Technical Components of the GTL Knowledgebase

Database Architecture and Infrastructure

Rapidly advancing available and emerging technologies for computation, data storage, and communications promise a wealth of aggressive and high-performance options for establishing the GTL Knowledgebase (GKB). To take full advantage of these opportunities, GKB technical requirements and operational needs must be well defined. In addition, decisions on system architecture and infrastructure will be influenced substantially by institutional requirements manifest in the GKB governance and management model and by the resultant roles of data providers, integrators, and users. Moreover, resources for creating, maintaining, and using the knowledgebase will arise from various elements of GTL research initiatives and from computing and informatics programs and institutions. These research and computing programs in turn will influence the choices and locations of hardware and software efforts and assets.

Building a successful knowledgebase will require evaluating and implementing numerous design choices for system architecture that impact models for GKB development, scalability, and GTL-relevant use cases. Investigation of these choices and requirements has revealed several viable options for GKB architecture (each with distinct strengths and weaknesses) that could meet at least part of the data needs of the GTL community. However, GKB architectural design ultimately must satisfy the full range of GTL researchers' data requirements and provide a foundational software platform for cost-effective software development and operations. In these capacities, system architecture is fundamental to the GTL Knowledgebase project.

To meet various user and operational requirements, optimal GKB architecture most likely would be a hybrid design combining elements of several basic architectural options. This solution could link, for example, a central data model (perhaps supported at multiple sites) with more heterogeneous and distributed data and analysis support accessible through a Web services model. Existing, proven architectural designs set precedents for the success of such a venture.

Since the early development of genome databases, system architectures have undergone revolutionary changes that the GTL Knowledgebase should exploit. Major features of these changes follow:

- *Unifying algorithms and data by integrating programming languages with a database system.* This creates an extensible object-relational system in which nonprocedural relational operators manipulate object sets.
- *Integrating Web services with a core database management system (DBMS).* Such integration has significant implications for how applications are structured, with DBMSs functioning more like object containers. Online analytic processing is now integrated into most DBMSs, and service-oriented architecture (SOA) models based on Web services can be leveraged successfully in this approach.
- *Progressively incorporating new services into DBMSs.* More of these systems now have frameworks for data mining, machine-learning algorithms, decision trees, visualization, clustering, time-series analyses, and modeling—with flexibility for adding novel, integrated analytical tools.
- *Increasingly using distributed and parallel approaches based on federated or clustered architectures.* Clustered architectures, in particular, have the advantage of removing

query bottlenecks by providing shared access to numerous data storage units from multiple clients.

- *Advancing object-oriented and deductive databases.* These databases enable many concepts to be integrated into mainstream object-relational systems.

GTL Knowledgebase developers must assess the value of a spectrum of choices in designing GKB architecture. The following sections outline some of these choices and describe how they might provide important capabilities for the GTL program. Since there are significant tradeoffs among various architectural options, establishing specific requirements with which to evaluate these options is critical. Typical architecture considerations include cost, scalability, flexibility, security, query performance and reliability, and management and sociological factors.

Architecture Design Driven by User Requirements

Findings

- The envisioned GTL Knowledgebase would support existing scientific communities (e.g., biologists as well as computational and information scientists) and would help foster the growth of a much larger research community: computational biologists.
- GKB architecture thus must address a wide range of user needs—an effort requiring constant interactions between the knowledgebase design team and scientists from all associated fields. In the past, enabling computational groups to apply analytical methods to biological data has been cumbersome because data were stored in multiple locations, saved in various formats using nonoverlapping systems of identifiers, and had widely varying levels of quality.

Recommendation

- The knowledgebase should enable computational groups to easily apply new algorithms on diverse platforms to GKB data.

Basic Architecture Choices

Designers of the GKB architecture must make fundamental choices regarding where and how data for the GTL program would be collected, organized, managed, and maintained. These capabilities conceivably could be focused at a single, central site or through coordination at multiple locations. A centralized approach generally has significant limitations because it requires building a huge infrastructure at one site. Moreover, the chosen site must employ—in one location—staff having all the biological expertise necessary to organize, represent, and curate every aspect of biological data and information. Such centralized groups also have to keep track of customer requirements and avoid becoming insular.

