# Hardware, Apps, and Surveys at Scale: Insights from Measuring Grid Reliability in Accra, Ghana

**Noah Klugman**[†], Joshua Adkins[†], Susanna Berkouwer[†], Kwame Abrokwah, Ivan Bobashev[†], Pat Pannuto[‡], Matthew Podolsky[†], Aldo Suseno[†], Revati Thatte[†], Catherine Wolfram[†], Jay Taneja[*], and Prabal Dutta[†]

ACM COMPASS, July 4th, 2019
Accra, Ghana

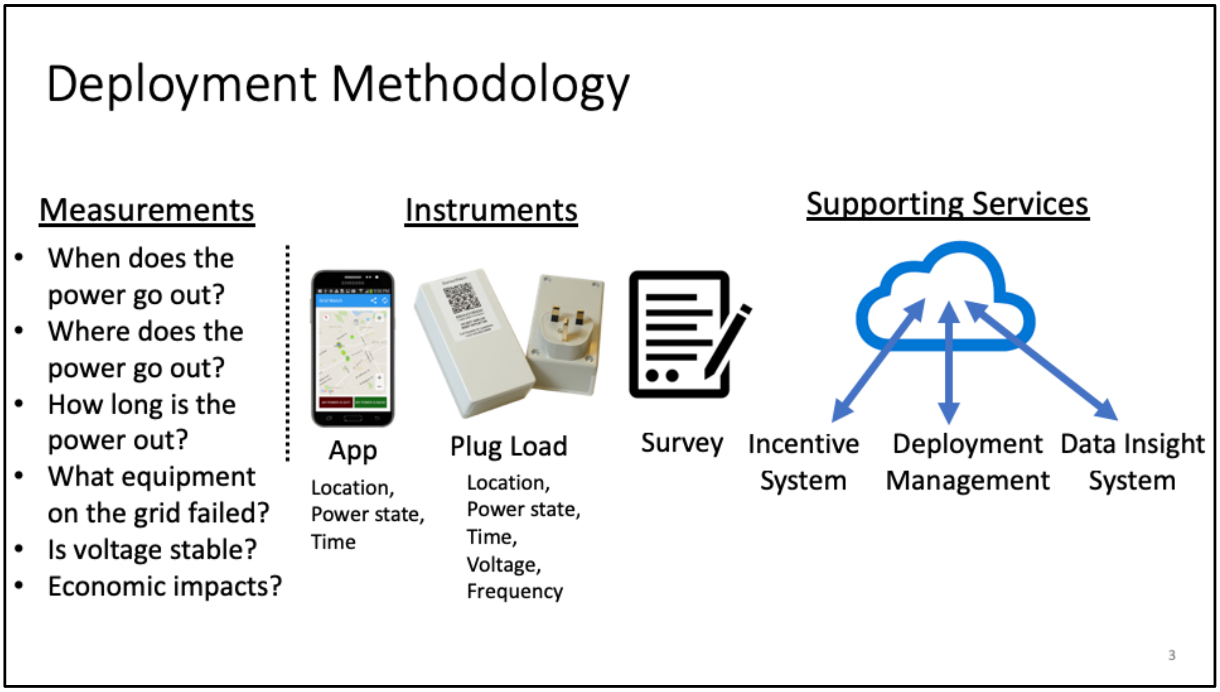
† * ‡

# Measuring electricity reliability in Accra

- Ghana Power Compact $498 million investment from US Federal Government
- Compact aims include reducing power outages, and stabilizing voltage
- Dumsor occurred from 2013 until 2015 due to under generation
- Current power situation could improve




So this is my 9th trip from California to Accra in 20 months… I had to look that up… and I'm really happy to be back to talk about our work measuring the electricity grid here. Along with most of our team, who are made up engineers and economists, I came to Ghana for the first time because of the Ghana Power Compact, which is a large investment (*) aimed at improving the reliability of the electricity grid throughout Ghana (*). On the right is a billboard in Accra sponsored by the compact. Now, from 2013 until 2015 Ghana suffered frequent, scheduled power outages that were such a part of life that they received a name, dumsor, meaning off and on. (*) Today, even after the dumsor period has ended due to more electricity supply coming on line, -  which is great - , Ghana experiences frequent power outages and voltage quality issues. (*) Our team was developing sensors to measure power outages and voltage quality, and we were excited for the opportunity to use these sensors to help those implementing the Ghana Power Compact empirically evaluate the reliability of the grid.




