A description of a Delta Real-time Enhanced Acoustic Monitoring (DREAM) Program

**Optimizing AT technologies and deployment methods that are linked to an information dissemination system based on pilot-level modern databases and Bayesian statistical models**

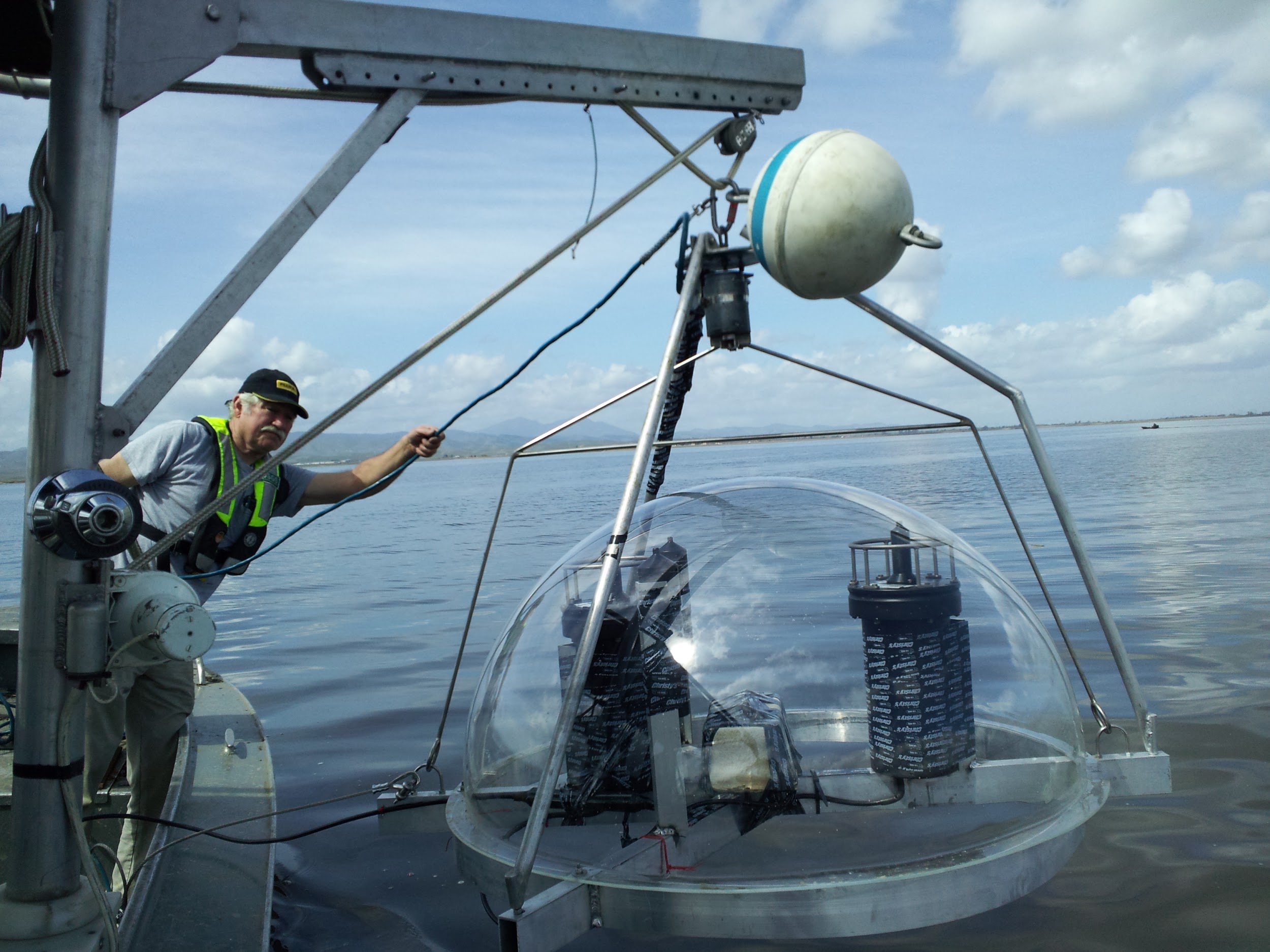
Jon Burau1, Aaron Blake1, Russell Perry2, Adam Pope2, Matthew Sholtis2, Tyson Hatton2, Chris Vallee1, and Noah Adams2

