Property Interests Product (PIP) Collaborative Research Project

Project Prototype Review and Technical Architecture Specification Overview

A Joint Government and Industry Research Initiative for the Development of the Queensland Spatial Information Infrastructure (QSII).

Information Integration Unit
Department of Natural Resources
Level 5, Landcentre
Woolloongabba, Brisbane

Table of Contents

1 Executive Summary

The purpose of this document is to:

1. provide a review of the PIP Prototype application; and

2. identify points that should be considered when developing an open specification.

The PIP project has provided an excellent opportunity to develop and evaluate state of the art systems that enable cross agency sharing of spatial data. The participants have gained valuable knowledge during the course of the project, which would be essential to any further initiatives. While some of this knowledge and experience can be documented as done so in this report, most is retained collectively by the participants as a group of individuals that worked together. It is therefore important to note that the value of this knowledge and experience diminishes exponentially with time. Collaborative research projects such as PIP, typically have a half life in the order of twelve months or less.

Obstacles encountered during the project could be categorised as technical and organisational. It is clear that the most difficult obstacles encountered were of an organisational nature, as they had the biggest impact on project schedule.

Some of the technical and organisational issues are listed below. The development of each interface to a custodian data source (ie. wrapper) introduced the project team to a variety of challenges such as:

- each wrapper was different and often used obscure data in the custodian business process, highlighting the detailed knowledge required by the custodial personnel;

- some of the development tools used were not compatible with existing custodian systems, requiring “workaround” solutions that led to inefficiencies;

- wrapper development was hindered primarily by lack of documentation; and

- wrapper efficiency could be improved if the wrappers called directly to the data bases rather than the legacy application environments.

The speed of the application was limited by the large size of the data sets involved and the relatively slow speed of a dial-up connection where:

- services of low bandwidth communication infrastructure provided limited spatial data transmission; and
- caching of spatial data was required to improve performance for services on low bandwidth communications.

There were a number of organisational issues that impeded the development of the prototype (further defined in the *PIP Technical Issues Report* PIP99d):

- lack of technical documentation - this particularly slowed the development of wrappers;
- lack of resources at participating agencies - this slowed the development of a secure environment for the prototype to operate within. (The prototype duplicated legacy systems.);
- organisational issues - such as legislative and privacy issues with data, lack of knowledge about integration processes at custodian agencies, and the lack of a single business unit (point of contact) at custodian agencies;
- organisation technical changes – such as changes to business relationships with collaborators and custodian computing infrastructure changes; and
- time constraints – such as the time required to replicate custodian systems, allow collaborators to develop software, and to provide programmers access to the computing environment through dial-up connections.

The PIP architecture consisted of custodian databases, an application composed of a browser applet and a servlet, wrappers that facilitated custodian data, and a network infrastructure.

The prototype determined a series of key outcomes that related to the:

- organisational aspects of custodial data
- size of data sets;
- speed of the supporting network; and
- application requirements.

The Technical Architecture Specification Overview provides a *descriptive* method of defining the PIP Architecture and elaborates on the various components of the architecture. A consensus was formulated between collaborators on the most appropriate way to describe an architecture where the components include:

- infrastructural facilities — composing of security, accounting, and quality of service and monitoring issues;
- applications layer — the layer with which the client interacts;
- integration services — provides accessibility and integrates marketable data; and
- custodian services — the provision of data by the custodian.

2 Introduction

The Property Interests Product Collaborative Research Project (PIPCRP) was a joint venture between five State Government Departments:

- Department of Natural Resources *
- Department of Environment and Heritage *
- Department of Mines and Energy *
- Department of Main Roads
- Department of Communication and Information, Local Government and Planning

and five Local Governments:

- Caloundra City Council *
- Balonne Shire Council *
- Cairns City Council
- Gold Coast City Council
- Brisbane City Council

in partnership with the:

- Commonwealth Scientific and Industrial Research Organisation (CSIRO)
- Distributed Systems Technology Centre Pty Ltd (DSTC)
- DASCOM Australia Pty Ltd
- Inprise
- CITEC
- CiTR Pty Ltd

*Note: the State Government Departments and Local Governments marked with an asterisk * following were the key collaborators in the prototype.*

A component of this project included the development of a prototype for the Property Interests Product over Queensland. The prototype tested the establishment of a land information marketplace that would provide streamlined access to property related information held by State and Local Government with a view to modernise access to current disparate land administration systems.