After meeting with the Ghana Power Compact team, we set a goal to (*) measure when (*) and where (*) the power goes out and for how long (*) it was out, where on the grid the power failed (*), and (*) any issues with voltage quality. We also decided to study (*) households and businesses in Accra, to learn how unreliable energy impacts them socioeconomically. To do all this (*) We designed two different sensors; one is an app (*) that is installed on personal use phones and (*) that attempts to detect power outages using on-phone sensors, and (*) one is a custom plug load sensor that plugs into the outlet at a household or firm and (*) reports information on the power over the cellular network. Together these sensors can tell us how well a transformer, feeder, or substation is performing. We also build survey instruments (*) to explore the socioeconomic questions. (*) Along with the sensors and surveys, we designed supporting systems for the deployment, which live in the (*) cloud. These included a (*) system to transfer incentives to participants for their time, we send them airtime to their phone (*) a deployment management system to keep track of sensors and participants, and a (*) data visualization and analysis system.

# Outline

- **Introduction**
- Small scale deployment
- Medium scale deployment
- Large scale deployment
- Conclusions



Right. So... With all that planned, we are ready to deploy our power monitoring system right?

# Outline

- Introduction
- Small scale deployment
- Medium scale deployment
- Large scale deployment
- Conclusions



Well, Let’s see.

# Small-scale: The first deployment

- Deployment goal: Do the sensors work?
  - GPS fix
  - Operated at 240v/50hz
  - Connects to cellular network

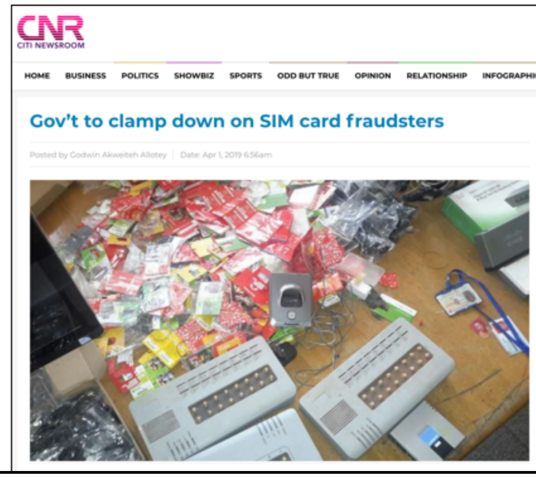
12 Plug-load sensors used in small scale deployment



Our first step was conducting a small scale deployment, with the initial goal to learn (*), if our new plug load sensors work (*) The deployment consisted of 12 plug load sensors (*), this is what they looked like, they are around 2 inches long and were placed in a enclosure and plugged into the wall. Specifically we wanted to learn whether theses sensor got GPS fixes (*), were able to handle the local grid voltage and frequency (*), and could communicate over the cellular network (*).