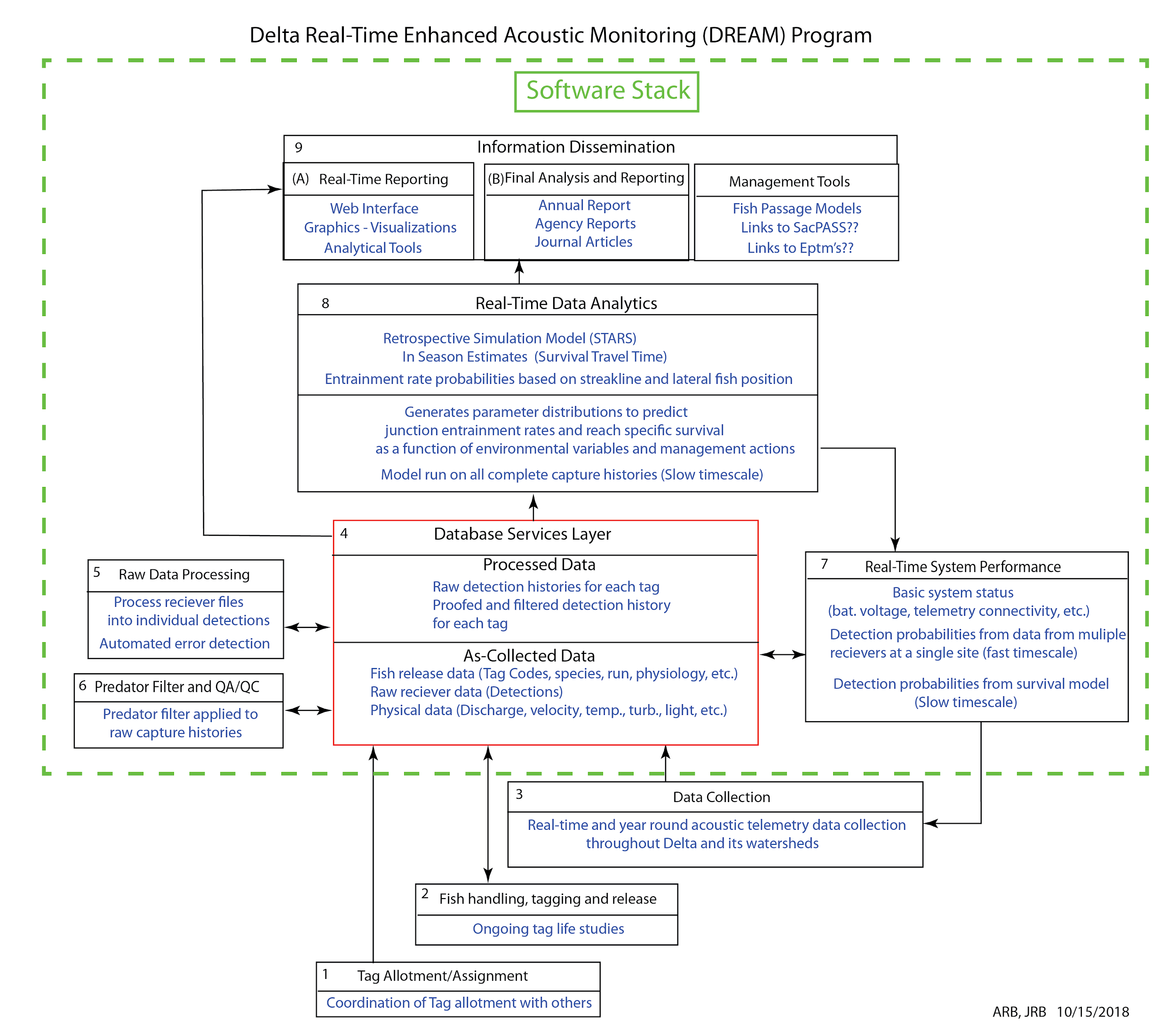
1California Water Science Center, Sacramento CA.

2Columbia River Research Laboratory, Cook, WA.

**U.S. Department of the Interior**

**U.S. Geological Survey**

Cover Picture - Deployment of acoustic telemetry receivers in Clifton Court Forebay, a regulating reservoir for the State Water Project pumping plants (Photo, courtesy of Chris Vallee, 2012).



