Genome Informatics, the *Sine Qua Non* of Genomic Research

Robert J. Robbins

Johns Hopkins University
rrobbins@gdb.org

&

Department of Energy
robbins@er.doe.gov

Key Issues

Informatics is Essential for HGP:

- size of problem
- complexity of problem
- Moore's Law to the rescue

Federation is a Requirement:

- diversity is a given
- domain cannot be bounded

Guidelines for Future:

- componentry
- anonymous interoperability
- value-additivity
- scalable systems
  - ✓ technical
  - ✓ social

Goals of the Genome Project

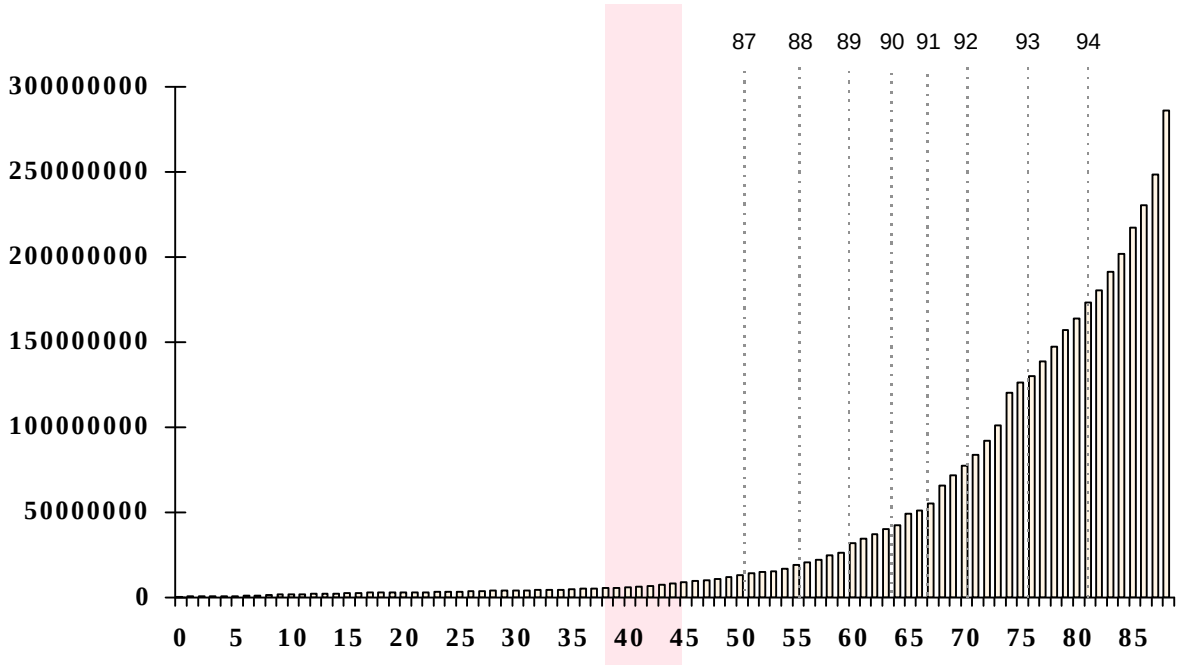
Biological Perspective

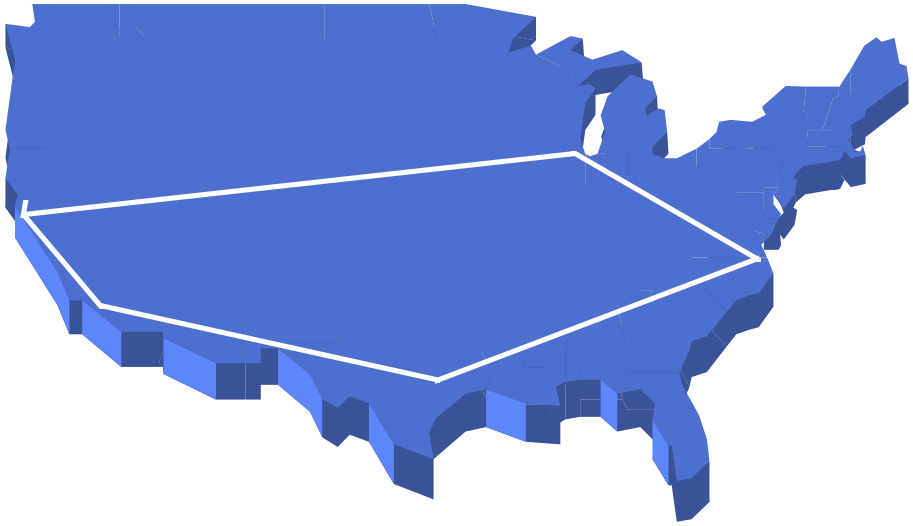
Official Goals:

- Construct a high-resolution genetic map of the human genome.
- Produce a variety of physical maps of all human chromosomes and of the DNA for selected model organisms.
- Determine the complete complete sequence of human DNA and of the DNA of selected model organisms.
- Develop capabilities for collecting, storing, distributing, and analyzing the data produced
- Create appropriate technologies necessary to achieve these objectives.

Size of the Problem

Growth of Sequence Data





Goals of the Genome Project
(short form)

Actual Goals:

- sequence genomes
- map genomes

Implicit Goals:

- understand genomes

Goals of the Genome Project
(short form: restated)

