Animal identification and tracking in Romania using open source software[1]

Introduction

The project on animal identification and registration in Romania started in 2004 with the objective of designing, building and operating an integrated information system to collect, process and store information about:

- holdings: farms, slaughterhouses, fairs and markets, rendering units, etc.,
- animals: cattle, sheep, goats, pigs, and horses,
- owners.

The National Sanitary Veterinary and Food Safety Authority organized a tender to find a partner able to manage the entire process, with tasks including:

- source, import, manage and physically distribute more than 20 million ear-tags for the existing stock of animals (whose numbers had already been estimated) and those to be born during the 10-year contract period;
- negotiate, contract and pay the staff needed for ear-tagging and data entry;
- evaluate the IT needs of the system; plan, contract and buy more than 3,000 workstation computers; contract and pay for communications services; buy servers in a data center; upgrade workstation hardware after five years; provide full hardware service for the duration of the contract for equipment distributed all over the country;
- analyze, program, implement and operate the entire information system for the next 10 years so as to deliver to the Authority all the information needed to ensure full compliance with national and European regulations on animal identification.

The tender imposed no technical solutions on the business approach, hardware, or software. For example, one of the offers proposed a number of regional data centers at which operators would key in data from documents received by post. The winning solution, on the other hand, was for a distributed database with hardware at the lowest level: the veterinarians.

Along with their technical proposals, the bidders had to state the costs of the entire system in a single, easy to compare manner: the total costs per animal tagged and recorded in a database that the authority can query to create all kinds of reports.

The contract was awarded to a consortium led by IQ Management SRL Romania, comprising:

- Caisley International (ear-tag supplier)
- Computerland Ltd. (hardware supplier)
- IQ Management (system integrator, manager and software supplier).

[1] https://joinup.ec.europa.eu/page/legal-notice

Organization and political background

The identification and tracking of animals involves the registration of holdings, fitting of ear-tags, and registration of animals in the National Animal Identification and Tracking System.

Under current legislation the National Sanitary Veterinary and Food Safety Authority is the central public authority responsible for the organization and control of animal identification.

Veterinary services at the county level include:

- local Sanitary Veterinary and Food Safety offices (DSVs)
- veterinary food safety areas
- veterinary districts

Veterinary districts are units organized under the law in which they work free practice veterinarians.

A veterinary district is a legal entity controlled by a single independent veterinary practice. Each district is responsible for all the veterinary actions required by law: vaccination, collection of samples for laboratory examination, decontamination, disinfection and de-infestation.

Under the new identification and registration scheme, independent vets must also visit each animal holding in their respective districts, ear-tagging the newborn animals to identify them and recording this information in the database.

The holdings, through veterinarians employed by stock dealers, are responsible for animal movement control. Independent veterinarians, however, are the only people authorized to issue certificates of animal health.

Independent veterinarians and dealers are together responsible for registering outbreaks of highly contagious diseases and taking action to contain these.

This system of independent and dealers’ veterinarians, and their support staff, creates a “doctor-client” relationship that is recognized throughout the Romanian livestock trade and has endured for more than 50 years.

Quick facts

Project name	SNIIA (National Animal Identification and Tracking System)
Sector	Animal health and food safety, public sector
Start date	2004
End date	Ongoing
Objectives	Collecting and processing data on animal births, movements, slaughter, health tests and laboratory analyses
Target group	Public bodies, industry
Scope	National
Estimated Budget	€78,000,000
Funding	National
Achievements	More than 2500 veterinarians are using a distributed database recording status and ownership of animals. More than 2000 other people are operating animal events. Two national agencies are using the information to manage subsidies and other activities relating to animal health.