**Figure 0.1 -** Dream Program component parts. All of these components were piloted, singly and as a system, as part of this report.

# Chapter 4: The DatabaseServices Layer: Clearinghouse for AT and Associated Data

## 4.1 Introduction

A telemetry technology was introduced to the Delta from the Columbia River system by the USGS in 2006 at the urging of Randy Brown (DWR, retired). Even though this technology has been in place for roughly 10 years, the deployment of AT receivers and the tagging and release of fish has been done in a somewhat ad-hoc, organic manner by a variety of groups (DWR, USGS, NMFS, USFWS, CDFW, UCD, several consultants, etc.) using a variety of funding sources (DWR, USBR, CDFW). Within the last few years, these studies have been coordinated through the Biotelemetry Project Work team under the auspices of the IEP. In most cases, deployments were made for specific purposes with deployments lasting several months during the salmon out-migration season (November – April) (DWR, 2012, 2015a, 2016, Plumb et al. 2016, Chapman et al. 2012, Perry et al. 2010). Historical AT efforts in the Delta mainly involved field-focussed efforts that essentially stopped at box 3 in Figure 0.1. Because most of these studies were loosely connected, the data they generated are stored in a collection of individual file systems and databases, making the analysis of the full suite of historical AT data and cross disciplinary inquiries, such as combining AT and physical data, challenging and time consuming. We propose to address this problem through the DREAM program by deploying the IT infrastructure required for the long-term storage and dissemination of AT and environmental covariate data collected in the Delta; just as the CORE array will provide a permanent foundation for AT studies in the Delta, this IT infrastructure will provide a permanent foundation for storing and disseminating this data.

Within this document we refer to the collection of computer hardware and software that will comprise this IT infrastructure as the Database Services Layer (DSL). We refer to a “Database Services Layer” instead of a “database” because achieving the DREAM project’s goal of facilitating broad adoption of AT mark-recapture methods within the Delta will require IT infrastructure that is more sophisticated than a single server running a database instance; it is likely that the DSL will consist of multiple servers supporting several database topologies, one or more file shares, and the software and servers required to support a public facing web based data portal.

## 4.2 Pilot Implementation

In order to make recommendations for the DREAM program, the USGS developed and deployed a fully functional pilot DREAM model stack. This effort included prototype versions of all of the hardware and software components required to interface with and model the data gathered by a network of AT receivers into statistically rigorous estimates of survival and route selection within the North Delta. The purpose of this development process was to test various architectures for the model stack components and to develop beta versions of many of the model stack components that could serve as a basis for the first roll out of the DREAM program.

The heart of the pilot model stack is a Database Services Layer comprised of three main components running within the Amazon Web Services (AWS) cloud ecosystem: A file store hosted in AWS S3, a MongoDB unstructured database running on AWS EC2, and a SQL database containing data to support statistical modeling hosted on AWS RDS (Figure 4.1). These components are supported and linked via software written by the USGS which is run on AWS EC2 instances. The structure and purpose of these components are described below.

### 4.2.1 Amazon Web Services: foundation of the model stack

We chose to develop the pilot model stack within the Amazon Web Services “ecosystem” of cloud services (the collection of these services is referred to as the “Amazon stack”). The Amazon stack goes well beyond the remote hosting of servers; AWS services provide infrastructure as a service (IAAS), software as a service (SAAS), and platform as a service (PAAS). In simple terms IAAS, SAAS, and PAAS allow users to purchase cloud based IT functionality, without needing to support, service, maintain, or scale the actual hardware or software supporting the desired functionality. The ability to purchase this type of functionality on AWS allowed us to focus on developing the components of the DREAMs model stack that required our specific expertise without expendingresources on provisioning and maintaining servers and software. We strongly recommend that the DREAM program model stack continues to be hosted/developed within AWS.

### 4.2.2 File store

The purpose of the file store is to archive the raw receiver files generated by the CORE array in a manner that is highly persistent (low probability of data loss), low cost, and extremely scalable. In addition, the DREAM program file store should be designed in a manner that supports global file addressing and quick access. While it would be possible to achieve these goals by storing receiver files within a database, a cloud based file store is a better choice because this architecture is more economical for storing large datasets that do not need to be searched, and which expand continuously while needing to be stored for extended periods of time. For these reasons, we chose to use Amazon S3 for the DREAM file store.

