Sustainable Business Models for Brokering Middleware to support Research Interoperability:

A Report from the Sustainable Business Models Team to the Brokering Governance Working Group of the Research Data Alliance (RDA)

Karl Benedict, University of New Mexico
Mairi Best, Consultant to EMSO and Ocean Observing and Environmental Sensor Systems
Sue Fyfe, *Co-Chair*, Geosciences Australia
Senay Habtezion, Global Change SysTem for Analysis, Research and Training (START) - International Secretariat
Clifford Jacobs, Clifford Jacobs Consulting, LLC
William Michener, University of New Mexico
Stefano Nativi, Institute of Atmospheric Pollution Research of the National Research Council of Italy (CNR-IIA)
Jay Pearlman, J&F Enterprise and University of Colorado
Lindsay Powers, *Co-Chair*, The HDF Group
Andrew Turner, Esri

Issued December 7, 2015

Executive Summary

The grand challenges of our age such as climate change and food security are creating a greater need for cross-disciplinary scientific insight to guide understanding and inform decisions. The evolution of technology, with the introduction of big data approaches, cloud computing and access to storage and processing, is also changing the way data is accessed and used. As a result, researchers are demanding more detailed earth observations for assimilation into increasingly sophisticated models and simulations, and there is a clear trend toward larger computational undertakings across a loosely coupled set of geoscience disciplines. This, in turn, introduces the demand for access to diverse data sets from a broad range of scientific domains. These data have many different formats coming from distinct community cultures. Bridging scientific disciplines through the introduction of a broker or mediation capability has been demonstrated for a number of cross-discipline applications [Nativi, et al 2013]. The development and use of brokering middleware in a research environment is a relatively new development and the sustainability of such a capability has not previously been addressed. Broker software now has a level of maturity that requires addressing sustainment of the middleware, including the question of which business models could be adopted to best serve this need. The Working Group, in addressing the questions of sustainability, does not presuppose which business/revenue or hybrid model might be embraced to sustain this middleware, but rather examines five classes of business models for consideration. For each example, the strength and weaknesses in the context of long-term sustainment of broker middleware are discussed. Examination of alternatives leads to a

prioritization among models with the recognition that no single model by itself is likely to provide the desired sustainment. We find a hybridized model incorporating aspects of three different business models over the lifespan of the brokering middleware, i.e. federally funded data facility guardianship in the establishment stage replaced or supported by a Consortium model and/or Software-as-a-Service as the broker matures, will likely provide the strongest model for sustainment.

Introduction

Research Context, History and Rationale

Advances in information and communications technologies have enabled an upsurge in open data sharing and collaborative, interdisciplinary, data-intensive research. Known as cyber-science or eResearch, this brings international players and their data together in collaborations that require disparate data and systems to interoperate so as to address the global grand challenges such as climate change. For example, the need for interoperability in cross-domain use of Earth observation data has rapidly evolved and this has been a major driver for electronically-mediated exchange of research information. Researchers are demanding more detailed earth observations for assimilation into increasingly sophisticated models and simulations, and there is a clear accompanying trend toward larger computational undertakings across a loosely coupled set of geoscience disciplines.

In the current environment, where research communities have evolved with their own data formats and cultures, the question of how to bridge these disciplinary gaps is of critical importance. More than a decade ago, the approach to interoperability was to establish standards and to seek agreement to use a federated approach, in which each repository would conform to defined interface specifications. In some cases, this was very effective but in other cases, repositories did not see a benefit for their user community and did not continue supporting federated solutions as the standards and technologies evolved. For multidisciplinary research interoperability (especially in a global dimension), it is unrealistic to expect that all software components or repositories of different disciplines will use the same specifications to interoperate. In addition, bridging the culture gap between the social, economic and natural sciences brings additional challenges that need to be met if we are to enable holistic solutions beyond those originally envisioned by the federated approach.

Brokering Middleware for Research Interoperability

Almost ten years ago, the cyberinfrastructure community began investigating alternatives to formal federation that would reduce the burden on repositories for supporting cross-discipline collaboration. One alternative was to introduce middleware to enable the translation of formats and data, streamlining these processes with a common model supporting translation (Nativi et al., 2013, see Appendix). Early validations of the brokering middleware approach for research purposes included the Group on Earth Observation's Interoperability Process Pilot Projects (IP3) in

2006-7 (Khalsa, et al., 2009). This led to the EuroGEOSS Project supported by the European Commission to mature the software implementation and demonstrate it across three study areas: drought, biodiversity and forestry. The resulting Broker was introduced into the Group on Earth Observation common information architecture (GEOSS) in 2011 and rapidly expanded capability for data discovery and access. The Italian National Research Center (CNR) has continued development of the Broker with support from the European Commission and through participation in the National Science Foundation (NSF) EarthCube program with the National Snow and Ice Data Center, University of Colorado (Figure 1).

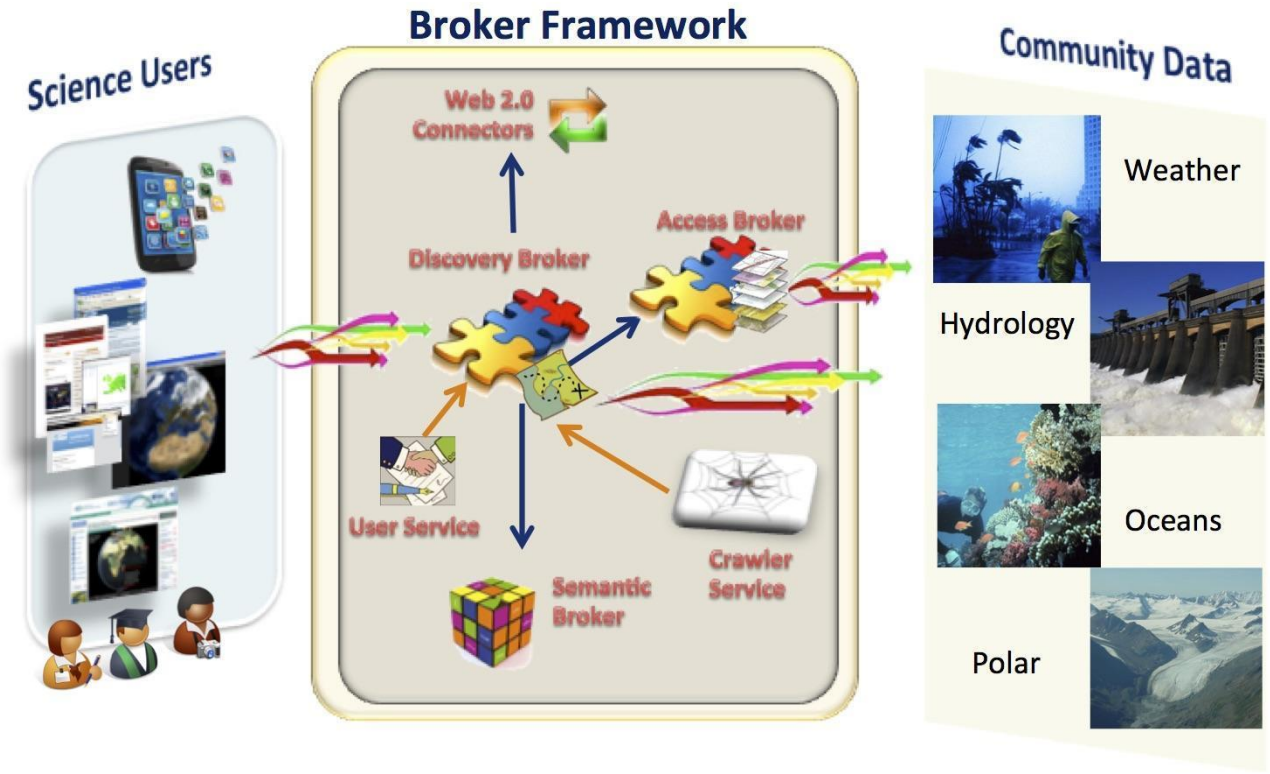

Figure 1 The Broker Framework supports interoperability among diverse disciplines

While there is still a drive to adopt common standards and specifications at the disciplinary level, mediation and harmonization between domains are essential to pursue multidisciplinary research in an effective way. Brokers operate across these interfaces, implementing mediation, distribution, harmonization, and transformation functionalities in a many-to-many context for the existing services and components managed by different communities. Brokering software is not a single piece of software code, but rather a conceptual paradigm with multiple components in a framework. The translation across research disciplines is simplified by having a common 'technical model' embedded in the broker framework with the translation to and from different disciplines handled by facilitators called 'accessors'. Figure 1 illustrates the approach applied by the National Research Center (CNR), which introduces a new middleware layer of service offerings. This layer should contain all the necessary existing (and new) components/services such as brokers to implement interoperability among present (and future) data infrastructures of different disciplines. Therefore, a Broker may be defined as an intermediary service dynamically implementing a many-to-many interconnection for a Client-Server framework. The Appendix to

this document provides further detail on the architecture and characteristics of information and data brokers.