Sequence a Genome

- equivalent to obtaining an image of a mass-storage device

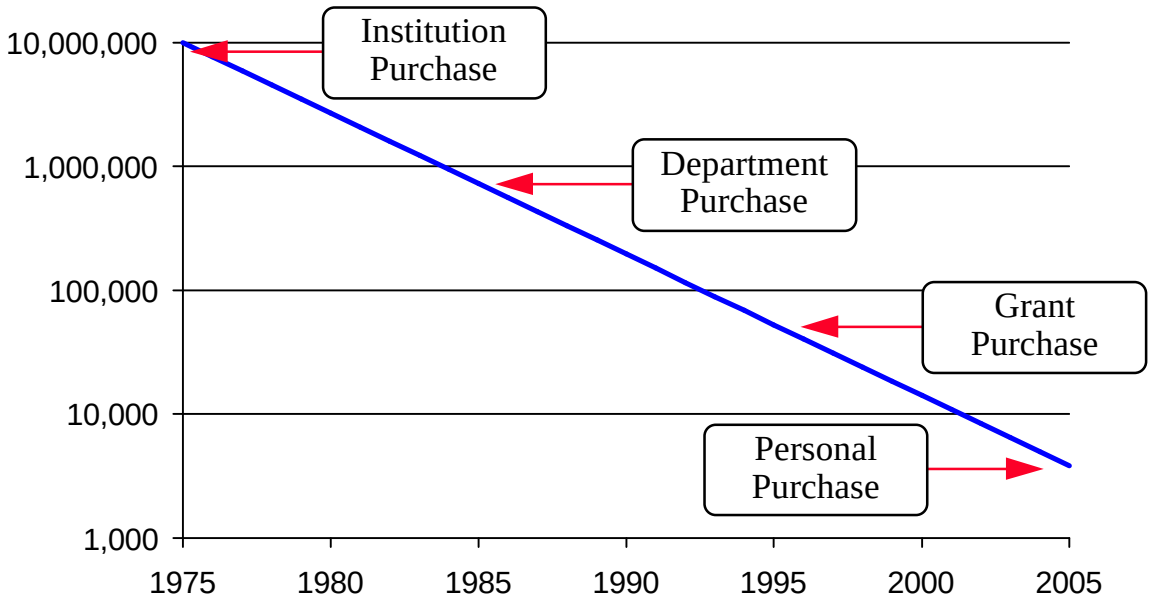
Map a Genome

- equivalent to developing a file-allocation table for the mass-storage device

Understand a Genome

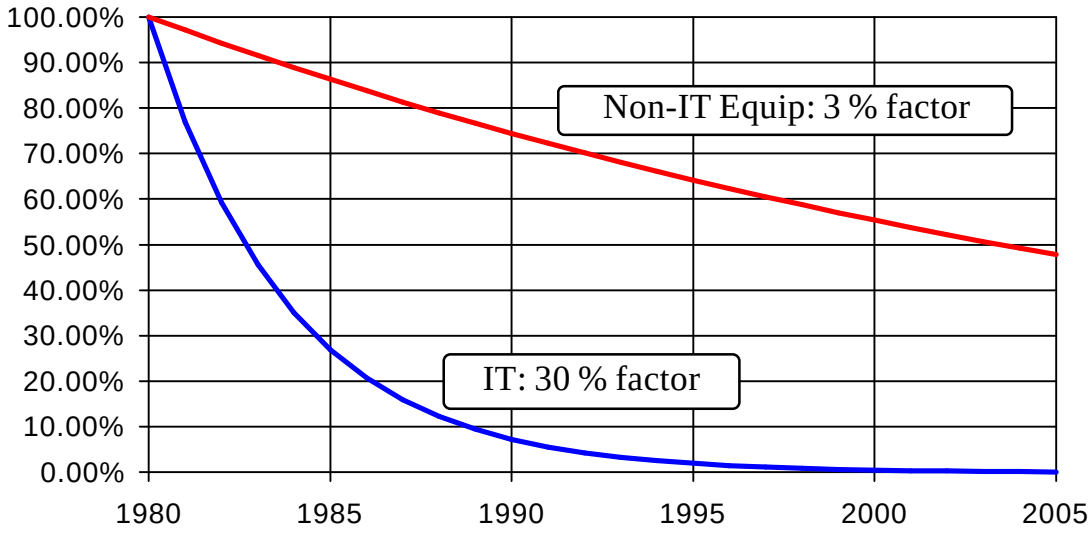