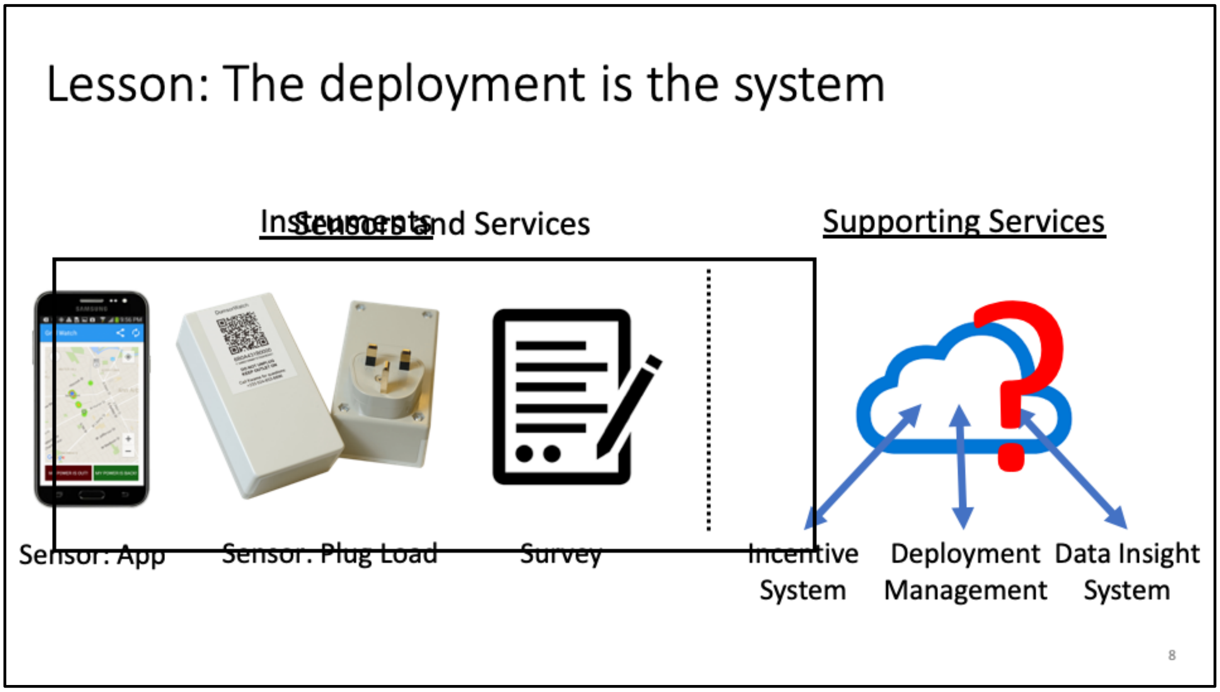
# Going beyond consumer needs

- Limit of 3 SIM cards due to security concerns




Now at this scale we only encountered one unanticipated challenge, which was when we hit a limit of SIM cards purchasable by one person(*). There was a 3-sim-card per person limit, set to address security concerns (*) which have been widely reported locally. Now, this might not be a problem for most consumers, but our sensors needed more. We got around this by going to multiple vendors, but circumventing customs or laws isn't really a scalable solution, although in this case it allowed us to satisfy our deployment goals by validating the sensors.




Now, in retrospect, our small-scale deployment focused only on validating the part of the deployment methodology within our area of expertise, (*) sensor and backend design. We assumed the rest of the deployment methodology was simple and would work itself out (*), when in reality, as we will see, the most significant issues with our next scale were instead related not to the core functionality of the sensors and backend, but rather from poorly designed information flow between subsystems and incorrect assumptions about the local context.

# Outline

- Introduction
- Small scale deployment
- Medium scale deployment
- Large scale deployment
- Conclusions

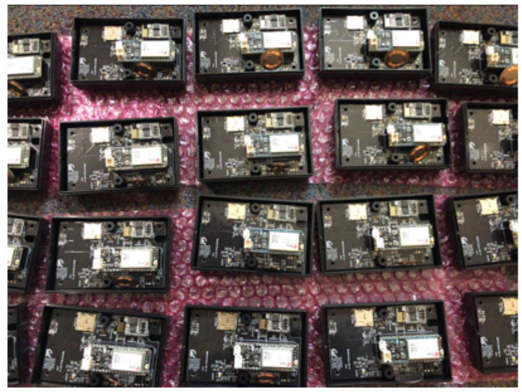


OK,


Let’s scale up and I'll tell you more about what worked and didn't

# Medium-scale: The second deployment

- Goal: Measurements power quality in a single district of Accra
- 2,000 app downloads and short surveys
- 165 plug-load installations and long survey
- Fully implemented deployment methodology



20 of the 165 plug-load sensors deployed at medium scale



Our medium-scale deployment had the (*) goal of taking meaningful measurements of power quality in a the Achimota district in Accra. We (*) recruited 2,000 individuals to download the app and take a short survey (*)  and 165 individuals to install the plug-load sensor and take a longer survey. (*) Here you can see 20 of the soon to be deployed plug-load sensors being assembled in our lab. (*) This scale required implementing our full deployment design, including hiring local support staff, recruiting and incentivizing participants, and implementing all of the technology.