There are multiple instantiations of brokering middleware in science, commerce and industry, supporting various facets of the business and management community as well as research. For example, IBM offers the WebSphere as a tool to facilitate connectivity between applications regardless of the message formats or protocols. IBS, a German company, supports customer resource management services and software through a Message Broker and engages the community of users by offering kits for creating connectors that customize the interfaces between user applications and the "core". The geospatial information company, ESRI, provides the Geoportal Server as an open source broker for discovery and use of geospatial resources including data, rasters and web services.

The research community must determine whether a single universal broker is desired or whether distributed brokers tailored by individual institutions can serve the needs for global interoperability. This question, which can be addressed from many perspectives, is a core issue in sustainability. Tailoring at the institutional level offers a degree of control and redundancy that is attractive, but may lead to distributed brokers losing elements of interoperability for reaching effectively across disparate disciplines. As this is a community and structural issue, it is raised here with the intent of creating a debate on steps forward.

Impetus for the creation of the Brokering Governance Working Group

Members of the Research Data Alliance (RDA) envisage a broad-based "distributed knowledge" environment to address key challenges in government, academia and the private sector. Middleware such as the CNR Broker can deliver capability for research data brokering as a component of this distributed environment, if transitioned from research software to operations.

The CNR Broker was originally available as open source software and was taken up by a number of research institutions. In the process, the middleware was adapted to local needs and cultures, with the result that certain core processes were altered and full cross-domain interoperability became more limited. This led to the recognition that a "broker-of-brokers", or potentially a stack of brokers, might be required to achieve cross-disciplinary interoperability in a completely open and unmanaged development and operation environment. To reduce this possibility, updated versions of the Broker software source code were not offered as open software. However, community needs for the Broker expanded with projects such as the NSF-supported BCube, and approaches to address this issue should be part of the broker evolution in support of scientific research. Large repositories, however, might be reluctant to commit due to concerns over a possible single point of failure for a function as critical as core broker middleware. This raised the question of an appropriate business model to support operational research brokering frameworks.

Thus, the RDA Brokering Governance Working Group (WG) was created to address the governance of brokering middleware frameworks used by the research community. The WG's Business Models Team has undertaken the task of investigating options for business models that would

potentially support sustainable, stable operational environments for the brokering middleware and related tools. This report summarizes the outcomes and recommendations of those investigations.

Addressing Community Requirements

The broader applications of brokering would include the private sector, government, research and educational institutions and individuals who need information for planning and decision support. In each of these cases, the appropriate sustainable business model may vary. In the context of the RDA focus on research data interoperability, the Business Models Team constrained deliberations to circumstances pertinent to broker software as it is used by researchers and educators, i.e. a functioning piece of software supporting a research environment. Although broker software does have commercial applications, the associated business and revenue models necessary for sustainment in other than research and education support are considered only to provide context for shaping our recommendations. The nature of the software and the culture of the target communities are also important considerations influencing this discussion.

Data interoperability supports research within a domain and across domains. Mature domains such as weather and seismic monitoring have standards that support interoperability and data repositories that allow ready access to data. Less mature and newer research areas generally need to evolve interoperability approaches. In this context, brokering can support domain data and information discovery and access. In addressing issues such as climate change that draw on many disciplines that may not have common standards and customs, the broker can provide important contributions to research. The framework for business model discussion in this paper addresses the range of needs and requirements specifically for the research community.

Software Maintenance and Sustainment

Software is an indispensable part of our professional and personal lives. Like many great inventions, it has disappeared from our consciousness. Broker software is commonly referenced to middleware thus making its presence even less apparent to the average user. It is only when this software ceases to exist or function that sustainability becomes an issue. Hence, for software developers and a subset of users, software sustainment is a necessary part of their mindfulness. The challenge is to deploy software under a business model that will maximize the chances for ongoing resourcing, stability, maintenance, support, enhancement and longevity of the software. In defining the sustainability options, we begin with the assumption that the broker “core” is a fully functional and well-documented middleware package, ready to be used for ongoing operations and having the capability to be customized for user needs.

In addressing sustainability, user support is a key element in the research community. There are two primary types of users: those that manage data and repositories and those that use data in their research. While the broker will be invisible to the individual research scientist, data managers providing a service to the research community will be aware of its capabilities and benefits. Members of the Research Data Alliance utilize brokering middleware and related tools in a variety of venues, for example, the CNR brokering system (Figure 1) is in use internationally in programs

and initiatives such as GEO GEOSS, ICSU WDS, NSF Earth Cube and IODE. However, with the exception of standards, there is little precedence for long-term sustainability of such advanced forms of interoperability tools. The issues surrounding sustainment are becoming increasingly complex and need to be addressed as the community engages with the issues of big data and cross-discipline research relating to the global grand challenges. It is sometimes seen as a chicken and egg proposition. Without sustainment, users such as major data repositories will not invest the time to gain experience with the broker and promote its use, whereas with a commitment to long term availability and effectiveness of the broker, the user base will grow past a tipping point toward significantly broader use. Thus, without a better understanding of how to sustain brokering software, there is risk that can potentially undermine the stability, enhancement, and longevity of this contributing software.

In considering software sustainment, a broad range of issues must be addressed including software maintenance, user support services, user base continuity and expansion, evolution of capability, etc. in an operational environment where the software is not visible to the individual user except when it fails, traditional indicators of success such as user adoption are hard to measure for middleware, so a successful business model must also address visibility. An example of a cultural issue is the desire for control of the user experience, since Data Repositories typically want control of the software that impacts their service capabilities. Archives are generally aligned with science disciplines and may want to tailor a broker capability to their own needs. As mentioned earlier, if this leads to many versions of the middleware, cross-disciplinary interoperability will be compromised.

The terms software maintenance and software sustainment are often used interchangeably. It is important to clarify the terminology when discussing software sustainment. The descriptions below help distinguish between these terms as they are applied to this report.

Software Maintenance

The *IEEE Standard Glossary of Software Engineering Terminology* defines “software maintenance” as follows:

> The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment [IEEE 90a].

To be complete, there is usually a fourth category of maintenance activities focused on anticipated problems, or preventive maintenance.

Software Sustainment

The Software Engineering Institute (SEI) (a federally funded research and development center), which is sponsored by the U.S. Department of Defense, has a working definition of software sustainment as follows:

> The processes, procedures, people, material, and information required to support, maintain, and operate the software aspects of a system.

To elaborate: “Successful software sustainment consists of more than modifying and updating source code. It must address many other issues such as documentation, operations, deployment, community development, security, configuration management, training (users and sustainment personnel), help desk, COTS product management, and technology refresh. It also depends on the experience of the sustainment organization, the skills of the sustainment team, the adaptability of the customer, and the operational domain of the team. Thus, software maintenance as well as operations and the customer mix/environment should be considered part of software sustainment” (Lapham et al., 2006).

We have defined a suite of attributes necessary for broker sustainment adapted from the Software Sustainability Institute attributes for software sustainment (Figure 2). These attributes will take on varying levels of importance during the life-cycle of software although some might not be as important at points in the life-cycle such as the initial development phase, but will receive equal consideration for sustaining mature software. Each of these categories requires some level of financial and human resource investment.

When assessing alternative business models for brokering software sustainment, an evaluation can be made as to how well the model will fulfill each of these requirements.

Figure 2. Required attributes for brokering software sustainability adapted from the Software Sustainability Institute's definitions for software sustainability (http://www.software.ac.uk/).

Attributes for assessing sustainability of the business models

Secure Financial Support – The efforts required to obtain and sustain funding for the broker. This may include proposal writing, venture capital gathering, reporting etc.

Engage user communities– The efforts required to identify and target, engage and sustain a class of institutional and/or science users such as solid earth geoscientist and their data facilities.

Marketing– The effort required to understand requirements and then pursue the case(s) for commitment to use a broker capability. This is focused on individual users or facilities and moves beyond the general engagement of a user community.

Human Resources – The personnel support and expertise required for market development, management and achieving the technical capability for software evolution and. sustainment

Software Engineering – The effort, including formal and informal processes, required to provide development and ongoing maintenance, improvement and technology assessment of products.

