WOSTMANN & ASSOCIATES, INC.

Development of a Cooperative Interagency Electronic Fishery Information Collection and Management Program in Alaska

Technology Assessment and Recommendations

July 2002

Submitted by

Wostmann & Associates, Inc.

Table of Contents

1 Introduction

This report is a supplement to Cooperative Interagency Electronic Fishery Information Collection and Management Program Needs Analysis report prepared for the Pacific States Marine Fisheries Commission by Wostmann and Associates, Inc. and Natural Resources Consultants, Inc. That report examines the need and potential for an integrated landing reporting system serving the needs of three fisheries management agencies: the Alaska Department of Fish and Game (ADF&G), the National Marine Fisheries Service (NMFS), and the International Pacific Halibut Commission (IPHC). It describes a vision for the integrated system. This report addresses the technical aspects of building such a system. It describes the existing technology and infrastructure available for implementing the system.

This report provides a brief overview of the information technologies currently in use at the three agencies. The currently utilized technologies and existing systems are evaluated for their potential to be used as part of the overall integrated system, and the effort necessary to interface them with the envisioned system.

The data communications infrastructure available in Alaska, both on shore and off shore, is described and evaluated for its ability to support an electronic landing reporting system.

This report examines the system architecture and components described in the system vision and evaluates them from a technical perspective in terms of feasibility, cost, and suitability to operational needs. Optional system elements are considered as alternatives.

1.1 Definitions, Acronyms, and Abbreviations

Term	Definition
ACS	Alaska Communications Systems, a leading telecommunications provider in Alaska. ACS currently holds the State of Alaska telecommunications provisioning contract.
ADF&G	Alaska Department of Fish and Game.
ASP	Active Server Pages, A technique for creating web applications and dynamic web pages that embeds processing script in the HTML for the page. ASP is proprietary standard created by

Term	Definition
	Microsoft and used with Microsoft server products.
ASP	Application Service Provider, an organization that provides the hardware, operating software, and service infrastructure to run computer systems. ASPs typically have data centers which run 24 hours a day, with onsite support for basic Internet connectivity, database connectivity, and system uptime monitoring, but not development or support services for the actual application.
ATM	Automated Terminal Machine, a small card-swipe terminal.
C	A commonly used programming language. C was created at Bell Labs in the late sixties for programming operating systems. It provides excellent control of the computer at the expensive of requiring the programmer to manually code almost all tasks.
C#	A new programming language created by Microsoft to support the .Net initiative. C# is similar to Java, but is optimized to exploit the features of Microsoft operating systems.
CFEC	Commercial Fisheries Entry Commission, the State of Alaska agency that licenses individuals and vessels to fish commercially.
CORBA	Common Object Request Broker Architecture, a set of protocol created by the Object Management Group, an IT industry consortium which allows distributed computer systems to talk to one another in standardized ways, eliminating incompatibilities between different vendor's computer architectures.
DB2	A relational database management system marketed by IBM.
DBMS	Database Management System. A DBMS is a software application that manages the storage, query, and updating of data, providing enforced data consistency and removing much of the

Term	Definition
	burden of managing data in files from the application programmer.
DMOS	DataCard Multitasking Operating System, the operating system that runs the DataCard ATM card-swipe terminals. DMOS is obscure and almost obsolete
FTP	File Transfer Protocol, a specification and software that implements that specification to transfer files between computers over the Internet.
GB	Gigabyte, a unit of disk storage equivalent to one million bytes. Disk drives currently available provide the capacity to store between 2 and 80 GB.
GHz	Gigahertz, a unit of computer central processor speed. High end PCs and servers typically have processors which operate at speeds of between 1 and 2 GHz.
HP-UX	Hewlett/Packard Unix, the proprietary version of the Unix operating system which HP provides with their high-end servers and workstations.
HTML	Hypertext Markup Language, the specification language for encoding documents which can be displayed by world wide web browsers.
IDE	Integrated Development Environment, software which runs on a programmer's workstation and which allows the programmer to edit, compile, and execute programs while developing their code. IDEs typically provide other programmer productivity tools to help the programmer write code efficiently.
IFQ	Individual Fishing Quota, a program which gives fishermen a share of the planned harvest which they can fish as they wish during an extended fishing season.
Inmarsat	A marine voice and data communications system that uses satellites to transmit and receive signals.

Term	Definition
IP Address	The unique address of each computer that is connected to the Internet.
IPHC	International Pacific Halibut Commission.
IT	Information Technology.
ITG	The State of Alaska Information Technology Group, an IT service provider for State agencies.
J2EE	Java 2 Enterprise Edition, a set of specifications and enhancements to the Java programming language which are intended to facilitate the development and deployment of enterprise wide distributed applications.
Java	A programming language developed by Sun Microsystems which can run on almost any vendor's hardware and operating systems, and which automates some of the low level programming tasks which a language such as C requires the programmer to perform explicitly in their code.
JBoss	A J2EE application server which is freely available for download from the Internet, and which is supported by a community of programmers who use and enhance it for their own applications, and who share the infrastructure code as a part of the open source movement.
JBuilder	An integrated development environment (IDE) for Java, marketed by Borland.
JSP	Java Server Pages, a technique for creating web applications and dynamic web pages that embeds Java as script in the HTML for the page.
Kbps	Kilobits per second, a unit of data transfer on networks and phone lines. Kbps is hundreds of bits per second. Typical modes transmit at 9.6 to 56 Kbps, while local area networks are generally 100,000 Kbps.
LAN	Local Area Network, the network in an office or

Term	Definition
	facility that ties local computers together. LANs may be linked via telecommunications providers to the Internet, or to other LANs.
Linux	A version of the Unix operating system that is freely available on the Internet as both an installable operating system and a set of source code. This allows any programmer to determine how the system works, and to identify the cause of problems. Linux is an open source project that accepts contributions from programmers across the Internet, as well as from corporations who are willing to adhere to the terms of its license, mainly to share the source code.
MS Access	A relational database management system marketed by Microsoft. MS Access is suitable for desktop and small workgroup applications, but is not intended for larger enterprise applications.
MS SQL Server	A relational database management system marketed by Microsoft. SQL Server is an enterprise level RDBMS, in the same class as Oracle or DB2.
.Net	A set of specifications and enhancements to the Microsoft product line which are intended to facilitate the development and deployment of enterprise wide distributed applications.
NMFS	National Marine Fisheries Service.
NRC	Natural Resources Consultants.
n-Tier Client/Server	A client/server systems architecture that has multiple layers of server software. N-Tier application can have processing logic located on either the client or the server, allowing for the development of thin clients that require lower processing power and which are easier to manage and maintain.
Oracle	A relational database management system marketed by Oracle Corporation. Oracle is the leading RDBMS for large enterprise class systems, but is frequently used for workgroup class systems

Term	Definition
	as well.
PC	Personal Computer.
PDF	Portable Document Format, a standard created by Adobe for document files, allowing them to be viewed and printed with the same appearance on many different types of computers and printers.
PHP	A technique for creating web applications and dynamic web pages that embeds processing script in the HTML for the page. PHP is an open source project and is freely available on the Internet.
PL/SQL	The language used for programming stored procedures in Oracle databases.
PostgreSQL	A relational database management system that is freely available on the Internet, and is supported by an open source community. Postgres is more capable than desktop databases such as MS Access, but does not provide the performance of full enterprise class databases such as Oracle.
PSMFC	Pacific States Marine Fisheries Commission.
RDBMS	Relational Database Management System, the de facto standard in database management systems. An RDBMS allows programmers to specify queries and update statements using the Structured Query Language.
RPC	Remote Procedure Call, the general process by which an application on one computer communicates with an application on another computer.
RTF	Rich Text Format, a word processing document format that can be used with Microsoft Word and with a variety of other word processing programs.
SOAP	Simple Object Access Protocol, a set of protocol created by an IT industry consortium led by Microsoft and IBM which allows distributed

Term	Definition
	computer systems to talk to one another in standardized ways, eliminating incompatibilities between different vendor's computer architectures.
SQL	Structure Query Language, a language which allows both programmers and non-programmers to query and update relational databases.
Sybase	A relational database management system marketed by Sybase Corporation. The original versions of Microsoft SQL Server were licensed from Sybase, but Sybase and SQL Server have evolved into completely separate products.
2-Tier Client/Server	A client/server architecture where the server acts as a database server only. The logic of a 2-tier application must be coded in the client software, although some logic can be imbedded in the database in the form of triggers and stored procedures.
TPC	Transaction Performance Processing Council, an IT industry group that specifies and reports on benchmarks for RDBMS performance.
Unix	A multi-tasking operating system originally developed at Bell Labs, which is commonly available on high-end servers and workstations, and which is available for PCs in the form of Linux, an open source version of Unix.
VAX	A minicomputer architecture and operating system marketed by Digital Equipment Corporation.
VB	Visual Basic
Visual Basic	A computer language marketed by Microsoft which allows the programmer to easily build graphical user interfaces, which has not historically provided the programmer with as much control of the computer as languages such as C and Java.
WAI	Wostmann & Associates, Inc.

Term	Definition
WAN	Wide Area Network, a set of local area networks in geographically separate locations which are connected by telecommunications links and which function as one network.
Windows 2000	A version of the Microsoft Windows operating system that is suitable for servers and enterprise applications.
Windows NT	An older version of the Microsoft Windows operating system. Windows NT was the first version of Windows that provided all the capabilities expected for enterprise type applications.

2 Technology Analysis

The practicality of the integrated landing reporting system described in the Needs Analysis will be determined by the technology infrastructure available for its implementation. In this section we review the technology base of the agency systems that would receive data from the integrated system, and the communications infrastructure available to connect to the integrated system.

2.1 Technologies in Use

The fishery management agencies that will receive data from the integrated landing system have systems in place that received and store landing information. In this section we describe and evaluate the technologies used in these existing systems, examining the implications for interfacing the existing systems with a new integrated landing system, and looking for synergies available from the combination of technologies.

2.1.1 Database Management Systems

All three agencies use relational database management systems as the data stores for the systems that handle landing data. ADF&G uses Oracle as the DBMS for the Fish Ticket Database. NMFS uses Oracle for the IFQ reporting system, the Electronic Reporting daily production tracking system, and for other processor information tracking. NMFS also uses MS Access as the local database for the Electronic Reporting client software. IPHC uses Microsoft SQL Server as the DBMS for their halibut logbook and ticket information tracking system. All of these database management systems use the structured query language (SQL) to access the data. All provide effective data stores appropriate to their usage. Staff at each agency is trained in the use of SQL. We know of no reason to use anything but a relational DBMS to store and retrieve data such as the integrated landing system will process.