During the development of the pilot software stack (hereafter referred to as the pilot development) we used AT software to upload raw receiver files from our test equipment directly to Amazon S3 buckets stored in AWS (“bucket” is a term for a S3 file address). After a new file was uploaded we used Amazon’s Simple Queue Service (SQS) to queue uploaded files for automated processing in LabVIEW software written by the USGS. This USGS software filters the raw data contained in the receiver files to extract tag detections which are stored in the primary database.he S3 address for the file generating each tag detection is stored along with the detection metrics in the primary database, which allows users to quickly find the raw receiver data comprising each detection. We did not prototype the ability to upload other raw file types to S3, but it would be simple to extend the prototype architecture we developed to process files from other types of instruments.

The final capability that should be built into the DREAM Program file store is the ability to archive old receiver files that are unlikely to be accessed in a low cost manner. We did not test this capability, but, there are two good options for low cost archival of data stored in S3. First, data can be transferred into Amazon Glacier, which is highly durable (99.999999999% average annual durability) cloud based storage. At the time of this writing storing a terabyte of data in Glacier costs about $50/year. Second, data stored in S3 can be transferred to physical media (e.g. hard drives) by Amazon, or, it can be downloaded to physical media directly.

## 4.5 Primary Database

Within the pilot development, the primary database is the main archive for AT information and associated environmental data and fish physiology information. While raw receiver files are archived in the file store, the information extracted from these files is archived in the primary database. We chose to use MongoDB for the primary database because MongoDB’s JSON based document store architecture allowed us to store a variety of complex data structures in a single database without extensive software development. Further, we feel that MongoDB or a similar NoSQL database will provide a flexible foundation that can be used for a full implementation of the DREAMs project. We chose to use a NoSQL architecture for the primary database instead of a more traditional relational database for the following reasons:

* Many NoSQL databases work on an aggregated data model, which means that they can store an arbitrary collection of data for a single primary key value.  This is a good match for a “fish centric” approach to storing all types of data associated with a single acoustic tag code, and allowing users to access this data without performing complex join operations.
* NoSQL databases are schema-less, which means that each record can contain different amounts of data, different types of data, and the structure of this data does not need to be known when the database is deployed.  We feel that this approach will reduce development time overall, as different research projects will collect different types of data, and new techniques and technologies may create data structures that we can not anticipate a priori.  For example, the initial data stored in the DREAM database may only consist of raw detections for each tag code, but a schema-less approach would allow future studies to add physiological data, release data, digital photographs (used to document fish condition), filtered detection histories for a variety of survival models, a 2D track from a high resolution tracking array, and acoustic camera footage to the data stored for each tag code, and both types of records could be stored in the same database.
* NoSQL databases are generally very easy to scale out through sharding.  Sharding is a method of increasing a database's ability to absorb and store data by breaking the database into multiple parts hosted on separate machines (generally in the cloud). This approach is a good match for the DREAM network’s scaling needs, as network writes are likely to scale significantly, while database reads are likely to scale very slowly.
* NoSQL databases generally allow users to store complex data without having to write software to translate the data into a relational database structure.
* Most mature NoSQL databases are open source and carry no licensing costs.

We deployed the primary database within Amazon AWS using a single MongoDB server instance hosted on a Windows Server instance within EC2. During the course of the pilot development we tested the primary database using both existing files from past studies, and to a limited extent, files collected in the field during equipment testing. During testing we found the flexibility of MongoDB’s document store architecture to be advantageous for storing AT detections from a variety of technologies that produce different types of detection data. We developed beta versions of the software required to process and store information files collected on HTI nodes and on HTI’s cabled multidimensional tracking systems, on JSATS receivers, and on VEMCO receivers of several different generations.

One of the design requirements for the primary database is that it be able to support public access (at least read only) to allow for broad dissemination of the CORE Array data and associated environmental covariates. We did not test this functionality because designing a public facing data portal was beyond the scope of the pilot development, however, MongoDB is used widely for very demanding customer-facing database applications so we are confident that it can support a public facing data portal (see <https://www.mongodb.com/who-uses-mongodb> for example use cases).