Budget and funding

Initially (2004–2006) the project was financed entirely by the state at an average cost of €0.96 per animal registered. This included:

- ear-tag procurement and
- paper forms distribution
- printing and distribution of paper forms
- an appropriate percentage of the users' IT infrastructure (hardware, software, and Internet connections)
- data warehouse
- communications
- labor for the application of ear-tags, form filling, and data entry

In 2006 the labor component was dropped from the SNIIA contract and replaced by a direct contract between the veterinarians and the National Sanitary Veterinary and Food Safety Authority. As a result, the SNIIA price dropped from to €0.41 per animal.

Since 2011 farmers have been required to supply their own ear-tags, and the contract cost has fallen to €0.30 per animal.

Technical issues

Choosing the system architecture

For each layer of the system we had to choose the software architecture so as to minimize data traffic, maximize system reliability, and maximize the ability for remote servicing in the event of a failure

Veterinarians' workstation operating system

With more than 3000 workstations costs mount up, so the cost of the operating system was one of the reasons we chose Linux. The second reason was reliability: this equipment is distributed across the country, often far from our service department, on veterinarians' desks where anyone can "just test" other programs or games that could bring the system down. Of course we had situations where someone's child, a bright future programmer, installed another operating system and hoped that nothing bad would happen, but there were fewer than 10 of these experiments. We never had to deal with viruses or other kind of attacks on workstations.

We tested many flavors of Linux before choosing Mandriva Linux for its stability, ability to run on all the brands of computers we had to deal with, and compatibility with CDMA modems. The kernel was recompiled to remove unnecessary modules, creating a slim, fast and responsive system. Many of the veterinarians operate in rural areas where power outages and other power problems are common. We therefore chose the ReiserFS journaled file system for its reliability and efficiency in handling many small files.

The desktop was KDE3 configured' to resemble Windows as much as possible, so as to create a friendly environment. Special precautions were taken to block users from messing up the desktop, icons, and folders. We allowed them to use OpenOffice programs and even media players. At the beginning we didn’t set up quotas for space usage so a couple of times we ended up with a lot of MP3s in users’ home directories.

Veterinarians’ database

We choose to use PostgreSQL databases because we intended to duplicate the full structure of tables from the small, local databases on veterinarians’ workstation to the big database in the data center.

PostgreSQL has been carefully tuned to deliver good performance even on a 256 MB computer. Cron scripts were programmed to back up the database daily, maintaining a full list of the last 30 days’ full backup images. The backups proved to be a good idea when it came to dealing with hard-disk failures, after which we succeeded in recovering information from the most recent local backups’. PostgreSQL proved to be a very solid, flexible and reliable database, and has allowed the application to make good use of database tricks such as triggers, conditional indexes, PLPgSQL and PLPerl functions.

Local application

The county DSVs also need access to the database for their work. To make this possible we chose to build a web-based front end. The veterinarians use this locally on their own workstations, which have Firefox pre-installed, while the DSVs use the same application over the Internet.

The local environment uses Apache as the web server, with many small PHP programs and libraries that also proved to be also suitable for upgrading the application via rsync.

Data transmission and database sync

Communications handled by the system include uploading documents created by veterinarians on their workstations; application updates from the server; mail-like messages between users and DSVs; and downloading of animal registers from the central database. We tested a lot of solutions but in the end decided to rely on the incremental ability of rsync. This has done the job well, including partial updates in the case of broken communications. Each workstation has an average of ten minutes of data communication per day – somewhat more than half our initial estimate.

The whole job was done by a Bash script on a cron job. Every workstation is initially programmed to automatically establish a communication channel with IQM’s data center between the hours of 21:00 and 06:00.

In the data center, another bunch of scripts (Bash, Tcl, Python) move the information to the “mirrored” PostgreSQL databases and then, using a java program, to validate and consolidate the information in the big PostgreSQL database.

The “communication server” is an 8-core Intel Pentium XEON 3 GHz with 16 GB of RAM and 1 TB of RAID storage running Ubuntu Linux (previously OpenSUSE). This keeps the last good database image backup of every local PostgreSQL database. These images provided a second line of recovery in a couple of situations where the information was impossible to recover from the workstations themselves, due to flood, fire, or theft.

