38.

360. *Id.* at 737.

361. Stephen Kent & Karen Seo, *Security Architecture for the Internet Protocol* 57–58 (Network Working Group Request for Comments 4301, Dec. 2005), *available at* http://tools.ietf.org/html/rfc4301.

362. RFC 2460, *supra* note 350, at 7; Stephen Kent & Karen Seo, *Security Architecture for the Internet Protocol* (Network Working Group Request for Comments 4301, 2005), *available at* http://tools.ietf.org/pdf/rfc4301.

363. Ed Jankiewicz et al., *IPv6 Node Requirements* 18 (Request for Comments 6434, Dec. 2011), *available at* http://tools.ietf.org/html/rfc6434.

represents a precommitment that the actual authentication be performed within one of the modules by limiting the amount of information that could pass through the interface.

c. Mobility

Another controversy surrounding IPv6 was the extent to which it would support mobility. As noted earlier, the IPv4 header includes a destination address, which identifies a physical location. The fact that mobile hosts change locations means that relying on the traditional architecture would require constant updates to the routing architecture to reflect each mobile host's current location. The problem is that updates to routing information take time to propagate through the network. That means that the network would struggle to maintain accurate information of the location of each mobile host.[364]

Instead, the mobility solution for IPv4 requires mobile hosts to designate a router on its home network as its *home agent* and asks that anyone seeking to reach it send to that address.[365] The home agent must also let other networks know that it is serving as the home agent for this particular mobile host.[366] The mobile host then registers with a *foreign agent* on the network on which it is currently located either by contacting the foreign agent or by responding to an advertisement sent by the foreign agent.[367] The foreign agent then sends a *care-of address* to the home agent to inform it where the packets addressed to the mobile host should be sent and to update that information if the mobile agent should transfer to a different node.[368] The home agent then encapsulates any traffic that it receives and forwards it to the foreign agent for delivery.[369] The mobile agent then decapsulates the traffic.[370]

This solution obviates the need for updating the routing tables, but it does incur some increased costs. The solution is quite complex, requiring no fewer than seven protocols and a large number of signaling messages between the hosts and the routers to set it up. In addition, it can lead to what is often called "triangle routing," because instead of passing directly to its destination, the packets must travel via the home agent. In extreme cases, the packets associated with a file being transferred between two people sitting next to each other in a conference room on the West Coast might have to travel to the East Coast and back. As such, it represents an apt illustration of David Wheeler's aphorism: "[a]ll prob-

364. Yoo, *supra* note 319, at 81.
365. *Id.*
366. *Id.*
367. *Id.*
368. *Id.*
369. *Id.*
370. Charles E. Perkins, *IP Mobility Support for IPv4, Revised* 10 (Internet Engineering Task Force Request for Comments 5944, Nov. 2010), *available at* http://tools.ietf.org/html/rfc5944.

lems in computer science can be solved by another level of indirection . . . except for the problem of too many layers of indirection."[371]

Some engineers suggested pursuing a more straightforward solution to this problem during the transition to IPv6. Instead of relying on the hosts to send the packets indirectly by way of the home agent, the updated location information could be included in the network layer by having packets include the address of the mobile host's current location.

The solution was never adopted because of two concerns.[372] The first was the difficulty in updating the mobile host's location should the mobile host shift to a new network.[373] The second was the danger that an imposter could masquerade as the mobile host and highjack all of the communications bound for it.[374]

Consequently, IPv6 declined to incorporate mobility support into the network layer and instead retained the indirect, host-based approach employed for IPv4. The IPv6 protocol did make some small refinements, such as eliminating the need for a foreign agent and routing return traffic directly instead of via the home agent (although this may be impossible if the foreign network's gateway routers check the source address of IP packets and discard those that did not originate on its subnets).[375]

The basic approach is not implemented through information in the primary interface visible to all other networks. Instead, mobility relies on the hosts to perform a number of complicated functions. Relying on hosts to handle mobility means that all of the hosts must be reconfigured in order to make any changes to the network. This can be quite cumbersome and tends to retard innovative change.