The technical hypothesis tested in the PIPCRP the principle that custodian legacy databases should not need to be significantly redesigned to be integratable into the Queensland Spatial Information Infrastructure (QSII). To achieve this, an open integrating environment was constructed to allow easy interfacing, in real time, between legacy systems and the provision of open access and connectivity to retailers. To test this hypothesis, the technical component of the PIPCRP used existing spatial information management environments (custodians) with an integration and web capability progressed from the Queensland Electronic Services Trial (QUEST) Project. The Quest project [QUEST98] was designed to test the concept of an electronic marketplace for spatial information with a major focus on access via the Internet.

The PIP prototype discussed in this report demonstrated the ability to integrate data between various State Government data custodians, Caloundra City Council and Balonne Shire Council. This document reports the experiences acquired from the development and use of this software system and establishes the needs and requirements that must be addressed in a future tender specification for a commercial variant of the full PIP infrastructure.

It is intended that the experiences gained from the development and operation of the PIP prototype software will provide QSIIC with a *true* understanding of the various pieces of information required to prepare an open technical specification, including the:

- various technical and information infrastructure component services;
- benefits of a collaborative integrated information management environment;
- collaborative relationships; and
- the costs and resources required.

3 PIP Prototype Review

The *PIP Prototype Specification* [PIP99c] contains detailed application specific requirements for the PIP prototype.

3.1 PIP Architecture

The PIP prototype architecture is based on the Internet Marketplace concept. This concept allows applications to access a wide range of services in an Internet environment that are advertised in a common 'registry' and accessed via a simple standardised interface. The PIP Prototype implementation of this architecture is shown in Figure 1.