Finding

- Given these limitations, a distributed approach to knowledgebase creation is highly desirable.

One option for GTL Knowledgebase development would involve integrated research teams such as the *Shewanella* Federation. In this approach—called a federated system—several sites specialize in a biological domain, but all members share a single data model or global schema (e.g., a homogeneous distributed system; see Fig. 4.1. Federated Database System, this page). This enables efficient data queries and retrieval across multiple sites without the need to translate formats across different schemas. Furthermore, this approach allows each site to establish expert curators in specialized areas of biology such as proteomics or metabolic pathways. Individual sites also can use the level of resources or even parallelization each needs to support query loads. Within this framework, the efficiency of knowledgebase development is generally good because the system is divided into smaller, more manageable parts (partitions). In this model, data queries can be highly efficient within a single site's biology domain but could be somewhat slower across multiple sites. Moreover, redundancy and fault tolerance are possible within a federated database system.

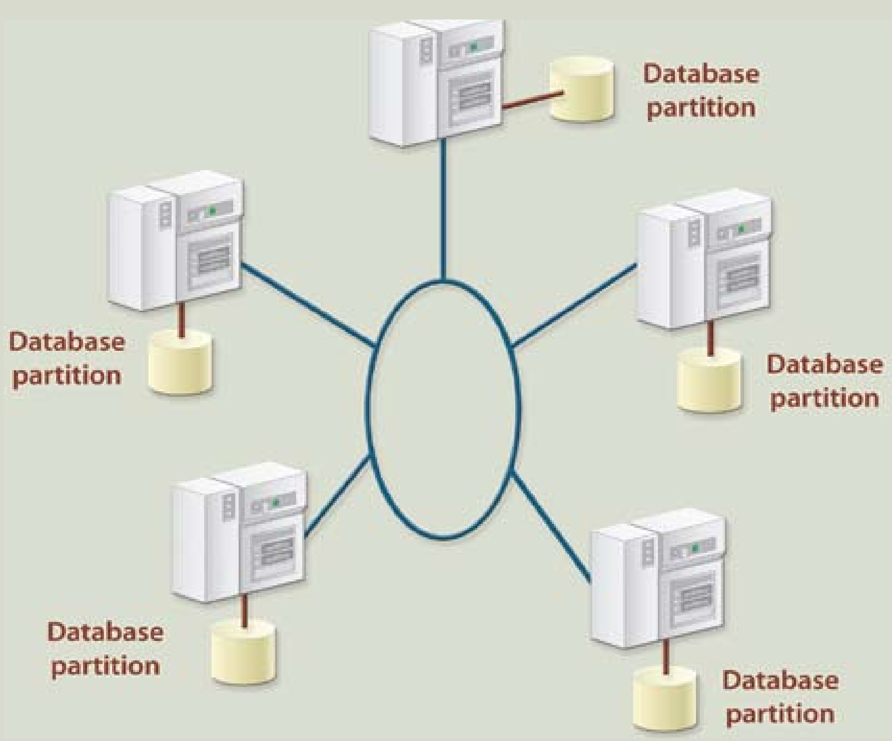

Fig. 4.1. Federated Database System. In this system, all sites share a common data model but "own" a particular part of the biology domain (i.e., horizontal partitioning) and have separate data. Queries can be directed against one or multiple sites as needed using the network.

A somewhat different approach to a distributed data system would involve a clustered architecture, in which multiple sites would mirror each other and have complete and equal access to all data through a shared group of data stores (see Fig. 4.2. Clustered Architecture Data System, this page). Such a framework would decrease redundancy and improve query efficiency, enabling optimal shared storage.

