In this years design project, we are tasked with creating a control system for the fictional Centertown's electrical grid. While the actual infrastructure for transmitting electricity is beyond the scope of this project, we will design the computing infrastructure that directs the flow of power and data, using a pre-existing configuration of hardware.

This system poses an interesting challenge, as there are many different layers which must all communicate effectively with each other. Additionally, the critical nature of the electrical system leads to the requirement that any design be robust against unusual circumstances. Our specific system is interesting since will enable solar power to be effectively used in Centertown, which will lead residents to see a decrease in cost of electricity, as well as decreasing the whole city's carbon footprint.

In this paper, I will describe the basic layout and function of the hardware in which our system will operate, highlighting the places in which interesting design decisions will need to be made. Then, I will discuss the desired properties of our design, both at an immediate level and from a more philosophical perspective. Finally, I will present key questions that arise from the stated design constraints and priorities.

I. Key Facts and Definitions

There are three key levels of hardware which we must handle the communication between: the central utility, the microgrid controllers, and the smart meters. Centertown is divided into a set of microgrids—smaller sub-divisions of residential houses, apartment blocks, or community buildings—managed by one central utility. Each microgrid has capabilities of power generation, either through solar panels on residences or the city's solar farm.

At the lowest level, each producer and consumer of electricity is separately equipped with a smart meter (sec 2.1) which measures the power generation, usage, and storage and communicates the relevant data to the microgrid controller so that it can be used to determine what the immediate flow of electricity should be and can be stored for later analysis. The available commands on the smart meter—e.g. for getting data and aggregating records—are fixed (sec. 2.2.1), but one major element that we must design is the way in which the microgrid controller and smart meters use this protocol to communicate.

One level above the smart meters, the microgrid controller transmits messages between itself and both the central utility and the smart meters using LTE cell service. It also facilities the sharing of excess power between members of the microgrid: if one member's battery is above their set minimum, and another's is below it, power will be shared with no associated cost or credit (sec. 2.1).

Lastly, the central utility is responsible for high-level operation of the system. It obtains data about the operation of each microgrid from the microgrid controllers, through a communication protocol which we must design. Designing this protocol effectively is vital, as there is a large amount of freedom in design, and it will thus impact the character of the final system. Using the data from the microgrid controllers, the central utility must buy power from New England Regional Utility when needed, handle the city-wide sharing of excess power, make long-term decisions, and produce monthly bills (sec. 2.2.3). It also has a larger amount of storage (2TB) than the other pieces of the system, which all have 64 GB (sec 2.2.1-2.2.3), making it vital to the goal of storing long-term usage data.

II. Desired Properties

The base, and arguably most important, property the system should have is the ability to handle all the necessary processes in a typical use case. This means being able to coordinate sharing within the microgrid, city-wide sharing for credit, and optimally purchasing power from the outside provider, all while collecting data and handling billing.

Our system must also be resilient to unexpected outages and changes in patterns of power usage. It should be able to able to adapt to isolated or long-term changes in power demand, and in the extraordinary case that a link in the data or electrical network is severed, the system should still function—so, for example, the microgrids should be able to function self-sufficiently.

While handling these systems, there are a few important trade-offs we will likely run into. Our system is limited in storage, processor speed, and the amount of data we can transmit over the LTE network in a month (sec. 2.2.4), so our design should take into account how to conserve and best utilize these resources. For example, the data we collect must be stored in a way that balances the interests of Centertown, which desires long-term data, and MIT researchers, who desire as fine-grain data as possible.

These trade-offs and desired use cases point to an additional level of considerations our system should address: the philosophical questions such an impactful system must consider. An import consideration is how electricity allocation should be prioritized in the case that there is a shortage. Similarly, we must decide how conservative overall our system should be, answering questions like how much extra electricity we should store, how much LTE data we should conserve for emergencies, and whether we should build any redundancies into our messaging systems.

III.Key Questions

Based on the described design constraints, I pose the following questions:

- How can we ensure that the un-credited sharing within the microgrid is fair, when a malicious actor may attempt to use it to get electricity at no cost?
- If the batteries in the central utility are full, is it able to sell excess electricity from solar panels to outside sources, or is the power wasted?
- Keeping fine-grained data on power usage could reveal personal information. What are the precise privacy concerns here, and how do we mitigate them? Are the channels through which we are transmitting this data secure?