

Taming the Internet of Things

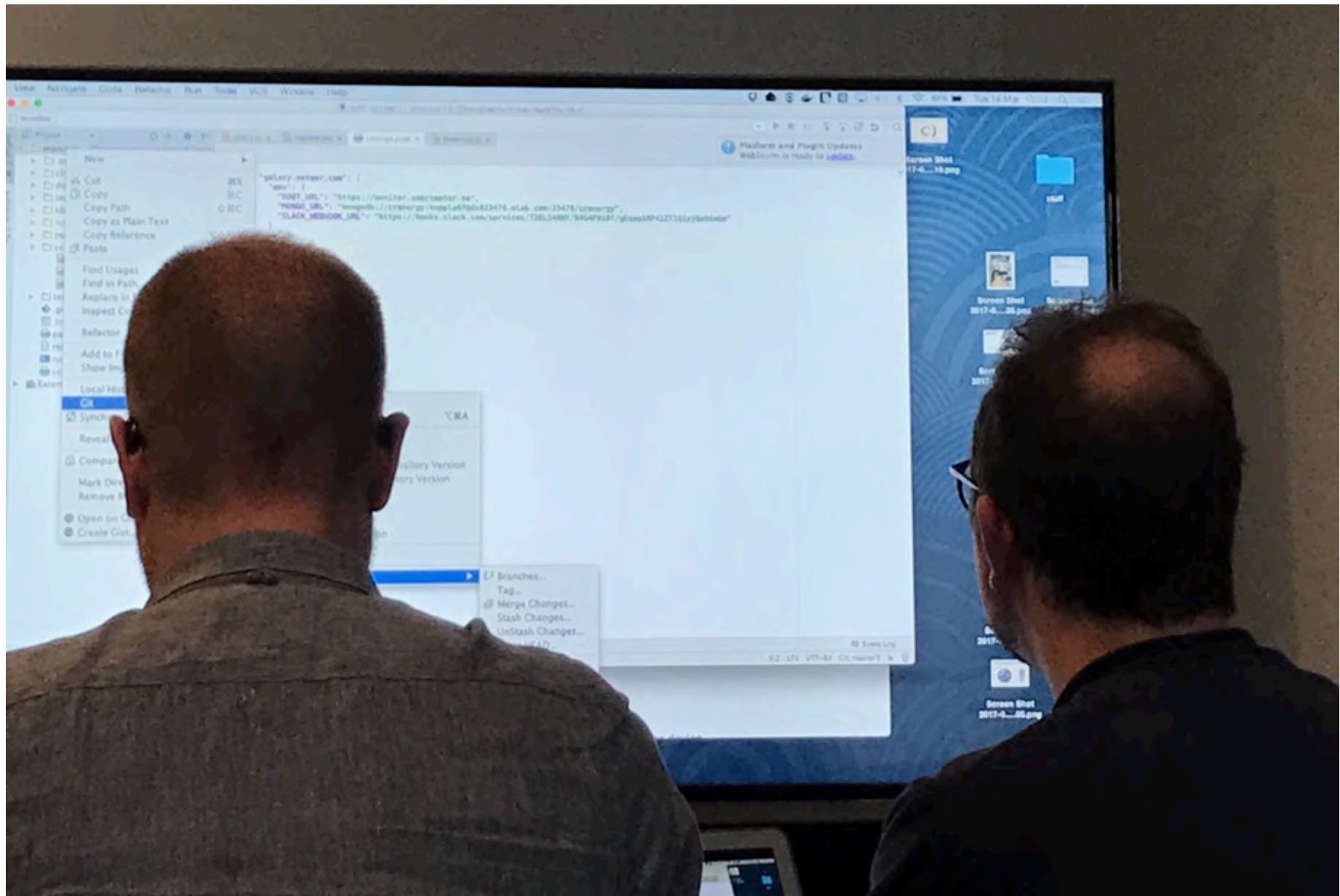
JFokus 2018 Stockholm



Henrik Kniberg

Hans Brattberg

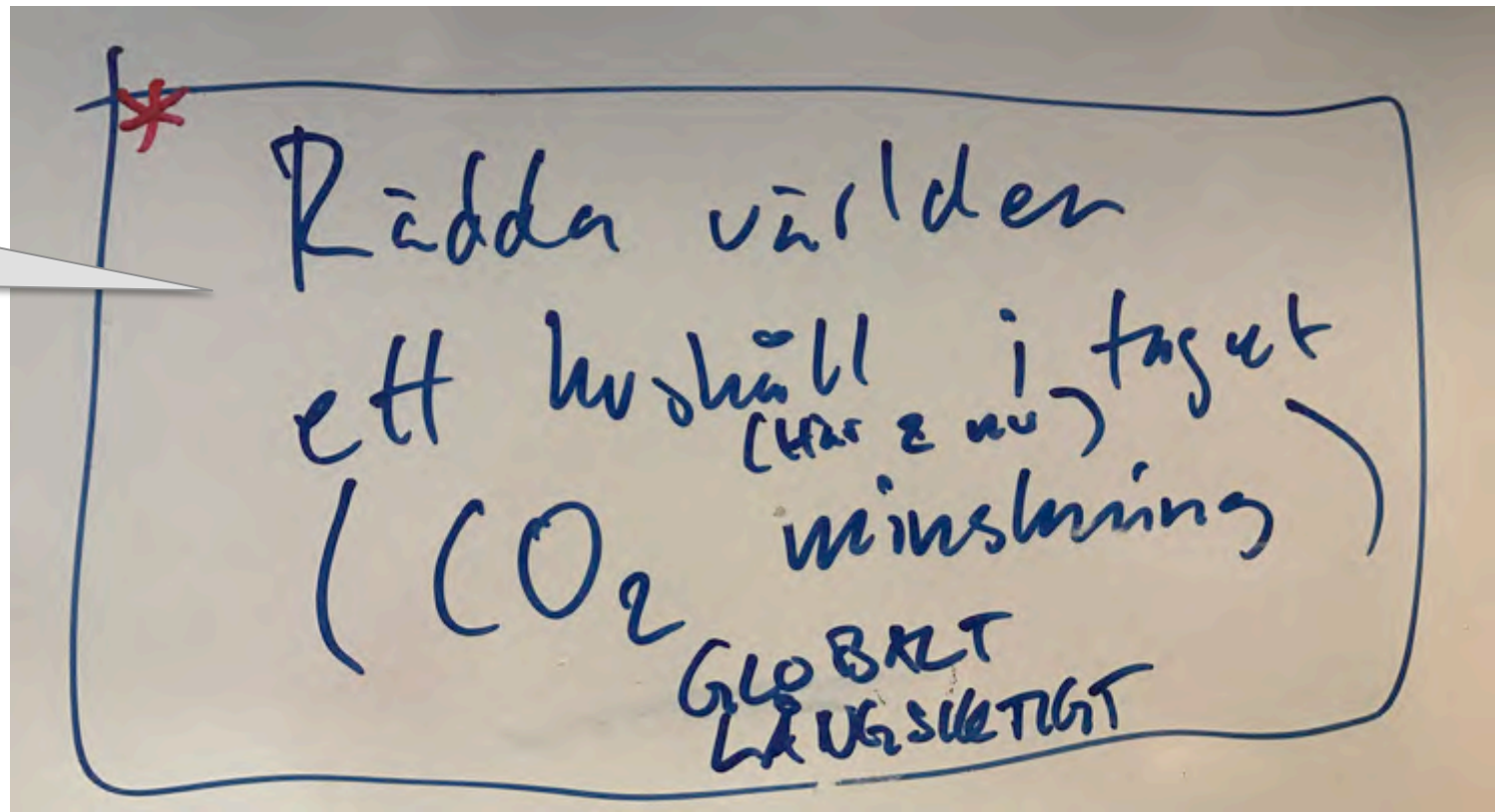




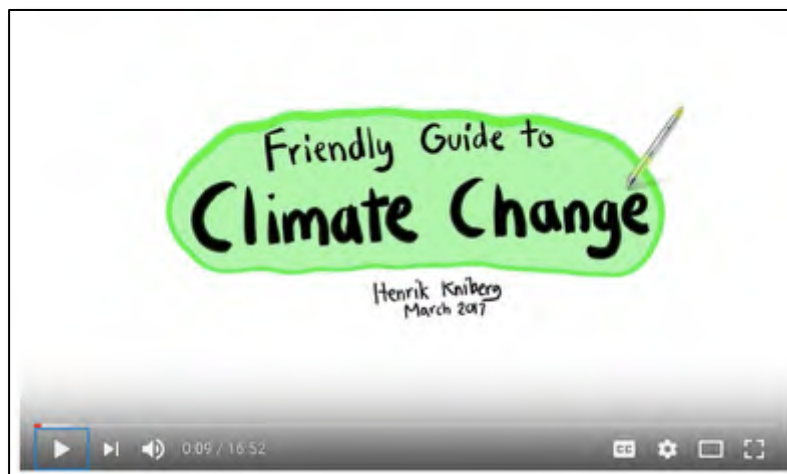
Henrik Kniberg & Hans Brattberg

What we do

Saving the world,