## Local insight is critical for participant recruitment

- Talking with local people helps understand how to legitimize our work in Accra

ACCRA METROPOLITAN ASSEMBLY

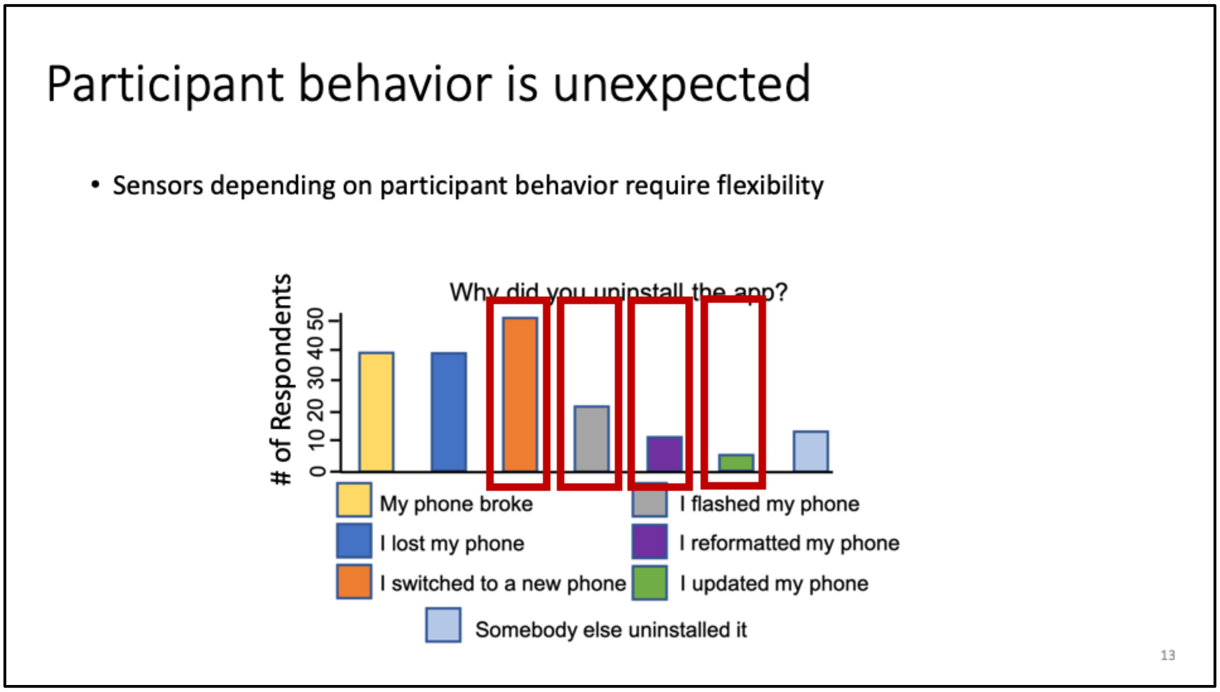
Assembly permission was important

Kelvin, a team lead, and the other field officers in the red, Dumsorwatch uniforms



Our team hired a local project manager, named Kwame, to help implement our deployment at this scale. As we prepared to deploy, Kwame himself hired a team of local field officers, who would recruit participants and install the sensors. This was all encouraged by the economists on our team, and was met with indifference by the engineers, who incorrectly viewed Kwames role as administrative. I'm a bit embarressed to say that our meeting with the field officers right before our deployment ended up being the one of the first times we heard significant local feedback on our deployment methodology, which we quickly realized was invaluable. (*) If you'll allow me I want to share three quick examples of this feedback, each which was critical for building trust with our participants. (*) Firstly, Kwame identified that permission from the government at the districts we were deploying would bootstrap trust with participants, as it did. We would have never considered seeking this approval without his input, Second, (*) we had bought uniforms for the field officers in the UC Berkeley school colors, blue and gold, which we found later to be the same colors of the Electricity Company of Ghana. The field officers explained that participants may not trust the Electricity Company of Ghana and strongly suggested we differentiate ourselves from them, so we begrudgedly bought new uniforms, only to later learn that this differentiation made participant enrollment much easier. Thirdly, at the urging of the field team, we changed the name of our project to

Dumsorwatch – again Dumsor being the local term for the electricity going off and on. This name has been a hit with participants and quickly contextualizes our project to them. In retrospect, seeking local feedback earlier would have allowed us to design our deployment with cultural considerations as a first order requirement, potentially further increasing participation take-up rates and reducing the cost of long and drawn-out deployment activities.




In a similar vein, it became clear that we did not properly anticipate participant interactions with our sensors, which led to some problems. (*) For example, (*) this is a graph taken from around 170 participants who we contacted to ask why they uninstalled our app. As you can see, people regularly switched (*) phones and restored phones (*) to factory settings. Now these actions were not anticipated and therefore not handled by our deployment system, and as a consequence lead to multiple compliance issues across different subsystems of our deployment. and this is only one of many unanticipated behaviors we observed. Moving forward, we would run a short survey over a large population to try to understand the true diversity of potential participant behavior, before implementing our system. This is fairly low cost to do, and would have led to our deploying more functional sensors with less effort.