Product management – The management of documentation, versions, licensing, distribution, security, and other administrative activities.

Before looking at the individual business models, factors exist that will impact the business model selection in addition to meeting the attributes above. One of these is the legal considerations of operating in a multinational or global environment.

Legal Considerations

An consideration for addressing sustainability is the need to conform to national and local legal mandates. This may not directly constrain the broker's existence, but will impact the effectiveness of the broker in delivering interoperability and, thus, its value to the research and development communities. Effective data interoperability faces a variety of challenges including those related to lack of common standards and approaches. Legal issues similarly arise when there is a need to combine, disseminate, and/or otherwise use/re-use data-sources. This presents difficult challenges similar to those posed by the technical, operational and semantic issues linked to data interoperability. A key question is how do we ensure that legal requirements (e.g. proprietary rights of others) are not infringed upon in the event of use/reuse of multiple data sources.

Legal requirements tend to be place-specific. Thus, in principle, data issues are automatically covered by existing national [in some cases international (WIPO) and regional (EU)] regulatory requirements and restrictions, such as laws of privacy, national security, intellectual property as well as other applicable rules and regulations. In some cases, these laws require that certain forms

of data (and data bases) fall into public domain (e.g. US government publications) although this is not always the case for data sets originated by private actors. In the latter case, licenses and contracts are more common, usually governed by agreements between 'rights-holders' and 'users', subject, in some cases, to public interest rules and limitations. These agreements (usually expressed in the form of 'policy' 'use agreements' etc.) govern different types and uses of research data for different scientific domains and communities of practice.

In sum, public research is subject to different place and domain based legal requirements as well as licensing and contract arrangements, which affect the interoperability of data. Where restrictions are more stringent, interoperability is hampered. There is a need therefore to do rigorous assessment of rules and norms that promote (or constrain) data sharing and/or interoperability across countries, domains and communities of practice and seek ways for promoting effective legal interoperability for data sharing. An extensive study of these issues is beyond the scope of this report and is being considered in other RDA working groups and interest groups.

Sustainable Business Models

Software Business Models

Software services for research provide intangible goods and services, and frequently employ business models different from those utilized by companies providing goods and services focused on financial, physical or human sectors. As they are currently configured, those entities responsible for brokering software serve or could serve four classic business archetype roles.

- A *creator* that transforms ideas into a product
- A *distributor* of software
- A *lessor* that provides the rights to use the software
- A *broker* that employs or operates the matching of data and information users with suppliers

An underlying premise of the archetypes with respect to software is that the provenance and intellectual property (IP) are established. This might not be true for brokering software if elements of the code are deployed in distributed operations or established by suppliers as embedded interface applications. A business model for future development, or modifications of brokering software and tools, needs to take account of these important aspects of software services.

Generally software businesses use a hybrid approach serving the role of more than one archetype function since they are acting as both the inventor and the IP lessor. Software companies can differentiate their business model by offering software as a product or, software as a service (SaaS), or a combination of both. As K.M Popp (2015) notes,

> SaaS means the software vendor does not deliver the software, but the customer gets both access to the software and usage rights …... The software vendor carries the cost of software support, maintenance, and operation."

Revenue Models

The software industry has developed a number of business models to generate revenue to support all or some of the attributes of sustainment outlined above. SaaS is considered in this review of viable revenue models, as well as non-commercial and hybrid funding models. The revenue models explored are: government funding through assistance awards and contracts; government funding through federal data facility guardianship; SaaS (including tiered pricing); Information and Ad sales; Corporate Support (including Foundation Support) and Product /Service Sales, (foundation support) and the Consortium model. Each of these revenue options are discussed in the following context and may have variants depending on the target market:

1) Description of the business model with examples of software currently supported under that model
2) Mapping Revenue Models onto Sustainment Attributes
3) Characteristics of the business model that would benefit the sustainability of brokering software
4) Challenges in the application of the model to support brokering software for the research and education communities

Government Funding Through Assistance Awards and Contracts

Model Description

For decades, governments have provided funding to support the development and sustainment of software. In the U.S., this support takes the form of contracts, grants, or cooperative agreements. Contracts provide services to the government, such as the development and maintenance of software and hardware used to ensure safe air travel. Grants are assistance awards that, in the context of the interests of the Research Data Alliance, are often used to "assist" university and non-profits in the pursuit of a research problem. The supporting government entities generally have little involvement in guiding the research activity that is funded by grants. Cooperative agreements are also assistance awards, which are generally reserved for use in the support of large and complex undertakings. In these cases, the government has substantial involvement in shaping the course of the research endeavor in cooperation with the investigators.

Aside from developing software in the research context of computer science such as compiler languages, most software development is a means-to-an-end to facilitate an inquiry, such as prove or disprove a research hypothesis, to gain insight into relationships within data sets, facilitating connections among essential but disparate data sets, or to simulate a physical phenomenon. Globus, NetCDF, and the Community Climate Model are examples of software that have been developed with government support and continue to be refreshed and modified with U.S. Federal dollars. The longevity of support for these software packages is measured in decades.

Occasionally governments will invest more strategically to support the national advancement of science research and development. In Australia, the Commonwealth Government provided 5-10 years funding for the National Collaborative Research Infrastructure (NCRIS) and Super Science

Initiatives, which have supported important national research data initiatives such as the National Computational Infrastructure, Terrestrial Ecosystem eResearch Network, Australian Urban Research Infrastructure Network and the Integrated Marine Observing System. More importantly, this funding has significantly enabled Australian researchers to participate in eResearch by supporting the development of IT infrastructure, software, tools and services for open publication, discovery, sharing and use of Australia's research data outputs through the Australian National Data Service (ANDS) and National eResearch Collaboration, Tools and Resources (NeCTAR). The US National Science Foundation is funding the EarthCube Program, which provides assistance grants for the development of Cyberinfrastructure to advance the geosciences. The program is expected to last a minimum of 10 years and has a strong focus on improving interoperability and data sharing across geoscience domains.

Mapping Revenue Models onto Sustainment Attributes

Funding – To obtain support from the government, a proposal is required. The home institution of the principal investigator bears part or all of the cost of preparing the proposal through its overhead charges (although overhead covers more than proposal development). Government initiated strategic investments, like the NCRIS and Super Science initiatives in Australia, can be of longer duration than research grants and cooperative research agreements, but are nonetheless limited in their duration.

Users – With respect to software developed as part of a research grant, the concept of market development is interpreted to mean understanding which research communities might find the software useful in the pursuit of their objectives. This is an important observation that could have an impact on sustainability. The design of the software could be influenced by the applicability of the software to other venues. This takes forethought on the part of the principle investigator either at the outset of the project, for example in the proposal, and/or realization that there is greater need for the functionality of the software than originally anticipated. Government funding does not necessarily encourage or discourage the principle investigators from engaging in market development [but some funding agencies, such as NASA, encourage and may require that software developed under the grant be made available to the community for reuse]

Communities – An important consideration by the government with respect to sustained support is the extent of community engagement. The engagement is not only concerned with the number of people using the software, indeed might depend on its availability, but also the influence that community of users has over the future directions of software capabilities and refinements. Often the government oversight of a project will encourage community engagement and might even support additional funding for outreach and engagement activities that might come in the form of funding for a workshop or seminar series. ANDS and NeCTAR have played a significant role in building enabling Australian researchers in the eResearch space, including building and maintaining the communities that will implement and shape data practices, standards, tools and infrastructure into the future.

Human Resources –Software developed under a research grant or cooperative agreement usually receives financial support for the duration of the award, which can range from one to five years. Support can come incrementally on an annual basis or be provided in entirety at the beginning of

the award. The incremental support is subject to adequate progress in research activities. Both grants and cooperative agreements allow modification of the original proposal, subject to approval by the funding agency. Most often the award is made to the institution which employs the researcher. Often this institution provides tangible and intangible contributions. Under these circumstances software developed under a research project 1) has the revenue stream for the work virtually assured for a limited period of time, 2) can be flexible in design and execution, 3) has a natural partner in the institutional home that often provides benefits useful to the project, and 4) has the potential to contribute to the educational mandate of the institution.

Software Engineering – The application of the best practices in software engineering is an essential element for sustainability. Although less rigor might allow short term objectives to be achieved, continued support might hinge on the ability of the software to be easily modified to meet researcher's changing needs and/or emergence of new data and technologies. Therefore investing in good software engineering is an important use of the revenue obtained from the Government.
However, government funding of a project, particularly in a university setting, does not assure that good software engineering practices will be used throughout the project. The practices of educational institutions to have students and research assistants write software usually means that the engineering practices are either unknown or not followed by the software authors. If the software is not meant for a broad audience, rigorous development practices are less likely to be followed.

