Appendix A: More about "What is Cyberinfrastructure?"

Cyberinfrastructure makes applications dramatically easier to develop and deploy, thus expanding the feasible scope of applications possible within budget and organizational constraints, and shifting the scientist's and engineer's effort away from information technology development and concentrating it on scientific and engineering research. Cyberinfrastructure also increases efficiency, quality, and reliability by capturing commonalities among application needs, and facilitates the efficient sharing of equipment and services.

Historically, infrastructure was viewed largely as raw resources like compute cycles or communication bandwidth. As illustrated by many activities in the current PACI centers and by the recent NSF middleware program, the scope of infrastructure is expanding dramatically beyond this narrow definition. For purposes of the ACP, infrastructure will comprise of a diverse set of technologies, facilities, and services and intangibles like design processes and best practices and shared knowledge. A major technological component is software that participates directly in applications and software tools that aid in the development and management of applications. A critical non-technological element is people and organizations that develop and maintain software, operate equipment and software as it is used, and directly assist end-users in the development and use of applications. The ACP seeks to bring about dramatic and beneficial change in the conduct of science and engineering research. Applications will greatly expand their role and become increasingly integral to the conduct of science and engineering research.

Cyberinfrastructure, as it captures commonalities of need across applications, incorporates more and more capabilities integral to the methodologies and processes of science and engineering research. Cyberinfrastructure will become as fundamental and important as an enabler for the enterprise as laboratories and instrumentation, as fundamental as classroom instruction, and as fundamental as the system of conferences and journals for dissemination of research outcomes. Through cyberinfrastructure we strongly influence the conduct of science and engineering research (and ultimately engineering development) in the coming decades.

Technologists are naturally the first to embed leading-edge technologies integrally with their research. The Internet—an inspirational example of this—was a new infrastructure defined initially with the narrow purpose of enabling new research in distributed systems, but which has now deeply impacted all research disciplines. The ACP seeks to replicate this type of dramatic change across a wide spectrum of disciplines and a wide spectrum of applications.

The ACP emphasizes infrastructure and applications. The overriding goals of infrastructure are several-fold:

More applications. Dramatically reduce the effort and the required expertise required to develop, deploy, and operate new distributed applications, encouraging more extensive development and use of such applications.

More capabilities. Provide facilities and supporting services that allow the community to do things not feasible otherwise.

More efficiency. Expand what can be accomplished for a fixed budget through sharing, reuse, and reduced duplication of both effort and facilities.

Reuse and multiple-use of designs. Infrastructure tries to capture commonalities across a range of applications. Functionalities and capabilities thus captured (manifested typically in a well-maintained software distribution) can be subsumed into many applications, reducing development time and effort and increasing quality.

Spread of best practices. Infrastructure can be a way to promulgate the best ideas, leveraging them in multiple places.

Achieving interoperability. Infrastructure offers a reference point for mediating the interaction among applications, defining common interfaces and information representations. The alternative of asking applications to interact directly with one another results in a combinatorial explosion of mutual dependencies, creating a house of cards that eventually falls of its own weight.

Tools. Infrastructure provides a set of software tools that make it easier to develop applications.

Services. Infrastructure provides, as an alternative to software that can be 'designed into' an application, services that can be invoked over the network by applications. When this approach is adopted, responsibility for the installation and administration of software and supporting equipment (and more generally provisioning and operations, as described below) is shifted to a service provider, where an aggregation of expertise and experience increases efficiency and effectiveness.

Shared facilities. Infrastructure allows the sharing of common facilities and equipment and instrumentation. This can be more efficient, due to statistical multiplexing[1] and as a way to reduce expensive duplication. Examples include sharing an expensive right-of-way for fiber optic cables or the sharing of a high-performance supercomputer with massive memory and input-output performance.

Assistance and expertise. Infrastructure provides direct assistance to end-users in making use of the available software, tools, and services, and does this efficiently and effectively through an experienced and shared pool of expertise.

Thus, it is critical to think of infrastructure as having several foundations:

Technological artifacts. These human-constructed artifacts include facilities (computers, mass storage, networks, etc.) and software. These artifacts sometimes provide services, and sometimes they are simply available to be 'designed into' applications.

Technological services. Various capabilities are provided as services available over the network rather than as software artifacts to be deployed and operated locally to the end-user.

Services from people and organizations. These include everybody who is providing a shared pool of expertise leveraged by the entire scientific and engineering research community to develop and operate the technological artifacts and provide advice and assistance to end users in making use of them.

[1] Statistical multiplexing is an efficiency advantage arising from sharing a set of work units in a single higher-performance facility rather than splitting them over multiple resources. Examples of resources are a communication link and a processor, where examples of work units are packet transmission or processing tasks. It is important to separate two performance factors: *throughput* (work per unit time) and *delay* (time elapsing from work request until that work is completed). For a fixed resource, average delay increases with average throughput (this is called *congestion*). The efficiency arises because for the same average delay, a larger shared resource can operate at higher average throughput. Alternatively, for the same average throughput, a shared resource can reduce the average delay. Thus, information processing and communication resources display increasing returns to scale even beyond any direct unit cost advantages from higher performance.