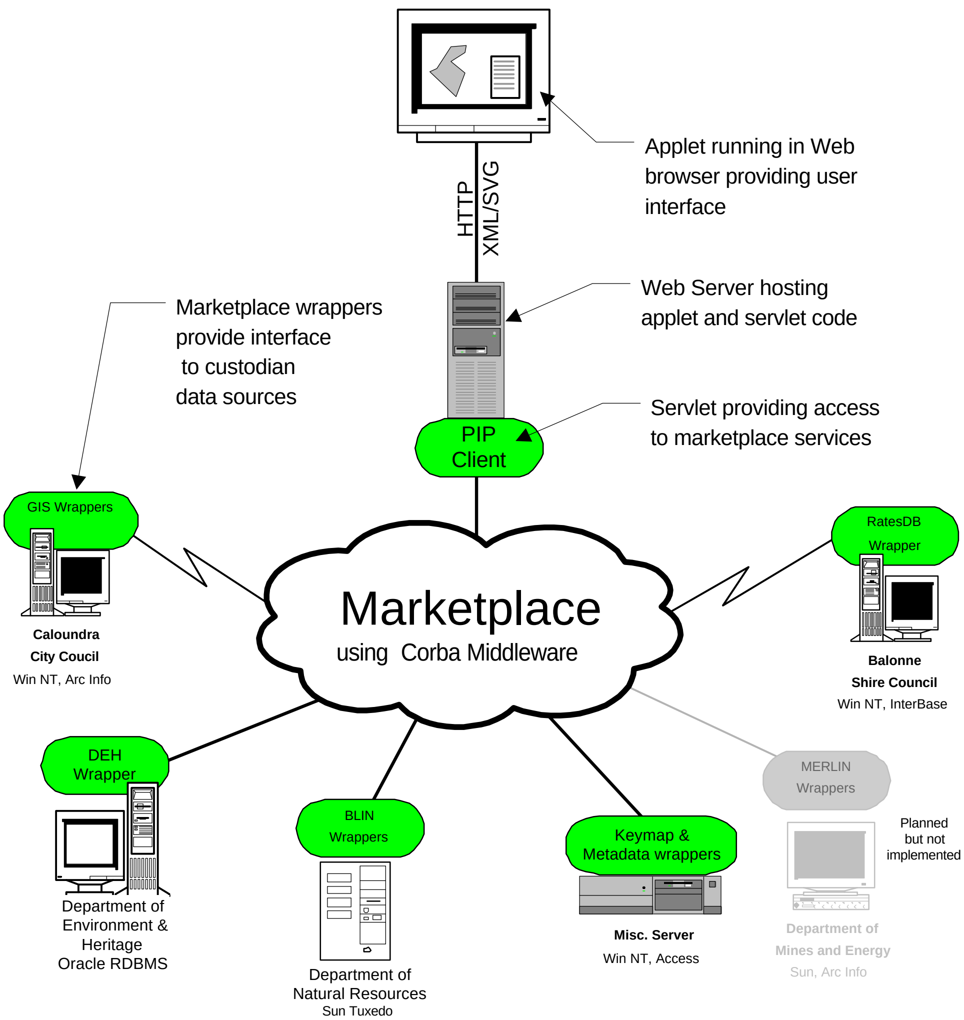

Figure -1: Major Components in the PIP Prototype

The PIP sits on top of a simple message-passing paradigm, whereby applications send requests to custodian service wrappers. Requests are expressed using the request specification language (RSL), while results are presented in a limited number of pre-determined formats.

The components are:

- **Custodians**: the custodians are the ultimate source of the data sets published for use within the marketplace. Data is published through the auspices of custodian services. Custodian services accept RSL queries and return data in a limited number of exchange formats via the wrapper interfaces.

- **Application**: PIP applications consisted of two components: the browser applet and the servlet.

  – **Browser/Applet**: constitutes the interface to the user; issues requests to the applet via standard HTTP protocols and is responsible for the display of any resultant data (returned as XML/SVG data streams).

  – **Servlet**: associated with the browser/applet is essentially a marketplace client and has access to the data servers. The servlet is in effect a proxy for the applet and enables more efficient processing of data and keeps the applet component 'thin' and simple. Both the servlet and applet are discussed in more detail in section 3.4.

- **Wrappers**: each custodian data set is advertised to the rest of the system and accessed via a wrapper service. The wrappers built for the PIP prototype are discussed in section 3.5.

- **Network infrastructure**: PIP utilises networking technologies including Telstra's Dial Connect dial-up modem/PPP service that connects to DNR's frame relay network as well as DNR's 'Ethernet' LAN links.

3.2 Key Outcomes

The key outcomes from this prototype cover many aspects, including:

3.2.1 Organisational

It can be argued that this was the major issue faced by PIP prototype. Obtaining access to data sets from custodians proved difficult in some cases. There were two main causes for these difficulties:

- Lack of custodian resources. In some cases data was replicated onto machines assigned to the PIP project, rather than being extracted 'live' from a custodian. (See PIP99d for further details.)

- Identifying the officer with the appropriate knowledge proved difficult. This can be traced to the popular idea of 'outsourcing', which means that a custodian may not be familiar with the operation and structure of their own data sets. An associated issue involves the difficulty of finding an officer with the authority to make data available. (See PIP99d for further details.)

The construction of the various developer teams is also of interest. PIP utilised one team for each custodian data source, plus a further team associated with integration of the data for presentation to the user.

Integrating the various software components of the PIP project took place quite late in the development process and was accomplished relatively easily due to the clear software interfaces and well defined areas of developer responsibility.

3.2.2 Technical

There were few technical issues arising from the PIP prototype.