## 4.6 Modeling Database

One of the main purposes of the DSL is to support statistical mark-recapture modeling. The USGS CRRL team that performed the statistical modeling for the pilot development use the R programming language extensively. While testing the primary database we found that it was inefficient to access detection data stored in MongoDB using readily available R libraries, while the R libraries for SQL queries are very efficient. In addition, we found that the database requirements for the statistical models were a good match for a relational database. For these reasons we chose to deploy a separate SQL database to support the statistical models. This databasecontains a subset of the data contained in the primary database, and has no provision for public access. During the pilot development we deployed the modeling database using Amazon Relational Database Service (RDS), which provides database-as-a-service for a variety of SQL implementations. The modeling database was populated automatically by LabVIEW software written by the USGS that searched the primary database for new detection information that met filtering criteria, and then added the new detections to the modeling database. This software was run on an Amazon EC2 instance; we expect that for the initial DREAM Program deployment this process would run once a day. One advantage of the pilot architecture is that the modeling database is completely separated from the design and operation of the primary database; each database can be built and operated to meet independent (and likely competing) engineering requirements.

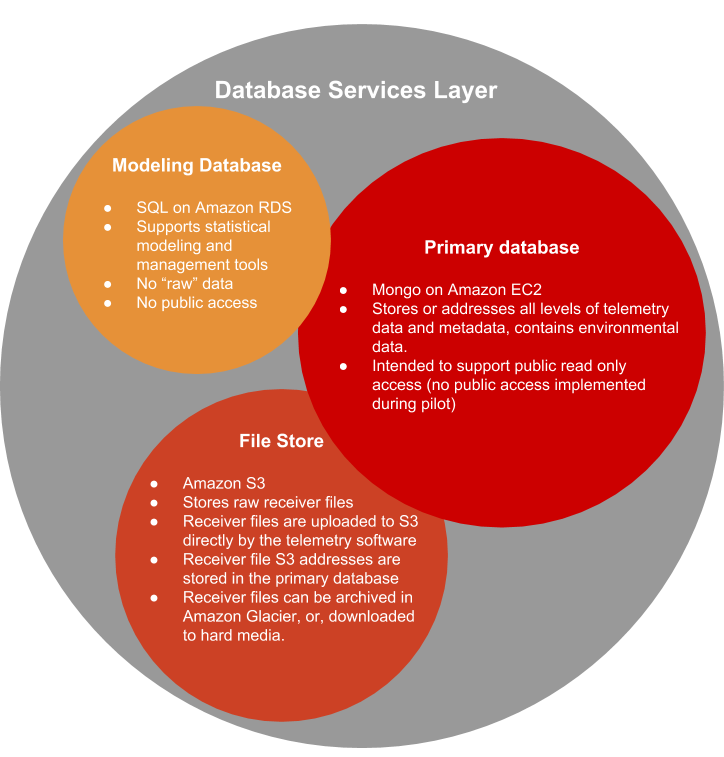
****

Figure 4.1 - Illustration of the components comprising the Database Services Layer deployed in the pilot DREAMs model stack implementation

## 4.7 Recommendations

One of the primary reasons to consolidate AT data and environmental data into a single data store is to reduce barriers-to entry for conducting AT studies so that a greater diversity of management questions can be addressed with limited research funds. If the DREAM project achieves this goal, a wide variety of research teams will be able to use the Core Array to simultaneously monitor juvenile salmon, adult salmon, resident and migratory predator species, sturgeon, and other species of concern. Additionally, researchers will be able to augment the permanent Core Array with temporary multi-dimensional tracking arrays to further leverage tag budgets by collecting high-resolution data on the same fish used to estimate survival and route selection parameters. If the Core Array achieves this level of utilization, massive amounts of data will need to be stored, processed, analyzed, archived, and served to the scientific community. For this reason the DSL that is implemented in the early years of the DREAM program will need to be designed around an architecture that can support the long term goals of the DREAM project, and these components will need to be deployed in an environment that will support rapid scaling.