one household at a
time



Youtube video:

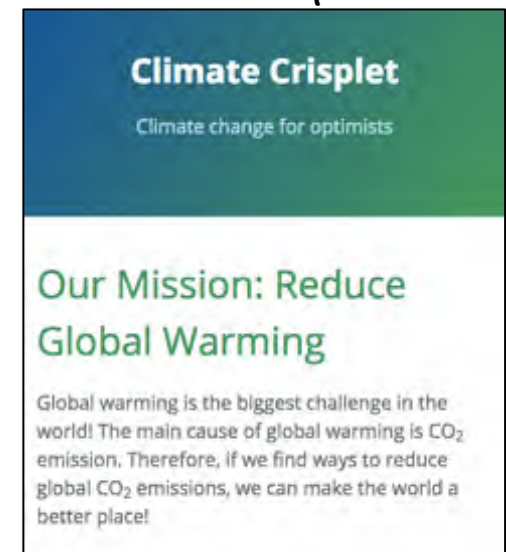


Henrik Kniberg & Hans Brattberg

GoClimateNeutral.org



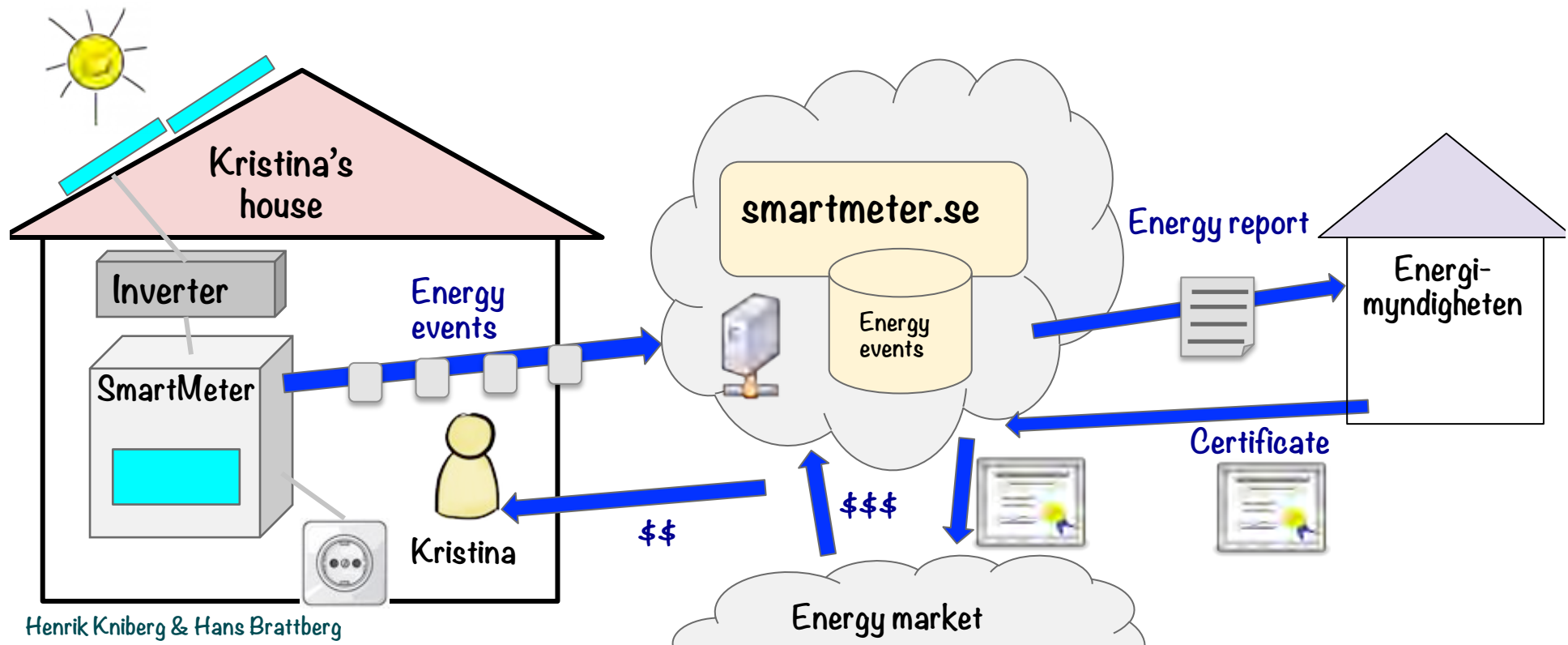
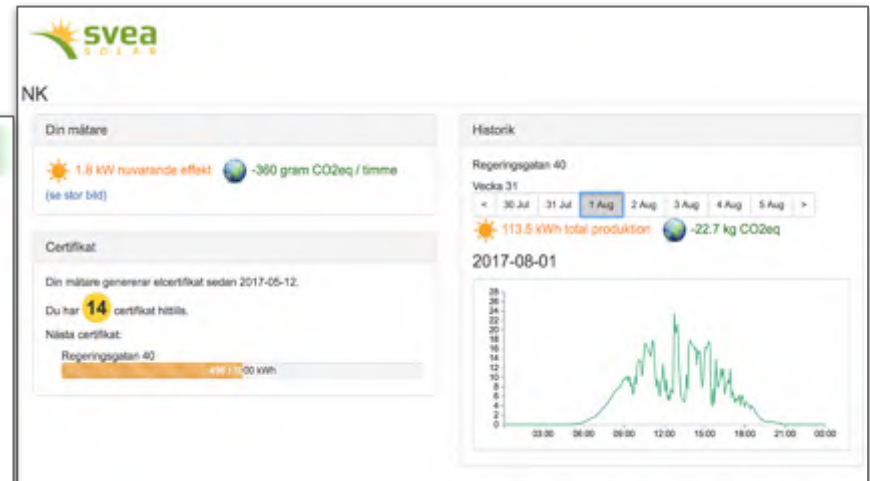
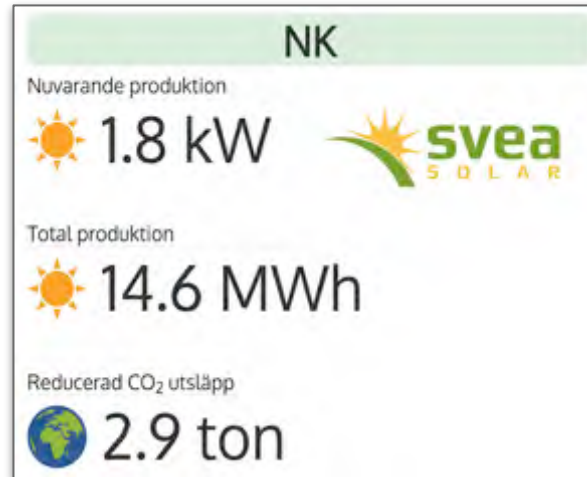
climate.crisp.se