Significant issues include:

- Currency and accuracy of documentation. In some cases, documentation was extremely minimal: more than once, the only way to capture vital information was from face-to-face meetings with key custodian personnel or contractors to the custodian. (See PIP99d for further details.)

- Validity of data sets. Since a data set is taken from each custodian "as is", there is a problem of ensuring the correctness of that data relative to the related data obtained from another custodian. This was most apparent in the registration problems that were seen when graphical data was amalgamated from the keymap service and BLIN. This issue has been examined in *PIP Project Information Infrastructure Modelling for the Property Interests Product* [PIP 99b].

3.2.3 Size of data sets

In the case of the Caloundra system, large graphical data sets were generated for any graphical query. This was traced to poor query response characteristics of the original data set. To improve performance, a caching system was developed that held the Caloundra graphical data in a separate store and directed all relevant graphical queries to this store. This proved much more efficient than presenting the query to the original ARC/Info data store. It is anticipated that further development of the custodian system could obviate the need for this separate cache.

3.2.4 Speed of supporting network infrastructure

The supporting network infrastructure covers various bandwidths such as a 10Mbps Local Area Network and modem-based links operating at between 42 and 50Kbps. In practice, it was found that the lower speed access links gave acceptable performance for non-spatial data. Performance was unacceptable for queries that resulted in large data sets (>200k bytes) being retrieved.

3.2.5 Application Requirements

The prototype's graphical user interface (GUI) was one of the main areas where improvement was requested. This is not surprising, given that it is the most visible component of the system and is thus very important for the system's end-users. Since the prototype was based on a manufactured example, it is difficult to judge the performance of the GUI without a clear understanding of user's task requirements. The need for constant, active feedback has proved important. Careful attention should be paid to user requirements, where end users should be actively involved with the integration process and the specification and development of the graphical user interface for a production system.

3.3 PIP Implementation

To a large extent, the current PIP prototype has been built on much of the infrastructure utilised in the earlier QUEST prototype [QUEST98]. However, a number of key differences existed between the two prototypes. The PIP prototype provided the following enhancements/extensions:

- integration of a larger number of custodian data sets from a wider variety of sources, including Oracle, ARC/Info, the SCDB, InfoBase and BLIN;
- adoption of the standards-based eXtensible Markup Language (XML) to describe and transmit spatial data and associated attribute data;
- operation across network components offering various bandwidths such as a 10Mbps Local Area Network and modem-based links operating at between 42 and 50Kbps; and
- a number of performance enhancements, including caching of some large spatial data sets.

3.4 PIP Applications

The PIP prototype consists of four applications:

- Development Proposal Analysis Service
- Tenure and Survey Analysis Service
- Built Environment Analysis Service
- Acquire Data Set Service

For reasons of efficiency and code optimisation, the PIP applications consisted of two components:

1. the applet (written in Java) which ran in the client browser and was responsible for user interaction, graphic and attribute presentation; and
2. the servlet (also written in Java) which ran at the web server and was responsible for data fusion, sending queries to data servers, and accessing the metadata repository.

This approach has a number of advantages:

- It minimises the code required to run in the web browser at the client machine —"thin" clients are generally faster and easier to manage in environments with many users.
- Common functions, such as query formulation and data fusion are better handled by the servlet, further reducing the load on the client applet.
- It reduces the bandwidth required by the client applet, in that the servlet is responsible for connecting to multiple data servers (or services). The servlet in effect is a 'proxy' for the applet.

- Both applet and servlet code are maintained by the web site host. Any organisation or department may choose to host a web site, which can be entirely separate to the data services hosts.

This approach allows for extensive user interaction with the datasets, without the need to reissue requests for data as the user pans and zooms a given region. Typically, for spatial applications to be supported across **all platforms**, it is not possible to provide the 'live' graphic data. Instead, a simple graphic is generated as an image embedded in an HTML document. This is then delivered to the browser. Such an approach limits the amount of user interaction and can be wasteful of bandwidth if the user needs to pan and zoom the image frequently. Since the applet used the 'Swing' component GUI library supplied with Java version 1.2, it is only able to run within a browser with the most recent version of Sun's Java 'plugin' software installed.