The “mirror server”, another server of the same size, is used to keep live mirrors of local databases for DSV’s supervising activity.

Central database administration and application

Information coming incrementally from the mirrored databases is stored in a big PostgreSQL database on the NAD (National Animal Database) server, a 24-core Intel XEON 3 GHz with 144 GB of RAM running Ubuntu Linux 64-bit (previously OpenSUSE). 2 TB of RAID storage is provided by 15,000 RPM Seagate Cheetah disks.

The database is backed up every night; the full compressed backup is 220 GB.

Statistics for the main database as of February 2013	
Table	**Number of records**
Holdings	2,467,000
Owners	2,463,000
Animals	82,832,000
Ear-tags lots	2,249,295
Events	3,274,000
Movements	58,227,000 (2 documents for each movement)
Slaughters	19,729,000
Messages	402,000
Internal data cross-references	329,853,000

Central database administration and reporting servers were written in Java in a JBoss environment, using first Echo-1 and then Echo-2 frameworks. For reporting we chose another open-source tool, JasperReports, and for task scheduling (long reports and queries) we used Quartz. Development was done in Eclipse for Java. Many PostgreSQL admin tasks are done using the PgAdmin tool.

To aid its work in regulating a large number of slaughterhouses and food processing companies, the National Animal Health and Food Safety Agency needed an online, web-accessible database of scanned documents. A web interface allows the local county inspectors to upload scanned documents into a document storage system based on Apache Hadoop.

Choosing the workstation hardware

IQM had the task of choosing the computers to be used by the veterinarians. The company asked for tenders from three major suppliers in Romania: IBM, HP and Fujitsu-Siemens. Each vendor was asked to quote for PCs based on the Pentium IV-2.8 GHz with 256 MB of RAM, 40 or 80 GB HDD, no CD-ROM, and 15-inch monitors.

Because all three suppliers agreed to lower their prices to approximately the same level, and because the main contract specified that the equipment should be changed after five years, we decided to buy 35% IBM, 35%HP and 30% Fujitsu-Siemens. This, we felt, would provide good feedback on the computers’ reliability and the quality of hardware service.

Choosing the data communication solution

Probably the most important choice we had to make concerned the communication solution. We needed to optimize costs and reliability, while ensuring national coverage. The problem was difficult because of the geographical distribution of veterinary districts all over the country, usually in rural areas where the only data carriers were old telephone lines.

At that moment, the costs of data communications via GSM were high and the upload speed was limited to that of GPRS (approximately 56 kbps). GSM coverage was also not good in many rural areas.

We found the best price/quality ratio from the provider Zapp using CDMA technology. Zapp supplied us with 1,800 Hyundai telephones which act as CDMA modems when connected to the PCs via USB. The data is sent through an L2TP secure channel directly to our communication server.

Where the CDMA signal was very weak, where possible we installed external Yagi antennas. The CDMA modems give a steady 156 kbps for both upload and download.

For many other locations we chose another Internet provider: Romtelecom. The former state-run Romanian telephone company, which is now majority owned by OTE of Greece, moves data through phone modems very slowly – 55 kbps, and sometimes even 33 kbps – but at a very low price: around €2 per month per workstation.

Software install and equipment distribution

After two months of intensive testing of various Linux distributions we succeeded in setting up a template for each of the three computer models we had chosen. Then we used SystemImager with FlameThrower software to configure 15 computers at a time over the local network. Each batch took 15 minutes including a personalization step, in which each PC received a pre-installed database of local animal owners created during a previous data collection phase.

The prepared computers were shipped individually, each with a modem and a brief “How to install” manual.

By 2012 it was time for the five-year hardware upgrade, which also required a great deal of work. A detailed calendar plan was made and each workstations underwent a “final sync and shutdown” operation before being transported to a central collection point.