- equivalent to reverse engineering the files on the mass-storage device all the way back to design and maintenance specifications

Moore's Law
(*To the Rescue*)



Price of equivalent systems

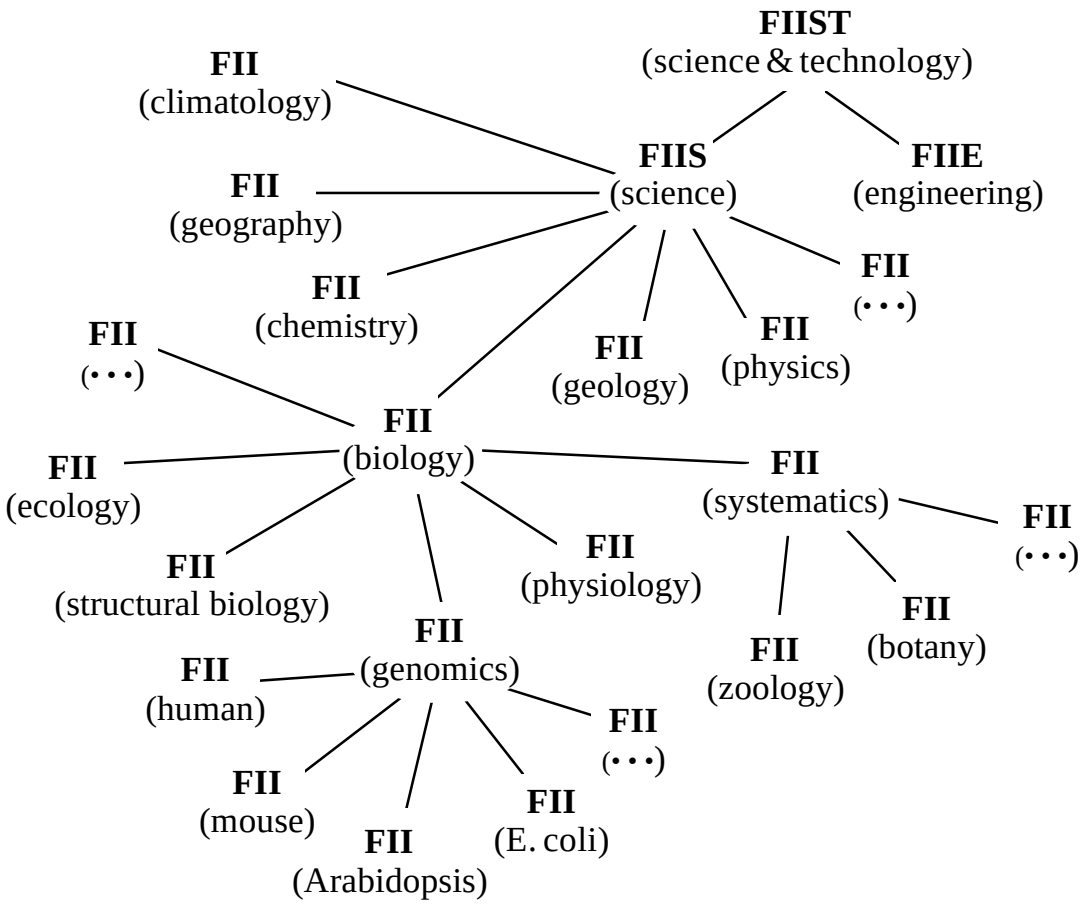
Moore's Law
(*To the Rescue*)