Client application development was a relatively straightforward exercise. The user interface requirement for each application is similar. Each application has a graphic display and a text display. The Java language was used to develop the applications that ran as applets within a web browser. However, these issues did arise:

- Requests for data generated by the user interface result in a single complete stream (after it has been collected from various servers) of information delivered to the client. For spatial data, lengthy delays can be experienced due to large volumes of data being transmitted over low bandwidth (dial up) Internet connections.

- Experience with the PIP prototype indicates that for dial up connections at speeds in the range 33.6-52kbps, PIP applications would be limited depending on the nature of the queries performed. The query window produced by the client application in a production system must be validated against available bandwidth. This will ensure realistic response times.

- While Java claims to be platform independent, its lack of maturity and rapid evolution over the last two years has resulted in a number of incompatibility and performance issues. It would be non-trivial to develop client applications for PIP to run on Windows NT/95/98/3.1 and Unix platforms using a variety of Web browsers. An alternative approach would be to allow applications to support the picture case (embedded image) for situations requiring cross platform portability, as well as the graphic element case where more sophisticated user interaction is required.

- The individual wrappers delivered graphic elements to the servlet as SHAPE files (the format used in the earlier QUEST prototype). These were then converted from this 'closed' format to XML/SVG[1] and then delivered to the applet that decoded and displayed the data. XML/SVG is a text based data format specifically designed for vector graphics. This format typically produces large data volumes. However, this is not a significant problem because text based data streams are often highly compressible.

- XML also helped simplify the handling of attribute data from participating services. Since the completion of the PIP prototype, DBMS vendors have released tools for automatically encoding database query results into XML.

[1] Currently at working draft stage. See http://www.w3.org/Graphics/svg

3.5 PIP Services and Wrappers

CORBA services were constructed to interface the various custodian data sets with the PIP system. These 'wrapper' services were built to deal with the mechanical aspects of retrieving data from the custodian data sets and also to perform the following functions:

- advertise the service to a naming service so that applications can locate and use the service;
- accept incoming queries (expressed in the Internet Marketplace's query language "RSL") and perform the operations required to satisfy the query; and
- format the resulting data from the query into a generic format to be delivered to the client.

Note that the wrappers performed little processing on the data that they deliver.

While the wrappers are relatively small and simple pieces of software compared to other software used in the PIP prototype, there are a number of issues that needed to be addressed:

- Following on from the lessons learned in the earlier QUEST project, it was decided that VisiBroker 3.4 would be used for the CORBA Java development environment rather than use the CORBA ORB supplied with the Java Development Kit. This decision was made to ensure that the CORBA system would be capable of dealing with very large data structures that may be created in response to a spatial query.
- The insufficient quality and quantity of documentation available for most services required the wrapper writers to have numerous meetings with the respective developers to obtain support.
- It was necessary to use various Java Virtual Machines. Microsoft's implementation was used for the Keymap component, while the other parts of the system used the standard Java 1.2.1 release.
- None of the wrappers required the full power of CORBA; most provided a simplified RSL parser implementation compared to QUEST wrapper software.

The development of each wrapper introduced the project team with a variety of challenges. The development of each wrapper required a different approach because they were accessing different system architectures and software products. The development times also varied due to the different complexities.

- The **BLIN and SCDB wrappers** required to handle both simple and transactional aspects of the underlying 'MAPPING' Tuxedo service.

  Each of these wrappers required approximately 10 person-days of effort.

- The **DEH wrapper** gained access to the EMR/CLR Oracle database in an obscure way. The wrapper had to make a query, throw the resulting data away and then check a table modified as a "side effect" of the original query—it is this "side effect" data that is returned by the DEH wrapper.