Product Management – Under government funding, product management is usually the responsibility of the institution that received the funding and the principle investigator employed by the institution. The government often encourages the sharing at minimum cost of software developed under a grant or cooperative agreement. However, modified versions of software might be offered as a commercial product. In general, the government usually allows the Intellectual Property rights to reside with the institution where the software was developed but this may require the government to have royalty free access to the software. The on-going review process employed by government agencies encourages principle investigators to demonstrate good product and project management.

Advantages

- Advances the conceptualization, prototyping, and initial testing of the software for such attributes as functionality and robustness, since there are no other sustainment attributes of the software that need to be considered during the initial development and demonstration phase.
- Brokering middleware development fits the Federal funding model well in that it requires capital (government), competencies (academia), and intellectual commons (university environment).
- There are some instances of longer-term support of software that the funding agencies deem necessary for significant community functions such as weather prediction and for these, concern must be paid to each of the sustainability attributes, but the assured

funding stream allows the project management to navigate the allocation of resources to addressing each of the attributes.

Challenges

- Funding term limits do not create sustainment. There is a significant risk that government funding is not a predictable or reliable long-term solution for sustainability.

An all too familiar scenario is the demise of software developed as part of a research grant after the research grant expires. Even though grants and cooperative agreements have limited duration (maximum of five years), perpetuation of activities under previous awards is possible through follow-on support. Long-term support of software as an essential element to achieve scientific research objectives is possible under certain conditions determined by the funding agency's priorities, policies and practices. The Globus, NetCDF, and Community Climate Model examples cited above share certain features: 1) They provided very useful cyberinfrastructure to advance research and education activities; 2) Their sustainment has been part of a larger set of activities, e.g. NetCDF is supported through the Unidata award and the Community Climate Model is maintained as part of the award to support the National Center for Atmospheric Research; and 3) the funding agencies recognize the need to invest in infrastructure (physical and cyber) to sustain the health of scientific enterprise. The challenge is that these examples are the exception and not the rule and Globus is now exploring traditional mechanisms of sustainment outside of government funding to complement its government support. This is particularly true when the software, such as middleware, is not as visible to users in successful operations. Furthermore, Government initiated strategic investments like the NCRIS and Super Science initiatives are often politically motivated and at the mercy of changes in Government leadership and party politics. The size, complexity and altruistic nature of the programs (e.g. ANDs and NeCTAR) established under such funding arrangements, makes it extremely difficult for these programs to continue under other funding arrangements after federal government funding is withdrawn.

As noted earlier, government support for software development does not ensure that good software engineering practices are employed, particularly if the software is developed as a means to an end when, for example, addressing research questions. It is not uncommon for software used in a project to be developed by graduate students with little if any software engineering experience and have almost no documentation. To be sustainable, software needs to be well engineered and well documented for its present purposes as well as future uses. In this case, future uses may mean crossing disciplinary boundaries to reach a broader user base. Grant funded software may not address this. The required software engineering must be explicitly planned for, but might not receive funding from the agency if the focus of the project is scientific research. The development should also address reliability and ease of operation. When software is created for one purpose and used by its developer, ease of operation by others or an outside organization is not generally a concern of the developer.

Brokering middleware has been developed and has been maintained under mostly government support either through assistance awards or contracts. However, the support has been of limited duration and piecemeal with no one award supporting the end-to-end maturation and

sustainment of the product. Using grant support there has been less need to identify to funding agents how resources were allocated toward the various attributes that are need to develop, maintain, and sustain the middleware nor to seriously consider the long-term implications of market development and product management including revenue models. This situation is generally true for most software developed under government funding and is and will remain a challenge.

Government Funding through Federal Data Facility Guardianship

Model Description

Federal applied science and data agencies such as the National Oceanic and Atmospheric Administration (NOAA), National Aeronautics and Space Administration (NASA), Australia's Bureau of Meteorology, the U.S. and British Geological Surveys, and Geoscience Australia develop software and systems to manage, analyze and provide data and information to government, industry, education, research and the public, so as to maximize the economic and social benefits the agency delivers nationally. Federally funded research data facilities such as those funded by the National Science Foundation and NASA, serve specific communities within scientific domains. Their mission is driven by the needs of the community they serve. Government data facilities supported by the NSF are typically governed by a Board of Directors who determine the budgetary priorities of the organization based on requirements put forth by the community and the data facility management.

In the guardianship funding model, the data facility may adopt and/or develop brokering middleware to serve their community. The costs of maintaining the broker would be incorporated in the operations and maintenance costs of the data facility. Success of this model requires a strong advocacy for the broker from the community as well as within the data facility. Sustainment is assessed by the Board of Directors based on continuing assessment of the value of the broker to the community.

Federal agencies will occasionally invest in software that delivers benefit to all government stakeholders, for example, in support of the open data agenda as a means of stimulating economic development. For example, Geoscience Australia developed and maintains the Australian Government's spatial data portal, FIND (find.ga.gov.au) and spatial data visualization tool, National Map (nationalmap.gov.au), that provide public access to a network of open government data from across federal, state and local government jurisdictions, and has also invested into development of an enterprise platform for data brokerage (mainly workflow based translation, transformation and ETL) based on Feature Manipulation Engine. Similarly, NASA has developed and sustained middleware that facilitate access and use of the data in its guardianship both through internal efforts and support provided to academia and industry. Other examples include:

- Unidata: AWIPS, McIDAS, THREDDS
- UNAVCO: IDV, Dataworks
- IRIS: IRIS-WS, IRISFETCH and many more including community authored software

Similar modalities are seen in Europe both at the national level and at the European Commission. Europe supports developments for e-infrastructure and complementary activities in research and application of data. The CNR broker was developed with support of the Italian National Research Center and with research and development grants such as EuroGEOSS and GEOWOW from the European Commission. The research and development grants do not address operational support. For the Data and Access Broker (DAB) of GEOSS, the Italian government as provided operational support for a fixed period of time, which may be extended. As mentioned above in the discussions of Australia, long-term sustainability of the DAB is not assured, in part due to the lack of resources for GEO.

Mapping Revenue Models onto Sustainment Attributes

Funding – Funding is provided by the funding agency of the data facility and budgeted as part of operations and maintenance. Agencies manage their budgeted allocations to deliver sustainable long- term objectives whilst delivering projects and the Government of the day's policies and strategic priorities.

Users – Not actively undertaken since the support for the broker is provided by the requirements of the user community. However, occasionally governments will solicit ideas for software development that would address unmet needs of current software and/or the development of software that expands or complements existing cyberinfrastructure.

Communities– US Federal research data facilities are very well connected to their primary user communities and are dependent upon these communities for guidance (Board of Directors, Steering Committees, strategic planning). These facilities cannot continue to exist without the strong support and use by their communities. Hybrid approaches have been tried in the US, for example, software development directly by the agency or through contract and/or grants is provided (source code and all documentation) directly to community groups to maintain and modify such that it best serves the needs of the community over time.

Human Resources – Software and systems development and maintenance is planned, budgeted and implemented on an annual basis to meet the prioritized needs and objectives of the data facility and community it serves. Externally developed software may also be incorporated into the operations of the data facility.

Software Engineering – Software development and/or maintenance is provided internally by an expert/advocate within the data facility, such as the central ICT unit, or more frequently is contracted to private companies.

Product management – Initial product planning and design may be undertaken through the support of a grant but long-term management is sustained through the data facility. This activity is guided by community driven requirements and implemented by the governing body of the data facility. The data facility manages the licensing, version control and distribution of the software.

Advantages

- Federally funded data facilities tend to be long-lived, secure, well-maintained and are appropriately resourced for as long as the provision of data remains a government priority. In the case of data repositories, it is the organization's mission to manage and maintain their data and associated data discovery and access tools.
- Federally funded data facilities are well-known, visible, accessible and respected by the research community. It is assumed that they represent the authoritative source of data and information.