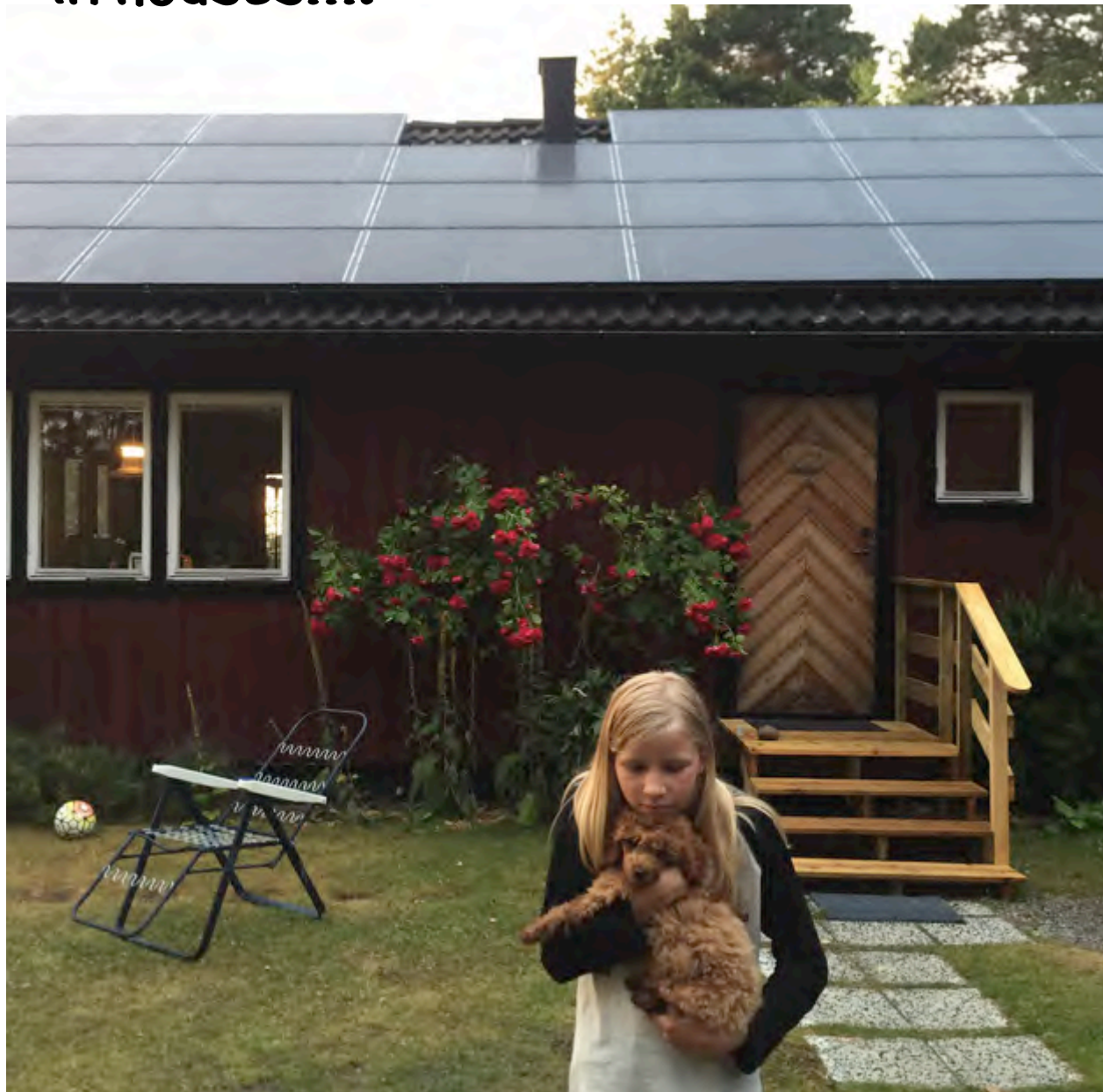
Show 'n tell



How it all fits together



In houses....



Henrik Kniberg & Hans Brattberg

On top of department stores...



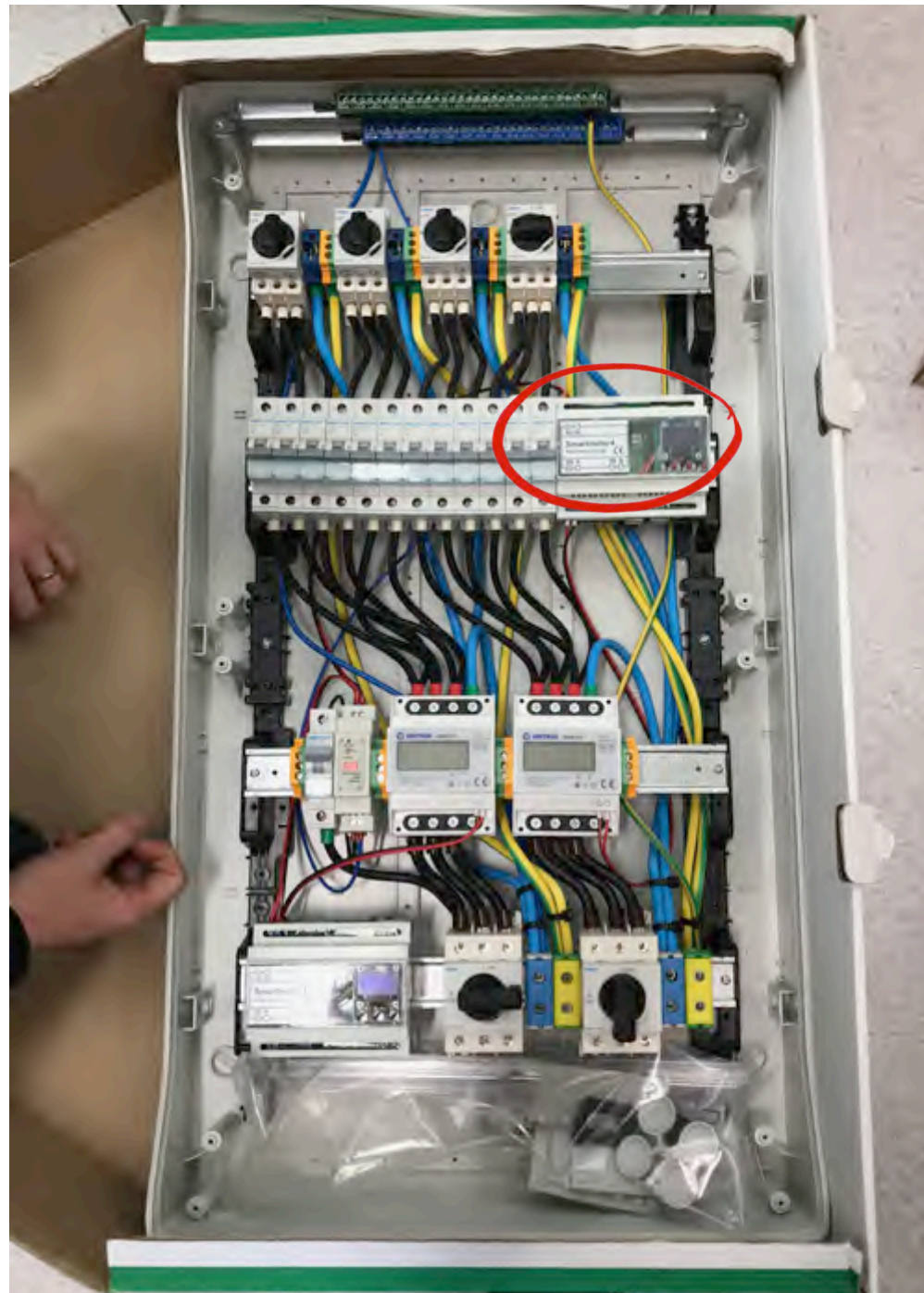
On fields....