2.1.2 Networks

All three of the agencies utilize IP based networks for internal networking, data exchanges, and to maintain an Internet presence with a public website. The ADF&G network is part of the State of Alaska Wide Area Network (WAN). The State WAN provides dedicated bandwidth through leased lines to all of the ADF&G field offices. Offices on Kodiak Island and to the west are linked via satellite; other offices use the terrestrial telecommunications cables. The NMFS network is primarily a local area network in the Federal Building in Juneau, supplemented by leased lines to their field offices. The NMFS electronic reporting system uses Internet mail delivery coming through their gateway to the Internet as the conduit of electronic reporting file transfers. The NMFS IFQ system uses a proprietary network based on telephone lines. The IFQ system has a modem bank in the Federal Building; with both local and toll-free phone numbers. The ATMs located at processors in fishing ports dial in to these phone numbers to establish connections for submitting landing reports. The IFQ dial-in network is the only non-IP based network used by the three agencies for landing information. The IPHC network is a local area network. It receives some data from the NMFS IFQ system via file transfer across the Internet. IP based networks are the industry standard, and are suitable for all data exchanges needed by the agencies. The NMFS IFQ dial-up network does not have any technical advantages over IP-based communications, and requires the use of non-standard protocols for transmission error checking and correction.

2.1.3 Hardware and Operating Platforms

The agencies use a mix of hardware and operating systems for their client and server system operating platforms. ADF&G uses Windows 2000 running on Intel-based PCs for their client workstations. The Fish Ticket Database system uses Intel based server hardware running Windows 2000 for their database server and Linux for their web application server. NMFS also uses Windows 2000 running on Intel-based PCs for client workstations, along with some Windows NT. The NMFS electronic reporting and IFQ systems run on a Hewlett-Packard HP9000 Unix server running the HP-UX operating system. This system runs the database and server application components. The IFQ system clients are dedicated card swipe terminals. The card swipe terminals are DataCard ATM 680 terminals running DataCard's DMOS operating system. The ATM 680 is no longer manufactured or supported by DataCard, and the DMOS operating system is an obscure variant of Unix with many proprietary extensions. The IPHC Halibut Fish Ticket system uses Intel-based Windows workstations and Intel-based servers running Windows NT. Intel-based PCs running Windows are the de facto standard for workstations and fully meet the needs of both the existing systems and the envisioned integrated landing system. Both Intel-based servers running Windows 2000 or Linux, and proprietary Unix servers such as the HP9000 are capable platforms with adequate performance for running systems such as the existing agency systems, the envisioned interface software, and the integrated landing system itself.

2.1.4 Application Software and Development Tools

While the agencies have similar networks, servers, and database management software, their application systems were built using a wide variety of application development tools and approaches.

The ADF&G Fish Ticket Database system is a 2-tier client/server system, with the database running on the server and all application software outside the database running on client workstations. The Fish Ticket Database system has a single database with four client applications. Each of the client systems is optimized for data entry of a specific type of fishery. The client systems are named Neptune for groundfish fish ticket processing, Venus for shellfish fish ticket processing, Zephyr for salmon fish ticket processing, and Triton for herring fish ticket processing. All the clients are designed for mass data entry so they cache records locally to avoid performance problems due to network latency when communicating with the database server. In addition, a web-based client is under development. The Fish Ticket Database system was developed between 1998 and 2000, with the web-based client in development in 2002. Oracle PL/SQL is the development tool used for processing in the database. The clients were developed using the Centura Rapid Application Development tool. The web client is being developed in PHP3. The Fish Ticket Database system is early in its lifecycle, and is expected to provide service for many years.

The NMFS information system is primarily a 2-tier client/server system, with some 3-tier components for the electronic reporting system and IFQ landing system. The NMFS system has a single database containing the core system used by the various applications. Much of the application logic resides in the database, and was developed with Oracle PL/SQL. Client applications for use by NMFS staff were developed with Oracle Forms. In addition, much of the reporting from the database is done using SQL reporting tools. The electronic reporting client used by processors was developed in Visual Basic. The electronic reporting client stores data in a local MS Access database. When ready to submit a daily report the data is packaged in a file and transmitted to NMFS as an attachment to an email message. A process on the server reads the incoming email, loads the data from the attached file into the database using Oracle SQL Loader, and sends back a confirmation email. This middle tier server process was developed in Perl. The IFQ client process runs on the ATM card swipe terminals. The IFQ client software collects the landing data, and then makes a dial-up terminal connection to the IFQ system server. A server process is started which receives the IFQ landing information and loads it into the database, sending transaction results back to the ATM terminal.

Once the transaction completes the connection to the server is disconnected. The ATM terminal software was developed in the C language. The server application that receives IFQ transactions was also developed in C. A web client for the IFQ system is currently in development. It is being developed using Java Servlet tools. An additional web application, which allows permit holders to pay fees online, was developed using the same technology. The IFQ system was initially developed in 1994 and 1994, and the ATM software was reengineered in 1997. The NMFS core database and electronic reporting system was developed between 1997 and 2000. The core database is early in its lifecycle and no replacement horizon has been set. The IFQ system is much further along in its lifecycle. The IFQ web application has begun replacing the ATM terminals. The integrated landing system envisioned by this report is expected to continue the phase-out of the ATM terminals. Likewise, the electronic reporting client may be partially or totally replaced by the integrated landing system.

The IPHC Halibut Fish Ticket system is a 2-tier client/server system. The database runs on the server and the application software runs on client workstations. The client/server system runs only on the IPHC LAN, which provides reliable database connectivity with good performance. The system was developed during 1998 and 1999. The client application was developed in Visual Basic. The system was developed as a replacement for a prior system that ran on a VAX system. It retains some of the VAX legacy data structures and development of a replacement is being contemplated for sometime in the 2003 to 2006 timeframe.

2.1.5 Other

Besides the client/server systems in use at the agencies both for internal access and external user input, the only significant technology is the use of plastic cards for permit holders. NMFS issues cards to IFQ quota shareholders and skippers who fish the quota shares of others. The NMFS cards contain a two track magnetic stripe that contains identifying data as well as some system information including the holder's PIN. The IFQ ATM terminals read the magnetic stripe. The Alaska Commercial Fisheries Entry Commission also issues cards for fishery permit holders. The CFEC cards do not contain a magnetic stripe, but they do have raised lettering that is used to create imprints on paper fish tickets.

The use of cards as tokens is intended to increase the security and integrity of the data collection systems. The card must be physically present to initiate a landing transaction or fish ticket. In the case of the IFQ card, the PIN is intended to allow only the cardholder to initiate transactions with their card. This is only partially effective, since most cardholders provide their PIN to the processors' staff who run the card swipe transactions. Likewise, although fish tickets are not supposed to be submitted without the CFEC card imprint this occasionally happens. The credit card industry has long since adopted the stance that the information on the card is more important than the physical card itself. If the information can be verified electronically then it is in most cases accepted.

2.2 Communications Infrastructure

An adequate data communications infrastructure is a prerequisite to a successful electronic reporting system. Although Alaska is geographically remote, it has significant electronic communications resources.

2.2.1 Wired Network

The wired communications infrastructure provides hard-wired connectivity to businesses and other locations in Alaska ports where electronic reporting would originate.

2.2.1.1 Undersea Cable

Three fiber optic undersea cables providing connections to the lower 48 States and the greater Internet serve the major population centers of Alaska. In the future additional cables may be built.

Cable System	Status	Route
Northstar	Existing	Anchorage to Whittier to Lena Pt to Tillamook, Oregon
Alaska United	Existing	Anchorage to Whittier to Juneau to Seattle (Valdez under construction)
North Pacific Cable	Existing	Japan to Seward to Pacific City, Oregon
Military	Proposed. May carry some commercial traffic	Shemya to Seward to Oregon or Washington (possible connections to Anchorage, Kodiak and possibly Juneau)
Flag Pacific	Proposed, but on hold	Japan to Whittier to Oregon or California

Cable System	Status	Route
Northern Light	Proposed, not likely to be build in the foreseeable future	Adak to Dutch Harbor to Kodiak to Seward to Whittier to Anchorage to Valdez to Juneau to Oregon or Washington

Coastal communities close to Anchorage and Juneau enjoy relatively high performance and reliability in their data communications as a result of these cables. Communities in Southeast Alaska and along the coasts of Prince William Sound and Cook Inlet have terrestrial cabling that is tied in to the fiber optic cables to give Internet and lower 48 connectivity. The communities of Anchorage, Juneau, Kenai, Nome, Seward, Sitka, and Valdez have broadband cable modem Internet access tied to the fiber optic cables.

2.2.1.2 Satellite

Alaska communities on Kodiak Island and westward do not have hard wired access to the national data communications infrastructure. Their communications for voice and data are dependent on satellite uplinks. The local networks in these communities are hardwired. The communities of Cordova and Nome have broadband cable modem service available, but these, too, travel over satellite links.

2.2.2 Wireless

For at-sea processors and tenders operating away from port communities the only alternative for data communications is wireless. Generally wireless communications are slower and more expensive than wired networks.

2.2.2.1 Satellite Phone

The following table shows the satellite voice and data systems that are currently in use.

Service	Equipment Costs	Usage Cost per Minute	Data Speed	Coverage
Iridium	$500-$1,500	$.99 to $1.50	10 kbps	global
Globalstar	$900	$1.69	9.6 kbps	regional

Service	Equipment Costs	Usage Cost per Minute	Data Speed	Coverage
MSAT	$4,000	$1.19	4.8 kbps	regional
Inmarsat A	no longer available	$2.65-$4.90	2.4 kbps	global
Inmarsat B	$25,000	$3.00-$9.00	9.6-64 kbps	global
Inmarsat C (email/data only no voice)	$2,500-$3,500	$0.25 per 32 characters	600 bps	global
Inmarsat Mini-4	$5,000	$2.50	4.8 kbps	spotty
Inmarsat M4	$8,000	$2.30-$7.25	64 kbps	global
Inmarsat Fleet 77	N/A	N/A	64 kbps	global

Most vessels that have satellite communications in Alaska use either Inmarsat A, B or C. Globalstar is making some inroads but they are in financial difficulty. Iridium was the first low cost/low Earth orbit system but they also experienced financial difficulty and were forced to declare bankruptcy.