In addition, since the Department of Environment and Heritage had no spare machine, it was necessary to replicate their Oracle 7.3.4 installation and load a copy of their data into the database. These were straightforward procedures.

This wrapper required about 8 person-days of implementation effort.

- The **Balonne wrapper** confronted the issue of documentation where very little information about their database format was available. It also became necessary to obtain a copy of Inprise's "Interclient JDBC Driver" software to facilitate access to the Balonne database.

  This wrapper required about 8 person-days of implementation effort.

- The **Caloundra wrapper** needed to interface with ESRI's ARCInfo software—a relatively 'closed' system. Help was required to design new access methods for the data set. The returned data suffered registration problems.

  The speed of retrieval for the spatial data also was too slow. The solution required the introduction of a cache to ensure adequate responsiveness.

  This wrapper required about 18 person-days of implementation effort.

3.6 Middleware Technologies

The intention of the PIP prototype system was to make a system as independent of the supporting middleware as possible.

Middleware refers to the software that sits between the user interface and any backend servers; enabling them to communicate with one another using agreed and understood standards. In addition to specifying a communications standard, middleware can also provide server replication, increased reliability, location transparency, and other "value-adding" features, at a far lower cost than if these were to be implemented directly by application developers.

Middleware can be provided in the form of standard services and/or high-level programming abstractions that insulate the client and server code from changes to the network, computer hardware, and protocols and make it simpler to extend some clients and servers without breaking others.

The function of middleware is discussed in detailed in the earlier QUEST report [QUEST98]. The PIP project follows on from QUEST and uses essentially the same middleware structure as implemented in the earlier project.

4 Technical Architecture Specification Overview

Given the experiences gained by the development of a prototype as discussed in the previous section, the specification of a technical architecture was tempered by two shortcomings of the prototype.

Firstly, a security system could not be tested and consequently detailed security requirements could not be identified through 'first-hand' experience. This shortcoming also removed the option of testing the performance of the prototype over the Internet.

Secondly, the development of the Department of Mines and Energy (DME) application and the proposed Caloundra Counter Tool were cancelled due to time constraints. This resulted in the inability to determine and test these specific requirements. DME can provide much valuable data to a Property Interest Product, likewise the reuse of existing wrappers as the case for a counter tool, could provide significant internal application to custodians. However, these restrictions on the development of the prototype should not hinder any attempt of providing an open specification for a technical architecture.

With this in mind, the architecture defined for the PIP prototype as discussed in Section 3.1 can be considered as having three fundamental layers, with an additional set of infrastructural facilities that are regarded as being *pervasive* throughout all elements of the PIP system.

The three fundamental layers are:

- **Application layer** — provides the revenue-raising applications that are directly related to the end users;

- **Integration services** — retrieves and integrates the various data sets published by the various custodians and is made available by the auspices of the custodian layer services; and

- **Custodian services** — provides the ultimate source to the data sets via the custodians. Data is published through the auspices of custodian services.

Figure 2 below, presents this logical layering structure diagrammatically.

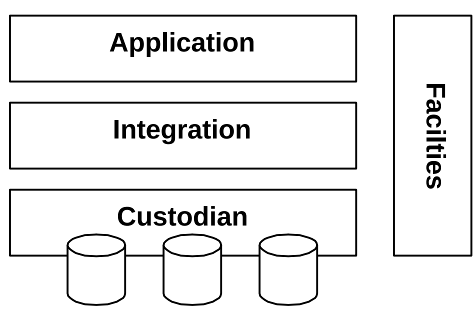

Figure -2: Logical Model for Integrating Services

The necessary functions of each of these layers are presented in the following sections.

Note that the concept of an *architecture* is a *descriptive* device—it lists the component parts of a system, but intentionally says nothing about their ultimate implementation. It may even be that facilities and even whole layers may be omitted from an implementation if it is felt that they are unnecessary or cannot be cleanly separated from other parts of the implementation. Nevertheless, it is important than an architecture canvass all possible matters and attempts to provide a full framework for discussion and design.