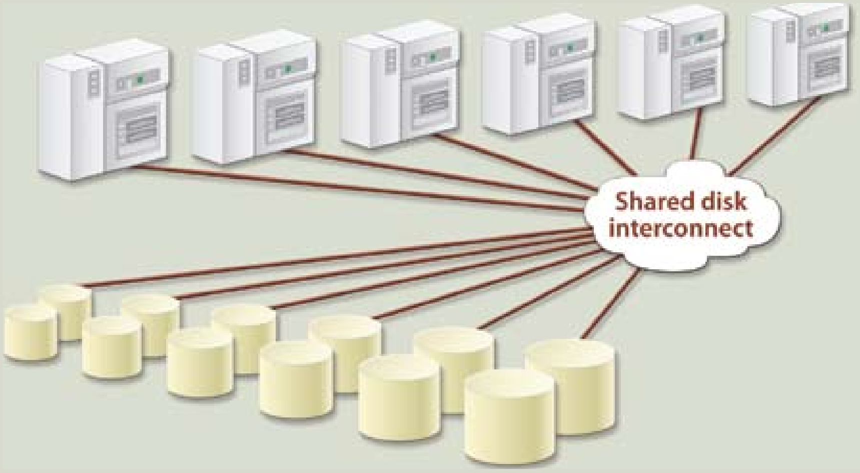

Fig. 4.2. Clustered Architecture Data System. A distributed clustered architecture has multiple server mirrors that all access a complete and shared data store. In addition, all database mirrors share a complete common data model and schema. Combinations of federated and clustered configurations also are possible and may have some advantages for knowledgebase design.

Although developers may be able to adopt a single data model for certain core information relevant to GTL, knowledgebase services would derive partially from linkages to sources of data and analysis tools outside GKB control. For example, GTL investigators could benefit from various external community databases potentially useful to their research and from other relevant analysis tools accessible through the Internet. Incorporating both external resources and the core data model into GKB architecture would require a logical partitioning of data and services as illustrated in Fig. 4.3. Conceptual Overview of GKB Architecture, p. 46. Core GTL data likely would be well understood and stable in terms of the data model; external data and services, however, are subject to faster evolution and potential instability that would need to be tracked by the GTL Knowledgebase. Community development of standards and ontologies is thus necessary for easy access and meaningful use of these external resources.

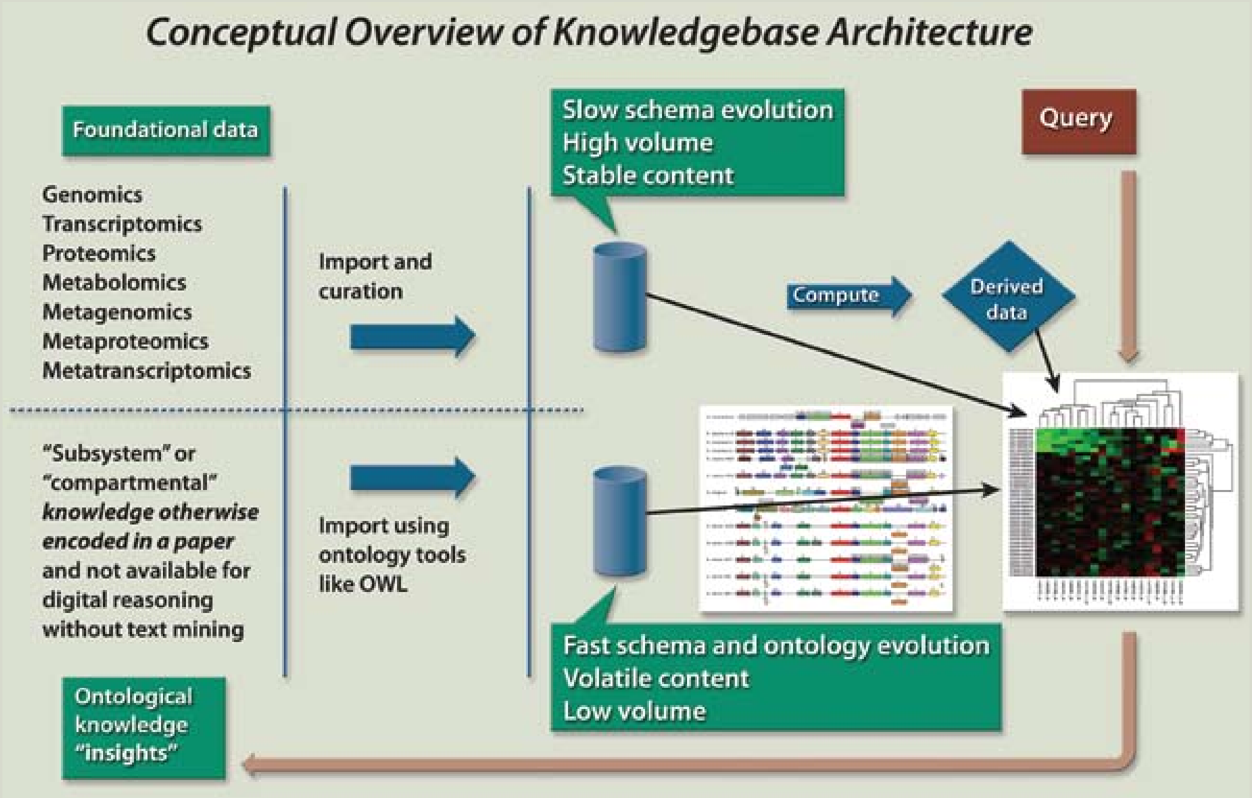
Conceptual Overview of Knowledgebase Architecture
Foundational data
Genomics
Transcriptomics
Proteomics
Metabolomics
Metagenomics
Metaproteomics
Metatranscriptomics
Import and curation
Slow schema evolution
High volume
Stable content
Query
Compute
Derived data
"Subsystem" or "compartmental" knowledge otherwise encoded in a paper and not available for digital reasoning without text mining
Import using ontology tools like OWL
Fast schema and ontology evolution
Volatile content
Low volume
Ontological knowledge "insights"