Capital Cost/Unit of Labor

Need for Federation

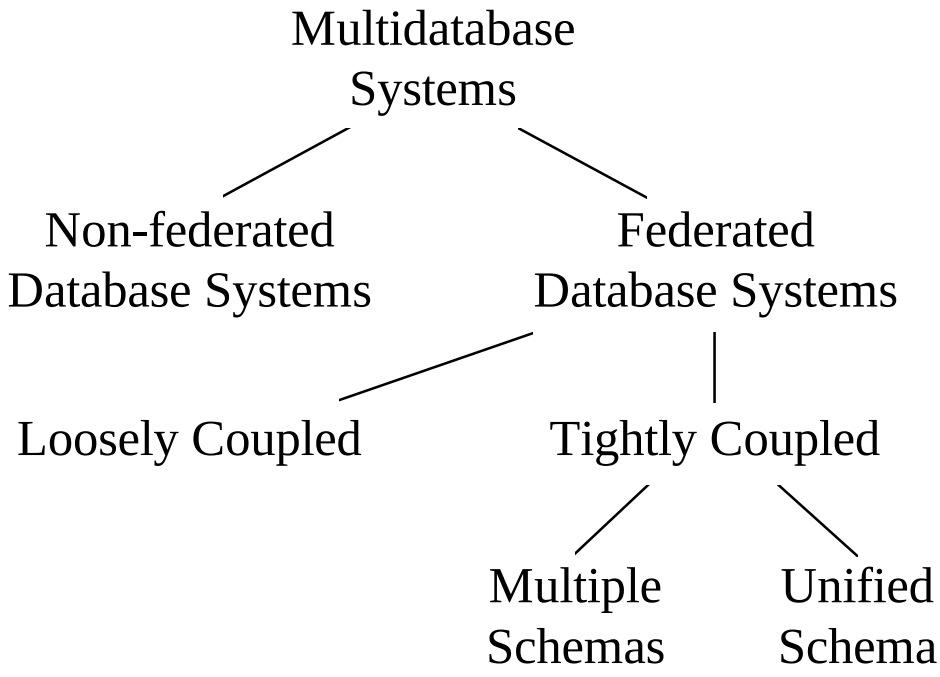
We must begin to think of the computational infrastructure of genome research as a federated information infrastructure of interlocking pieces, including both data resources and analytical tools.



But building federated database management systems is considered an unsolved research problem in computer science.

The solution may to think in terms of a federation of database publishing systems...