Inmarsat C is email and data only. The system in Alaska is called Fleetnet. Trident uses it as does many of the industry associations for send small files attached to email messages, mainly things like crew share tally's, equipment orders and product on board reports. However, Inmarsat C doesn't handle files larger than 32 kb or about six pages of text and at $0.25 per 32 characters, it can get very expensive.

Inmarsat B is available on the larger vessels and offers 64 kbps, the highest speed available but data transfers cost $7.00 to $9.00 per minute depending upon the time of the day and the hardware is $25,000.

Inmarsat M4 is not yet usable for shipboard applications, it needs the development of a tracking antennae.

Inmarsat Fleet 77 is the newest Inmarsat product. It promises "always on" voice/data communications where vessels pay only for what they send and receive. Their web site states:

> "The Fleet F77 service is now commercially available, offering unparalleled flexibility when it comes to global maritime communications technology. Inmarsat, which has provided reliable world-wide safety services for 21 years, can now connect ocean-going vessels - whether merchant, fishing or luxury yachts - to corporate IT networks and the Internet at super-fast ISDN speed. Inmarsat's Fleet F77 is a single integrated voice, fax and data service offering a choice of

> communications, including Mobile ISDN and a revolutionary Mobile Packet Data option that allows ships to stay "always connected" during a voyage anywhere in the world. With Mobile Packet Data, users pay for the amount of data they send and receive, rather than how long they stay online."[1]

However, dealers are not yet able to provide equipment or information on costs but indicate it will probably be similar to Inmarsat C.

ICO-Teledesic Global is still in the development stages after an initial bankruptcy but advertises eventual Internet browsing capabilities are prices significantly below current costs and at an order of magnitude increase in speed using wireless phone technology on a series of 288 lower atmosphere orbiting satellites. It seems unlikely that this system will be able to deliver on these promises in the next 3 to 5 years.

The marine electronic experts indicate that all these systems are considerably slower than a 56k modem and browsing the Internet is not practical.

2.2.2.2 Cell Phone

Cell phone service is available in much of coastal Alaska. Most of Prince William Sound in the vicinity of Whittier, Valdez, and Cordova has service, as does most of Cook Inlet. Much of Southeast has service, although the geography of the Alexander Archipelago results in many dead spots. Larger coastal communities such as Ketchikan, Sitka, Juneau, Seward, Kodiak, and Dutch Harbor have good cell phone coverage. A number of remote villages have cellular service, including Emmonak, Hooper Bay, Togiak, Toksook Bay, and Savoonga.

Cellular service is line of sight to towers, so is only usable inshore, and can have dead spots on mountainous coasts. Cellular signals have a maximum range of 30 to 40 miles. Cell phone connection speeds have a maximum rate of 14.4 Kbps, and may be limited to 9.6kbps. Actual rates in less than optimum conditions can be half the maximum rate, due to retransmission times. Cell phones may have slow uplink rates of about 1kbps.

Cell phone service plans usually require an annual commitment. Monthly costs can range from around $50 to $200 or more. Higher rate plans provide more minutes of airtime. In almost all cases it is less expensive to purchase a higher rate plan than to pay per minute charges for exceeding the plan airtime. Per minute charges are usually in the range of $.20 to $.40 if plan airtime is exceeded.

[1] http://www.inmarsat.com/fleet/

2.2.3 Ability to Support Electronic Landing Reporting System

The data communications infrastructure in Alaska is sufficient to support an electronic reporting system. Most of the large and mid-size shore based processors already have Internet connections, and many small processors do as well. Many of these processors already use the NMFS Electronic Reporting system to submit required reports electronically.

At-sea processors must use a much more costly and limited data communications infrastructure based on satellite technology. The expense of satellite communications limits the practicality of an electronic reporting system based on its use to only the larger at-sea processors. A system designed to meet the needs of at-sea processors would have significant technical constraints limiting the size of data transfers. These technical constraints would limit design flexibility, and require greater custom development than would be needed for a system using the shore based infrastructure. However, a significant percentage of landing reports for at-sea processors are made when they are in port, offloading processed product. At these times they have access to the shore-based communications infrastructure.

3 Alternatives Analysis

The Needs Analysis report describes a system that would meet the agency needs for integrated landing reporting. It includes alternative architectural approaches and multiple client subsystems. This section examines the individual system elements and architectural alternatives. It evaluates each option according to how well it meets the operational needs of the integrated landing system; how its development would affect the existing agency systems and databases; the cost of the option to both the agencies and to processors; the security implications; and maintainability concerns.

IT costs are notoriously difficult to estimate, particularly during stages of the system lifecycle before rigorous requirements specifications have been documented. The cost estimates in this section are based on the scope of the application and its subsystems as can be seen at this time, and the professional experience of the analysts with systems using similar technologies. In most cases a cost range is provided, but should not be as the maximum possible range, particularly on the high end. Rather, it provides useful information for system development planning. Costs estimates are for the specification and implementation of the software capability, unless otherwise noted. Studies show that large software projects frequently exceed their budgets and are otherwise troubled.[2] Prudent IT planners and project managers make allowances for this unfortunate fact.

Costs which industry would bear are identified in terms of computer hardware, software licenses, and development projects required. Dollar amounts are not estimated since these items would not be isolated to supporting the electronic landing system, but would also typically be used for other business purposes. The information presented with the alternatives provides the relative cost impact to industry system users.

3.1 Server and Database Architecture

The integrated landing system envisioned in the Needs Analysis has a client/server architecture. This section examines components of the server portion of the architecture.

---

[2] Perlman, Ellen, "Technotrouble", Governing, Sept. 1998, Vol. 11, No. 12, pp. 21-24

3.1.1 Rejected Alternatives

During the development of the system vision described in the Needs Analysis several alternative server system architectures were considered and rejected as infeasible.

3.1.1.1 Full Replacement System

The envisioned solution positions the integrated landing system as a front-end to the existing agency systems. The existing agency systems are the ADF&G Fish Ticket Database system, the NMFS IFQ system, the IPHC Halibut Fish Ticket system, and the NMFS electronic reporting system. An alternative would be to replace all the agency systems with a single shared system. While this approach would eliminate any data consistency and reconciliation problems it would be extremely difficult to implement and maintain. The different agencies have differing mandates and diverse existing systems. A replacement system would have to duplicate all the functionality of the existing agency systems in addition to developing the integrated landing functionality. The cost to develop the existing systems is estimated to have been between $1.2 and $1.8 million. A full replacement would likely be more because the resolution of issues would involve reaching consensus between three agencies, rather than just within one. The management and support of such a system would be similarly complex, with prioritization of support work needing to be established by all three agencies cooperatively. Although these are also issues for the envisioned system they would be more difficult to resolve for a full replacement system since large portions of such a system would not be areas of common concern for all the agencies, but instead would be agency specific.

3.1.1.2 Gateway to Agency Systems

A variation of the envisioned solution would have the integrated system server using the existing agency systems as the data stores. In this case the integrated system would have no database of its own. While this would eliminate the development and support effort for the integrated system database it would vastly complicate the system interfaces to the existing agency systems. Where the envisioned system requires relatively simple interfaces which can received new landing records and updates to existing landing records, an integrated system without its own database would need interfaces which retrieve existing landing records as well as accept them. The system interfaces would also have to provide data in real time, and the performance penalty of having to access up to four remote systems would make adequate response time problematic. Since none of the existing agency databases have the change tracking and interface control features the integrated system will need these would have to be added, effectively developing the integrated landing database in one of the interfacing systems, so the potential cost savings of not developing the database would not be realized.

3.1.1.3 2-Tier Client/Server System

A 2-tier client server system has client software running on the desktop or web application server communicating directly with a database running on the database server. The 2-tier architecture works well when both the client and server are on the same organization's network. It is less desirable for applications where the communications take place across the Internet, and where multiple client types are needed. Exposing the database access directly to the Internet exposes the system to unacceptable security risks if a vulnerability is discovered in the database management system software because no application software on the server isolates the DBMS from Internet hackers. In addition, a 2-tier client/server application requires that all processing that cannot take place inside the database through database procedures must be delegated to the remote client. This increase maintenance effort if multiple client types are needed.

3.1.2 Server Application

The server portion of the landing system application will provide the client interfaces, business rules, and processing to implement the functions and features of the envisioned system. The server portion of the envisioned application system can be fully realized with available server technology and programming design patterns and techniques. The development cost of the server application capabilities is estimated at between $40,000 and $70,000.

The server capabilities described could be built with standard and commonly available system development tools and techniques, and would require reasonably skilled but not extraordinary IT support personnel. The server business rules should embody all the validation rules of the system, even those that will also be implemented on the clients. This "belt and suspenders" approach will minimize the potential for incorrect data being inserted into the database as the result of client misbehavior.

3.1.2.1 User Management and Security Subsystem

The server must provide the user management and security subsystem. It will allow users to authenticate to the system, and will control what functions each user can execute, depending on their privileges. The agency client interface must provide user management functions to allow agency personnel to add, change, and delete user data, passwords, user groups, and privilege assignments. User authentication via passwords and privilege assignment via user groups are standard IT industry best practices. Further security of the server itself depends on the firewalls, intrusion detection systems, and system hardening of production support installation.

3.1.2.2 System Interfaces

The server will have interfaces with five systems: the ADF&G Fish Ticket Database, the NMFS IFQ system, the IPHC Halibut Fish Ticket system, the NMFS Electronic Reporting system, and the CFEC Licensing system. The CFEC interface will receive permit card information and vessel ADF&G number and name information to be used in validations. This interface will receive the data on a periodic basis. All the other interfaces will send landing data to the respective systems. The data transmitted will be in accordance with the receiving system's needs. The user action will initiate data transmission. In the case of the NMFS IFQ and Electronic Reporting interfaces the data submitters will initiate the data transmission. In the case of the ADF&G Fish Ticket Database and IPHC Halibut Fish Ticket system interfaces agency users will initiate the data transmission. For the latter two systems data transmission could also be initiated automatically on a timed basis.