Fig. 4.3. Conceptual Overview of GKB Architecture. Partitioning of data and services between the components sharing a single data model defined in GTL and external heterogeneous data and services.

Findings

- A principal GKB requirement is the need for *separation between high-volume data* (usually from high-throughput experiments) *and low-volume data* (e.g., information on protein structures and transmission electron microscopy images).
- Another important knowledgebase requirement is the need to maintain large bodies of derived data to support queries on vast amounts of information. Good examples of such information are the data needed to rapidly display large sets of chromosomal clusters in prokaryotes; the volume of these data exceeds that of input data by two orders of magnitude.

Recommendations

- For both high-volume and low-volume data types, the GKB should provide capabilities for performing machine reasoning and user-driven queries (e.g., via a simple Web interface). Data volume will constrain storage and query mechanisms for high-volume information to a more limited set of possible implementations (see Fig. 4.3, this page).

Service-Oriented Architectures and Ontologies

Findings

To take advantage of a wide array of Web-based resources such as data stores, visualization environments, and analysis tools, architectures based largely on Web services models—so-called service-oriented architectures (SOAs)—have evolved and are significantly applicable to the GTL Knowledgebase. Driven by massive commercial data stores like

Amazon and Google and by the need to represent and present unusual data types (e.g., multimedia), this architectural trend considers Web content and services as databases. In Web services models, additional logic beyond that which resides in standard database engines is built to access distributed Web resources. Such models have many attractive features for distributed biological data and services and thus offer the potential for biologists to create analysis pipelines that automatically link experimental data to multiple computations, resulting in new insights. One example of such a resource is MeDICi (Middleware for Data-Intensive Computing; Gorton et al. 2008), which represents a workflow tool for biologists based on SOAs.

SOA-based models are not without drawbacks. For example, they are subject to failures of individual Web resources on which queries depend and sometimes are associated with query performance problems in accessing heterogeneous Web resources. However, well-understood design approaches and supporting technologies can address SOA drawbacks and could be leveraged to build successful SOA features into the GTL Knowledgebase.

