Overview

# What is Heraldry?

Heraldry is a network application that accepts text input from users. These messages are stored in a database and distributed in real time to other users.

Heraldry was developed as a tool for an IT operations department. Changes needed to be logged for auditing purposes and the other staff must be informed of those changes. Heraldry combines these two requirements.

# Communication Workflow

A user creates a message on a client and sends it to the central server (1). The server stores the message in a database to support searching these messages.

The message is then published to other users (2). Some users, for example the helpdesk, can only receive messages (2a). Users can also choose to receive only messages that belong to certain categories, for example only messages that are marked as important, messages that are related to the mail service, or that are related to a single server machine.

The latter filtering is also used on servers. These servers receive messages that are related to them and they add the messages to an audit log, for example a simple text file, or the Windows event log.

# How is Heraldry build?

Heraldry uses asynchronous messaging for exchanging the methods. The messages are transported via Microsoft Message Queuing (MSMQ). This allows a client to go off line and get the messages when it is back on line.

Storing messages in a database and distributing them among the clients is the responsibility of a server process. All clients send their messages to the server. The messages are stored in a database by the server and are then distributed to the other clients using a Publish/Subscribe pattern.

# Which technology is used?

Heraldry uses three main technologies:

1. Messaging is implemented with NServiceBus, a configurable service bus for .Net that builds upon MSMQ and exposes a simple yet powerful API over it.
2. Database access is done with Castle ActiveRecord which is an ORM framework built upon the well-known NHibernate ORM.
3. GUIs are implemented in WPF, which allows rich, customizable graphical interfaces.