As a result, calls for embedding mobility support in the network layer have persisted.[376] One major problem is that addresses serve two distinct purposes: labeling both the identity of the host and its location. The combination of these functions was unproblematic when hosts were immobile personal computers. The growing importance of mobility has led many engineers to call for an identity-locator split, in which distinct addresses are used to specify the identity of the host and its location.[377] New proposals along these lines continue to appear,[378] although other en-

371. *Talk:Indirection*, WIKIPEDIA (15:46, Nov. 9, 2010), http://en.wikipedia.org/wiki/Talk:Indirection (last visited Sept. 20, 2015).

372. *See generally* Charles E. Perkins, *Mobile Networking Through Mobile IP*, 2 IEEE INTERNET COMPUTING 58 (1998) (discussing Mobile IP and the transition to IPv6)

373. *See id.*

374. *See id.*; TANENBAUM, *supra* note 299, at 472.

375. Charles E. Perkins et al., *Mobility Support in IPv6,* at 8 (Request for Comments 6275, July 2011), *available at* http://tools.ietf.org/html/rfc6275.

376. For a survey, see Chakchai So-In et al., *Future Wireless Networks: Key Issues and a Survey (ID/Locator Split Perspective)*, 8 INT'L J. COMM. NETWORKS & DISTRIB. SYS. 24, 30–31 (2012).

377. For a survey, see W. Ramirez et al., *A Survey and Taxonomy of ID/Locator Split Architectures*, 60 COMPUTER NETWORKS 13 (2014).

378. *See, e.g.*, Arun Venkataramani et al., *MobilityFirst: A Mobility-Centric and Trustworthy Internet Architecture*, ACM SIGCOMM COMPUTER COMM. REV., July 2014, at 74.

gineers have questioned that approach.[379] Still others place the responsibility for exchanging signaling messages with the home agent on proxy agents instead of the mobile host.[380] Empirical studies suggest that network-based approaches may improve network performance.[381]

For the purposes of this Article, whether the architecture ever embraces network-based mobility support is immaterial. The ongoing debate underscores how shifts in the underlying technology and demand for wireless devices are changing the demands that people are placing on the network. It also provides an apt illustration of how influential decisions about what information to include in interfaces can affect the functionality of the network.

C. Open Application Programming Interfaces ("APIs")

Modularity theory also sheds new light on calls for platform providers to open up their *application programming interfaces* ("APIs"). APIs are the interfaces that application providers use to give instructions to the underlying platform.[382] APIs increase the modularity of the platform by providing a uniform set of commands that applications can use to communicate with the platform. For example, Microsoft maintains a number of APIs that application providers can use to produce software for the Windows operating system. Apple's mobile operating system has a rich set of APIs that has unleashed a torrent of fascinating applications for the iPhone.[383] So many people are producing applications for Facebook that some observers have speculated Facebook's APIs will displace the Internet suite protocols as the dominant communications platform.[384]

There have been increasing calls to require platform owners to make their APIs open.[385] Strategic denial of access to APIs to certain outside developers was the heart of the antitrust case against Microsoft.[386]

379. *E.g.*, Dave Thaler, *Why Do We Really Want an ID/Locator Split Anyway?*, Keynote Address at the 3rd ACM Workshop on Mobility in the Evolving Internet Architecture (Aug. 22, 2008), http://research.microsoft.com/en-us/um/people/dthaler/dthaler-mobiarch-keynote.pdf.

380. Sri Gundavelli et al., *Proxy Mobile IPv6*, at 4 (Network Working Group Request for Comments 5213, Aug. 2008), *available at* http://tools.ietf.org/pdf/rfc5213.

381. Nitul Dutta et al., *Survey on Mobility Management Protocols for IPv6 Based Network*, 1 Advances in Network & Comm. 1, 15 (2013), http://www.humanpub.org/ANC/ppl/ANC6PPL.pdf.

382. *See* PETERSON & DAVIE, *supra* note 4, at 31.

383. *See* Kevin J. Boudreau & Karim R. Lakhani, *How to Manage Outside Innovation*, MIT SLOAN MGMT. REV., Summer 2009, at 69, 69.