Henrik Kniberg & Hans Brattberg



Big Things and Small Things

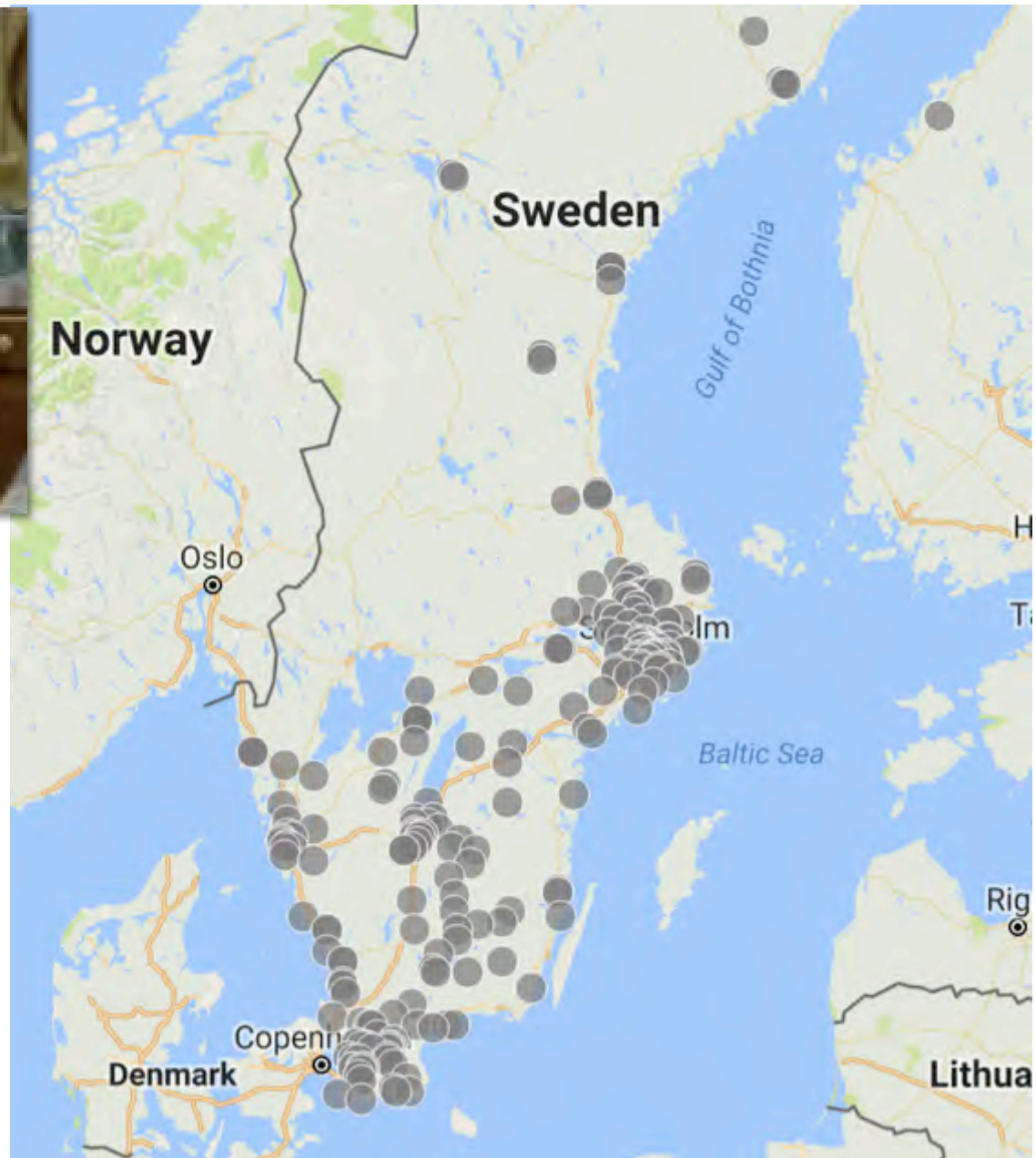


Things all over the place

er... how do we
update the
software?



Continuous Delivery on IoT !