During the upgrade the original desktop PCs were replaced with notebooks (see below). The duplication was done using Clonezilla, which allowed us to do 20 notebooks in 15 minutes. This time included running scripts to install the previously backed up database, convert it from SQL_ASCII encoding to UTF, and configure it.

Operating and maintaining the system

Maintenance and upgrade operations that apply to every workstation are managed through scripts which synchronize each machine with a central repository . For troubleshooting purposes we are also able to access individual computers remotely. Both bulk and individual connections are made through SSH tunnels in text mode, or in graphic mode for PCs connected to high-speed networks at county DSVs. With the help of dedicated “rescue” scripts, the remote access approach saved a lot of time and service interruptions by allowing a computer to be remotely diagnosed and fixed in minutes instead of days.

The architecture was designed to be operable with no network access at all, minimizing the quantity of information to be transmitted around the loop. Information in documents and requests submitted by veterinarians in the field passes through a validation process before being used to update the national registers and then passing down to field level again in the form of updates to the local databases. We use MediaWiki as a knowledge database for common problems and solutions, "how-to's" and other short documents. Redmine is used for project management and bug tracking.

The technical support framework was designed to respond quickly and with a minimum of interruption to services. 14 staff in a call center use an in-house application (web based, with PHP and a PostgreSQL back-end) to track problems, isolate their causes and route the information to the right department in real time. More than half of all problems are solved without moving the computer from the veterinarian's office, just by accessing the PC remotely and running maintenance scripts from a dedicated library.

More than 800 versions of scripts and application files have been updated in six years of service. We use Subversion to keep track of these.

From 2007 the system was extended to interface with a laboratory information management system (LIMS) which delivers information about the animals. The service was implemented using web services through secure communication channels.

In 2010 the whole central hardware, taking up two 48U racks, was moved to the IBM-Petrom data center in Bucharest.

In 2012 a new framework was developed for a specific subsidy scheme. This web-based system uses HTML, pure JavaScript/jQuery on the client side, and on the server side the CouchDB NoSQL database for document storage, with a little help from Lucene for fast queries. The middleware is written in Go. Reports are generated as LaTeX documents and rendered to PDF using XeLaTeX.

Equipment

Upgrade

In 2012 IQM started to upgrade the veterinarians' workstations as specified by the contract. A lot of new equipment was tested: notebooks, netbooks and even tablets. We decided to upgrade to HP notebooks with 15.4-inch screens, AMD dual-core processors, 2 GB of RAM and 300 GB HDDs.

We also tested a lot of new Linux distributions, including ArchLinux, SalixOS, Fedora, Ubuntu Linux (with Lubuntu and Xubuntu flavors), and Scientific Linux (a RedHat clone). The final choice was Linux Mint 13 "Maya" (32-bit version), due to its consistent look and feel, good performance, up-to-date repositories, and an exceptional match with hardware. There was no need to recompile the kernel: using the stock kernel everything – wireless chip, CDMA and GSM modems – worked right from the beginning with no need for extra drivers. The new EXT4 journaled file system was used.

For the record, we also tested OpenSolaris and three BSD-like operating systems (FreeBSD, OpenBSD and DragonFly BSD) on the same notebook. All the BSD systems offered very good performance in database tests (even better than Linux), very good battery life, and a very good user interface thanks to Xfce. However, the lack of automatic wireless network management eliminated them from the contest.

Taking advantage of plentiful hard disk space we installed a lot of packages we considered might be used in the future, including OpenJDK, the Go language, and a full set of scripting tools such as Perl, Python, Tcl, Ruby and Lua. The last version of CouchDB was installed by default, preparing the workstation for the next application framework that would be based on CouchDB sync mechanism. The Node.js environment with a lot of other goodies (CoffeeScript, SocketIO, Express) was also installed.

Legal issues

Identification of cattle and buffaloes

In April 1997, in response to the BSE crisis, the Council of the European Union implemented a system of permanent identification of individual bovine animals enabling reliable traceability from birth to death.