384. *See* Jason Chew, *Is Facebook the New Internet?*, PROACTIVEINVESTORS (Oct. 1, 2011, 9:55 AM), http://www.proactiveinvestors.com/companies/news/19122/is-facebook-the-new-internet--19122.html; Mark Cuban, *Is Facebook the New Internet and How Long Before Microsoft Tries to Buy It?*, BLOG MAVERICK (Apr. 22, 2010), http://blogmaverick.com/2010/04/22/is-facebook-the-new-internet-and-how-soon-before-microsoft-tries-to-buy-it/; Andrew Yee, *Why Facebook Is the New Internet*, EBIZQ (Mar. 6, 2011, 11:55 PM), http://www.ebizq.net/blogs/cloudtalk/2011/03/why_facebook_is_the_new_intern.php.

385. *See, e.g.*, *WhatsApps Reason for Not Opening up API Is Just Not Informed and Short-Sighted*, API EVANGELIST, (Mar. 27, 2015), http://apievangelist.com/2015/03/27/whatsapps-reason-for-not-opening-up-api-is-just-not-informed-and-shortsighted/.

386. *See* Massachusetts v. Microsoft Corp. 373 F.3d 1199, 1216–22, 1240–41 (D.C. Cir. 2004). Note that the district court rejected the states' request that Microsoft be ordered to modularize Windows so

Calls for open access to smartphones such as the iPhone[387] and social networking sites such as Twitter and Facebook[388] have also emerged. Opening APIs would yield many potential benefits of faster innovation and greater flexibility.

At the same time, mandating open interfaces would have a number of drawbacks. For example, Apple has traditionally insisted on closed APIs in order to guarantee that the end-users' experience remained positive. Although it has loosened this policy somewhat with the iPhone, it still reviews all software before making it available through its App Store and insists on a share of the revenue that app developers generate through its product.[389] Moreover, following the same approach as one's competitors is generally bad business strategy; the mere fact that one's competitor had adopted an open platform provides a substantial reason to adopt a closed one.[390] The presence of a popular open platform, such as smartphones based on Google's Android operating system, helps ensure that Apple's iPhone business practices will not harm competition or innovation.

The advent of smartphones has also unleashed a dramatic increase in the heterogeneity of applications running on mobile operating systems. To meet this demand, mobile operating system providers are experimenting with a wide range of new functionalities, many of which are implemented in widely different ways. For example, instead of providing video chat as a separate application, Apple's FaceTime builds that feature into the underlying operating system.[391] Google Wallet goes a step farther, taking a payment-system functionality traditionally provided as an application and building it into the chip.[392] The growing heterogeneity of demand and dynamic nature of this platform make any attempt to mandate open access to any fixed set of APIs inherently problematic.

that outside developers could replace portions of the actual Windows operating system itself. New York v. Microsoft Corp., 224 F. Supp. 2d 76, 162, 251–52 (D.D.C. 2002), *aff'd sub nom.* Massachusetts v. Microsoft Corp. 373 F.3d 1199 (D.C. Cir. 2004).

387. ZITTRAIN, *supra* note 3, at 1–5.

388. The Federal Trade Commission has investigated Twitter's recent moves to limit access to its APIs. *See* Amir Efrati, *Antitrust Regulator Makes Twitter Inquiries*, WALL ST. J., July 1, 2011, at B4, *available at* http://www.wsj.com/articles/SB10001424052702304450604576418184234003812. Facebook previously restricted access to its APIs, but provides more access to counter Google's Open Social Initiative. *See* Brad Stone, *To Counter Google, Facebook Opens Its Code*, N.Y. TIMES BITS BLOG (June 2, 2008, 4:33 PM), http://bits.blogs.nytimes.com/2008/06/02/to-counter-google-facebook-sets-code-free.

389. *See* Jonathan M. Barnett, *The Host's Dilemma: Strategic Forfeiture in Platform Markets for Informational Goods*, 124 HARV. L. REV. 1861, 1920 (2011). Apple reportedly restricted the programming techniques that iPhone application developers could use, only to back off in the face of inquiries by European antitrust regulators. *See* Sascha D. Meinrath et al., *Digital Feudalism: Enclosures and Erasures from Digital Rights Management to the Digital Divide*, 19 COMMLAW CONSPECTUS 423, 461–62 (2011).