Field demo



Field demo

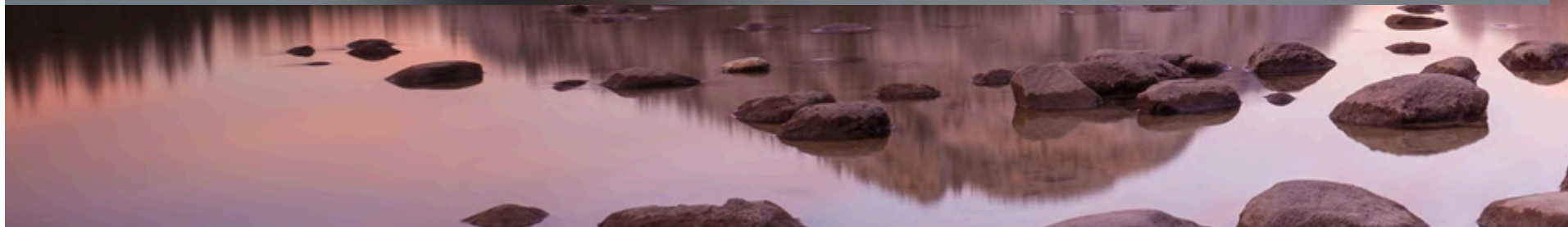


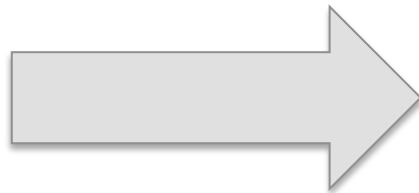
Henrik Kniberg & Hans Brattberg



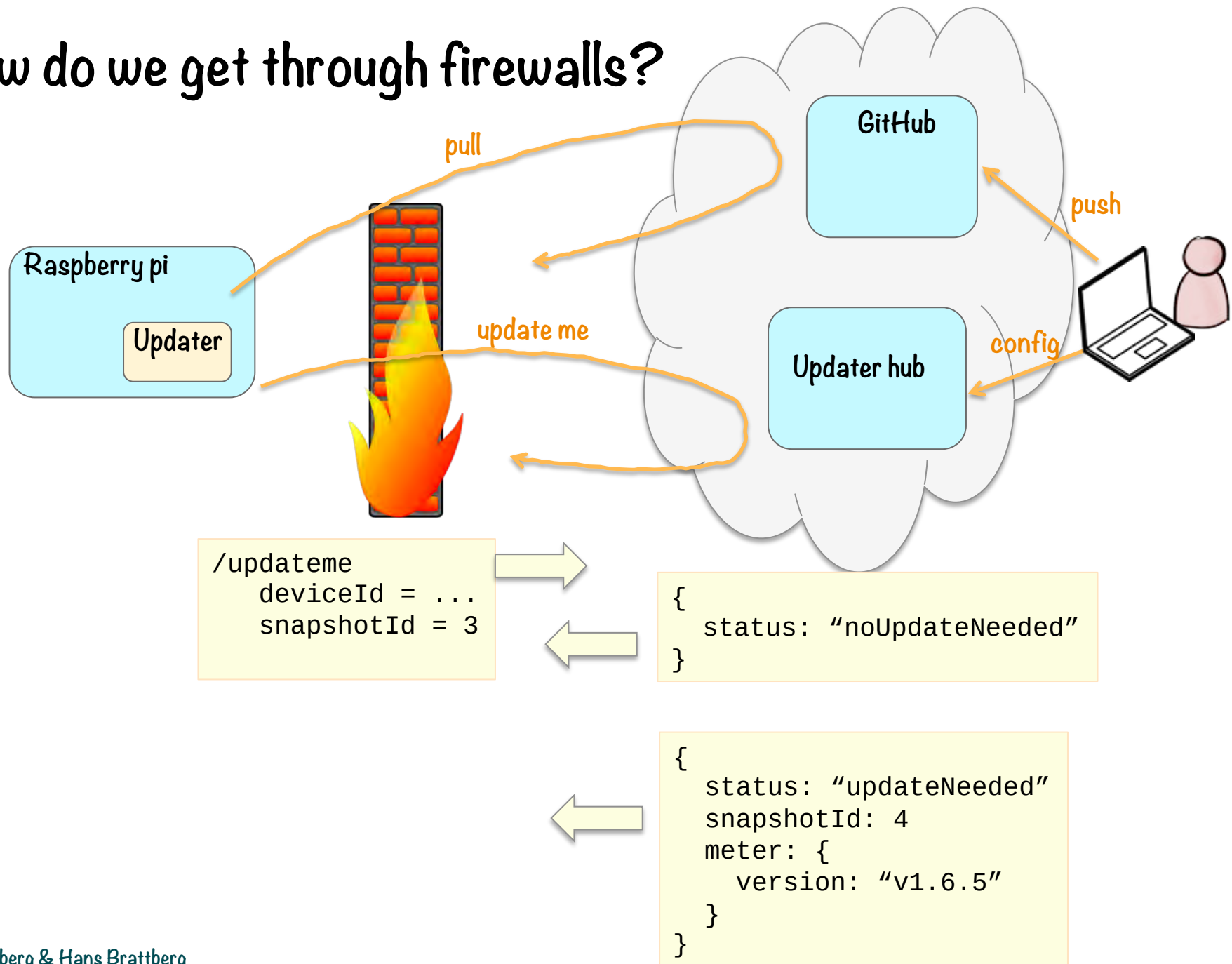
Sophia Kniberg

Ringing...



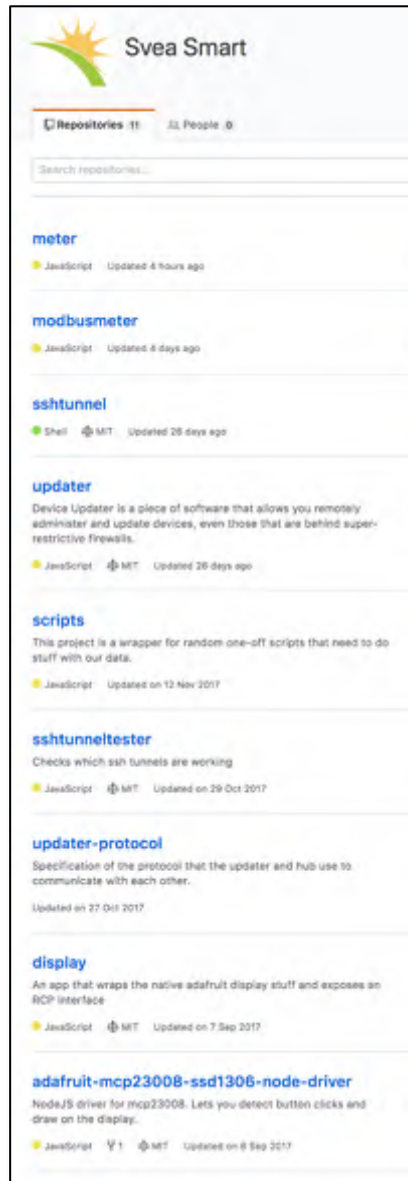


How do we get through firewalls?