The basic objectives for Community rules on the identification of bovine animals are:

- localization and tracking of animals for veterinary purposes, which is of crucial importance for the control of infectious diseases,
- traceability of beef for public health reasons, and
- management and supervision of livestock subsidies as part of the common market in beef and veal.

The system for identifying and registering individual bovine animals includes:

- ear-tags for each animal with an individual number
- maintaining a register for each holding (farm, market etc.),
- cattle passports,
- a computerized database at national level, and
- legislation:
    - Regulation (EC) 1760/2000 - Basic Regulation
    - Regulation (EC) 911/2004 - Detailed rules on ear-tags, holding registers, passports
    - Regulation (EC) 1082/2003 - Control measures
    - Regulation (EC) 494/98 - Sanctions
- double identification before six months of age - two ear-tags or one ear-tag and a tattoo, mark on the pastern or electronic identifier
- maintaining an up-to-date register on each holding;
- a movement document for each movement of groups of animals;
- a central register of all holdings or computer database at national level.

Identification of sheep and goats

Community rules on the identification of sheep and goats were reinforced following experience gained with foot-and-mouth disease (FMD) in 2001. The system, which was adopted in December 2003 and entered into force in July 2005, is based on the principle of individual traceability and includes:

- double identification before six months of age: two ear-tags or one ear-tag and an electronic identifier
- maintaining , an up-to-date register on each holding,

- a movement document for each movement of groups of animals,
- a central register of all holdings or a computer database at national level, and
- legislation:
  - Council Regulation (EC) 21/2004 - Main regulation
  - Regulation (EC) No 1505/2006 - Control measures

Change management

One of the most important aspects of a project on this scale, started in 2004 and continued since then within a continuously changing legislative framework, is the need for continuity in development and operation.

Procedural and legislative changes at both national and European levels (e.g. goat and sheep identification) have had to be implemented quickly. Changes in the Romanian political situation over the last ten years have often brought significant corresponding changes among the national authorities who ultimately manage the project.

Romania’s transition from EU candidate country to full Member State has also raised a lot of problems, but the versatility of the technical solutions that have been chosen allowed us to solve these at low cost, with good quality and short implementation times.

Effect on government services

Finance and budget

The National Animal Identification and Tracking System has made possible the efficient allocation of budgetary funds for livestock. Services provided by veterinarians, such as vaccination and serological tests, can now be planned based on the number of animals of each species. Knowing the exact number of animals in each category enables efficient tracking of public expenditure and accurate budgeting.

Rapid alert system and crisis management

Knowledge of the exact location of animals and holdings, combined with modern GIS technology, can reduce the impact of infectious diseases. The ability to obtain geographical information in a timely way has strengthened the response capacity of the public institutions involved in crisis management during disease outbreaks and reduce the resulting costs.

Public health and consumer protection

A number of diseases that affect animals are transmissible to humans, and contamination of animal food can cause serious illness in large numbers of people.

The National Animal Identification and Tracking System allows rapid identification of risk areas and supports prompt reaction by the authorities to withdraw contaminated food from the market and identify people at risk from transmissible diseases.

Cooperation with other public and private bodies

The main actors in the animal identification system are:

- farmers and farmers' associations,
- animal husbandry and food industry companies,
- veterinarians working for private companies or in independent practice,
- The Romanian Veterinarian Chamber (the professional organization for veterinarians),
- public authorities: the National Sanitary Veterinarian and Food Safety Authority, the Payment Agency for Rural Development (APIA), and the Ministry of Agriculture.

Information from the SNIIA database is delivered to APIA so that this agency can check the conditions needed for subsides under different schemes. Since 2010 more than 500 operators in APIA local centers have been using applications written in Google Web Toolkit (GWT) and accessed through Google Chrome and Firefox browsers. The application server is JBoss and all development is done in Java with Eclipse, with the back-end accessing the same PostgreSQL database.