Challenges

- Finding a home for brokering middleware within a data facility that serves a specific community will be challenging because the very nature of the broker is that it should serve numerous communities at once, and therefore may not be highly valued by a single community willing to take on the responsibility and expense of maintenance.
- Federally funded data facilities are typically long-lived but not necessarily permanent. Many operate on a 5-year funding review cycle. For example, Australian Government Departments and their Portfolio Agencies are not guaranteed stable entities. Restructures can result in the loss of entire functions and their associated ICT infrastructure, systems and services.
- These facilities are under constant budgetary constraints and the value of the broker must remain high to justify continued support. Even within a relatively long term, stable agency like Geoscience Australia, budget allocations and strategic priorities are volatile and can result in the removal of resources supporting software, systems and services if other activities are considered to be a higher priority for the fiscal year. This tends to result in limited or no software development or system maintenance after the launch of a system or tool, and the agency's ICT unit typically maintains an increasing number of incomplete, outdated and often duplicated legacy systems and web applications.
- The brokering middleware requires a strong advocate and expertise within the data facility to be maintained.
- Facilities may wish to develop brokering software in house (provided more control and in-depth knowledge of the code)

Software as a Service (SaaS) Business Model

Model Description:

A common model for software and services sustainability in the private sector is called "Software as a Service" or SaaS. While there are multiple definitions of SaaS, a common one that is useful for the present discussions is:

> **Software as a Service (SaaS)** - software that is owned, delivered and managed by one or more providers and is available remotely. The provider offers access to the software that is consumed in a one-to-many model at any time on a pay-for-use or subscription basis.

While the definition focuses on “pay-for-use”, there are almost as many variants as there are markets. This will be discussed below. However, there is a general set of characteristics for SaaS that are crosscutting:

- Hosted
- On demand
- Integrated (operates on a platform)
- Subscription or other form of relation
- Multi–tenant (simultaneous use)
- Supports Network effect i.e. builds more rapidly, leveraging marginal benefit and tipping point evolution (viral adoption)

These characteristics become particularly interesting as research and applications move to using Big Data, and cross-discipline modeling and processing emerge as essential needs to address global issues. From a business perspective, the cloud paradigm and implementation provides a rich environment for customer interactions and support. This allows for more dynamical support and removes geographic boundaries, which could inhibit more traditional hardware businesses.

SaaS has found wide applicability in a range of markets. Examples of SaaS implementations include WordPress (a website resource), Survey Monkey, Smartsheet, Customer Relations Management, GLOBUS, iPlant, GitHub, ArcGIS Online, FME Server and many others. Generally, these provide direct and visible utility to end-users and are proving themselves to be an economical replacement for in-house software and infrastructure in government science agencies and data repositories. There is a challenge in comparing pricing and the markets mentioned above and the market environment for middleware because the level of visibility is different, and in fact, the customer base may be different. For example, if we consider the need for interdisciplinary research, there may not be a broad base of existing resource infrastructures that reach across research domains.

This challenge does not preclude a service-based approach and related pricing. For example, the pricing can be based on incentives to procure higher-level capabilities and services. The initial service offering can be a free service that engages the community. Once engaged, options for more capabilities are provided (while still retaining the free baseline option) and the user or institution becomes part of a paying base for a sustainable business. The pattern of free offering followed by paid enhanced capability is known as “Freemium”. Examples of various models are given in Table 1 (Pricing Models).

Table 1: Pricing Models (includes Freemium pattern)

Capacity-based Model	Customer cost scales as capacity, usage, or number of users reach certain thresholds.
Feature-based Model	Customers scales according to the number of key features available
Time-based Model	Subscription with a fixed (extendable) time

Use-case Model	Fees are based on specified categories of customers (non-commercial use, educational, non-profit, etc)
Quality of Service Model	Customers get a certain level of response time or service level

A typical example would be the product smartsheet, an 'intuitive online project management tool enabling teams to increase productivity using cloud, collaboration, & mobile technologies" offered by Smartsheet. This online version of Microsoft Project that enhances collaboration capabilities. The customer has a free trial period and then has the option to chose from of a basic ($14/month), team ($39/month) or enterprise package for continuation. This is a combination of time-based and feature-based pricing models.

Mapping Revenue Models onto Sustainment Attributes:

Funding – The source of funding for SaaS business models includes complementary mix of contributions mentioned above. In a reasonably large number of cases, the approach to funding is a "freemium" model; the fees for enhanced services are not generally high and thus the transition from free service to enhanced services has a low barrier. A SaaS organization benefits from the scale of the number of transactions or subscriptions. There is a question of whether these models apply to middleware. IBS is using this model for its Message Broker middleware but since the Message Broker is not a standalone business, this example does not validate the application of the business model to brokering and middleware for research.

Users – Unlike services to a broad range of end users, the business model for middleware will need to target large institutions and businesses that will benefit from the interoperability offered. For the academic and research community, the market could be institutional clients and data/information repositories. The market development must follow the traditional cultural pattern of the target market and thus will need to be adapted across a range of markets to be effective. For example, for institutional data repositories, the issue of open source software arises frequently and thus the business model must address how to provide enough unique benefits from the service that customers will not simple download the software and apply their own software engineering.

Communities – For customers from the research community, community engagement through presentations at conferences, working sessions, demonstrations and other direct interactions are part of the culture. Community advocacy can support market development.

Human Resources – The SaaS model, particularly for open source modalities, can adopt support and contribution from user and supplier communities. This can come in the form of software recommendations, feedback on usability and adaptation to the cultural norms in target research audiences. There are specific examples of such contributions that have significant impact on the software application effectiveness. Both the Broker Framework and the Message Broker offer open software kits to the community of users to build connectors or adapters that can then be incorporated into the core broker offerings. This is a model with community software

contributions, but it is not a fully open environment. Other models, such as WordPress, have a fully open software core and actively encourage community engagement. Like the evolution of crowd sourcing, community contributions are done on a voluntary basis. For the research community, the mixed model of community contributions and a SaaS modality can be complementary and provide a broad customer engagement for sustainability.

Software Engineering – The role of software engineering is to develop and maintain an operational capability of the software. In the open community model, software can be developed through community contributions and in widely used systems, there can be community inputs on quality. However, software engineering is usually a focused effort and may need more structure than occurs in a virtual and distributed community effort. This should be addressed as part of the program management activities.

Product management –"Traditional" SaaS implementations generally use conventional product management approaches. Typically changes are prioritized and then an agile process may be used to facilitate fast development. Documentation for users is usually web based and accessible through the internet. Chat rooms are a common means of offering answers to users' questions. The application of such management processes to open source software has occurred and is done on an application-to-application basis.

Advantages:

- SAAS models cover a wide range of both open source and proprietary software capabilities.
- Flexibility of pricing from very modest costs for supporting an open source community to more structured implementations.
- Shorter development cycles when software can be pushed out to the cloud with smaller batch sizes, faster feedback and high overall quality.

Challenges:

- New capabilities may not be relevant to a subset of users and they must adapt to the change.
- The research community may be a low volume customer. There will be issues to be addressed on how to get higher volumes and flow through.
- A broker needs a comprehensive set of software interfaces or application interfaces to provide a broad interoperability capability. Building such a base can be time and resource intensive as it moves beyond traditional standards.

Information and Ad Sales

Model Description

Online information and ad sales are forms of marketing that use the Internet to deliver marketing messages to consumers and to provide contacts for targeted marketing - individuals that may be interested in purchasing products and services. Online advertising includes web banner advertising, search engine marketing, and other forms. Income may be generated by pay per click, search and web analytics, cost per action, revenue sharing, email and referral marketing, and

related mechanisms. Information may be collected on data usage, and may be of interest to either data providers (e.g. usage stats) or users (popular tools or data), potentially providing an additional revenue stream.

Information and Ad Sales is the business model used to support well known software products such as:

- Google
- Facebook
- YouTube
- annotation/comment service (e.g. Pinterest)

Mapping Revenue Models onto Sustainment Attributes

Funding – Funding is provided by the business or industry that benefits from information collected on users by the broker (e.g. email addresses) or exposure to corporate ads that result in traffic to the business or industry and subsequent sales. Business contracts must be sought and created by the broker and the business or industry that is seeking information or placement of ads.

Users – Considerable effort may be required to identify businesses or industries that desire the information that can be obtained by brokers and to identify appropriate advertisers. A small number of consulting firms offer services that enable brokers and other institutions to connect with one or more (i.e., bundle) advertisers.

Communities – Caution must be exercised in selling user information to external entities or integrating ads into broker services as such activities may be viewed as violating personal privacy or creating negative associations with businesses and industries that are not favorably viewed by users of broker services.

Human Resources – Additional human resources may be required to capture and package data and information desired by external business and industry interests. Likewise, substantive investments of personnel time and money (e.g., contractual services) are normally required to match brokers with appropriate corporate advertisers.

Software Engineering – Software development or contracts with third parties (e.g., Google analytics, web developers) may be necessary to capture/package information and support advertisements.