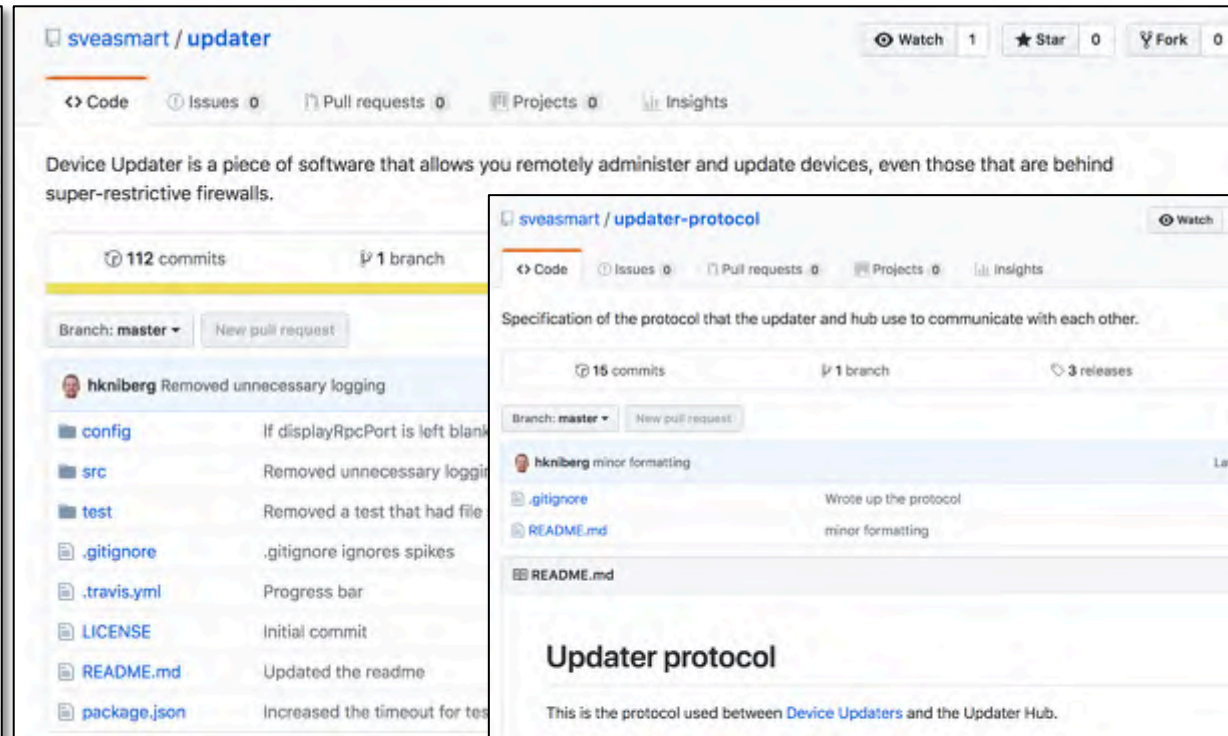


Most of our code is open source on github

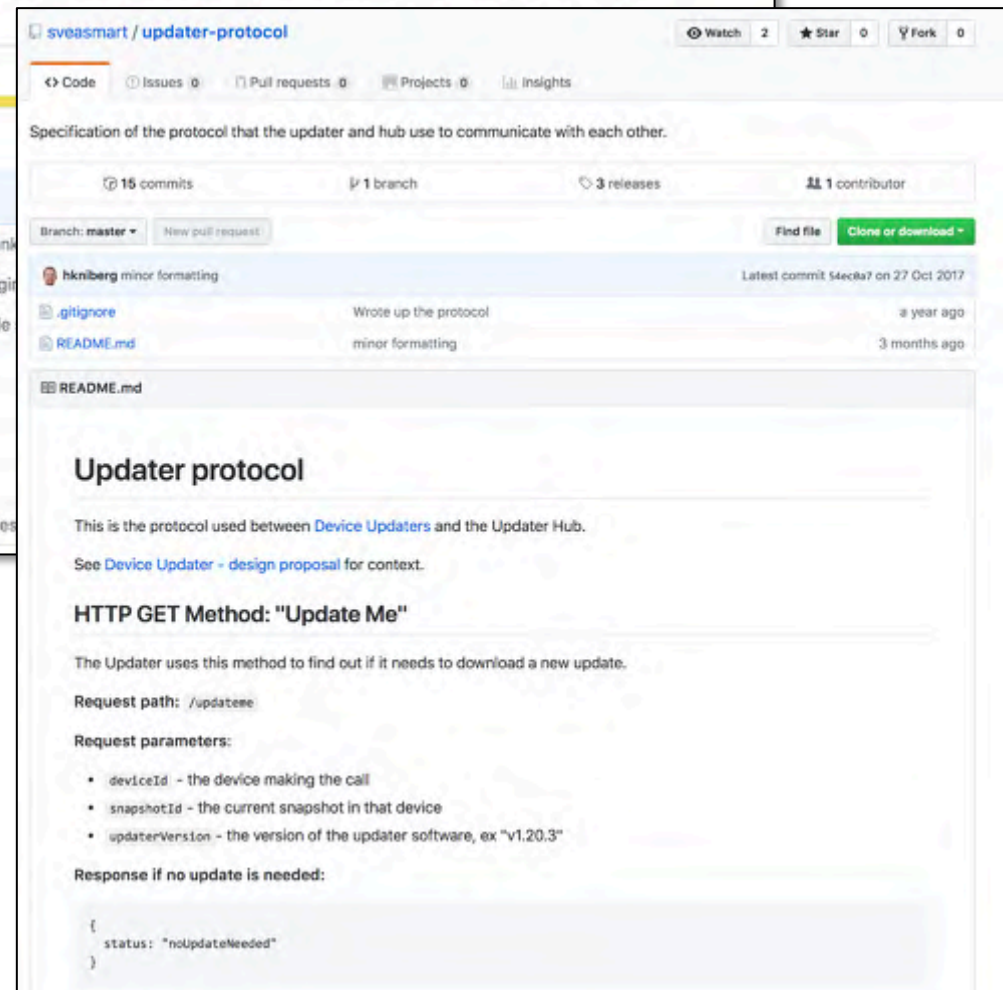
... but updater and updater-protocol are probably the most useful



The screenshot shows the Svea Smart GitHub profile page. It features a list of repositories on the left side, including 'meter', 'modbusmeter', 'sshtunnel', 'updater', 'scripts', 'sshtunneltester', 'updater-protocol', 'display', and 'adafruit-mcp23008-ssd1306-node-driver'. Each repository entry includes its name, a brief description, and the last update date. The 'updater' repository is highlighted, showing its description: 'Device Updater is a piece of software that allows you remotely administer and update devices, even those that are behind super-restrictive firewalls.' and its last update date: 'Updated 26 days ago'.



The screenshot shows the GitHub repository page for 'sveasmart/updater'. The repository has 112 commits and 1 branch. The description states: 'Device Updater is a piece of software that allows you remotely administer and update devices, even those that are behind super-restrictive firewalls.' The file list on the right includes 'config', 'src', 'test', '.gitignore', '.travis.yml', 'LICENSE', 'README.md', and 'package.json'. The commit history shows a recent commit by 'hkniberg' titled 'Removed unnecessary logging'.



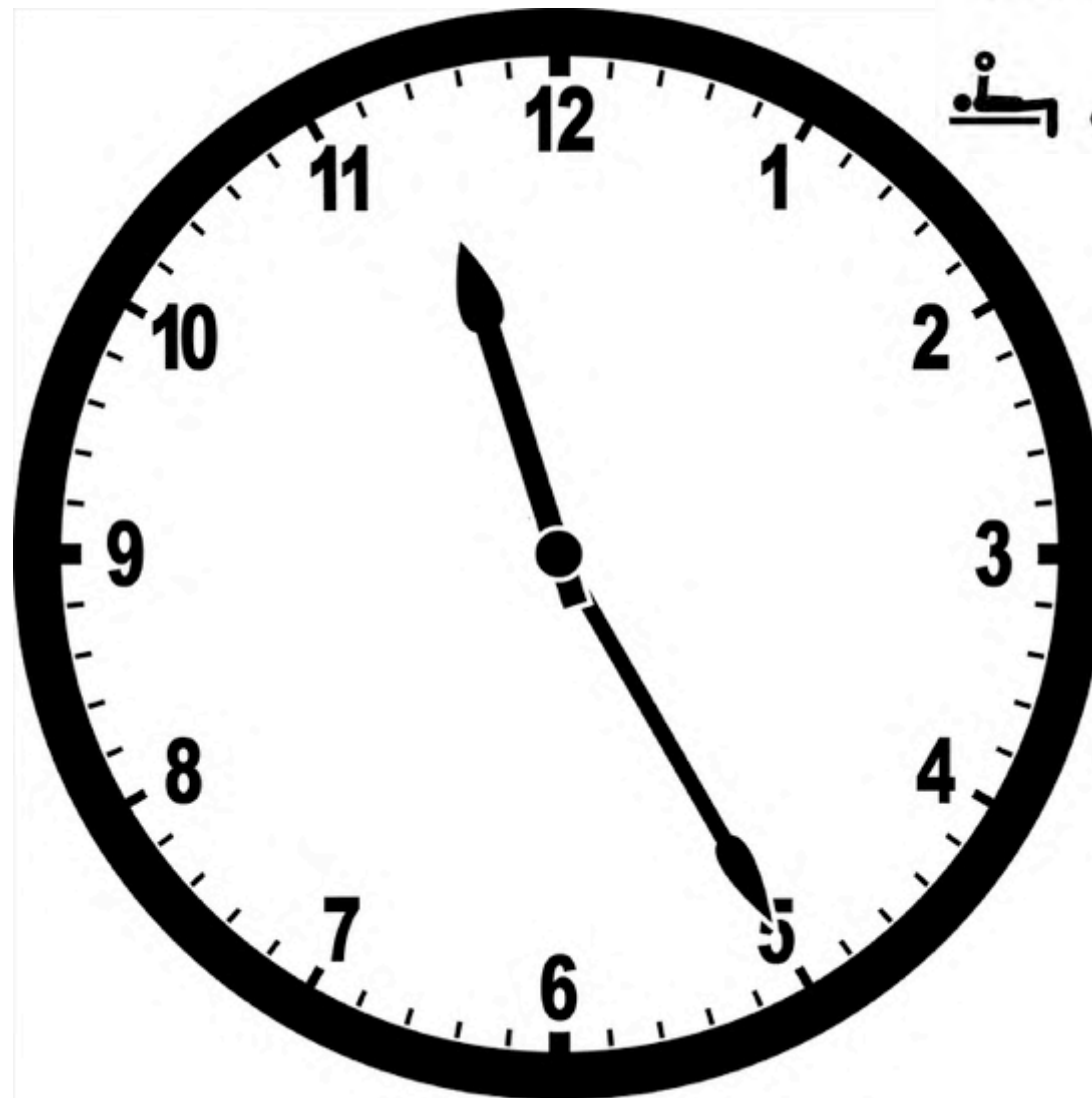
The screenshot shows the GitHub repository page for 'sveasmart/updater-protocol'. The repository has 15 commits, 1 branch, 3 releases, and 1 contributor. The description states: 'Specification of the protocol that the updater and hub use to communicate with each other.' The file list on the right includes '.gitignore', 'README.md', and 'package.json'. The commit history shows a recent commit by 'hkniberg' titled 'minor formatting'. The README content is visible, starting with 'Updater protocol' and 'This is the protocol used between Device Updaters and the Updater Hub.' It also includes a section for 'HTTP GET Method: "Update Me"' and a list of request parameters: 'deviceId', 'snapshotId', and 'updaterVersion'.

