Nature creates complex systems from simple components. For example, a vast variety of proteins are

linear chains assembled from 21 amino acids, the building blocks of proteins. Twenty amino acids are

naturally incorporated into polypeptides and are encoded in the genetic code.

Imitating nature, man-made systems are assembled from subassemblies; in turn, a subassembly is

made from several modules, each of which could consist of submodules, and so on. Composability

has natural bounds imposed by the laws of physics, as we saw when discussing heat dissipation of

solid-state devices. As the number of components increases, the complexity of a system also increases

for the reasons discussed in Section 10.1.

The limits of composability can be reached because new physical phenomena could affect the system

when the physical size of the individual components changes. A recent paper with the suggestive

title, “When Every Atom Counts” [245], discusses the fact that even the most modern solidstate

fabrication facilities cannot produce chips with consistent properties. The percentage of defective

or substandard chips has been constantly increasing as the components have become smaller and

smaller.

The lack of consistency in the manufacturing process of solid-state devices is attributed to the increasingly

small size of the physical components of a chip. This problem is identified by the International

Technology Roadmap for Semiconductors as “a red brick,” a problem without a clear solution – a wall

that could prevent further progress. Chip consistency is no longer feasible because the transistors and

“wires” on a chip are so small that random differences in the placement of an atom can have a devastating

effect, e.g., can increase power consumption by an order of magnitude and slow the chip by as much

as 30%.

As the features become smaller and smaller, the range of the threshold voltage – the voltage needed

to turn a transistor on and off – has been widening, and many transistors have this threshold voltage

at or near zero. Thus, they cannot operate as switches. Although the range for 28 nm technology was

approximately between +0.01 and +0.4 V, the range for 20 nm technology is between −0.05 and

+0.45 V, and the range becomes even wider, from −0.18 to +0.55, for 14 nm technology.

There are physical bounds on the composition of analog systems; noise accumulation, heat dissipation,

cross-talk, the interference of signals on multiple communication channels, and several other

factors limit the number of components of an analog system. Digital systems have more distant bounds,

but composability is still limited by physical laws.

There are virtually no bounds on composition of digital computing and communication systems

controlled by software. The Internet is a network of networks and a prime example of composability

with distant bounds. Computer clouds are another example; a cloud is composed of a very large number

of servers and interconnects with each server made up of multiple processors and each processor made

up of multiple cores. Software is the ingredient that pushes the composability bounds and liberates

computer and communication systems from the limits imposed by physical laws.

In the physical world the laws that are valid at one scale break down at a different scale, e.g., the

laws of classical mechanics are replaced at atomic and subatomic scales by quantum mechanics. Thus,

we should not be surprised that scale really matters in the design of computing and communication

systems. Indeed, architectures, algorithms, and policies that work well for systems with a small number

of components very seldom scale up.

For example, many computer clusters have a front-end that acts as the nerve center of a system,

manages communication with the outside world, monitors the entire system, and supports system

administration and software maintenance. A computer cloud has multiple nerve centers of this kind, and

new algorithms to support collaboration among these centers must be developed. Scheduling algorithms

that work well within the confines of a single system cannot be extended to collections of autonomous systems where each system manages local resources; in this case, as in the previous example, entities

must collaborate with one another, and this requires communication and consensus.

Another manifestation of this phenomenon is in the vulnerabilities of large-scale distributed systems.

The implementation of Google BigTable revealed that many distributed protocols designed to protect

against network partitions and fail-stop are unable to cope with failures due to scale [73]. Memory and

network corruption, extended and asymmetric network partitions, systems that fail to respond, and large

clock skews occur with increased frequency in a large-scale system and interact with one another in a

manner that greatly affects overall system availability.

Scaling has other dimensions than just the number of components. The space plays an important

role; the communication latency is small when the component systems are clustered together within a

small area and allows us to implement efficient algorithms for global decision making, e.g., consensus

algorithms. When, for the reasons discussed in Section 1.6, the data centers of a cloud provider are

distributed over a large geographic area, transactional database systems are of little use for most online

transaction-oriented systems, and a new type of data store must be introduced in the computational

ecosystem, as shown in Section 8.8.

Societal scaling means that a service is used by a very large segment of the population and/or is

a critical element of the infrastructure. There is no better example to illustrate how societal scaling

affects system complexity than communication supported by the Internet. The infrastructure supporting

the service must be highly available. A consequence of redundancy and of the measures to maintain

consistency is increased system complexity.

At the same time, the popularity of the service demands simple and intuitive means to access the

infrastructure. Again, system complexity increases due to the need to hide the intricate mechanisms

from a layperson who has little understanding of the technology. Another consequence of the desire

to design wireless devices that operate efficiently in terms of power consumption and present the user

with a simple interface and few choices and satisfy a host of other popular attributes is an increased

vulnerability to attacks from a few individuals who can discover and exploit system weaknesses. Few

smartphone and tablet users understand the security risks of wireless communication.