Product Management – Services such as provision of data and information and supporting advertisements require significant relationship management, with both the business/industry and the users of the broker services. For instance, particular attention must be paid to ensuring that users are not disenfranchised because of infringement on privacy and association with annoying or distasteful ads.

Advantages

- automated mechanism and requires little human intervention

- scales easily as user numbers increase
- may include a user model where users pay to not see ads
- there is potential for annotation service provided by this model to become part of institutional quality assessment process
- this model can contribute to a business diversification strategy for resilience and growth

Challenges

- requires sales to advertisers and/or people that wish to use the service -- requires a sales team and an identified large consumer base
- requires good planning of communications and branding
- ads may disenfranchise user base
- sales of information is a major privacy issue and may disenfranchise user base
- brokering services may not be amenable with this model
- questions about who owns the usage and annotation data

Corporate Support and Product /Service Sales

Model Description

Corporate Support –

In the context of support for a data brokering service provided through funds contributed by corporate entities there two primary models: sponsorship and membership, where membership may be broken down further into direct and indirect models. The sponsorship model is one where corporate entities provide general or specific funding for the system as a whole or some subset of the brokering system for which their sponsorship (and acknowledgement) is seen as being aligned with their mission or business model. Often the sponsorship model does not include substantial participation in the governance of the system.

The membership model may provide more direct participation in system governance, depending upon the specifics of the model. One type of membership model is one in which corporate entities "join" an organization that is responsible for the development and maintenance of the brokering system, potentially at different membership levels that entail different rights and privileges. This "direct" membership model may have substantial member participation in system governance as a benefit of membership. A second type of membership model is an indirect one in which members of corporate entities (e.g. professional organizations, research organizations, universities) are provided access to services of the brokering platform that they would otherwise need to pay for individually or not have access to at all if it weren't for their organizational affiliation. In this indirect membership model the organizations to which the members belong would provide support for a brokering system capability.

Product/Service Sales –

A product or service sales model is another approach for obtaining funds needed to sustain a brokering service. In this instance, the business model includes an outright sale of software for customer platforms with the customer having ownership of the software and then service to support operations on the customer's platform is provided by the vendor. The service could

include a schedule of software updates or responding to special customer needs. SaaS, on the other hand, is where the vendor retains ownership and operates the software in a “cloud” or vendor platform. The service modality is quite different with automatic updates of software capabilities, etc. In the fee for service variant users would pay for access to some or all of the services provided by the brokering service, with the amount paid potentially falling into access tiers associated with levels of service or differential access to service types. In this respect there are some similarities between the fee for service mode and the Software as a Service model described above, with the potential differentiating factor being that the financial contributions in this case are primarily coming from brokering system product users while the SaaS model has a mixture of end user revenue and brokering platform organizational users. In a fee for transaction model the brokering platform provides an e-commerce platform through which requested data products or services may be ordered and accessed on an item-by-item basis. In this model there may be a revenue sharing component with the data and service providers that are contributing to the brokering system’s content.

Examples of how this model has been used to provision brokering middleware or similar tools and services include:

- In-kind contributions of computational or storage services from providers such as Amazon, Google, HP or Microsoft. (Sponsorship model)
- The Open Geospatial and iRods Consortia are membership-based and include corporate sponsors at different levels (direct membership model)
- AGU and other professional organizations offer access to subscription-based content as a benefit of membership (indirect membership model)
- Freemium model - RedHat, WordPress, Evernote (added value for premium level - e.g. in a brokering context this could include value added services on top of data vs. basic download) (fee for service - similar to SaaS above)
- iTunes store / Google App Store (Fee for transaction model - product sales)

Mapping Revenue Models onto Sustainment Attributes

Funding– In the corporate support funding model substantial outreach and negotiation is required to first establish a relationship with potential corporate sponsors/members, then demonstrate the value of the system to the organizations, and finally negotiate agreements that will meet the needs of both partners in the arrangement. This is a long-term process that requires a high degree of interaction and maintenance through time.

In the product/service sales model there is a requirement for marketing and promotion of the service as visibility and awareness of the service is a prerequisite for its use. Once awareness is established sufficient documentation and interface capabilities about the products and services of the platform must be produced to both demonstrate the capabilities and value of the system while also lowering the barriers to use.

Users – Market development is a core component of both the corporate support and product/service sales models. Without a clear understanding of the target market, potential corporate partners/sponsors can't be identified and therefore developed as potential supporters of the brokering system. Once the market for the brokering service is identified the relationship

building, promotion and documentation activities described above must be used to actively develop and maintain connections with the identified market of users.

Communities – As highlighted above, engagement with the community of potential users of the brokering platform is critical to the success of both the corporate and product/service sales models. As the funding generated by both of these models requires demonstration of community engagement (including use) or actual use lack of engagement puts both models at serious risk of failure.

Human Resources – The revenue generated through corporate support would typically consist of negotiated funding amounts that are delivered over a specified period of time, in many cases providing an opportunity for renewal if the continuation of the agreement appears beneficial to both parties. In the case of corporate sponsorship the agreement may be for a fixed amount while an agreement that provides in-kind contribution of resources may provide resources in proportion to needs up to specified cap. In the case of a corporate membership model the funding provided may potentially fall into different membership levels, with the membership tiers providing differential member benefits to the member companies. The indirect membership model may provide either fixed funding or proportional funding based on the number of organization members that will have access to the brokering service/platform.

The product/service sales model provides a revenue stream that is directly proportional to the number of items "sold" or delivered. In this case the actual revenue generated may scale with demand on the system, but may also fall below needed minimum levels for maintenance of core system functionality (i.e. base operating costs).

Software Engineering – In the case of both corporate support and product/service sales the system must have capabilities that are well aligned with the needs and requirements of the sponsoring organizations or customers. In both cases the development of an agile development model that provides for an ongoing exchange between users/stakeholders and system developers must be developed. In such an agile development model emerging needs can be rapidly developed and tested and if proven beneficial can be integrated into the system with a minimum of time or effort.

Product management – The product management model must align with an overall platform that is flexible and extensible. This is required to meet the (often rapidly) evolving needs of the users and partners from whom funding is directly obtained. Without a responsive product management model the alignment of system capabilities with user/sponsor needs can drift apart, yielding an opportunity for rapid drop-off in obtained funding.

Advantages

- Model has the capability to provide steady, long-term funding
- This moedel can contribute to a business diversification strategy for resilience
- Could build partnerships with membership organizations (i.e. corporate [non-profit or otherwise] sponsors) to provide premium level service access

Challenges

- May require significant promotion and effective communication with corporate sponsors
- Expectations of “free” on the internet works against buy-in for the premium tier of freemium
- Tension between profit motive for Corporate partners and productive revenue capacity
- Challenge of differentiation between free service level and premium service level
- Association with some corporate sponsors may disenfranchise some users
- Providers of free data will not want distributors to profit from it if there is no added value

Consortium Model

Model Description

Organizations (Universities, non-profits, for-profits, foundations, individuals, etc.) establish a consortium based on a common mission to create efficient and enhanced use of resources. In this case, the Consortium provides a sustaining environment for brokering middleware through provision of infrastructure, financial support, maintenance and community engagement. The Consortium can be a single infrastructure or may provide distributed support through its members.

Successful Instantiations include:

- iRods
- OGC
- CUAHSI

Mapping Revenue Models onto Sustainment Attributes

Funding – Start up funding for the Consortium is often provided by a single entity (Foundation or other funding source). Continued support comes from Consortium membership fees (annual or other). Sustaining funding requires sustained membership by a critical mass of organizations or external funding sources may be required due to membership instability.

Users – Significant ongoing effort may be required to identify and maintain the user community beyond the Consortium members. Alternatively, the Consortium may represent the primary user communities and further marketing efforts may not be required.

Communities – Community engagement is a significant effort in the consortium model and may require full time staff to maintain and increase membership.

Human Resources – The Consortium is typically governed by a Board of Directors who provide oversight and long range planning. Software and systems development and maintenance is planned, budgeted and implemented on an annual basis to meet the prioritized needs and objectives of the Consortium. The support infrastructure may be distributed across member organizations, or can be centralized in a single facility.

Software Engineering – Software development and/or maintenance is provided internally by an expert/advocate within the consortium body or may be subcontracted. Software may also be developed externally and adopted by the Consortium. There is also the possibility of Consortium members contributing to the development and maintenance of the broker as well as infrastructure support.

Product management – The brokering middleware is managed to meet the needs of the Consortium and the communities it serves. Input to this process comes through the Consortium members. There may be a single product manager within the consortium umbrella or a Consortium member may agree to provide the product management services.