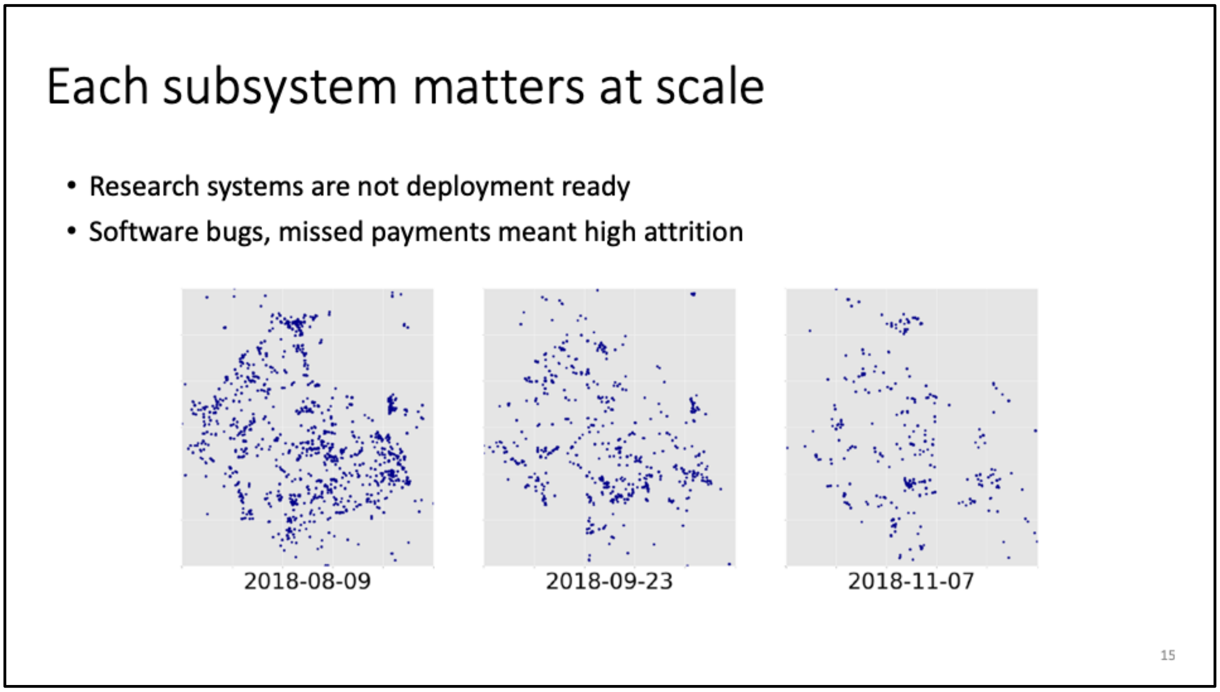
## Scale doesn’t bootstrap trust

- Local resources were made available slowly
- Acquiring resources required in-country presence

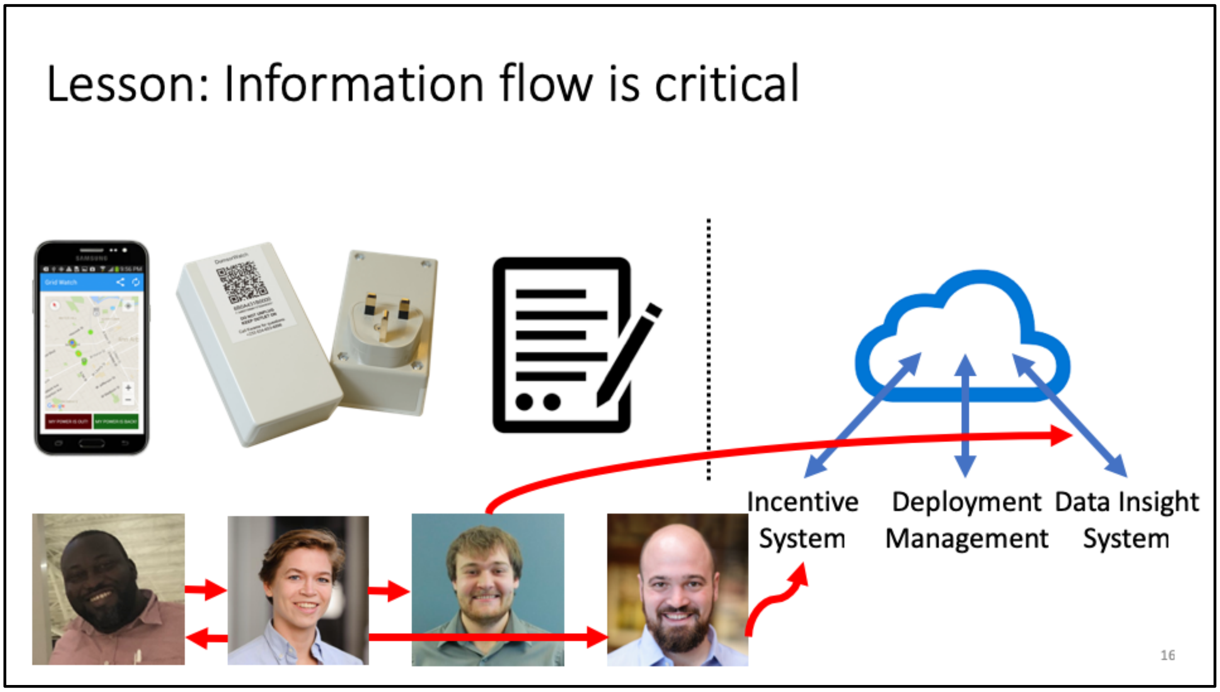
400 MTN prepaid SIM cards acquired after months of effort

14

Our earlier issues acquiring SIM cards foreshadow that local services could be difficult to purchase, but we brushed this off assuming that at scale we would be able to approach service providers at an enterprise level, and benefit from different procurement channels. (*) In practice, procuring SIM cards (*), along with the other major Ghanaian service, the API to transfer airtime incentives to our participants, first required establishing trust with the service provider that our project was legitimate. Attending meetings and drafting documentation for these service providers (*) created an unexpected dependency on our team being in Ghana, leading to months long delays in accessing critical services. Based on this, we would suggest starting procuring local equipment needed for scale well before any actual deployment.