Federated Information Systems



Tightly Coupled: single organizational entity overseeing information resources relevant to genome research

•
•
•

adoption of common DBMSs at participating sites

shared data model across participating sites

common semantics for data publishing

Loosely Coupled: common syntax for data publishing

Federation as a Continuum

Impediments to Federation

Technical

- Integrating distributed, heterogeneous databases is not easy.

Conceptual

- Semantic mismatches exist among databases.

Sociological

- Local incentives encourage competition, not cooperation among database providers.
- Few incentives encourage intellectual participation by research community.

Federation-Ready Systems

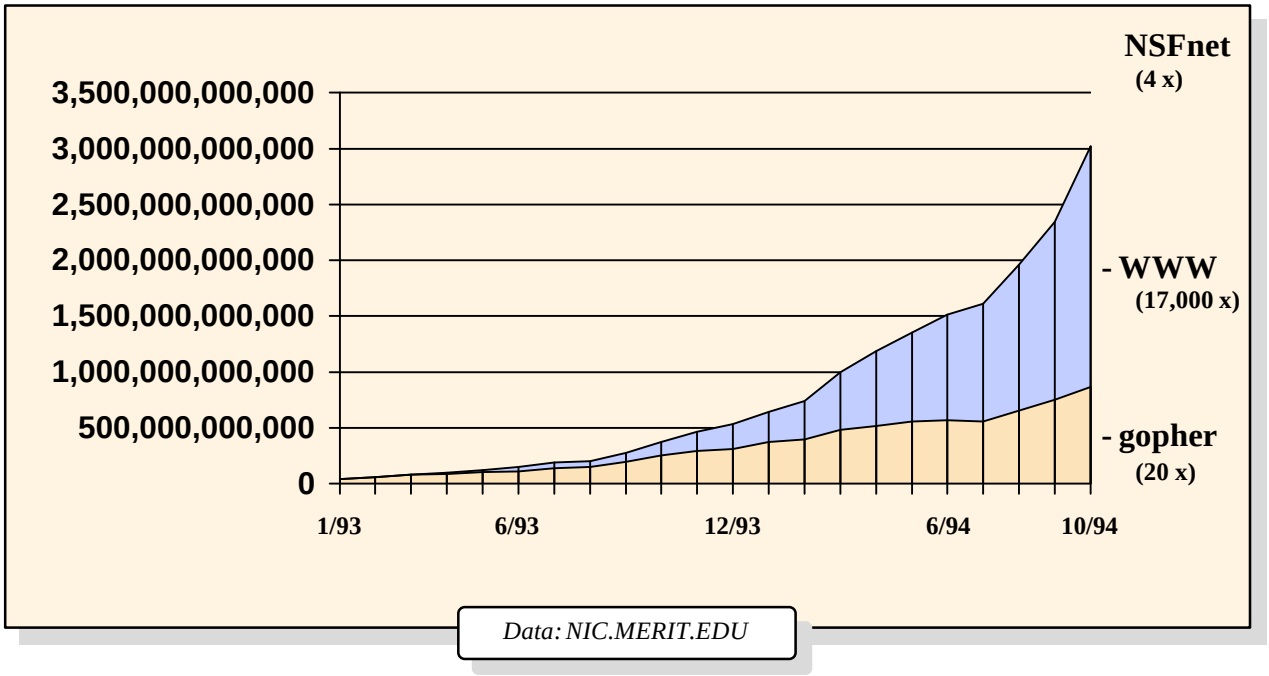
Requirements:

- technical interoperability
- semantic interoperability
- social interoperability

Guiding Principles:

- componentry
- anonymous interoperability
- value additivity
- scalable systems
  - ✓ technical
  - ✓ social