The server side of the system interfaces will not introduce significant security risk. The receivers of the data will be fixed IP addresses. The data can be encrypted, either by passing it through an encrypted tunnel or encrypting files for transfer if desired. The maintenance effort for the server interfaces should be low, since data is primarily being transferred from the server to the interfacing systems. Likely maintenance and enhancement activity would be the result of adding data elements to the system or an interfacing system needing data that is captured on the server but not transferred in the interface. The second case can be mitigated by transferring all data of interest, even if the receiving system does not currently store all data elements.

3.1.2.3 Client Interfaces

The server will provide two client interfaces, one for data submitter clients and one for agency user clients. Many types of clients may be developed which use these two interfaces, but they will all conform to one or the other of the interface types. The data submitter client interface will support functions to allow data submitters to enter and amend landing reports and to view their own previously entered data. The agency client interface will allow agency users to view and edit all data submitters' reports, as well as to data enter reports on behalf of data submitters.

The client interfaces are the point of greatest security risk on the server, since they are the entry to the server for users on the Internet. Having two separate client interfaces reduces the risk because the agency client interface, which has much more powerful functions, can be limited to agency users in specified locations by firewalling. Both client interfaces will be password protected for all users. Such multiple layers of security for the more sensitive agency user interface will allow the security risk to be managed. Having two client interfaces will increase maintenance effort only marginally since the data submitter client functions are for the most part distinct from those the agency user clients need.

3.1.3 Database

The integrated system database will provide the staging area for client to use while building up complete landing and daily production reports. It should store all updates to submitted reports, providing a clear history of all reporting and amending activity. It will provide a buffer for the agency systems interfaces, allowing them to accept data at the rates they prefer. The database will allow the system to coordinate updates to the interfacing systems and to log the results of those interactions as an aid to problem determination and resolution.

A fully developed data model, based on the data definitions in the Needs Analysis report and implemented on a relational database management system, should fully meet the operational needs of the integrated landing system. The integrated system database will minimize changed to the existing systems' database architectures, since it will be able to store and manage all the data itself, allowing those systems to peel off only the data they require. The cost to design and develop the database for the integrated landing system is estimated to be between $35,000 and $50,000.

The database of the integrated landing system will represent a security risk since the server may be subject to attack from the Internet and a security compromise would allow attackers to modify or delete data. However, with good software development practices and survivability as design criteria the risk can be managed. The integrated system database will be at no more risk than any other database being used for electronic commerce on the Internet.

The integrated landing system database will require typical database maintenance such as daily backups, regular monitoring, and periodic performance tuning. It will require staff with database administration skills in addition to application programming skills.

3.1.4 Summary

The following tables summarize the cost and characteristics of the system components of the server application.

System	Cost	Remarks
Server application	$40 - 70K	Server framework, business logic, user management, system interfaces, and client interfaces.
Database	$35 - 50K	Data model, implementation, triggers, and administration.

3.2 System Interface Architecture

The existing agency systems will require some development to allow them to receive data from the integrated landing system, and in some cases provide data back to that system. The envisioned system is intended to have relatively low impact on the existing system in terms of additional development and software maintenance, in order to make efficient use of existing capabilities and investment in systems development.

The technology alternatives that could be used are discussed in section 3.4 along with other infrastructure choices. The specification and design of the system interfaces would be part of the integrated system project, with a high degree of input from the IT support personnel responsible for the existing systems. The actual implementation of the interfaces on the existing systems would be the responsibility of each organization. The system modifications could be done by in-house IT support staff or contract resources, similar to the way the organizations implement any enhancement to their existing systems.

3.2.1 ADF&G Fish Ticket Database

ADF&G will need to create an interface to allow the integrated landing system to load records to the Fish Ticket Database. The interface may be a real-time interface or a file transfer interface. ADF&G will need to decide how record transfers are initiated. They could be triggered by data submitter actions such as submitting or amending a landing report. Agency users could initiate them either as a part of normal processing, for example when a landing report is updated with the paper copy received information, or when data is updated. Alternately, agency users could request data load to the Fish Ticket Database explicitly. Finally, records could be transferred in batches on a specified schedule.

The Fish Ticket Database interface will need processing logic to map Landing Weight and State line items and Disposition line items to Fish Ticket Database Ticket Item records. It will need to handle both inserts of new reports and updates to existing report data. ADF&G will need to determine the rules for this logic. The cost to develop the interface for the Fish Ticket database is estimated at $30,000 to $45,000.

The interface will have a low additional security risk for the Fish Ticket Database. Communications for the data transfer will originate from a single, known IP address so rogue requests can be filtered at the firewall. The maintenance support for the interface should be low to moderate, and less than for any one of the four existing Fish Ticket Database clients.

3.2.2 NMFS IFQ System

The NMFS IFQ system will need to provide an interface to allow the integrated landing system to submit IFQ landing transactions and to receive the results of the transactions. The interface must provide real-time service. Its processing logic will be similar to the existing web application in scope and complexity. It will fully meet the IFQ system requirements of running IFQ transactions, and will allow data submitter needs to be fully met since multiple IFQ transactions will be able to be run from a single landing report, for example when multiple stat areas are entered. The interface will have to handle updates to previously submitted transactions, a function which the system does not currently provide to data submitters. How it handles these updates will need to be determined by the agency. It could elect to store and report updates for manual review and action, or for data fields that do not require review the changes could be made to the data in the database. The cost to develop the interface on the IFQ system is estimated at $15,000 to $20,000.

The IFQ system interface will have a relatively low additional security risk for that system. The communications will originate from the single, known IP address of the integrated system and can be suitably firewalled. The interface software itself will use the existing stored procedures, which will isolate it from the actual database updates. The maintenance effort to support the interface can be expected to be low, significantly less than for the existing web application.

3.2.3 IPHC Halibut Fish Ticket System

The IPHC Halibut Fish Ticket system will have to provide an interface that accepts landing reports from the integrated landing system. The interface need not provide real-time feedback since data transfers will be initiated either automatically at a scheduled time or by agency users on demand. The interface could use file transfers since the data submitter will not need or expect feedback from the IPHC system. Alternately, it could be a real-time interface, which might be desirable if agency users initiate data transfers to the IPHC system. The IPHC system interface will receive the data and load it to the IPHC database. It may need to hold data for user review or report activity to IPHC staff because some data elements such as packer flag may require staff input. The interface will have to handle updates to previously submitted landing reports. IPHC will have to determine how their system will handle these updates. It could elect to store and report updates for manual review and action, or for data fields that do not require review the changes could be made to the data in the database. The cost to develop the interface on the IPHC system is estimated at $10,000 to $15,000.

The IPHC system interface will have low additional security risk for that system, particularly if a file transfer interface is implemented. However, even a real-time interface will not introduce significant risk since the communications will originate from the single, known IP address of the integrated system and can be suitably firewalled. The interface software will load the data to the database or make updates according to IPHC criteria. The maintenance effort to support the interface can be expected to be relatively low.

3.2.4 NMFS Electronic Reporting System

The NMFS Electronic Reporting system will need to provide an interface to allow the integrated landing system to feed it daily production reports. The current interface between the Electronic Reporting client and the Electronic Reporting database uses email file transfer and batch loading to move the data into the database. This delays feedback to the user and makes problem resolution more time consuming. The interface should be able provide more real-time response, informing the integrated system, and thus the data submitter, if problems are detected when data is transferred. However, the integrated system will not need a real-time response from the Electronic Reporting system in order to allow the data submitter to complete a daily report. If the Electronic Reporting system interface does not respond the data will be stored on the integrated system and forwarded periodically until the interface accepts it. The actual processing logic will be similar to the existing data load application in scope and complexity. It can fully meet the Electronic Reporting system requirements of receiving data. The interface will have to handle updates to previously submitted reports. The system currently allows amended reports so this will provide similar functionality. The cost to develop the interface on the NMFS Electronic Reporting system is estimated at $30,000 to $40,000.

The NMFS Electronic Reporting system interface will have a relatively low additional security risk for that system. The communications will originate from the single, known IP address of the integrated system and can be suitably firewalled. The interface software itself may need to provide additional security rules for controlling when the database is allowed to be updated. The maintenance effort to support the interface can be expected to be approximately the same as for the existing file transfer and batch load application.

3.2.5 CFEC System

The CFEC system will not necessarily receive data from the integrated landing system. However, it will have to provide the set of valid permit numbers and the associated card data for the integrated landing system to use in validating and deriving landing information. CFEC currently makes this data available to ADF&G and IPHC. The interface data would be used to validate input data for landing reports. A natural lag exists from the time data is inserted into the CFEC database to the time fishers begin making landing with their new permit, so the interface does not have to be real-time. Daily or possibly weekly updates would suffice. The interface would be relatively straightforward, providing cross-reference data for permit numbers to permit card information and vessel numbers to vessel names. The cost to build the interface is estimated at $5,000.

3.2.6 Summary

The following tables summarize the cost and characteristics of the system interfaces that will need to be developed on the existing systems.

Agency	System	Cost	Remarks
ADF&G	Fish Ticket Database	$30 - 45K	Insert and update records in database
NMFS	IFQ System	$15 - 20K	Similar to interfaces for existing ATM and web application.
IPHC	IPHC Halibut Fish Ticket System	$10 - 15K	Insert and update records in database
NMFS	Electronic Reporting System	$30 - 40K	Similar to existing interface to electronic reporting client
CFEC	Licensing System	$3 - 5K	Read only periodic provision of permit and vessel cross-reference data

3.3 Client Architecture

The system envisioned in the Needs Analysis document provides for a number of clients for both agency users and data submitters. The two client interfaces described above support all the envisioned clients. All of the clients can potentially be built; none are mutually exclusive. However, some clients depend on features implemented with other clients. These dependencies are noted in the following sections. In addition, some clients implement some features more effectively than others.

3.3.1 Data Submitter Web-based Client

Web-based clients are an increasingly popular choice for extranet systems. Web applications are familiar to many people from their use of sites such as amazon.com. Two web-based clients have been described for the envisioned landing reporting system. The data submitter web client would allow buyers and processors to fill out and submit landing reports on the web. The web clients would not affect the database design or implementation.