Finally, much of the early effort from the engineering team was focused on getting sensors out the door, and other supporting systems got sidelined under the assumption that they could be improved later. One particularly painful example of this was the supporting system to send incentives to participants, which was still immature when it came online, and experienced major bugs, leading (*) to participants receiving late incentives and subsequently to a higher than expected level of participant unenrollment. (*) This is visualized here, where we can see spatial data from app users degrading over a couple of months as they uninstall the app. Going back, we recognize (*) we would have saved considerable cost, time, and goodwill to accept deployment delays to ensure that every subsystem in the deployment methodology was production ready at the point of deployment, rather than developing on the fly.




Now, thinking back on our medium scale deployment, we recognize that it (*) depended on team members to transfer information between subsystems. For example, if a participant withdraws, Kwame (*) would call Susanna (*) who would notify Josh (*) to update the plug-load sensor registrations (*). She would also notify Pat (*) to update the incentive payment system. Then she would notify (*) Kwame that all is good. Now these people are great, but If any of these people drop the ball, there will be a bad effect. This ad-hoc information flow sort of worked in practice at this scale, because the deployment was still fairly small and information transfer wasn't all that frequent, but in retrospect, it was bound to fail for two reasons. The first is that the team had to notice state changes, which is difficult to sustain when life gets busy or when a deployment has been running for many months. The second is again that in order to correctly propagate state and keep all subsystems in sync, no single team member could ever mess up, and if they did, it wasn't immediately obvious. This gets to the key lesson from our medium scale deployment, which is that a deployment of this size has enough information moving through it to require clearly defined interfaces between subsystems, and to require that subsystems automatically propagate and syncronize any state changes, removing humans from the loop when possible. Anything else won't scale.

# Outline

- Introduction
- Small scale deployment
- Medium scale deployment
- Large scale deployment
- Conclusions



Alright, with that in mind

# Outline

- Introduction
- Small scale deployment
- Medium scale deployment
- **Large scale deployment**
- Conclusions



lets scale up one last time

# Large-scale: The third deployment

- Goal: Measurements power quality in two more districts of Accra
- 1,400 app downloads and short surveys (total 3,400)
- 292 plug-load installations and long survey (total 457)

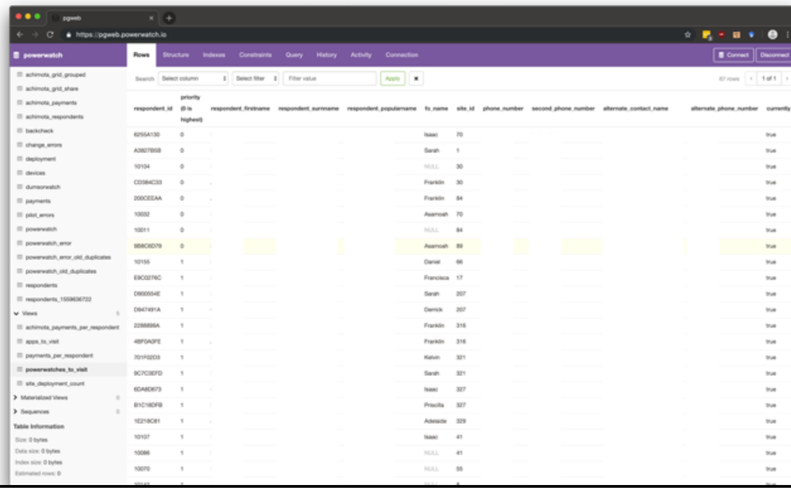
A shelf of assembled plug-load sensors



Our large-scale deployment has the goal (*) of expanding our measurement area to a total of three districts in Accra, by adding Kaneshie and Dansomon. This led to 1,400 more individuals downloading the DumsorWatch app (*) and 292 more individuals installing the plug load sensor (*). This larger total scale brings new problems, the most notable being what I just introduced about information flow becoming critical.

