6.1800 DP Updates, Spring 2023

All of the issues discussed below were present in the DP spec, but many teams did not address them (and so they are more clarifications than updates). Many of the issues relate to failures and security. It's not too surprising that those topics went un- (or under-)addressed in the DPPRs, since we cover fault tolerance and security in the second half of 6.1800. But now that we're there, these issues should be top of mind.

Design Goals and Modularity

- Many teams identified their design goals, but did not mention them throughout. As you design elements of your system, these goals should always be in mind. Which of these goals am I addressing with each particular design choice I'm making? What is the tradeoff among these goals or others if I had made an alternate choice?
- Your design choices impact users. Are there aspects of your system that will make it difficult to use? Would users *like* the system you're designing, or merely tolerate it?
- It's tempting to define the modules of your system as the municipal machines, some VMs hosted in the cloud, etc. But these entities represent the elements of the underlying facility, not necessarily the modules of *your* system. What would be the major modules in the software you would build to create your system? Those modules should be front and center, not the underlying resources.

Moving and Replicating Data

- If you're moving data to the cloud: Why? Does it need to be there? Does it improve performance? Etc. There are good reasons to store certain data in the cloud! It's important that we understand your purpose in moving or replicating data in the cloud.
- If you are replicating data: what are the tradeoffs you're making? Reliability might improve, but at the cost of what? Is that trade-off worth it?

Failures

- Suppose users aren't submitting their forms uniformly during the two months of data collection. What fraction of the population needs to be simultaneously filling out a form to cause problems, and is it plausible that such a fraction of the population would be trying to do the same thing at the same time? Can your system handle this increased load, especially if failures are also present?
    - Note also that, because Fictlandia has multiple timezones, the preferred time for users to enter their data is not uniform.
- The deadline for data collection is March 1. Are there sequences of failures in your system that might make meeting this deadline impossible? If so, can you fix them?
- The deadline for distribution of data is April 1. Can your system handle failures during that time? If you are moving data from municipal machines to elsewhere in your system, can you be sure that it arrived at its destination? Keep in mind the end-to-end argument:

even TCP connections can fail. If a TCP connection fails midstream, something must be done at a higher layer to “clean up”. TCP will have transmitted only a subset of the packets and perhaps only acknowledged fewer than that.

- Consider the (perhaps arbitrary) limitations of your system. How will your system handle names that are larger than the size of the allocated name field, or is there a clear and convincing argument that such a situation is impossible? How will you handle families that are larger than the “largest possible” family? How many genders can your system represent, and why?

Security

- What sort of threats does your system protect against (another way to ask this: what is your *threat model*)? In particular, if you are using encryption, how is encryption protecting against particular threats? Note that we’re not requiring you to protect against every conceivable threat; that would be impossible. It’s important to be clear about what your system *does* protect against.
- If you are using encryption, who has the key(s) to encrypt and decrypt? How will you distribute those keys?
- How much privacy does your system provide? For example: encrypting a bunch of individual census records and then handing them over to someone who has the decryption key does not do a lot to prevent misuse of those individual records (though it does prevent people without the encryption key from viewing them). Are there places where your system should provide aggregate statistics instead of individual records?
- If you are making design choices in order to protect the data, are those justified? What does “protected” mean in your system, and why is it important for this type of data?

Common mistakes/misunderstandings + Clarifications

- One of the requirements of the system is that the data needed by the different government organizations must be delivered to them. Unsaid, but generally true is that they should not have access to data that is *not* theirs to have.
- Each of the government organizations discussed in the spec — the municipal, state, and national census organizations as well as the School and Election Boards — run their own data management systems. Each needs some part of the census data delivered to it. Take as an example the Election Boards. Although the census data provides one opportunity for cleaning out the voter rolls of people who have moved away, the primary source of enrolling new voters has nothing to do with the census, nor is the census in any way directly involved in determining the legislative districting of the voters. So, the Election Boards significantly benefit from the census information as an input to their process, but they run their own data management systems. This is true for each of the types of organizations.
- The national government is building the cloud service for its own use in its own census system. It is depending on the municipalities for collection of census data, but otherwise it expects to use the cloud for its own census analysis and activity. (You read a bit about the national census before Spring Break. The national government in the USA does a

huge amount.) Out of generosity, the national government is allocating a number of Virtual Machines equivalent to the number of physical machines, for each municipality, supported using national cloud resources. But that does not mean that all of the remaining cloud resources are also available for managing the municipalities' responsibilities. If you believe the municipalities require more resources than they have, it is important to make that explicit and well-justified.