New Technologies



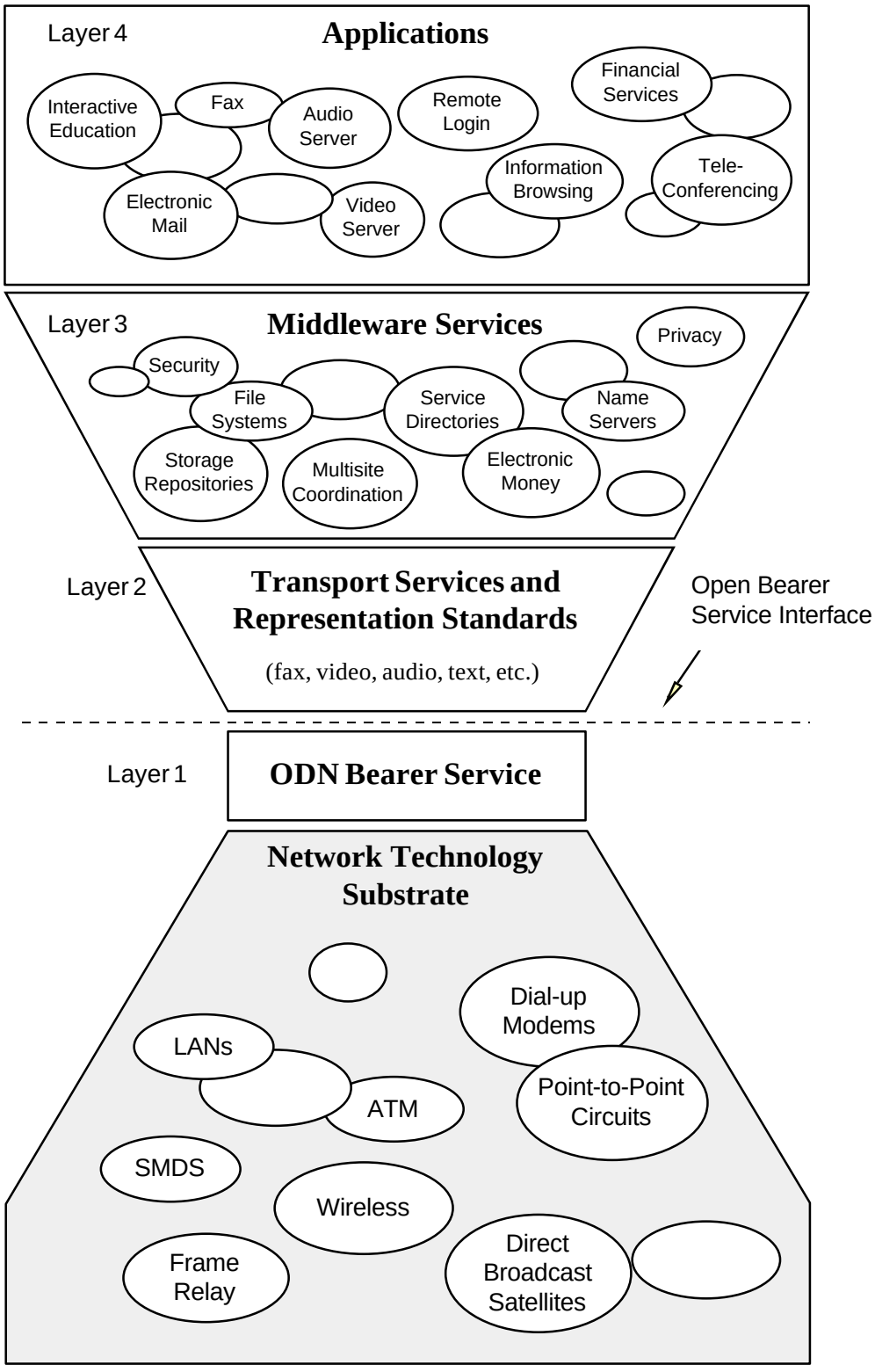
Data: NIC.MERIT.EDU

The run-away success of the World-Wide Web points to a solution, but the Web as presently constituted is not the answer.

