

**Carter-Baker Commission**  
**June 30**  
**Paula Hawthorn**

My name is Paula Hawthorn. I received my PhD in EECS from UC Berkeley many years ago, and have since then for the most part been running the software development groups of database startups. I am now retired, volunteering my time to various organizations, very occasionally teaching a database course at UC Berkeley or consulting.

As a part of my volunteer work with an association of computer professionals, the ACM, I have been looking into the HAVA-mandated state-wide voter registration databases. Today I have been asked to talk with you about the potential for interoperability of these databases from a technical standpoint.

These databases are being implemented with mostly different vendors, with different database designs, different database system software and different hardware, according to <http://www.electionline.org/Portals/1/EB11.FINAL1.pdf>. There is great strength in having this diversity of systems so that single attacks cannot work against all of them, the same bugs do not occur in all of them, and so they can all act independently during the high-stress times of voter registration and voter verification. So this is great for reliability, security and privacy.

However, this diversity means that interoperation is a challenge. In computer science database research, this type of system is called a *loosely coupled federated heterogeneous database system*. It is loosely coupled because the system is composed of separate, independent entities. It is heterogeneous because the entities are all different.

Research on how to make a loosely coupled federated heterogeneous database system interoperate first began at least 30 years ago, and it is for the most part a solved problem. There are three parts to the solution: the control structure, the interconnecting network, and the software interconnection modules that act as the glue connecting the systems. We will discuss the details of those three parts of the solution later. But once those parts are in place, which I am going to call the mechanical parts, there is still the larger task of getting all the parts to actually do something useful.

Think of the databases as 50 different people, many speaking different languages. Getting the mechanics of interoperation done is necessary to get the people all in one room, but then we have to get them all to do the same task. This requires an agreed upon or mandated common language, and tasks.

My advice is to build this federated VR database system slowly, starting with simple tasks first. I would start with automatic re-registration.

**1) Automatic Re-registration** is a simple application if we don't worry about fraud, and if we rely on the voter to notify the state when they are moving. Then a voter could send a change-of-registration card to the state VR office when they move, and the state office

could delete the person from their roles, and, perhaps weekly, send files to the other states with the names of people to add to their lists. The states receiving “automatic additions” would perhaps require different data (some states require date of birth, some only the day and month, etc) and some data might not be easily translatable between systems. An example of a perhaps hard to translate data item is an image of a person’s signature. Some states are storing an image of the voter’s signature in the voter registration database, so that if the person votes absentee an election official can verify the signature on the envelope with the signature in the computer file. If the format used by one state is incompatible with that used by another, or if there is any missing data, the new state might need to send a note to the new voter requesting additional information or a new signature. But as long as fraud is not an issue, automatic re-registration is a straightforward application.

**Fraud Detection for Re-Registration:** But of course fraud is an issue: we can imagine some scheme where someone decides to move a group of voters from one state to another, without the voters knowing that is being done. Either all the states would need to store signature images, so that the signatures on “change of address” requests can be verified, or each voter should have a unique, national voter id number which only they knew, and which they could give as proof that they were the person they said they were.

**2) Multiple Registration or Voting Fraud Detection.** A second, more complex task for the federated VR databases is detecting when someone has registered or registered and voted in more than one state. This task requires that all the registrations in all 50 states be compared to each other.

As you probably know, the VR databases keep track of whether or not a person voted so they know, for instance, if a person has already voted absentee and is trying to vote again on election day. (This happens. People change their minds). I would recommend just catching the people who actually cast votes in more than one state, rather than those who are registered in more than one state, because the state data includes lots of people who have once been registered there, moved and have not been dropped from the rolls. So there would be many false positives. I would concentrate on only those who have actually voted in more than once in the same election.

But to do that still requires that the voter registration data in each state all be compared to each other state. The only viable way to do this is to maintain a federal database of all 300 million voters. That is not a huge database, and is easily implementable. See [http://research.microsoft.com/~gray/papers/MSR\\_TR\\_102\\_VOF\\_Technology\\_Forecast.doc](http://research.microsoft.com/~gray/papers/MSR_TR_102_VOF_Technology_Forecast.doc) for references to truly huge databases. The problem is not the size; the problem is the need for a unique identifier. If we are to learn that John Smith has registered in two states, we need to know that it is the same John Smith. Each state has its own unique identifiers, usually the driver’s license number. But all that John Smith has to do is to get a driver’s license in two states, or, in one state to claim that he does not have one, and the fact that it is the same John Smith will be undetectable. So fraud detection requires a unique, verifiable national identifier for each voter.

Many states are keeping images of the voter's signature, to verify absentee ballot signatures. It may be possible to use a combination of signature analysis software and a voter's name as a unique id, and avoid the issue of a "national voter ID".

Now, we have discussed two different applications. Let's discuss the mechanics of how to get the databases working together to accomplish those applications. There are three parts to the solution: the control structure, the interconnecting network, and the software interconnection modules that act as the glue connecting the systems.

**Control structure:** Control can be *centralized* or *distributed*. In a centralized system, there is a single control entity (say, some federal computer system) that sends commands to each of the state databases to perform certain tasks. In a distributed control structure, each state VR database system decides for itself when a task will be performed. For instance, re-registration can clearly be done in a distributed fashion, but fraud detection needs to be centralized.

**Interconnection network:** The interconnection network can be something as fast and available as the internet, or as slow as sending files on disks by postal mail. Between these two extremes are leased lines, phone lines, etc. Connecting all the VR databases on the internet will expose them to internet hackers. The type of interconnection will be dictated by the speed that is required for those tasks that the interoperating databases will be performing. So, if the interoperating databases are processing voter registration transfers, and if the requirement is that a voter should be able to vote in his new state the day after he files a change-of-registration, then very fast on-line interconnections like leased lines or the internet should be used. But, if the rule is that he should vote absentee in his old state until the new registration takes effect, and that the new registration does not take effect until the residence requirement in the new state is met, then there can easily be a 2 week or more period for the "change" transaction, and a slower method of interconnection can be used. In general, slower interconnection methods, such as mailing a disk, or sending an encrypted file of data over a leased line, are less complex and more secure than having the VR databases always up on the internet.

**Software Glue:** It will be necessary for an agency or consortium of states to define the standard format that all the VR databases will use to communicate data. Then each VR database system will need glue modules that translate their interior data syntax and semantics to the standard format. How complex the glue modules are will depend on the complexity of the data that needs to be exchanged, but in general this task is not a hard one. The only hard part is agreeing in how to uniquely identify voters.

**Conclusion:** What we have found in practice is that these loosely federated heterogeneous database systems, while not hard to implement, can be extremely slow in operation if the tasks require a lot of communication between the different systems. So the tasks need to be made very simple, and the design needs to be kept very small and clean.