# Automating deployment management

- At scale, lots of state is always in flux
- Input errors are more common and have a big impact
- A single, safe interface for all non-expert users





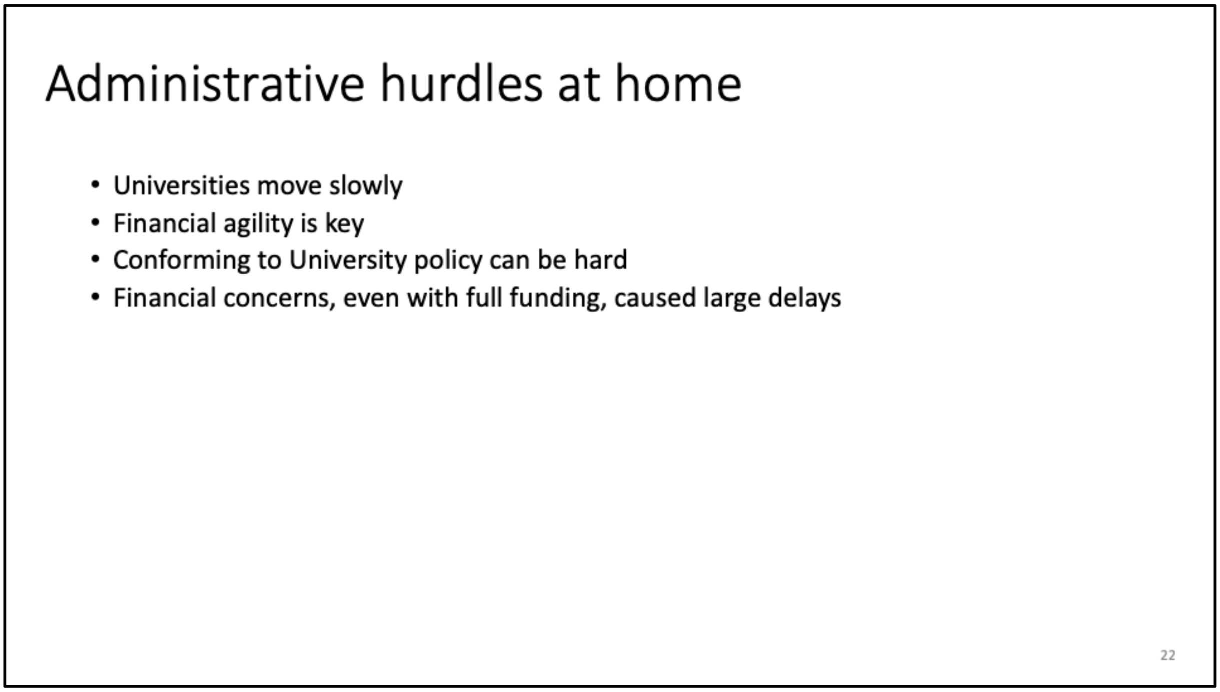
and this is because, at this large scale, a bit due to the law of large numbers, exceptional cases happen frequently (*). For example, in our medium scale deployment, only a small handful of participants elected to opt out after starting, allowing for human in the loop removal of individuals from subsystems. However, at our larger scale this previously rare event happens regularly, and clearly had to be automated. (*) Further, the consequences of input errors at this scale were more significant as they were less likey to be fixed due to their frequency. For example a field officer inputting a wrong participant phone number, would lead to incentives being sent to this wrong number, which would lead to the sensor being unplugged by an angry participant who never got paid, which would lead to our wanting to contact the participant, which we couldn't because the phone number was wrong. If this happens for 50 people, its much harder to go back to the field and correct. (*) So with the goals of removing humans from the loop, keeping state synchronized across deployment subsystems, and providing guards against input error, we built a deployment management system to be single front end for all field officers and non-expert user researchers to interact with.

# Deployment management in practice

Using the Deployment Management System in the field

21

Here we can see our deployment management system being directly accessed by Kwame and the field officers to track state in real-time. We gave Kwame and the field officer team leads laptops to access our deployment management system in the field, and in all this system has dramatically increased the ease of running our deployment, and has been critical for operating at the current scale.




Stepping back, deploying at this scale stressed the limits of the university financial systems, which we did not anticipate. UC Berkeley experienced very long delays (*) sending wire transfers, at times leading to capital shortfalls for critical services in the deployment. One way around this to request invoices earlier,(*) but costs oftentimes couldn't be predicted. Further, the invoicing method required by the University was difficult to convince local companies to follow as it doesn’t seem to be standard practice in Accra (*). The impact of these problems was that the project (*) team learned to anticipate costs well before they were due, at times leaned on discretionary funding with less restrictions, and took on personal debt when needed, lest services be shut off that are critical to the deployment. Moving forward, we suggest working closely with the University on how to meet the non-standard financial requirements of a deployment early on, with the hope that special mechanisms might be developed to make this less painful, although we are not convinced this is possible.