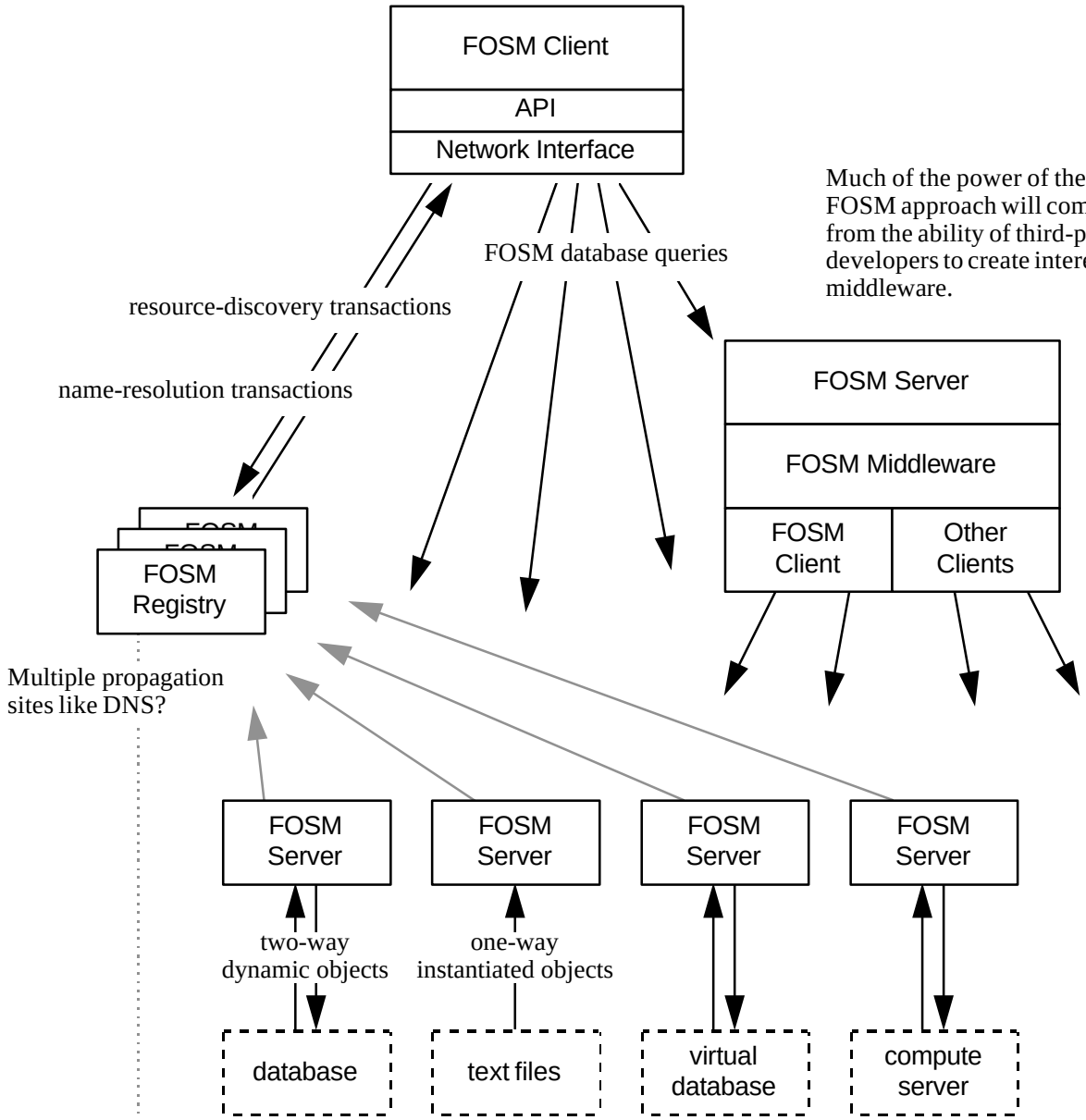
Needed is a Web-like system capable of delivering structured data, with its semantic structure intact, to a client capable of performing basic structured data manipulations.

Also needed is essential technical and social infrastructure to support value-adding activities by anonymous third-party developers and to support information classification, discovery, and filtering.