What can possibly go wrong? **EVERYTHING!**

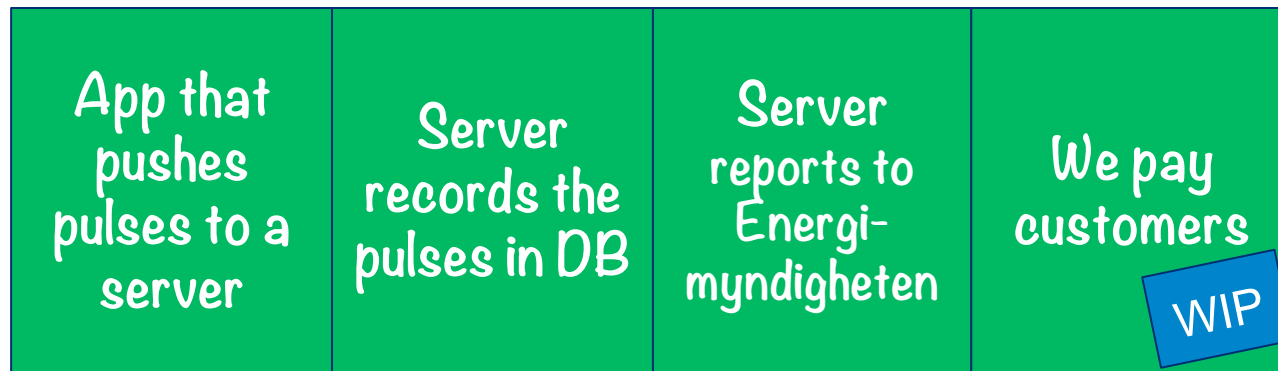
IOT



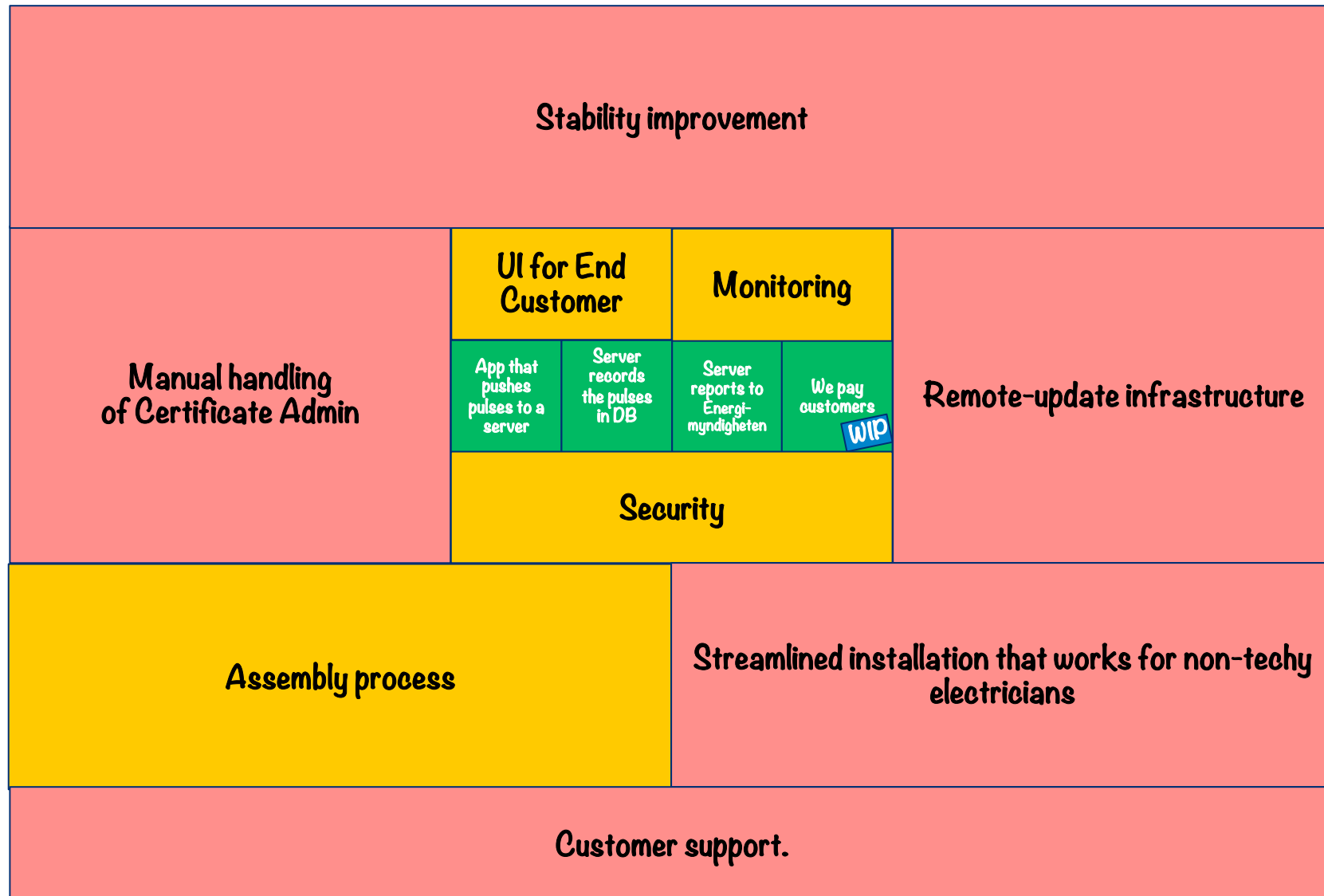
The time should now be:



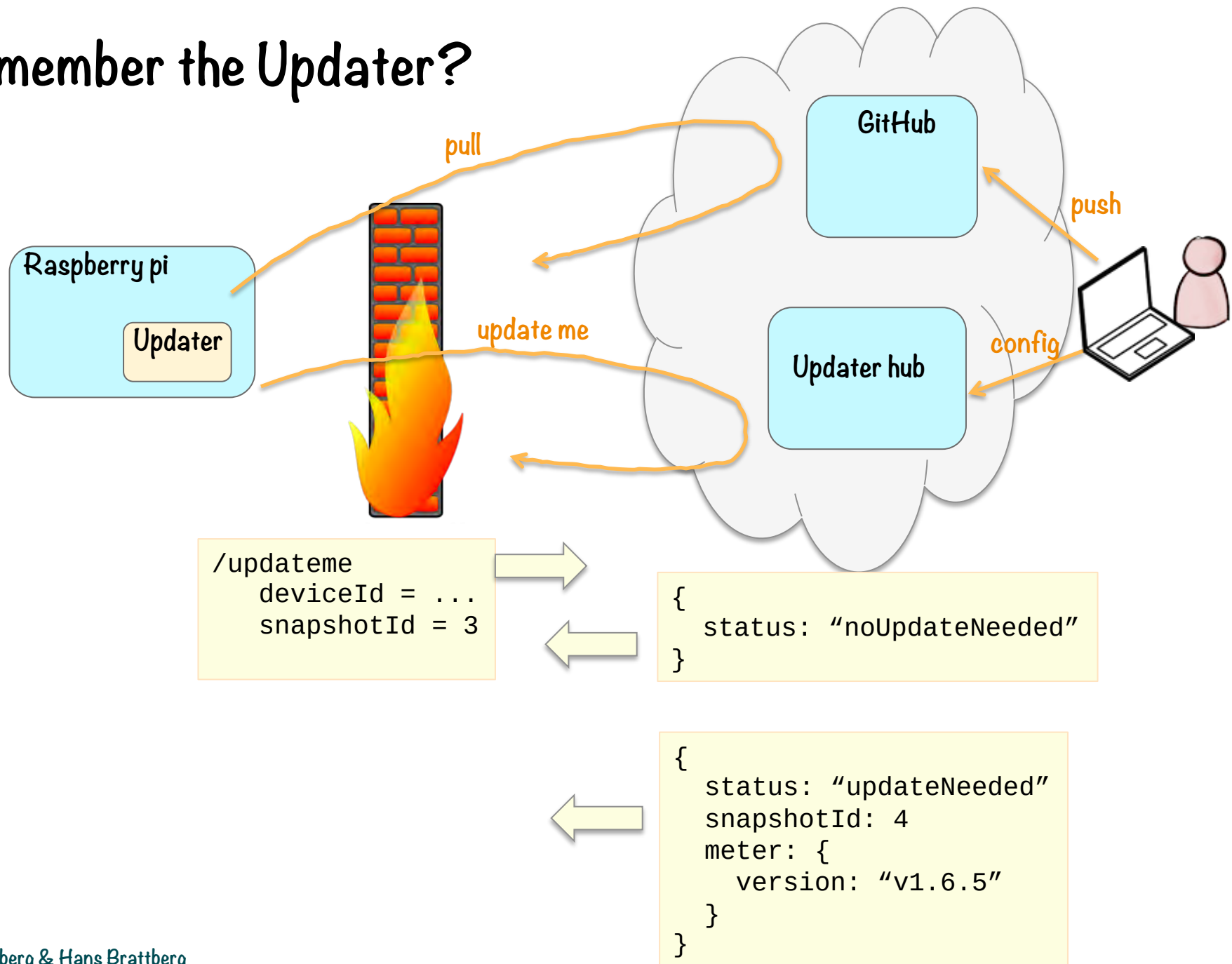
How hard can it be?