While many existing biology databases use a simple architecture, the GKB would require a combination of architectures, including SOA for Web services; application programming interfaces (APIs) for data retrieval; database clusters; online analytical processing; and carefully crafted, flexible data models. Current data systems employing this combination or hybrid approach are, for example, the *Shewanella* Knowledge Base and MicrobesOnline, both of which integrate several data resources (see sidebar, *Shewanella* Knowledgebase, p. 24, and Fig. 4.4. Data Types and Resources Integrated by MicrobesOnline, this page).

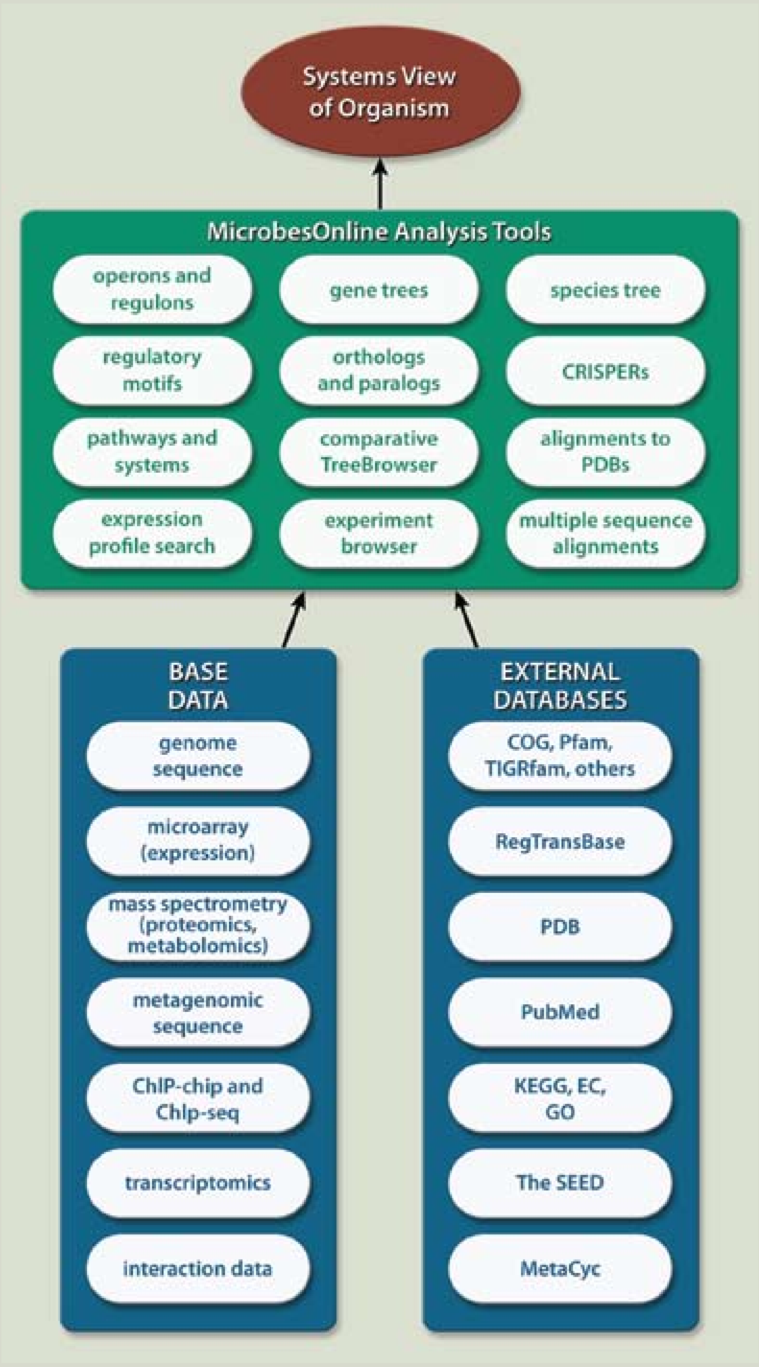

Fig. 4.4. Data Types and Resources Integrated by MicrobesOnline. A hybrid, distributed, and Web services model for data integration and management.