Open Data Network Model



FOSM Reference Architecture



FOSM servers should be able to provide different standard "prunings" of their objects. Thus, FOSM naming conventions must support versioning.

Registry of FOSM servers, FOSM objects (& versions & prunings), FOSM links, FOSM subfederations, FOSM editorial records, FOSM methods, FOSM names, FOSM classifications, etc.

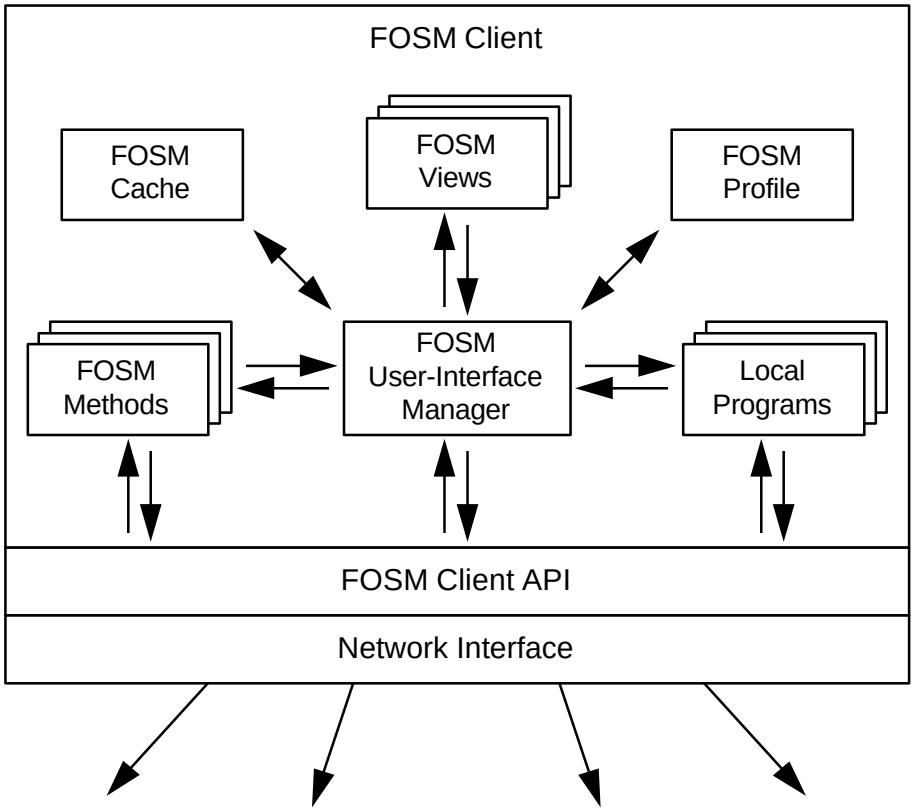
For a more detailed discussion of the FOSM reference architecture, see Robbins. R. J., 1995, An information infrastructure for the human genome project, *IEEE Engineering in Medicine and Biology*, in press.

FOSM Clients

To build a FOSM interface, the client must first query a server to obtain necessary type and format information. This, and other FOSM metadata, should be storable in a local cache. The size of the cache should be user-settable. Normally, the cache would be first-in, first-out, but the user should be able to set caching priorities, perhaps even to specify certain cached elements that are never to be flushed.

FOSM views will allow users to create local views on FOSM objects or to build virtual FOSM objects.

A FOSM profile system will allow users to customize the behavior both of the local client and of remote servers without requiring servers to maintain registries of users and preferences.



FOSM methods are local, hardware-specific software packages that are invoked to "view" objects obtained from FOSM servers. For example, one of the standard local methods would display and operate HTML documents; another would build, display, and operate query interfaces for FOSM objects.

The FOSM User-Interface Manager (UIM) would probably be some kind of script interpreter, possibly a generic script interpreter so that more than one scripting language could be used.

The FOSM API should allow easy development of local programs that can interact directly with the client API, without requiring assistance from the user-interface manager. This would facilitate the development of third-party bulk-data-transaction modules for special markets: DNA sequences, finance, etc.