390. CARL SHAPIRO & HAL R. VARIAN, INFORMATION RULES: A STRATEGIC GUIDE TO THE NETWORK ECONOMY 23–27, 298 (1999).

391. Open Internet Advisory Committee, *AT&T/FaceTime Case Study*, FED. COMMC'NS COMM'N 1, 1 (Jan. 17, 2013), *available at* https://transition.fcc.gov/cgb/oiac/Mobile-Broadband-FaceTime.pdf.

392. *Forensic Security Analysis of Google Wallet*, NOWSECURE, https://www.nowsecure.com/blog/2011/12/12/forensic-security-analysis-of-google-wallet/ (last visited Sept. 20, 2015).

Even more interesting from the perspective of modularity theory is the lack of coordination. Modularity focuses actors on optimizing their individual interests, which may or may not in the aggregate optimize the performance of the system as a whole.[393] Given that all companies are drawing off a common pool of surplus that they jointly create, they have the incentive to want others to do more.[394] As was the case with GPTs,[395] this can lead to suboptimal investment in the entire modular ecosystem.

The rise and fall of the IBM PC provides a useful example of the importance in the shift. Unlike Apple, who was the early market leader, IBM adopted an open modular architecture for its PC.[396] Initially, this strategy proved spectacularly successful, as the open architecture reduced the magnitude of IBM's upfront investment, allowed it to market its PC in only fifteen months, and unleashed a wide range of innovation within its platform.[397] Fairly quickly, however, IBM's strategy began to evince a number of downsides. The same modularity that unleashed parallel implementation permitted IBM's former partners to capture an increasing amount of the value and eventually to ally with other computer manufacturers that would ultimately displace IBM.[398] The result was a lack of innovation in the platform. Similar complaints have been levied at Android.[399] This contrasts starkly with Apple, which continues to both manufacture and innovate in computers and smartphones.

Companies must think carefully about which parts of the value chain to continue to develop internally.[400] Most notably, modularity allows imitation as well as innovation.[401] The lack of coordination can lead to significant economic harms.

D. Future Internet Architecture Proposal

Policymakers and commentators frequently assert that the Internet's architecture has played an instrumental role in its success and argue that that architecture must be preserved.[402] Interestingly, this assertion is contradicted by a steady drumbeat of articles in the engineering literature noting the functions that the Internet does not perform well.[403] These

393. *See* Chesbrough & Teece, *supra* note 183, at 66.
394. *Id.* at 67–69.
395. *See supra* notes 197–200 and accompanying text.
396. *See* Chesbrough & Teece, *supra* note 183, at 68–69
397. *See id.*
398. BALDWIN & CLARK, *supra* note 5, at 267–68; Chesbrough & Teece, *supra* note 183, at 69–70.
399. *See, e.g.*, Nirave Gondhia, *Android Innovation: Sony and HTC Risk Falling Behind Even More*, ANDROID AUTHORITY (Apr. 21, 2015) ("Is innovation dead in Android?").
400. Chesbrough & Teece, *supra* note 183, at 71.
401. Ethiraj et al., *supra* note 29, at 941.
402. *See supra* notes 3–7 and accompanying text.
403. *See, e.g.*, Jon Crowcroft, *Net Neutrality: The Technical Side of the Debate*, COMPUTER COMM. REV., Jan. 2007, at 49, 51; Handley, *supra* note 317; Raj Jain, *Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation*, PROC. MIL. COMM. CONF. (2006), http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4086425; Thrasyvoulos Spyropoulos et al., *Future Internet: Fundamentals and Measurement*, ACM SIGCOMM COMPUTER COMM. REV., Apr. 2007, at

include security, mobility, quality of service, mass media distribution (multicasting), and support for multiple connections to the same location (multihoming).[404] Although these were not important when the Internet first emerged as a mass-market phenomenon during the mid-1990s, they have now become critical.[405]