# Laboratory as a factory

- Assembly too small for traditional factory
- 10 undergraduates, 98% yield



Finally, I want to point out the number of plug-load sensors we needed for the large scale was too small to get professionally assembled (*) and a bit too big to build with a small team of graduate students. (*) Here is a picture of our solution, where we built an assembly line including different stations and quality control checks and hired a team of 10 undergraduates to work it. (*) We assembled around 300 sensors with a yield of around 98%, comparable to lower end box build services.

# Outline

- Introduction
- Small scale deployment
- Medium scale deployment
- **Large scale deployment**
- Conclusions

24

OK, so what are some take aways

# Outline

- Introduction
- Small scale deployment
- Medium scale deployment
- Large scale deployment
- Conclusions



Well

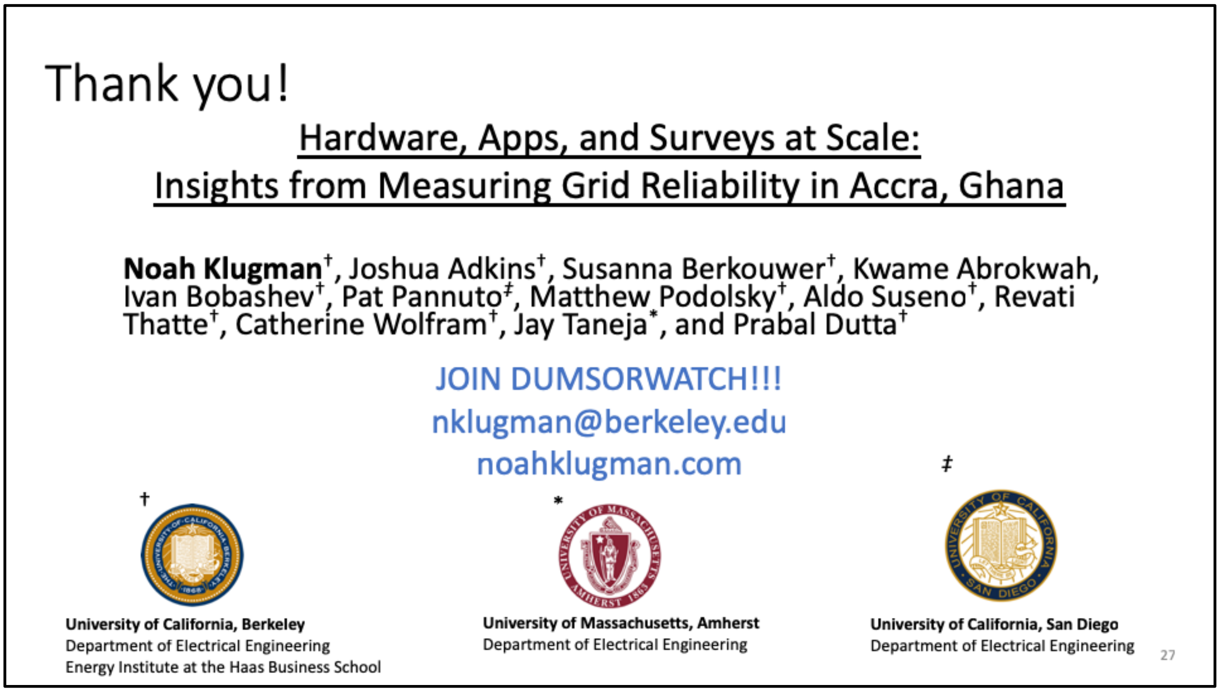
# Conclusions

- Scale requires automating interfaces and information flow
- Test local assumptions early
- Deploy

| Scale | Districts | Plug-load Sensors | DumsorWatch App | Deployment Date | Number of FOs |
|---|---|---|---|---|---|
| Small | 0 | 12 | 5 | May 2018 | 0 |
| Medium | 1 | 165 | 2000 | Aug 2018 | 11 |
| Large | 2 | 293 | 1400 | Feb 2019 | 14 |
| Total | 3 | 457 | 3400 | | |

26

Our key take-aways, which we would suggest for any deployment that contains aspects of continuous sensing interaction, are to (*) plan how information about the deployment passes through each subsystem, where dependencies lie, and how this should be optimized. Iterate on this design as problems arise rather than manually patching the issue, which won't scale, (*) test the whole system at every stage, especially assumptions about the local contexts. Talk with local people, meet with local service providers, anything you can do to understand all parts of working in a new area. Finally, deploy. (*) This (*) this experience has taken our work so much farther than we imagined it would go, exposing new monitoring applications, an area of systems research related to solving deployment problems, and most importantly a novel data set from Accra that is currently being used by multiple impact evaluations and directly by the utility company to improve improve energy reliability in city.




Thank you so much. I'll say quickly that we are continuing to scale in Accra over the next two years, and would love to get people in this room involved. Come talk with me. And with that, I’m happy to answer your questions.