There are many possible design approaches for a DSL that will support the DREAM project’s goals. The pilot DSL that we deployed is one such design, and our pilot implementation can be scaled up to support the DREAM project. We recommend that the pilot DSL be considered for use in the initial DREAM Project deployment as this will speed up the model stack deployment. In addition, we make the following general recommendations for developing and deploying the DREAM DSL

### 4.7.1 Design philosophy

It is important to recognize that a wide variety of database topologies can be used to achieve a particular software engineering goal, and often the tradeoff between these databases comes down to the nature of the software development required to integrate each database into the overall project. In developing the pilot DREAM Program DSL our design philosophy could be summarized as “Allow scientists to focus on scientific programming; if there will be a software engineering burden, place it on professional developers.” Thus, we chose to use multiple databases, each of which was chosen to make integration with our existing scientific programming tools as fast and easy as possible. The tradeoff is that if our pilot DREAM DSLdesign is deployed in the future, these design choices are likely to create a software engineering burden for the developers who are tasked with developing a public facing data portal. We feel that this tradeoff is desirable, and we recommend that the DREAM Program model stack and DSL be designed to minimize the programming burden placed on scientists.

### 4.7.2 Platform

We strongly recommend using Amazon Web Services (AWS) as the platform for most of the DREAMdatabase and model stack components because AWS allows users to rapidly deploy and scale reliable IT infrastructure, and provides a variety of services that can be combined easily to create complex systems. If AWS is not used, we recommend using a similar cloud based platform that provides a “functionality”-as-a-service model.

### 4.7.3 File store

We strongly recommend using Amazon S3 as a file store, both due to the functionality of S3, and the ability to combine S3 with other AWS services such as the Simple Queue Service and Amazon Lambda, which make integrating file storage into AT and database systems easy and efficient. If Amazon AWS is not used, we recommend using a similar cloud based storage system such as Google File Store or Windows Azure.

### 4.7.4 Modeling database and other application databases

Given the widespread use of the R programming language amongst statisticians and modelers, we strongly suggest using a relational database for the portion of the database services layer that supports the statistical models used in the DREAM Program. In addition, as the DREAM Program develops we recommend that other members of the DREAM consortium consider creating similar “application databases” that contain a subset of the data stored in the primary database and which utilize a database topology that best matches the needs of their specific application.

### 4.7.5 Primary database

We chose to use MongoDB for the primary database during the pilot development because it met our design goals (flexible storage of a wide range of data types, easy sharding, easy to setup, etc), and was based on a document-store model that allowed us to integrate database connectivity into our existing suit of data processing software with minimal software development. However, there are a myriad of database designs that could be used for the DREAM Program primary database (including SQL), and we are not sufficiently experienced in database design to make a strong argument for choosing MongoDB over other plausible database options. Instead we make the following recommendations: 1) it is important to consider how potential database designs will impact the teams responsible for collecting and processing CORE array data, and, 2) if MongoDB is used for the DREAM Program we recommend testing cloud services that offers MongoDB-as-a-service, such as MongoDB atlas, instead of deploying a MongoDB server in EC2. In the course of our testing we have found that database as a service platforms tend to have lower operating costs and are markedly easier to maintain and scale.

### 4.7.6 Public data portal

In order to fully realize the DREAM Program goals the model stack will need to incorporate a public data portal to provide outside users with easy access to the AT data and environmental data stored in the primary database. We did not prototype this capability during the pilot development, but we expect that building a robust public-facing web-based interface for the primary database will be a significant software development project. We make the following recommendations:

1. We strongly recommend that a professional software development firm with demonstrable experience in developing web-based interfaces for complex data sets be contracted for this work.
2. We recommend waiting to engage with development teams until the DREAM Program model stack and Core Array have been deployed and operating for at least a full AT season. It will be most efficient to engage with outside developers when the primary database is populated with actual AT and environmental data, and when the DREAM Project team can articulate clear use cases for the data portal.
3. We recommend planning for significant upfront and recurring costs associated with developing, deploying, maintaining, and supporting the public data portal. It is important to recognize that if DREAM Program scientific team members deploy and operate the primary database (e.g. the USGS, NMFS, etc.), these staff members will need to spend a significant amount of time working with the team developing the public data portal.

### 4.7.7 Budgeting

If the pilot database services layer developed during this effort is deployed for the DREAM Program, we expect that continuous operation of the DSL to support the Core Array for the first two years can be achieved for around $600/month in cloud/hosting costs; we would recommend budgeting closer to $1000/month to allow for inefficiencies during the development and deployment process. In addition, there will need to be staff time budgeted for deploying the DSL and improving the software used for data processing and for optimizing the software linking the modeling database to the primary database.