It is of course possible for an architecture to take a very different structural path, but it is important to note that the *same functions will essentially be present*. The architecture proposed by the Internet Marketplace (and exemplified by both the QUEST and PIP prototypes) takes the structuring approach shown in Figure 3:

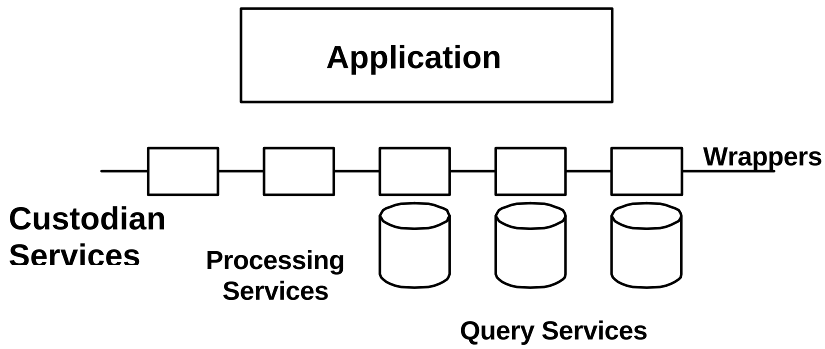

Figure 3: Logical Model for Integrating Services as used in the PIP Prototype

In this case, there are two major layers:

- **Application layer** — provides the end-user application and involves the integration of data from the custodians; and

- **Custodian services** — provides the source of the data sets.

This view combines the integration layer into the application layer. The PIP prototype used this model, where the servlet forms part of the application layer.

This approach (as defined by the Internet Marketplace) may provide benefits in a situation where responsibilities and capabilities are not clearly defined. This is often the case when a system is composed of elements that cross-organisational boundaries.

4.1 Infrastructural Facilities

Infrastructural facilities concern those features that can be regarded as being pervasive throughout a system—that is, they tend to cross the architecturally defined layers. For example, it may be important to determine security access to the overall system, commonly called 'top-level' security, as well as determine what the user can access once the 'top-level' access has been granted. As noted in [PIP99a], 'Security of the system as a whole depends on the security of information sources that are assets controlled by a number of voluntary participating agencies. The reliability of each agency in properly maintaining and using its asset becomes not only a matter of self-interest but also one of mutual concern.'

It is clear that there is a need for the coordinated provision of many of these facilities.

4.1.1 Security

The security component should provide:

- access control to the network as a whole and to each individual custodian service;
- end-to-end security with a strong trust chain;
- mutual authentication (authentication of user to system *and* system to user);
- for groups of users and allow for the assignment of different security levels to groups;
- choice of authentication mechanisms (at minimum, a strong username/password mechanism such as Kerberos and X.509 PK certificate support). An ability to incorporate other authentication mechanisms in the future would be valuable;
- application-level VPN tunnelling with cryptographic message protection (encryption and/or integrity checking);
- fine-grained access control (protected resources may range from network services to application-level objects);
- a flexible model for assigning roles/privileges to users;
- a comprehensive auditing capability;
- suitable management tools for maintaining security information (the ability to delegate control over different parts of the user space or access control rules to different administrators may be important when data custodians are not closely related);
- scalability and fault tolerance; and
- security facilities should be based on standards where possible.

Ideally, integration of security and accounting should also be possible.

4.1.2 Accounting

The accounting services of the PIP should have the following characteristics:

- be flexible and configurable;
- account for a client's use of the system, perhaps on a per-custodian-level service basis;
- Integrate with the security system; and
- provide support for external reporting and financial management systems.

4.1.3 Quality of Service and Monitoring Issues

The PIP should provide basic feedback to users and have detection and monitoring functions to determine the level of service being given to the client. These functions should indicate service failure and provide alternatives to the service.