Since 2010 more than 1,200,000 subsidy applications have been registered using this software and subsidy payments of more than €1.2 million have been made.

A special web interface has been designed for the Romanian police, who need to check animal identification codes in cases of fraud or theft.

Since 2010 private companies have been allowed to manufacture and sell ear-tags, and the system was extended to accommodate these changes. New applications have been developed, and more than 2,400 distributors are now using the web interface to enter information such as ear-tag models, numbers and destination farms into a central PostgreSQL database.

Evaluation

Future plans

Over the last four years a lot of other open-source solutions have been tested in preparation for an upgrade that would scale horizontally better than the current system.

NoSQL solutions for distributed computing, storage and search that have been tested include: Cassandra, Riak, BigCouch, ElasticSearch, Solr, Membase and then Couchbase. During tests, we submitted important feedback to developers and even patches. We have also closely watched the development of projects including HBase, Hypertable, and VoltDB to see where they might fit into our own development plan.

Talend Open Studio has also been evaluated recently for job integration, scheduling, various automation tasks, and data flow.

Conclusion

Open source has become a strong and viable alternative to proprietary software and has reached the level at which it is usable for large and important projects. Not only the Linux operating system, but a whole pack of other software – relational and NoSQL databases, application servers, compilers and utilities – are mature, reliable and suitable for public sector use. For active and maintained projects, technical support comes straight from the developers: quickly and with the best expertise. In this project, the availability of the source has been extremely important in some cases where it was necessary to understand exactly what a specific piece of software is doing, tune it for maximum performance, or adapt it to our own needs.

The freedom from proprietary license costs becomes more important as the number of computers and users increases. A significant part of the money saved by not having to pay for software should always be used to keep a highly qualified team of developers, maintenance staff and system administrators close to the center of the project.

Open source applications that have been used in the project

Package name/Software	**Use**
Linux Mandriva	OS for workstations/stage 1
Linux Mint 13 32-bit	OS for netbooks/stage 2
OpenSUSE Linux	OS servers/stage 1
Ubuntu Server 64-bit	OS servers/stage 2
ArchLinux, CentOS	OS servers/development various sub-projects
Apache	local and servers web server
Nginx, mod_jk	load balancer, Tomcat connector
PHP	local application and various management modules, call center
Firefox, Google Chrome	browsers for all web applications
PostgreSQL	local, mirror and central databases
PgAdmin	PostgreSQL database administration
Slony-I	early “master-slave” PostgreSQL replication (have moved to native async replication)
OpenJDK	Java environment
JBoss	main application server
Tomcat, Jetty	servlet containers for other web services
Tcl, Bash, Python, Ruby	scripting for various cron tasks at all levels
Rsync	main incremental sync mechanism
Subversion	version control for all sources
OpenSSH	tunneling, encryption, remote access
Google Web Toolkit (GWT)	extremely efficient HTML application framework for Payment Agency applications
Eclipse	highly productive development environment
Echo-1, Echo-2	old HTML application frameworks for early and current administrative application
Quartz	job scheduling (long reports and queries)
JasperReports	world's most popular open source reporting engine
Go	powerful compiled language from Google for middleware for Payment Agency applications
CouchDB	versatile NoSQL document database with REST interface and nice replication for Payment Agency applications and next data & sync architecture
CouchDB-Lucene	Google-like indexer for data in CouchDB for Payment Agency applications
Lyx	LaTeX editor for report templates used with Go
Tex, XeLaTeX	typesetting system for rendering PDF reports
jQuery, Coffeescript	JavaScript frameworks
Hadoop	distributed computing and data storage for scanned document library
MediaWiki	searchable knowledge database, tech docs, how-to’s, solutions
LibreOffice	office documents
Redmine	project management, bug tracking system
MySQL	database for MediaWiki and Redmine
Memcached, Redis	memory cache for various applications