A data submitter web client would meet the needs of many buyers and processors, particularly for groundfish landings where the number of fish tickets per day is small. The web application server, acting as a client to the integrated system server, would have the most up to date access to data for validations, since it is directly under control of the system maintainers and could be updated whenever they update validation data. The web client would have the easiest setup for submitters; all they would need would be a computer with an Internet connection, a web browser, and a valid userid. As the Needs Analysis user stories show, a majority of processors already have PCs with Internet connections, so the cost to the processors would be minimal. A web client would relieve the submitters of the need to make backups, at least as far as the landing reporting system is concerned. Likewise, they would not have to make any effort to keep software versions up to date, the web server would always provide the latest version. The cost to develop a public data submitter web client is estimated at $40,000 to $50,000.

Putting a critical application on the public Internet introduces some security considerations, but the principles that produce secure web applications are well understood, they just need to be followed. Data transmitted across the Internet can be encrypted with standard tools. User ids and passwords provide acceptable user authentication. Monitoring invalid logon attempts and unexpected client requests can identify attackers whose IP addresses can be blocked. Many banks and financial services provide Internet accessible web applications which exchange data which is considered at least as sensitive as landing reports, therefore the IT industry considers web applications an acceptable risk for the type of service this application would provide.

Maintenance efforts can be expected to be lower than other client types, since all maintenance is done on the server, and the client is relatively insensitive to the configuration of the desktop PC upon which it runs.

Web clients typically have slower response time than desktop applications, but this would not be a serious problem when submitting less than half a dozen landing reports per day. The web client could be optimized for slower Internet connections, with a minimum of graphics. However, the primary disadvantage of the web client for data submitters is the application would not be available when either the user's Internet connection or the server system was down.

3.3.2 Agency Web-based Client

The agency web client would allow agency users to review submitted landing reports, modify those reports, enter landing reports received on paper, manage user accounts, and do various types of summary reports for in-season management and enforcement. It would meet many, but not all, agency user needs. It would be most effective for users reviewing the data for a particular landing report, requesting in-season summary reports or statistics, making corrections to landing report data, and for casual data entry of the odd landing report received as a paper fish ticket. It would also be effective for managing user accounts and security attributes. It would be less effective for mass data entry where many landing reports are being data entered together as a batch. Like the submitter web client, it would require no effort to setup for agency users, and the software version would always be up to date since it is run on the web application server. The cost to develop an agency web client is estimated at $60,000 to $80,000.

The agency web client would be easier to secure than the public data submitter client. Since client requests would be coming from known agency networks access control by IP address could limit usage to only the agency locations desired. A virtual private network could be used if further security was desired.

Like the data submitter web client, the agency web client can be expected to be easier to maintain than other client types because effort is focused on the server, and the client PC configuration issues are for the most part eliminated.

3.3.3 Data Submitter Desktop Client

The desktop client application running on the user's PC and communicating with the server provides the most flexible user interface option. Desktop applications can fully exploit the window system features such as single key selections on drop-down lists and context sensitive data and window control changes such as unlocking IFQ information fields only when the species selected is halibut or sablefish. The key advantage of the desktop client is ability to function even when the server or the user's Internet connection is unavailable. It also provides the best performance for data entry.

A desktop client would meet the needs of most buyers and processors, but it would require them to provide more computer resources than the web client. In addition to an Internet connection and userid, they will need a reasonably powerful PC with sufficient free disk space and memory to run the application. This probably means a machine less than four years old, or if older one upgraded to current memory and disk standards.

The desktop client will affect the overall database architecture of the system because it will need a local database for storing data to allow disconnected operation. However, this will have negligible effects on the architecture of the server database. Rather, it will require database design for the local client database and design effort in the mechanisms that allow the local data to be kept in sync with the server data image. The cost to develop a desktop client for data submitters is estimated to be in the range of $50,000 to $70,000.

The desktop client introduces additional security considerations since the application must cache data locally, including userids and passwords for authentication, in order to function while disconnected from the server. This subjects the data to compromise in the event of the theft of the machine, or in cases where unscrupulous users otherwise gain access to it. Even encrypted the data can be compromised since the decryption algorithm and keys will need to be a part of the application for its own use. Data transmission to the server is less of a security concern since it can be encrypted with standard tools such as SSL. The communications channel to the server can also be a target of attack, primarily for denial of service attacks where attackers flood the server with bogus data which is rejected, but which clogs the connection port, interrupting service to legitimate users.

The installation and maintenance of a desktop client can be expected to be greater than for a web client. The software will need to be packaged with an installer program in order to minimize end user effort to install the application.

Desktop applications are subject to workstation configuration issues, and can be disabled if a user unintentionally deletes a needed file, for example when cleaning up their hard disk to create more space. The user must also upgrade the software when a new version is released, and will need support for more complicated upgrades that require local database changes and data conversion. In addition, the both the client and the server portions of the system will need a mechanism to detect their version, and what other versions with which they are compatible. This allows the system to accept data input from an older version while warning the user that they need to upgrade, giving the user a long period of time in which to accomplish the upgrade before their client stops working. The desktop client could recognize when it is out of date and prompt the user with nag message to encourage downloading a new version.

While having a local database provides performance and disconnected operation advantages, it introduces some liabilities. In addition to complicating software upgrades, the local database is subject to data loss if it contains data that has not been uploaded to the server when a database crash or corruption occurs. The intent of the system vision is that all clients should stay connected to the server in most cases, so this should not be considered a critical issue. The ability for a desktop client to continue working while disconnected from the server introduces the potential of validation data cached on the desktop machine to become out of date, allowing data entry of invalid information which would not be identified until the records are uploaded to the server. An additional limitation of the desktop client is that if more than one workstation at a processor is used to submit landing reports then the local database must be setup on a server, and all of the workstations doing landing reports must interact with the same database. This increases the complexity of installing and maintaining the client, and introduces the possibility of users installing the local database on their own workstations. If that occurs then the data needed for making NMFS daily production reports would not complete on any workstation and erroneous reports could be inadvertently submitted. Resolving such problems would be difficult and time consuming, particularly if no IT support resources are available at the processor's site.

3.3.4 Agency Desktop Client

The agency desktop client would provide agency users with the highest performance client option for doing batch data entry of landing reports. In addition it would allow users to review submitted landing reports, modify those reports, manage user accounts, and do various types of summary reports for in-season management and enforcement. It would meet all agency user needs. Like the submitter desktop client, it would require more desktop PC resources and support effort than the agency web client, but this should not be significantly more that that required for the existing agency systems. The cost to develop an agency desktop client is estimated at $80,000 to $100,000. The agency desktop client performance advantages arise because data can be entered and stored locally even if the server or the user's network connection is down. When the client re-established communications with the server all entered data could be automatically uploaded. This technique would also hide the effects of slow communications by allowing the user to continue entering data while it is uploaded in the background. Other advantages would be similar to those of the data submitter desktop client.

The desktop client will affect the overall database architecture of the system because it will need a local database for storing data to allow disconnected operation. This will require database design for the local client database, but should not require changes to the server database architecture.

Like the agency web client, the agency desktop client would be easier to secure than the public data submitter client. Since client requests would be coming from known agency networks access control by IP address could limit usage to only the agency locations desired. A virtual private network could be used if further security was desired.

The installation and maintenance of a desktop client can be expected to be greater than for a web client. The software will need to be packaged with an installer program in order to minimize end user effort to install the application. Desktop applications are subject to workstation and user induced configuration issues. Software version issues are less than for the data submitter desktop client since the agency users can be directed to upgrade their software, and will be expected to do so. The system will still need a mechanism to detect the user's software version.

The ability for a desktop client to continue working while disconnected from the server introduces the potential of validation data cached on the desktop machine to become out of date, allowing data entry of invalid information which would not be identified until the records are uploaded to the server. However, the Internet connections that ADF&G, IPHC, and NMFS provide their users are quite reliable and completely disconnected operations should be uncommon.

3.3.5 Spreadsheet Upload

A spreadsheet upload client would provide an electronic version of the paper landing form that would run on the data submitter's PC as an electronic spreadsheet. Many data submitters are already familiar with spreadsheets and use them in their business. The client would be a spreadsheet format that would look very similar to the paper form. Users would fill out the form in their spreadsheet program, just as they fill out the paper form. The format would validate the data entered, and prompt the user for missing data. Once the data was entered the format would be saved as a spreadsheet file on the user's PC. The saved files would be uploaded to the server using a simple web application. The server would read the data out of the uploaded file and use it to populate a database record, similar to how it would process uploaded landing reports from other clients. The database records would be the same as for data that other clients upload, but might have additional fields to track specific spreadsheet properties such as the version of the spreadsheet, and to store the actual file if desired.

The spreadsheet client upload would meet many of the needs of data submitters who have no network connection when they are collecting the landing data, but who are later in an office where they can make an Internet connection. The advantage of the spreadsheet client over a paper form is the data validation and consistency checking which the spreadsheet client can do at the point of data entry. The validation would not be as extensive as that available on the web or desktop clients, but the spreadsheet client would be simpler for most users. The cost to develop the spreadsheet client is estimated to be $10,000 to $15,000. The cost to the data submitters would be the cost of the spreadsheet software if they do not already have it. In addition, if they wish to use the spreadsheet client in locations where they currently don't have computers then new systems would be needed. The spreadsheet client would require the web client to be developed, since the spreadsheet could not provide user authentication, an upload mechanism is needed, and the spreadsheet itself could not provide desirable features such as the ability to download summaries of submitted data.

The primary security risk with the spreadsheet client is the possibility that files with viruses may be uploaded. If the server processes the file, extracting the data, storing it in the database, and then deleting the file, then this risk is mitigated. If the original file is saved in the database for later review then the file should be scanned with an anti-virus system such as Norton Anti-Virus or McAfee.

Like the desktop client, the system will need to be version aware, and will need to notify users when they upload a file, which was created from an older format. Insuring that users download new versions of the spreadsheet format and dealing with versions that can no longer be processed will increase support efforts, particularly since a spreadsheet format cannot itself recognize that it is out of data and therefore cannot prompt the user with nag messages to encourage downloading a newer version.

3.3.6 Custom and Commercial Software Interface

The custom and commercial software interface on the server would allow large processors with their own IT organizations and commercial software vendors to develop clients which could submit landing reports as well as provide other processor business management functions. The custom client systems could provide for all the processors' business system needs, and would greatly reduce data entry needed for making landing reports. For processors who already have custom business systems in place, the software interface would allow them to modify their existing systems to submit landing reports. The custom software interface would use the same interface components as the desktop client, the only change to the database would be to track the particular client software name and version along with the reported data, in order to identify software explicitly if it causes errors.