Knowledgebase planners also anticipate that semantic Web technologies can be employed to augment core capabilities of GKB architecture. Such technologies include standard ways for defining Web services using controlled vocabularies (e.g., with UDDI or SOAP) and ontologies for describing data objects (e.g., based on OWL). These semantic Web capabilities make access to distributed knowledgebase services technically easier and more meaningful for researchers. Furthermore, with such technologies, query and retrieval tools can intelligently determine which information and services on the Web have data relevant to a query because knowledge in each domain has been described using a formal ontology. For example, Web resources describe themselves with rich semantics amenable to reasoning by external automated agents, and machines can assume much of the burden of data and service discovery, engagement, heterogeneous assimilation, and integration. Theoretically, scientists thus can devote their time to more creative decision making in advanced stages of the information-gathering process.

Developing robust, expressive ontologies and using them efficiently are extremely challenging endeavors requiring the collaboration and commitment of computer scientists and highly qualified domain experts. For these activities, only limited advancements have been achieved globally, and such efforts are nascent in the biology community. However, experts have created a strong and ever-expanding family of ontologies. These vocabularies range from environmental markup language (EML), which is highly relevant to numerous emerging high-throughput datasets of environmental sequences, to relatively new protein functional annotation (PRO) ontologies. Based on its pervasiveness among PubMed references, Gene Ontology (GO)—which describes the molecular and cellular functions of protein complexes—is the most successful ontology in biological research today. An ontology achievement related to DOE's Bioenergy Research Centers involves carbohydrates in plant cell walls. Formal description languages have been created for carbohydrate structures (e.g., at the University of Georgia's Complex Carbohydrate Research Center, http://www.ccrc.uga.edu), and these ontologies potentially could be extended to encompass the composition and structure of the entire plant cell wall. This would enable descriptions of analytical results on biomass and detailed queries of observed cell-wall structure and dynamics.

Integrating bioinformatics software with databases will be essential. The Gaggle (Shannon et al. 2006), for example, uses standardized descriptions of data items and Web services to integrate data resources and tools to support biologists' analytical needs (see Fig. 4.5. Communication in the Gaggle, p. 49). This approach represents a small-scale prototype of the kinds of systems the GTL Knowledgebase should incorporate to support data-driven analysis, modeling, and visualization. Other useful concepts include notification services that provide updates to biologists if data relevant to their research change or become available.

Additionally, Web-accessible systems increasingly are handling enormous datasets (e.g., in Earth fly-by or geographic information systems and multimedia). These have given rise to an architectural trend characterized by very large databases having simple schemas and high efficiency. The GTL Knowledgebase inevitably will contain datasets to which such technology can be advantageously applied.

Recommendations

- A GKB architecture study group should be established to analyze options and priorities for designs. The study group should include stakeholders from various GTL-relevant areas of biology, representatives of DOE funding entities, developers of software tools, and the core GKB architecture team. Additionally, an architecture configuration board should be established to oversee system choices and performance.
    - Knowledgebase technical and operational requirements must be analyzed in detail to fully exploit the tremendous opportunities offered by computing, informatics, and communications technologies.
- Well-proven architectural technologies and configurations should be used whenever possible to reduce costs and increase system robustness.