Advantages

- Consortium members provide a ready user base and connections to community
- Consortium members may provide expertise to contribute to open source software sustainment

Challenges

- This model requires a home infrastructure base and technical support
- Significant and ongoing community and membership development effort
- Needs advocate and expertise within the communities supported to maintain broker

Software Lifecycle and Sustainment

The development and use of brokering middleware in a research environment is a relatively new development. More mature middleware such as the libraries associated with the Android operating system have adopted business and revenue models that promote sustainment. Often the approach to sustainment is to apply functionalities from various theoretical business/revenue models to create a hybrid approach to secure the longevity of the software. Broker software has now attained a level of maturity that necessitates consideration of how the middleware will be sustained. The Working Group does not presuppose which business/revenue or hybrid model might be embraced to sustain this middleware, but rather offer examples of endeavors in support of research and educational activities that rely on middleware for their success. Some of the examples can be related to the class of revenue models outlined above and others represent a hybrid of business/revenue models. For each example, the strength and weaknesses in the context of long-term sustainment of broker middleware are discussed. In all examples, the middleware represents a functional element of a large activity for which support is obtained because of value of the layers on top or below (or interest in both) the middleware interface. For example, middleware might connect a user interface (top layer) to several databases containing unique and valued data sets (bottom layer). Sustained support for activities that connect users to data sets come from an interest in advancing knowledge through these connections and not in support for the middleware that facilitate these connections. To make the discussion and ultimately the recommendation more clear, two use cases are discussed in this section. One (Unidata) has a base in government funding and the other (WordPress) is an open source web tool that exemplifies a SaaS operation across a broad range of communities.

Unidata (government funding)

The mission of Unidata is to provide the data services, tools, and cyberinfrastructure leadership that advance earth-system science, enhance educational opportunities, and broaden participation. The program had its origins in the community lead activity to provide near-real time atmospheric data to academic institutions to enhance the research and educational experiences of faculty and students. For the past thirty years the program received sustained and continuous support from the National Science Foundations, because it assisted that agency in fulfilling its mission and demonstrated a record of broad community engagement and governance. In addition, the managing organization is a consortium of US academic institutions with established credentials in atmospheric and related sciences.

Unidata's vision calls for providing comprehensive, well-integrated, and end-to-end data services to meet the needs of the geosciences community. To achieve this vision, several middleware programs were developed, refreshed, and sustained to allow seamless integration of display and analysis interfaces with near-real-time data streams and static data sets. NetCDF and Local Data Manager (LDM) are two examples of Unidata created and sustained middleware. At least one of these products, NetCDF, with over 300,000 downloads, has found a user community much larger than the stakeholders it was original created to serve with over 300,000 downloads. These middleware examples are mature software products that demonstrate the characteristics relevant to sustainable software:

Funding – long term government support is justified because the Unidata program helps the National Science Foundation achieve its strategic objectives which include but are not limited to integrating education and research to support development of a diverse STEM workforce with cutting-edge capabilities and providing world-class research infrastructure to enable major scientific advances.

Community engagement – there as thousands of world-wide users of Unidata software and data streams from hundreds of academic institutions and government agencies. Unidata established a policy and user committees made up of stakeholders that help guide the program's activities.

Human resources – the program supports a diverse staff that range in expertise from software engineers to technical project managers. In addition, stakeholders from the academic community contribute to guiding the software development and are activity engagement with the Unidata staff in the implementation process.

Software Engineering - the development of the middleware was done by professional software engineers using the best practices that include extensive testing and documentation. Software refreshment and sustainment continues under the leadership of information technology professionals.

Product management - several Unidata personnel are engaged in management the middleware software used by the community. There are many activities such as managing licenses, announcing new project releases, and tracking response times to user requests for help.

Clearly a potential weakness of the Unidata program is that it depends almost entirely on one source of revenue, the Division of Atmospheric and Geospace Sciences of the National Science Foundation. Attempts to broaden the user base and funding sources over the years have met with limited success. However, Unidata has a long history of continuous support and continues to provide high value to its community.

WordPress (Software as a Service)

WordPress started in 2003 to enhance the typography of everyday writing and "with fewer users than you can count on your fingers and toes". Since then it has grown to be the largest self-hosted blogging tool in the world, used on millions of sites and seen by tens of millions of people every day. WordPress is an Open Source project; everything from the documentation to the code itself was created by and for the community; the core software is built by hundreds of community volunteers, and there are thousands of plugins and themes available to transform a website site into almost anything "that can be imagined". Though largely developed by the community surrounding it, WordPress is closely associated with Automattic, and in 2010, Automattic handed the WordPress trademark to the newly created WordPress Foundation, which is an umbrella organization supporting WordPress.org. In addition, WordPress.com was created to provide business and community support for more advanced web design and web communication. This sells subscription services at various levels ($99, $299, …) to help build websites and handle communication and also adapt open source developments to the business marketplace, engineering open source contributions for stable, high leverage operational environments. Thus we see SaaS complying with both the open source community and services to a business customer base.

Funding – Long- term support comes from SaaS sales to the business community and individuals/organizations that desire an effective and consistent message. This was a bootstrapped effort over a ten-year period. Current revenues are about $45M/year and venture investments have started recently as the originators look at "exit strategies".

Community engagement – Community engagement starts with "happiness engineers." To quote Andrew Spittle, a WordPress happiness engineer: "we are passionate about making the web a better place." The success of this approach is seen in the numbers. WordPress continues to grow fast, with 50 million users today compared with 4 million five years ago. It has built the community through open source developments with a management approach that both encouraged community participation and understood that there needed to be a layer of formal software development and management in order to support institutional demands. The result is that WordPress.com has about 500,000 paying customers.

Human resources – WordPess runs their business with about 400 employees. They are a virtual company with no headquarters/buildings. They offer substantial benefits and propagate a strong team spirit.

Software Engineering – WordPress has professional software engineering that work alongside customer relations staff. WordPress has an effective customer blog to understand feedback on software developments.

Product management – WordPress has a strong product management culture closely connected with their user communities. They do not manage the open source component, but do encourage development and are responsive to understanding and capturing/hardening open source contributions. They do feed some of their commercial development back to the open source community.

WordPress is an example of the new generation of software innovations with a focus on customers and community support. With its commercial component focused on a key marketplace – branding and communications – it has captured a major market share across a broad customer base. However, the market environment required to support middleware designed for research purposes may be different as the public face of brokering would not be addressing a visible need such as branding. The research community does not offer the same depth of opportunity. On the other hand, brokering is gaining acceptance in diverse research and business applications and if these markets can be tied together to create a critical mass, serious consideration of SaaS is justified.

Consideration of Community requirements

Although business, revenue and/or hybrid models for sustaining software can be identified, these paradigms will be most successfully applied to the research community (as represented by the RDA) when they are melded to fit community cultures and practices. Researchers and educators, who have significant representation in RDA, often favor software that is low cost, open source, and with a development and refinement cycle driven by community to evolving community needs. Open source software is favored because it is relatively inexpensive to acquire and refine for local use. The governance mechanisms that often develop to support open source software tend to create practices and standards that are amenable to community practices. These, if well supported over a long duration, may lead to wider adoption as well as to promote interoperability standards across communities. It has been suggested that open source software in the research community has more sustainability since there may be many stakeholders may be involved in the software development and use, and because the voluntary contributions of time and ideas help contain the cost of sustainment. However, there are other factors such as (1) community culture and practices and (2) the burnout of volunteers needed to provide long-term continuity that might mitigate these potentially positive attributes.

Synthesizing a Business Model for Brokering Software

After considering the advantages and disadvantages of the business models for software sustainment, the RDA Brokering Governance Working Group recommend a hybrid business model to support future sustainment of brokering middleware for research data interoperability. Our convergence upon the hybrid model was based on factors that we believed could best contribute

to the sustainment of the broker while operating within the research and education cultural environment. One important factor was that we believed that researchers would not be willing to pay significant usage fees for the broker service since they are used to working in a funding limited environment. Additionally, a broker is designed to serve many different communities (in fact working across communities) and therefore finding a single community willing to support it would be unlikely. Finally, establishing the critical mass of users is essential to sustainment. This could not be accomplished by putting an unknown entity into operations without a substantial investment in marketing and community development. These were the primary considerations that have led us to a hybrid model recommendation.

Consistent with these considerations, we summarize the strengths of three business models that we feel are most relevant for sustaining a brokering middleware for research.