The custom client software using the customer and commercial software interface has the potential to best meet the needs of individual processors, since they can specify the exact features they need to efficiently run their business. The cost to provide the interface is relatively small, estimated at $5,000 to $10,000. Most of the interface components will be developed for the desktop client. The cost to processors to develop or modify custom systems to use the interface will depend on the amount of customization required to gather data elements that are not currently being tracked, and to actually communicate with the client interface. The source code of the desktop client can be provided as a reference to help IT developers to build the custom interface software efficiently.

The custom client software interface introduces minor security risks above and beyond those associated with the desktop client. These primarily arise due to exposing more information about the interface to IT personnel outside the agencies. While security through obscurity is considered a poor practice to depend on, public interfaces using software that is not well know are seldom subject to determined and extensive attack. However, the interface should not depend on obscurity as its primary security risk mitigating factor. If the distribution of interface information is limited to known processors and their IT staff personnel the additional risk is small.

The custom and commercial software interface will increase the maintenance effort for the integrated system since changes will need to be tested with the clients using the interface as well as clients which the maintenance organization controls. Rigorous test specifications and procedures may reduce the amount of testing with the custom systems that is actually needed. Custom systems using the interface will constrain the rate of change of the interface itself, since rapid change would be detrimental to the utility of a custom interface available to third parties.

3.3.7 Electronic Logbook Extract File Interface

Electronic logbooks could provide the capability to generate a file with set-by-set or haul-by-haul location, time, and estimated catch information. The integrated landing system could accept this data and use it to generate very accurate statistical area worksheets and at-sea discard line items. Electronic logbooks are being developed, in time they will become common. The electronic logbook interface would not be a complete integrated landing system client. It would depend on the data submitter web client or data submitter desktop client, and would enhance those client applications. The database architecture would not be affected by this interface if the data is only used to populate the stat area worksheet and discard items.

The additional cost to develop the electronic logbook file interface for the integrated landing system is estimated to be $5,000. This does not include the development cost of an electronic logbook system itself, or modification of such a system to produce the extract file of line-by-line hauls or sets. The cost to use such an interface would be the time and effort to export the data to a floppy disk from the vessel electronic logbook, and to deliver the floppy disk to the processor.

The primary security risk with the electronic file interface is the possibility that files with viruses may be received. If the server processes the file, extracting the data, deriving the landing information from it, and then deleting the file, then this risk is mitigated. If the original file is saved in the database for later review then the file should be scanned with an anti-virus system.

Maintenance effort would be increased because maintainers would need to coordinate with the developers of the electronic logbook software to keep the format of the export file in sync. The software will need to detect version differences, and handle out of date versions appropriately.

3.3.8 Summary

The following tables summarize the primary advantages and drawbacks of the different client alternatives described above, their estimated cost to develop, and their interdependencies.

Client	Pro's	Con's	Cost	Remarks
Data Submitter Web	Easy setup No backups needed Version always current Web browser infrastructure in place Validation data always current	Internet connection and server must be up	$40-50K	
Agency Web	Easy setup	Performance insufficient for mass data entry	$60 - 80K	

Client	Pro's	Con's	Cost	Remarks
	Version always current Validation data always current	data entry		
Data Submitter Desktop	More user friendly user features Better data entry performance When connection to server not available has ability to cache data	More complex to install and maintain Workstation configuration can cause problems Cached data used for validation can go out of date Local data can be lost if not backed up Keeping software versions up to date requires effort	$50 - 70K	
Agency Desktop	Best data entry performance for mass data entry	More effort to install and keep versions up to date	$80 – 100K	
Spreadsheet Upload	Simple to use, looks similar to paper form Able to use without network connection when gathering data	Validation at point of entry not as complete as with other clients Generating Landing Report ID numbers problematic	$10 – 15K	Requires web client. This client may be best suited for salmon fisheries
Custom and Commercial Software Interface	Custom systems can best meet processor's needs	Since other organizations will write software to interface specifications, changing those specifications must take into account rate at which those organizations can absorb change	$5 – 10K	Uses same server interface as desktop client, so desktop client must be developed first.
Electronic Logbook Extract File	More accurate stat area and at-sea discard data Less effort to data enter stat areas and	Need to coordinate file formats with logbook software vendors.	$5K	Requires web or desktop client.

Client	Pro's	Con's	Cost	Remarks
	discard information			

3.4 Infrastructure Elements

The technical infrastructure that the system will be built upon must be chosen before design can begin. The hardware and software infrastructure must be able to scale to meet the expected transaction volume. The number of paper fish tickets submitted per year is about 250,000. Of these, approximately 30,000 are groundfish. If each took 10 transactions on the part of data submitters and agency users to complete, then the transaction volume expected is between 300,000 and 2,500,000 per year or 6,000 and 50,000 per week. This establishes the scope of the system transaction volume. If transactions arrive only during business hours this gives an average of between 2 and 20 transactions per minute. The Transaction Processing Performance Council[3], an IT industry organization which benchmarks computer system performance, lists even its lowest performing system configuration at above 10,000 transaction per minute, so the demands of the integrated landing system should be well within the capabilities of common configurations of standard computer hardware and operating environments. In this section alternatives for technical foundation of the system and key system elements are identified and compared.

3.4.1 Computer Hardware

ADF&G currently estimates that fish ticket data requires 1 to 2 gigabytes per year of storage space. The integrated landing system will store similar types and volumes of data. Doubling that figure allows for additional data that the integrated system will store. Allowing for 5 years of data before any expansion might be required and doubling the data space requirement to account for mirroring the data on disk yields a data storage requirement of 40 gigabytes.

Both Intel-based server hardware such as servers marketed by Dell, Compaq, and IBM, and Unix hardware such as Sun and HP are capable of this level of processing and data storage. Servers with approximately 1 GHz

3 http://www.tpc.org

processors, 1 GB of RAM, and arrays of 10 to 20 GB hard drives would be capable of comfortably handing the processing load. The cost of this type of machine is estimated at $5,000 to $10,000 for the Intel based products and 2 to 3 times that for proprietary Unix solutions. While a single server could support all server functionality, and might be an appropriate choice if proprietary Unix hardware is selected, dividing the processing load between 3 servers is a common configuration. One database server, one web server, and one application server would provide the ability to isolate and performance tune for separate functions. Isolating some server functions to separate hardware, particularly the web server, enhances security due to the ability to limit the capabilities of the most likely to be attached machine. Hardware maintenance would be only slightly greater with three servers, and spare parts would be interchangeable if all were the same brand.

3.4.2 Operating Software

The operating software for the server system is somewhat dependent on the hardware chosen. Proprietary Unix hardware is provisioned with the vendor's version of the Unix operating system. Such operating systems are fully capable of supporting the application software envisioned for the integrated landing system, and often have fault tolerant features. Intel-based hardware gives the choice of Microsoft Windows or Linux as the operating system. Either of these is capable of providing the service needed for the application. The cost for Windows is about $2,000 per server. Linux is significantly less, and can actually be installed at no cost, but with vendor support and tools is typically $600 less than Windows. The operating system selection is related to the database management system chosen. Both Oracle and Microsoft SQL Server are in use at the agencies, Oracle at ADF&G and NMFS, and SQL Server at the IPHC. Other relational DBMS systems such as IBM's DB2, Sybase, and PostgreSQL would be capable of hosting the integrated landing system application, but offer no compelling advantages. Microsoft SQL Server is less costly to license, between $5,000 and $10,000. However, it requires Microsoft Windows as the operating system and is considered to be less scalable as transaction rates increase. Oracle is estimated to cost between $15,000 and $25,000 to license. Oracle will run on Microsoft Windows, Linux, and most of the proprietary Unix variants.

The Microsoft Windows operating system is considered to be less secure than Linux or proprietary Unix since it has drawn the attention of Internet attackers in recent years and many exploits have been discovered. However, all these operating software alternatives can be adequately secured if competent systems administrators use industry best practices for hardening the operating systems and applications. Maintenance efforts would be similar for all choices, although Microsoft Windows would be more likely to require upgrades as Microsoft releases new versions and sunsets support for older ones.

3.4.3 Software Development Environment

Although distributed systems can be developed using a wide array of programming languages and development environments the leading software development environments for large scale distributed systems are the J2EE Enterprise system development architecture and the Microsoft .Net initiative. J2EE is based on the Java language and is a more mature enterprise system platform. It runs on Microsoft Windows, Linux, and the proprietary versions of Unix. Oracle supports J2EE with their Application Server. The cost of a J2EE development environment can range from zero licensing cost to thousands. Many open source development environment tools such as the Eclipse Java IDE and the JBoss application server are available without charge on the Internet. Commercial tools such as Borland's JBuilder are available for costs ranging from around $500 to $4,000 per developer, depending on options. The .Net environment is newer and not as well proven. It requires Microsoft Windows and SQL Server. The cost of the .Net development environment is approximately $1,000 to $2,000 per developer.

While all development environments provides features for security, programming practices have a much greater impact on the security and vulnerability of the completed application. Design time, and training for programmers in developing survivable applications, is more likely to provide desired security benefits than marketing features of particular development platforms. Impact on application is similar, being more dependent on the programming practices employed than on the particular development environment.

The development environment for the client software will likely, but not necessarily, be the same or related to that used for server development. The communications protocols will isolate the client code from the server code. This will enable the custom system client interface to be developed using whatever tools the processors' IT providers prefer, as long as they are capable of writing to the communications specification. For the other clients, the development tools will likely be related to the server development environment. For a J2EE environment, desktop clients can be developed in Java. Web clients can be developed as Java Servlets or as Java Server Pages (JSP). For a .Net environment, desktop clients can be developed in Visual Basic or C#. Web clients can be developed as Active

Server Pages (ASP). In either case alternative tools such as PHP for web clients can be employed. No additional cost for developing clients should be required, beyond that of procuring the server development environment.

3.4.4 Client/Server Communications Protocols

After the development environment, the choice of communications protocols will have the greatest influence on the capabilities and characteristics of the integrated landing system. The choices fall into two major categories, Remote Procedure Calls (RPC) and file transfers.