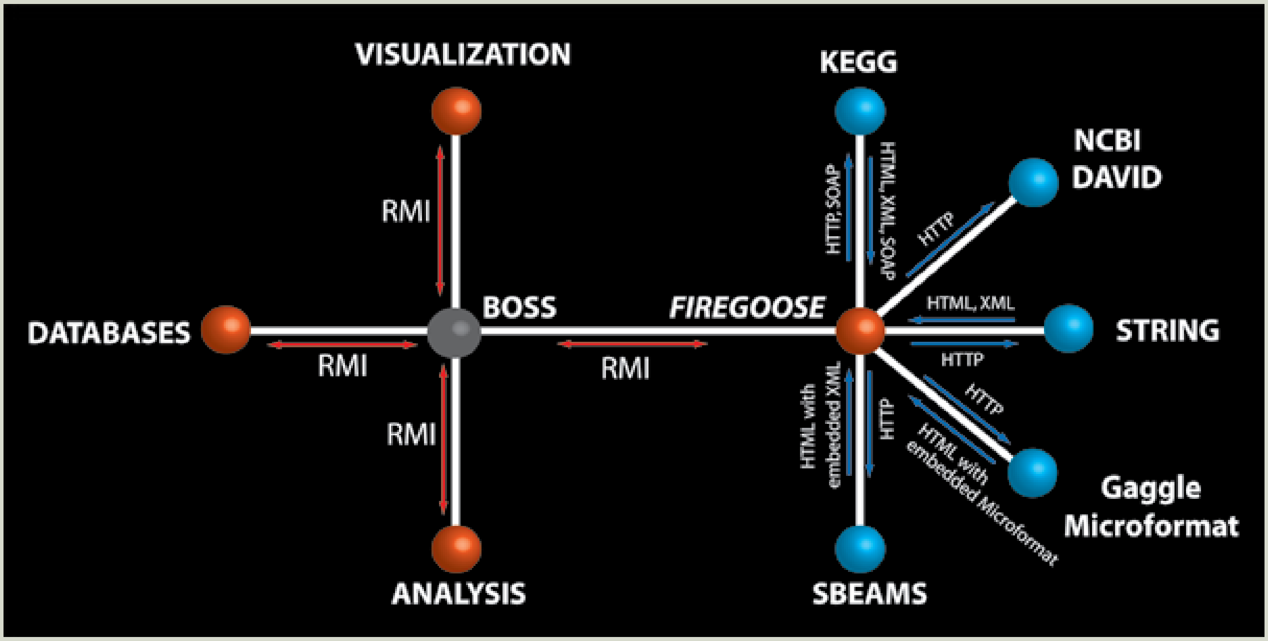

Fig. 4.5. Communication in the Gaggle (http://gaggle.systemsbiology.net/docs/). Software and databases shown as red dots send and receive broadcasts via Java remote method invocation (RMI). The blue nodes are Web resources connected to the Gaggle through the Firegoose and accessed using HTTP with other protocols and formats such as HTML, XML, and SOAP layered over top. Analysis tools in the Gaggle framework include R, MatLab, and MeV; the visualization tools include Cytoscape, BioTapestry, DMV, and Genome Browser. A central strength of Gaggle and Firegoose is the ease with which they can be extended to include third-party tools and databases that have been developed using varied platforms and programming languages. [Source: Adapted from the following two documents: Bare, J. C., et al. 2007. "The Firegoose: Two-Way Integration of Diverse Data from Different Bioinformatics Web Resources with Desktop Applications," *BMC Bioinformatics*, **8**(456). Shannon, P. T., et al. 2006. "The Gaggle: An Open-Source Software System for Integrating Bioinformatics Software and Data Sources," *BMC Bioinformatics* **7**(176).]

- GKB performance, scalability, and latency requirements must be carefully defined and analyzed.
- The GKB should be designed to facilitate cost-effective upgrades associated with anticipated changes in requirements.
- Detailed data requirements—such as rapidly evolving versus stable schemas and large versus small volumes of data—must be defined and the underlying architecture and transport mechanisms built accordingly.

Data Access and Security

Findings

The envisioned GKB would promote the formation of collaborative groups that both informally and formally share data and insights to advance their scientific investigations. Such collaboration is extremely important for integrating analyses of large datasets across multiple groups and allowing sensitive, accurate curation and analysis of data prior to public release. These activities will facilitate the construction of various user interfaces ranging from simple Explorer-type tools to next-generation collaborative tools comparable to contemporary social networking sites such as Facebook.com.

Recommendations

- GKB architecture must implement security policies and practices supporting GTL data and information sharing. These should include procedures for protecting pre-public data for periods specified within architectural guidelines.
- Also, users should be able to incorporate into the GKB additional private data based on their analyses and protect this information using security mechanisms provided by knowledgebase architecture.
- Furthermore, the GKB security model should allow biologists to share their private data with a selected set of collaborators in the GTL community.