Federally Funded Data Facility Guardianship

The federally funded data facility model is a strong support model for the sustainment of brokering middleware. It offers a mechanism for assessing ongoing community value as well as stable funding for operations and maintenance. The data facilities are a proven model with the required infrastructure necessary for sustainment. and in addition they provide an authoritative reputation that engenders trust in the user community. The broker reaches across communities and thus bridges the discipline specific user bases of traditional facilities. In responding to global challenges, the broker could offer increased leverage to data facilities.

SaaS

SaaS is a service model that works either in an open software or proprietary environment. It adapts well to customers and end users needs because of its flexibility and scalability. This offers significant advantage in permitting the broker to support multiple communities, leveraging the startup in the research domain and gaining expanded support though applications in business and government arenas. With the increasing need for access to cross-disciplinary data, brokering has recently seen applications in the business community and has been supported when the broker also supports core organizational missions. Thus a SaaS model synchronized with a guardianship or consortium model is consistent with current modalities of operation.

Consortium Model

The Consortium model requires significant commitment and investment to develop the membership and establish an infrastructure to support the broker and consortium itself. This is a viable model but is likely best implemented under the umbrella of an existing organization (e.g. UCAR, iRods etc.). One benefit to the Consortium model is that it provides the opportunity for members to contribute to the future development and direction of the brokering middleware.

Broker Business Model Synthesis

No single solution appears to provide a sustainable business model for brokering software. Through this analysis however, we have converged on aspects of individual business models that together form a hybrid model that we believe could contribute to the sustainability of a broker for research. In defining the recommendations, we began with the assumption that the broker “core” is a fully functional and well-documented middleware package, ready to be adopted by an

organization for ongoing operations and maintenance, with the capability to be customized to user needs. Also, we assume that in its early operations, the broker will be focused on serving the research and education (RDA) communities.

The hybrid model can therefore be described in terms of recommendations for the sustainability of a broker for research.

1. The broker must live within a larger enterprise and not standalone. That is, as middleware, the broker will not have the visibility on its own to generate and maintain sufficient awareness and support among the diverse user communities.
2. The government guardianship model provides the best opportunity to establish a strong research user community while providing stability of support in the early to mid phases of the lifespan of the broker. This model can and should be supplemented with external support (financial or in-kind) to diversify the funding model and the user community commitment. A diversified user community is essential to the sustainment of the broker. This may take the form of a Consortium or other contributive support. During this phase of support, it is essential that a long-term support strategy is planned and eventually executed. The timing of the transition to the long-term support strategy will be dependent upon the circumstances of the participating support organizations.
3. The mature phase of sustainment for the broker may be provided by a continuation of the Consortium or by offering the broker as Software as a Service. We believe both of these models may be successful independently or together, complementing each other.
4. Expansion of the user base for brokering across communities that have needs synergistic with scientific research and development can provide a critical mass of users for sustainment. This can include, for example, applications in geospatial data applications, social science, humanities, environmental management and/or business applications.

A hybrid business model provides the greatest flexibility to meet the requirements for sustainment of a broker for research. Here we have considered the sustainment of the broker to have two post-development stages, an initial phase where there is a need for stability while the user community is established and the value of the broker is demonstrated, and a second phase encompassing a more mature broker, in which there is ongoing maintenance and improvement to continue to meet the needs of the user community. We also recognize that all software, including brokering middleware, has a finite lifespan that is determined by its utility to the user community, and the community's willingness to provide support resources.

References

Daconta, M. "Cloud broker software an emerging force for enterprise migrations." *Retrieved May* 2 (2013): 2013.

Lapham, et. al. 2006. Sustaining Software-Intensive Systems, Technical Note: CMU/SEI-2006-TN-007. May 2006

S. J. Khalsa, S.J., S. Nativi, and G. Geller, "The GEOSS Interoperability

Process Pilot Project (IP3),” IEEE Trans. Geosci. Remote Sens., Special Issue on Data Archiving and Distribution, vol. 47, no. 1, pp. 80–91, Jan. 2009.

Dr. Karl Popp: Software Industry, Software Ecosystems and partner business models in the software industry

Marshall, James J., Robert R. Downs, and Shahin Samadi. "Relevance of software reuse in building advanced scientific data processing systems." *Earth Science Informatics* 3.1-2 (2010): 95-100.

Mattmann, C.A.; Downs, R.R.; Marshall, J.J.; Most, N.F.; Samadi, S., "Reuse of software assets for the NASA Earth science decadal survey missions," in *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, vol., no., pp.1687-1690, 25-30 July 2010
doi: 10.1109/IGARSS.2010.5653018

Nalley, David 2015. The Tragedy of Open Source, presented at the Cybersecurity Summit, Arlington, VA http://www.slideshare.net/ke4qqq/the-tragedy-of-open-source

Nativi, S., M. Craglia and J. Pearlman. 2013. Earth Science Infrastructures Interoperability: The Brokering Approach. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 6, NO. 3, June 2013 pp. 1118-1129.

Open Source Initiative, 2015. http://opensource.org, 855 El Camino Real, Ste 13A, #270, Palo Alto, CA 94301

Popp, Karl Michael, 2015 “Software Industry , Software Ecosystems and partner business models in the software industry”, http://drkarlpopp.com/revenuemodelssoftwareindustry.html#.VmRRXOnRn0A

Williams, Roy H. 2011. Earth Science: New Methods and Studies, CRC Press, ISBN-13: 978-1926692579

Appendix
Overview of Data Brokering

The broker paradigm

The Broker model is used to structure distributed systems with decoupled components, which interact by remote service invocations. Although the broker provides needed functionality in structured systems, it provides the greatest benefit when components are in less structured systems where components still need to know each others location and other details that are necessary to exchange information. Broker middleware is responsible for the coordination of communication among components: it forwards requests and transmits results and exceptions. Using the Broker paradigm means that no other component other than the broker needs to focus on low-level inter-process-communication. Thus brokering middleware can be used to add functionality to the exchange of information to a relatively unstructured and uncoordinated set of components containing data sets, for example.

Different definitions

Traditional definition (akin to a stockbroker or commodity broker)

An intermediary software that assists a client application to navigate through a complex supply environment of many options.

Alternative (and more general) definition

A single piece of software making other heterogeneous pieces of software (i.e. software systems) work together.

Information/Data Services Broker in SOA (more specific) definition

In the service-oriented architecture (SOA), Broker is a software agent that brings together service consumers (clients) and providers (servers) for data/information sharing. Broker is deployed on a third-party tier –i.e. is a “middleware” component.

Examples of implementation

The brokering paradigm can be applied to a variety of technological environments and architectural approaches such as:

Brokers in SoS (System of Systems)

In a SoS (or metasystem), a Broker can be defined as a software framework that supports the interconnection of the constituent autonomous (and heterogeneous) enterprise systems by providing mediation, transformation, and quality of service (QoS) control capabilities in order to simplify cross-disciplinary discovery, access, and use of data and information.

Brokers in Web 2.0

In the Web 2.0 paradigm, Brokers can act as Business and Data Mashup Enablers addressing heterogeneity challenges, such as: (a) providing access to the different data/content suppliers, (b)

providing harmonization services to support the mashup process, and (c) transforming incompatible IT-based resources into a form that allows them to be easily combined.

Cloud Broker software

A new type of software that sits on top of cloud providers to abstract, simplify and map various cloud offerings to your environment. Cloud broker software assists organizations in creating solutions in the cloud, migrating solutions to the cloud and moving solutions between clouds (Daconta, 2013).

Data/information broker traits

The Broker that (effectively) works in a heterogeneous environment containing multiple remote objects that interact synchronously or asynchronously typically demonstrates the ability to:

1. Finalize requests on behalf of its clients against a vast supply system –e.g. by transforming different interoperability protocols;
2. Support many clients at the same time in a dynamic way;
3. Access large, distributed, and heterogeneous supply systems in a dynamic way;
4. Be fully autonomous from its clients and accessed supply systems;
5. Be flexible and configurable (even at run-time);
6. Be extensible.

Note on Open Source Software

Open source software is commonly used, and often favored by, the research community. Since open source software is developed through community contributions, it is likely to conform more closely to community customs, standards and practices. In addition, open source software may implicitly promote sustainability since there may be many stakeholders and the voluntary contributions of time and ideas help contain the cost of sustainment. Experience shows that governance models as well as interoperability standards are often engendered through the process of creating and sustaining open source software [Opensource Initiative 2015; and Williams, 2011]. Balancing these benefits is the fact that sustainment over the long term is not guaranteed and there are examples of important community software that is maintained by only one or two people (Nalley, 2015). Voluntary contributions is often motivated by volunteers having concurrent grants to further the evolution of software tools. When these grants end volunteer participation may decline.