3.4.4.1 RPC

Remote Procedure Calls provide real-time interaction between the client and the server. Common RPC protocols, which are used for communications across the Internet, are J2EE, CORBA, and SOAP. J2EE is specific to Java Enterprise development. CORBA is the Common Object Request Broker Architecture specified by the Object Management Group, an industry consortium that has been developing it for more than 10 years. CORBA is used for both componentization of applications running locally connected and for remote object communications across LANs and WANs. SOAP is the Simple Object Access Protocol, a newer protocol based on XML. SOAP is most often associated with Web Services, applications that provide data to other applications across the Internet. Both CORBA and SOAP are platform and language independent, J2EE is platform independent, but is limited to the Java Language. CORBA is considered a higher performance protocol, both because it is more mature and because it transmits some information in binary form, while SOAP translates all information, even numeric data, into ASCII text. While both CORBA and SOAP are supported by industry consortiums, SOAP is a much more active area of development and more products are being developed to use SOAP-based web services than are being CORBA-tized.

Both CORBA and SOAP can have their data streams encrypted, relieving the applications of this task. Open source implementations of both CORBA and SOAP are available for download from the Internet. In desired, commercial implementations are also available if purchased development software is desired. All RPC protocols are complex, and require maintenance programmers to have training in the specific protocol.

3.4.4.2 File Transfer

File transfer protocols transfer the data between client and server, but provide much less feedback than RPC protocols. The two most common methods used transfer files are FTP and email. The File Transfer Protocol (FTP) is a well-established method for moving files between computers on the Internet. FTP is a reliable protocol. It guarantees delivery of the file. If the file cannot be successfully transferred the sender knows that immediately and can take appropriate action. FTP requires a stable connection between the client and server for the duration of the file transfer; it cannot be initiated if any part of the path between client and server is down. FTP is tolerant of low speed, high latency connections, although this affects performance.

Email is the other commonly used technique for transferring files. Files are attached to email message and sent to a recipient. Email is an unreliable protocol. The Internet makes its best attempt to deliver the message, but can throw it away if undeliverable, sometimes without any feedback to the sender. Email does not require a stable connection between the sender and receiver, the message can be sent as long as part of the path is up, and the protocol will attempt to send it along as the next segment of the path comes up. Because of this, email is tolerant of unreliable networks, but at the price of not guaranteeing delivery.

Both FTP and email are one-way transfers. The only feedback the sender receives is the indication of successful delivery of the file in the case of FTP. All response to the processing of the contents of the file must be provided by the application using another file transfer. The response is necessarily delayed in time since a batch process must run to handle the received file and generate the response. Neither FTP nor email provide encryption of their contents, but files to be transferred can be encrypted by the sending application and decrypted by the receiving application using agreed upon encryption algorithms and keys. Neither FTP nor email require additional development software purchases.

3.4.5 Printed Report Documents

The integrated landing system will be required to print landing reports for signature and submission to ADF&G. Several approaches can be used to produce the printed documents. The client application could print the documents using its native print capabilities, either writing directly to the printer for desktop clients or producing HTML for web clients. The client application could export a file in Rich Text Format or MS Word format which the user could open in MS Word or another word processing program and then print. The client could export a Portable Document Format file, which could be opened using the Adobe Acrobat reader and then printed. None of these options require purchase of development tools; the capability to write to file and printer formats is available in the integrated development environments, or in example code that can be downloaded from the Internet. Using native printing does not require any particular software on client workstations beyond the drivers required for any printing. Using a word processing file format requires that the client workstation have a compatible word processing package. In practice, most processors and agency users already have Microsoft Word or a similar word processing package. PDF files require the Adobe Acrobat reader, which can be downloaded without cost from the Internet, and which has become a de facto standard for documents transmitted across the Internet.

Printing documents does not subject the system to security risks, but the ease with which the information being printed can be changed is a security concern. Using native printing has the least risk of having information changed. PDF files used with the Acrobat reader are read-only, but software programs are available on the Internet that allow PDF documents to be modified. Word processing programs by their nature can modify the files they print, and therefore introduce the greatest risk of data being inadvertently or intentionally changed.

Native printing typically requires the most maintenance effort of any document printing approach in a distributed system because many different client types and clients introduce variations and complexities that the application software must handle. Using a proxy program to print documents, either a word processing program or the Acrobat PDF file reader, insulates the application from the print process, reducing maintenance effort for print problems.

3.4.6 Application Hosting

There are a number of alternatives for where the system will be run once it is developed. One of the three agencies could host the system on their network on behalf of the other agencies. A commercial Application Service Provider (ASP) could host the system. A third party agency could host the system acting as an ASP.

Application Service Providers can provide several levels of service. The base level is to provide hosting for the hardware and software that runs the system. This keeps the hardware physically located at the ASP's site, ensures the system has connectivity to the Internet, and ensures that the system is running. An additional level of service would provide administration support for the system. This would be the system administration for running backups and doing database performance tuning. The next level of support is to provide help desk coverage. Help desks should be able to handle system connectivity issues and availability problems. The help desk also provides the first line call taking for application problems, but transfers those problems to application support staff who are not typically part of the ASP. The help desk may also provide a problem tracking system that tracks both system level and application problems from report to resolution.

3.4.6.1 Agency Computer Room

The integrated landing system could be co-located with one of the existing agency systems in the responsible agency's computer room. The integrated landing system would receive the same service level and security provisions as the other agency systems with which it is co-located.

Co-locating at ADF&G, NMFS, or IPHC has some advantages. If agency IT personnel are part of the application development and support team this would put the integrated system closer to support resources. The network connection to the hosting agency's system would have higher performance and would be have higher ultimate reliability than any physically remote connection since there would be less network to traverse between those two systems.

However, there are several disadvantages of co-locating at an agency operations room. Currently, none of the agencies have fully supported 24 X 7 computer operations. While application support personnel might be closer to the system, they might be less motivated to resolve problems that were not impacting their own agency directly. A cost transfer arrangement would have to be negotiated between the participating agencies and the hosting agency. The hosting agency might need to increase its network and system administration support staff.

3.4.6.2 State of Alaska Data Center

The State of Alaska Information Technology Group operates a 24 X 7 data center in Juneau. They have begun offering server hosting services to other agencies on a fixed fee basis. The ITG data center is one of the largest in Alaska. Their service offerings are similar to commercial ASPs. Telecommunications are provided under contract by Alaska Communications Systems. ACS also manages the firewall behind which the system would run. ITG would provide the computer hardware, enterprise class Dell 2550 servers running Microsoft Windows or Linux. The servers have a 3-year replacement cycle, thus avoiding most problems associated with aging hardware. ITG provides 24 X 7 monitoring, hardware, and operating system support, as well as help desk service. Service operations would include backups, application restarts, and operating system patches. The help desk would take first line application problem calls, but would route them to whatever organization the agencies choose to do application development and support. ITG does not offer application support directly.

The cost for hosting the integrated landing system at the State of Alaska data center is estimated as at approximately $30,000 per year, based on a charge of $8029 per host per year and $10 per host per gigabyte of tape backup per month.

3.4.6.3 Commercial ASP

Commercial ASPs such as ISI of Beltsville, MD, RackSpace, and SkyNetWeb operate large state of the art data centers where applications can be located under contract. They appear to have an extremely secure and rigorous operation. Their service offerings include hosting the application on agency-supplied hardware, at their data center. The data center provides 24 X 7 application monitoring and help desk support. System administration such as backups, offsite storage of tapes, server reboots, and minor system operations is included. They would provide T-1 access to the Internet. They do not provide application programming support, but their help desk would provide first line application support and would route trouble reports to an agency specified application support group. They provide firewall security and intrusion detection, as well as physical security.

The cost for hosting the integrated landing system at a commercial ASP is estimated at approximately $48,000 to $56,000 per year. Charges are on a monthly basis, and many ASPs offer optional service in addition to base hosting.

The disadvantage of commercial ASP hosting is that if service is inadequate the only alternative is to move the system to another ASP or to bring it back in-house, which is a complicated and costly undertaking. Also, if the ASP selected is unstable or goes out of business this would adversely affect the application.

4 Recommendations

In the Cooperative Interagency Electronic Fishery Information Collection and Management Program Needs Analysis report we concluded that an integrated landing reporting system should be developed. In this section we offer recommendations to give the development project the best chance for success. In a frequently quoted report published in 1995 the Standish Group documented the dismal failure rate of IT projects and identified a set of commonly acknowledged success factors.[4] In priority order they are:

- User Involvement
- Executive Management Support
- Clear Statement of Requirements
- Proper Planning
- Realistic Expectations
- Smaller Project Milestones
- Competent Staff
- Ownership
- Clear Vision and Objectives
- Hardworking, Focused Staff

In the following recommendations we attempt to address these factors in a manner that fits the needs and vision of the integrated landing system.

4.1 Staged Development Recommendations

In order to provide for smaller project milestones, we recommend that the integrated landing system be developed in stages. The nature of the envisioned system, with a central server and multiple clients, facilitates this because not all client components are needed for the system to operate. Staged development will produce a deployable system more quickly than if

[4] The Standish Group, "The Chaos Report", 1995, http://standishgroup.com/sample_research/chaos_1994_1.php

all features are developed before system test and deployment begins. In addition, during each stage the lessons learned in the prior stage can be used to improve the system. We recommend the stages detailed in the following sections, but the actual contents of each stage should be decided during the planning for that stage.

4.1.1 Preliminary Development – Technology Demonstrator

Since an inter-agency software development team is not yet in place, and the integrated landing system will be built using technologies that the agencies have not used extensively, a small initial development project will provide an opportunity to exercise tools and development procedures. The objective of this development should be a server, and client software that can be distributed to agency field offices and selected processors. The software would simulate the data traffic between clients and server that is expected in the integrated landing system. The software should record the performance of communications and processing, and the incidence of communications and system errors. This will confirm that the performance of the chosen protocols, software development environment, and infrastructure are adequate to support the integrated landing system. If they are found to be inadequate it gives project managers an opportunity to make decisions to correct the situation or redirect the project resources.

4.1.2 Stage 1 Limited System

The limited version of the system would be the minimum complete system that could be deployed and that would provide value to the data submitters and agencies. The NMFS Electronic Report functions might be a part of the limited system, but probably would be deferred to the next stage. NMFS IFQ reporting functionality would be included.