These analyses suggest that TCP/IP is a forty-year-old technology that was designed for a very different context and is being asked to do more than its designers ever imagined. Many engineering scholars complain that the Internet has become ossified and impervious to significant architectural change.[406] Others are pursuing a variety of "clean slate" initiatives examining how the Internet's architecture might better meet these needs if it were designed from scratch today.[407] Government agencies in the U.S., Europe, and Asia have all launched programs to explore alternative architectures better suited to supporting the new demands that end users are placing on the network.[408] For example, the National Science Foundation's Future Internet Architecture program asked all participating projects to "mak[e] security an integral part of the architecture" and "build security into the design."[409] For example, both the NEBULA and Named Data Networking projects required that each packet be cryptographically signed in the network layer, effectively making each packet self-authenticating.[410]

101, 101; Sixto Ortiz, Jr., *Internet Researchers Look to Wipe the Slate Clean*, COMPUTER, Jan. 2008, at 12.

404. Crowcroft, *supra* note 403, at 54.

405. *See* YOO, *supra* note 329, at 3–4.

406. For citations to eight representative examples, see Paul Laskowski & John Chuang, A Leap of Faith? From Large-Scale Testbed to the Global Internet 1, 2 (Sept. 27, 2009) (unpublished manuscript presented at 37th Telecommunications Policy Research Conference) (on file with author). *See also* Olivier H. Martin, State of the Internet & Challenges Ahead 29 (2007) (unpublished manuscript), *available at* http://www.ictconsulting.ch/reports/NEC2007-OHMartin.doc (noting that "there appears to be a wide consensus about the fact that the Internet has stalled or ossified").

407. *See, e.g.*, Steven Bellovin et al., Report, *A Clean Slate Design for the Next-Generation Secure Internet*, NSF Global Environment for Network Innovations Workshop (2005), *available at* http://sparrow.ece.cmu.edu/group/pub/bellovin_clark_perrig_song_nextGenInternet.pdf; Jon Crowcroft & Peter Key, *Report from the Clean Slate Network Research Post-SIGCOMM 2006 Workshop*, ACM SIGCOMM COMPUTER COMM. REV., Jan. 2007, at 75, 75; Anja Feldmann, *Internet Clean-Slate Design: What and Why?*, ACM SIGCOMM COMPUTER COMM. REV., July 2007, at 59; Ortiz, *supra* note 403; Thomas-Rolf Banniza et al., *A European Approach to a Clean Slate Design for the Future Internet*, 14 BELL LABS TECHNICAL J. 5 (2009); International Center for Advanced Internet Research (ICAIR), *Grand Challenges in Advanced Networking Research*, (May 10, 2010), http://www.icair.org/mission/grand-challenges.html; STANFORD UNIVERSITY CLEAN SLATE, http://cleanslate.stanford.edu/ (last visited Mar. 14, 2011). For a more cautionary assessment, see Constantine Dovrolis, *What Would Darwin Think About Clean-Slate Architectures?*, ACM SIGCOMM COMPUTER COMM. REV., Jan. 2008, at 29.

408. For an overview, see Jianli Pan et al., *A Survey of the Research on Future Internet Architectures*, IEEE COMM. MAG., July 21, 2011, at 26, 28–34. *See also* David Clark et al., *New Arch: Future Generation Internet Architecture*, Final Report (2003), *available at* http://www.isi.edu/newarch/iDOCS/final.finalreport.pdf.

409. Darleen Fisher, *A Look Behind the Future Internet Architectures Efforts*, ACM SIGCOMM COMPUTER COMM. REV., July 2014, at 46, 46, 47.

410. Tom Anderson et al., *A Brief Overview of the NEBULA Future Internet Architecture*, ACM SIGCOMM COMPUTER COMM. REV., July 2014, at 81, 82; Lixia Zhang et al., *Named Data Networking*, ACM SIGCOMM COMPUTER COMM. REV., July 2014, at 66, 68.