These functions include:

- display and control of the overall state of the system, possibly in a graphical manner;
- support for replication of integration level services;
- notify of exceptional circumstances (service goes down, for instance…);
- report on activity over the system; and
- provide mechanisms for management of services (start, stop, etc.)

4.2 Application Layer

The application layer can be regarded as the ultimate consumer of the data sets that are made available through the various custodian services. These are then processed and integrated under the control of the integration services. In the general (and increasingly important) case, applications may include automated processes as well as human users.

Some issues for the specification of the Application Layer include:

- the need for active and constant feedback (timeliness feedback) when long-running processes are active, particularly when the application is being used by a human operator;
- support for heterogeneous environments;
- the need to interface with common off-the-shelf software such as spreadsheets and databases;
- interfacing with the various infrastructural facilities such as security and accounting; and

- the application's GUI (if appropriate) should be well regarded by the user community (user acceptance) and should be easy to use and consistent with other applications that the user may utilise in the courses of their work.

4.3 Integration Services

This layer ensures that 'all data that is marketable should be accessible to appropriate

The major issues for integration services are:

- the provision of a standard set of Application Programming Interfaces to the higher level applications; and
- the provision of a set of Standard Services to the higher layer, such as:
  - allow the discovery of services by published metadata and interface;
  - provide a repository of metadata information;
  - provide a standard "white pages" style of service lookup;
  - integrate with the various supporting infrastructural facilities;
  - facilitate easy development of new applications;
  - compose a single application request into multiple requests onto multiple custodian services, as necessary;
  - provide minimisation techniques (compression, etc.) to reduce the size of the data sets being used and keep performance within acceptable limits;
  - perform data fusion such that a request for a given data set can be satisfied regardless of the location and format of the actual individual elements of that data set; and
  - adhere to applicable open standards for data and processing.

4.4 Custodian Services

The custodian of a data set is ultimately responsible for offering the data used in the PIP system. A data set may be offered for read-write or read-only access via a custodian service. Each custodian service should be defined through a System Programming Interface such that each service can be invoked in the same way by the higher-level functions of the Integration layer.

Issues here include:

- the publication of a data set should not require any change to a custodian's existing data storage system;

- there should be a mandatory requirement for a single format, across all the infrastructure, with providers at their discretion offering optional extras;
- security and confidentiality should not be compromised;
- any data offered for access remains the property of the originating custodian;
- the custodian determines issues such as access to, and charges for use of the data;
- the custodian is responsible for the accuracy and validity of the published data; and
- the definition of the System Programming Interface should allow flexibility in dealing with numerous disparate services while presenting a standardised interface to the rest of the system.

5 Conclusion

This document has examined various aspects arising from the development of the PIP prototype system including:

- the integration of custodian data sets;
- the performance of the prototype across high and low-speed network connections; and
- the development time required and problems associated with development.

In addition, a Technical Architecture Specification Overview has been described that highlights issues that should be examined in preparation for the production of a full architecture specification.

The experience gained in developing a PIP prototype and the information reported in this document as result of this experience can be used as basis to produce a specification document for a Property Interests Product technology architecture.

6 References

QUEST98 *The Queensland Electronic Services Trial—QUEST Project System Evaluation Report*, July 1998

PIP99a *Property Interests Product (PIP) Collaborative Research Project Market Infrastructure Framework Report*, June 1999, available from http://www.qsiis.qld.gov.au/data/qsiisDocument/published/930191566578/pipcrp/market_framework_report_31mar.pdf

PIP99b *Property Interests Product (PIP) Collaborative Research Project Information Infrastructure Modelling for the Property Interests Product*, June 1999, available from http://www.qsiis.qld.gov.au/data/qsiisDocument/published/930191566578/pipcrp/circulated_report_integrity_and_integration2.pdf

PIP99c *Property Interests Product (PIP) Collaborative Research Project Prototype Specification* contains detailed application specific requirements for the PIP prototype.

PIP99d *Property Interests Product (PIP) Collaborative Research Project Prototype Technical Issues Report,* April 1999.