The server and database for the system would be developed during this stage. The agency interfaces for the ADF&G Fish Ticket Database, NMFS IFQ system, IPHC Halibut Fish Ticket system, and CFEC system would be developed. The agency desktop client should be developed as the client for agency users. The web client for data submitters should be developed as the first client for those users, providing the least support intensive client first.

4.1.3 Stage 2 Full Basic System

The full-featured basic system development would add the NMFS Electronic Reporting features to the system if they were not developed in the prior stage. The clients would need to be enhanced to provide Electronic Reporting functions for the users. The server and the Electronic Reporting system itself would have to be modified to create the system interface. In addition, the data submitter Desktop Client would be developed.

4.1.4 Stage 3 Enhanced System

After the basic system has been developed it will be substantially full-featured. Additional development will enhance the system to make it more usable for end users, and to provide features identified during development and production use. The system and its interfaces will be relatively stable at this point, so custom and commercial systems interface can be developed with confidence its specification will not be volatile. The agency web client can also be developed to access to the system to a larger population of agency users with less support demands than the desktop client would demand.

4.1.5 Stage 4 Additional Features

Additional features such as the Spreadsheet Upload capability and the Electronic Logbook extract upload could be developed in an additional stage. However, these features are likely to be developed only when needed. For the electronic logbook interface this would be only after electronic logbooks come into common or required use. The development of that feature would have to be coordinated with the logbook software developers.

4.2 Phased Deployment Recommendations

Just as we recommend developing the system in stages so to we recommend deploying it into production use in a series of releases rather than all at once. Not only will this allow for smaller project milestones, but it will also provide benefit to some users sooner than would otherwise be possible. A phased approach to deployment would also reduce risk, since if the initial version of the system has problems a much smaller user community will be affected.

The phased deployment cuts across the development stages. They can be mixed and matched according to how they best meet agency goals and preferences, but the following proposed deployment phases use the NMFS Electronic Reporting development to tie stages and phases together.

4.2.1 Phase 1 – Groundfish, Including IFQ Fisheries

We recommend the initial deployment and production use of the integrated landing system to be for groundfish landings. The relatively low volume of groundfish landings poses less risk for the system. The integration of landing reporting and IFQ reporting offers a significant reduction in reporting effort for the submitters. ADF&G, NMFS, and IPHC will all participate in the system deployment and will begin to receive its benefits. The NMFS Electronic Reporting features of the system can be deferred to the next deployment phase with minimum impact to the agency, but a significant reduction in development effort. This will provide system benefits to the agencies and data submitters sooner than if Electronic Reporting is included.

4.2.2 Phase 2 – NMFS Electronic Reporting

The second phase would deploy the NMFS Electronic Reporting functionality to the same data submitters who had begun using the system in the first phase.

4.2.3 Phase 3 – Westward Region Crab

The following deployment would bring the Bering Sea crab fisheries onto the integrated landing system. At that point the Bering Sea crab and groundfish processors would be able to do all their landing reporting using the integrated system.

4.2.4 Phase 4 – Other Shellfish

Once the Bering Sea crab landings are being processed with the integrated landing system other shellfish landings can be added. This will involve additional processors and will reveal how the system responds to additional volume before bringing on salmon and herring in the next phase.

4.2.5 Phase 5 – Salmon and Herring

Salmon fisheries generate the greatest volume of landings in the shortest time period. Leaving salmon and herring until this phase will allow the software to stabilize and will allow system developers to tune performance to the level needed before attempting to service the greatest transaction volume. The volume of herring landings is incrementally low and can be accommodated in the same deployment phase.

4.3 Development Team Recommendations

A strong development team will be important to this project, both the IT specialists who will build the software, and the agency policy makers and project managers who will decide on directions. This is particularly important in the case of an inter-agency project where the lines of authority will be outside traditional agency organizational structure. Specific recommendations for the development team are:

- Maintain an inter-agency steering team to provide overall direction and to represent and report to agency management, PSMFC, and the North Pacific Fishery Management Council. Agency management is needed to provide the executive management support for the project, PSMFC provides administrative grants management, and the Council approves record-keeping initiatives for FMP fisheries.
- Include agency users on the requirements definition team. These users should continue their participation once requirements specified by participating in test procedure development and system test activities.
- Include data submitter users on the requirements definition team, or a supplementary advisory team. They, too, should later participate in test activities.
- Include agency IT staff members on the development team. They are intimately familiar with the issues of processing landing data, as well as their own systems. That knowledge will be highly valuable to the development team.
- Staff the balance of the development team with contract developers. The scope of the integrated landing system is greater than can be addressed by the current agency IT personnel unless they are relieved of their current responsibilities for supporting the existing systems. Contract developers can be used to provide greater resources than would otherwise be practical, and the number of contract developers can be reduced during lulls in development between phases or stages.

Contract developers should include personnel highly skilled in the chosen technologies who can act as mentors for other staff. IT mentoring is the best way to manage the adoption of new technology.

4.4 Technology Recommendations

In terms of the success of a development project the technology chosen is perhaps the least important of the critical decisions to be made. By that we mean that a skilled development team with strong project management and well-defined requirements is likely to produce a successful system with any appropriate technology; the specific technology chosen is seldom the determining factor in the success of a project. That said, we offer the following recommendations as good choices for the integrated landing system project:

- Java as the Development Environment – Java and the J2EE enterprise architecture extensions is a very widely used development environment for large scale distributed systems. It is a relatively mature technology and both NMFS and ADF&G have already begun using it.
- Oracle as the Database – Oracle is a highly capable RDBMS and NMFS, ADF&G, and CFEC already have made it the foundation for their systems.
- Linux or Windows 2K as the Server Environment – Either Linux or Windows 2000 can provide a suitable operating platform for the envisioned system at the level of query and update activity anticipated.
- SOAP as the Communications Protocol – While SOAP is a relatively immature technology it has a great deal of support from major players in the Information Technology industry and can be expected to improve rapidly. SOAP will likely become the de facto standard for distributed system communications across the Internet over the next 3 to 5 years.
- State of Alaska Data Center as the Hosting Location – The State of Alaska data center is located in Alaska, has the 24X7 support level needed for the envisioned system, a help desk for first line support, and a reasonable cost.
- License the Data Submitter Desktop Client Software as Open Source – The source code for the data submitter desktop will be a useful reference for custom or commercial system developers building software that will submit landing reports. Open source licenses have come into being that encourage the redistribution and improvement of

software while preventing the derived software from then being distributed under restrictive licenses. This allows commercial developers to use the source code in their own software if they provide their source code and improvements to licensees with no restrictions on redistribution. Alternatively, commercial developers can use the source code to understand the program, and then develop their own source code to license and sell as a commercial product.

4.5 Project planning and methodology recommendations

A strong development methodology can make great contributions to project success. Activities that should be addressed are:

- Project Plan – A software project plan is generally concerned with how a system will be designed, constructed, and deployed, rather than what will be built. A strong project plan is especially critical for this project since the direction of the project and ultimate ongoing support will be the responsibility of not one, but three agencies. The details of the project team responsibilities, conflict resolution procedures, agency IT group participation, technology direction, and development methodologies should be decided on and documented in a project plan.
- Requirement Specification – A clear statement of requirements is a key success factor for software development projects. Requirements should be specified in sufficient detail so that developers are not left to make business decisions, and so that project scope can be effectively managed. We recommend using a standard such as the IEEE guidelines to organize requirements efficiently and to reveal where more detail is needed[5]. Requirements specification can be done once at the start of the project, or can be done incrementally, with a new requirements specification written at the start of each development stage.
- Design and Implementation - We recommend a lightweight design and implementation methodology such as eXtreme Programming[6]. XP is a set of practices including short development cycles, pairs of

[5] Institute of Electrical and Electronics Engineers, Recommended Practice for Software Requirements Specifications, ANSI/IEEE 830-1998, New York, 1998.

[6] Beck, Kent, Extreme Programming Explained: Embrace Change, Addison-Wesley, 1999.

programmers working together, automated unit testing, and iterative design improvement intended to help a development team produce software quickly while allowing user team representatives to prioritize features and manage change.

- Deployment and Support – The testing and acceptance activities, user training, production startup, and ongoing support of the system must be managed with as much attention as the initial development in order to assure a trouble free system. Rigorous testing by the user team using documented test plans and procedures should be undertaken to validate that the users are getting the system they need. A beta test program with selected processors should be conducted before general release of the system, to exercise the system under actual production conditions while allowing problems to be managed.

4.6 Recommendations to Support System Development

Finally, the Needs Analysis Challenges section identified a number of things that would have to be addressed during the development of the system. The following are specific items that are external to the system, but are critical to its operational usage.

- Add product code/delivery code values for the proposed concepts of condition code and disposition code. Condition code would identify the physical form of the fish coming across the dock and for which the scale weight at landing time is taken. Different condition codes would require different product recovery rates to be used to calculate such things as round weight or net weight for halibut. Disposition code would identify what happens to the fish after the landing. Disposition codes could identify the results of processing such as fillets or meal, and can identify non-processing dispositions such as discard at sea, and reason information such as discard flea infested.

- Change regulations or obtain interpretations to allow the electronic landing report to be used in place of State of Alaska Fish Tickets.

- Change regulations or obtain an interpretation to allow the CFEC card not to be imprinted on landing reports if it is verified electronically. The financial industry is able to handle the processing of credit card transactions without imprinted receipts, electronic verification should be possible for permit cards.

- Change regulations or obtain an interpretation to allow the initiation of electronic landing report to satisfy the requirement to start fish ticket before the offload is complete.

- Change ADF&G groundfish/shellfish statistical area or IPHC/NMFS regulatory area boundaries to align them, in order to eliminate ambiguity in mappings. This will allow the system to derive the larger regulatory area given the statistical area.

- Change regulations or obtain an interpretation to allow the initial landing report entered and printed on the system at the completion of the offload to be the document that is signed, and to allow it to be enhanced with product disposition and economic data later without requiring another signature.

- Change regulations or obtain an interpretation to allow landing report data to be transmitted back to the processor that submitted the original report. The current interpretation of ADF&G confidentiality regulations only allows the data to be supplied to the holder of the permit card under which the fish ticket was submitted.

226 Seward St.
Suite 210
Juneau, AK 99801
Phone: (907) 586-6167