Modularity theory emphasizes that all designs are the products of their times rather than the result of some preconceived vision of the ideal architecture.[411] In addition, changes in needs and technological interdependencies can lead to remodularization. Although they are rare, remodularizations do occur.[412] For example, although Windows once represented a form of middleware running on a DOS operating system, it has now displaced DOS as the primary operating system platform. Similarly, although packet transmission began as an application riding on a voice network, packet transmission has now become the relevant platform, and voice has become an application riding on a data network rather than vice versa.[413]

The costs of remodularization dictate that any such proposals should be approached with considerable caution. Modularity theory does offer some guidance as to the types of circumstances under which remodularization may be justified, including changes in the nature of technological interdependencies, increases in the degree of technological opportunity, and the growing heterogeneity of demand. Moreover, the inherent costs of remodularization raise the real possibility that an architecture may become locked into a modularization that is suboptimal. As the literature exploring how the presence of a large installed base can dampen innovation demonstrates, the real risk may be too little change rather than too much.[414]

VII. CONCLUSION

The benefits of modularity are frequently invoked as a justification for the imposition of regulations designed to preserve the status quo. There is no question that existing modular architecture can reduce complexity, speed innovation, facilitate the division of labor, and promote flexibility. At the same time, modularity carries with it certain drawbacks, a fact that is rarely acknowledged or understood. The balance of countervailing forces determines whether a particular modular architecture is beneficial as well as how it should be implemented in terms of the number of modules, the division of tasks, and the construction of the necessary interfaces. Modularity theory underscores that this tradeoff cannot be determined *a priori*. Indeed, some scholars argue against a "presumption of 'promodularity' bias."[415]

Policymakers should therefore avoid treating the existing network architecture as if it were a natural construct that must be preserved at all

411. ABBATE, *supra* note 296, at 51 ("The ARPANET's builders did not start out with a specific plan for how functions would be divided up among layers or how the interfaces and protocols would work. Rather, a layered model evolved as the ARPANET developed.").

412. RFC 817, *supra* note 159, at 25.

413. Dave Clark et al., *Overlay Networks and the Future of the Internet*, 63 COMM. & STRATEGIES 1, 1–2 (2006).

414. *See, e.g.*, Joseph Farrell & Garth Saloner, *Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation*, 76 AM. ECON. REV. 940, 941 (1986).

415. Ethiraj & Levinthal, *supra* note 10, at 172.

costs. Instead, modularity's contingent quality underscores the importance of developing heuristics for determining how and when a modular architecture should evolve.

Even determining that modularity is the preferred policy is not sufficient by itself to justify regulatory intervention. Network effects already provide strong incentives toward compatibility. Governmental action is necessary only if systems refuse to adopt modularity when it would be beneficial to do so. Moreover, the policy question should not be posited as a choice between modularity and nonmodularity. The optimal solution may be a hybrid, such as the one created by the coexistence of the open Google Android platform and the closed platform of the Apple iPhone. Purely as a matter of business strategy, the fact that one's competitor had adopted a modular strategy makes it beneficial to pursue the opposite course. Moreover, consumers benefit from having more choices, and the presence of a major open platform protects against anticompetitive harms.

On a broader level, the increasing frequency with which architectural concepts such as modularity are invoked during policy debates makes understanding their conceptual underpinnings all the more important. Otherwise, the opacity of the engineering concepts will obscure debates rather than provide insights, while at the same time permitting advocates to introduce normative assumptions without appearing to do so.[416]

416. *See* Marjory S. Blumenthal, *End-to-End and Subsequent Paradigms*, 2002 L. REV. MICH. ST. U. DET. C.L. 709, 710 (noting that "[a]lthough the embrace of engineering principles . . . appears to impart a legitimacy to certain kinds of advocacy, that advocacy reaches beyond the engineering to the ideology long associated with the Internet"); Tarleton Gillespie, *Engineering a Principle: "End-to-End" in the Design of the Internet*, 36 SOC. STUD. SCI. 427, 450–51 (2006) (noting that "[b]y highlighting certain features of the technology and obscuring others," framing policy debates in terms of architectural concepts has "the power to frame the entire technology in terms of an assumed set of priorities" in a way that is "often cloaked in a discourse that performs its political neutrality," but actually "embody political standpoints, even as they obscure them").