Security Reflections

We have chosen to make heavy use of a pure DBMS setup with regards to servers in the system. In the early phases of development we were discussing whether we should construct a custom C# server as a front-end to our databases. While it is an interesting problem to work on, we had a hard time finding any arguments against using a DBMS directly. At the end of the day we are working on a classical database management problem and such over-engineering adds a great number of potential points of failure.

However it turns out that one can keep things too simple. Towards the end of development we obtained domain information that made some of our assumptions look over-optimistic:

1. Preparing 'VoterBoxes' (DB servers) centrally in a closed network environment and then distributing them physically to polling stations later on is not likely to be approved in reality. Apparently the system is expected to run on all kinds of reused hardware, depending on the individual municipality/polling station. Furthermore; the hardware can't be collected and configured at a central location.
2. To make a local network at a polling station physically impregnable is too ambitious. There will always be cases where volunteers completely disregard security protocols regarding placement and surveillance of hardware.

We think that these are somewhat poor requirements to a software system of this importance. They are not at all impossible to fulfill but they signal a maybe too careless approach to the serious security problem an election poses.

The first issue about distribution and configuration of voter databases is matter of configuring our voter box manager to output in a different fashion than local network communication. One way would be to output the data as an encrypted SQL file which is then to be distributed via a 'USB flash drive' together with the technician assigned to setup the database server at the given municipality. Doing this without putting a lot in the hands of the technician is tricky though. We would much prefer that the servers left a central location fully setup and without the keys to truly modify them. To be more specific, it should only be possible to modify the 'voted' column in 'voter' and even this requires a key which is given to a single individual at the polling station.

The second issue could be mended with extra cautious local network communication. Security sensitive database transaction is not a new problem and DBMS developers have spend decades incorporating solutions into their systems. We have used MySQL this far which can be set up with private and public keys for SSL communication. It is a convenient DBMS to use because ITU can provide this. But in reality we would probably deploying some commercial DBMS with further security measures enabled. It is only a matter of swapping a connector in the DAO implementation to support a different DBMS.