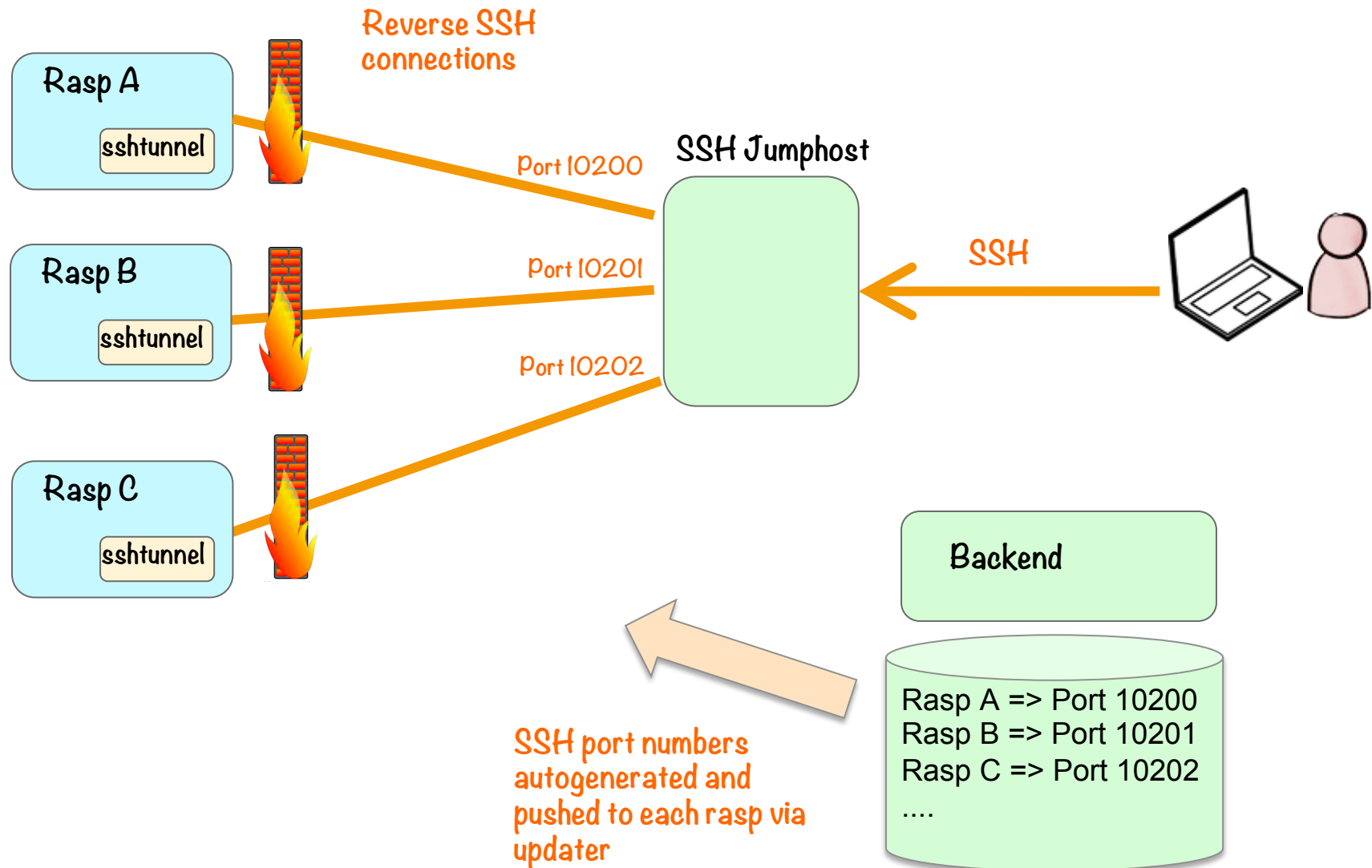
Reality – where we spend our time



Remember the Updater?



The other way in: SSH tunnel



Ansible

```
create_cloneable_sd_card.yaml x
158 - name: install node
159   apt:
160     name: nodejs
161     state: present
162     when: correct_node_version_installed == false
163
164 - name: Check newly installed node version
165   command: nodejs -v
166   register: node_version
167
168 - name: Assert that the correct version of node is installed now
169   assert:
170     that: "expected_node_version in node_version.stdout_lines[0]"
171
172 - name: "Download updater version {{ updater_version }}"
173   unarchive:
174     src: https://github.com/sveasmart/updater/archive/v{{ updater_version }}.zip
175     dest: "{{ root_dir }}"
176     creates: "{{ root_dir }}/updater-{{ updater_version }}"
177     remote_src: True
178
179 - name: Symlink the updater
180   file:
181     src: "{{ root_dir }}/updater-{{ updater_version }}"
182     dest: "{{ root_dir }}/updater"
183     state: link
184     force: yes
185
186 - name: Install the updater (npm install)
187   npm:
188     path: "{{ root_dir }}/updater"
189
190 - name: Test the updater (npm test)
191   shell: "npm test"
192   args:
193     chdir: "{{ root_dir }}/updater"
194
195 - name: Copy in the production config for updater (overwriting if it already exists)
196   copy:
197     src: "updater/local.yml"
198     dest: "{{ root_dir }}/updater/config"
199     force: yes
200
```

```
TASK [Copy in the production config for updater (overwriting if it already exists)] *****
changed: [43q9rkz8vk]

TASK [Copy in the updater start script] *****
changed: [43q9rkz8vk]

TASK [Copy in the updater service setup script] *****
ok: [43q9rkz8vk]

TASK [Copy in the updater change-hostname.js script] *****
changed: [43q9rkz8vk]

TASK [Install the updater service, stop the updater service if its exists] *****
changed: [43q9rkz8vk]

TASK [Clone Meter v2.6.0 from github] *****
changed: [43q9rkz8vk]

TASK [Install the Meter (npm install)] *****
ok: [43q9rkz8vk]

TASK [Test the Meter (npm test)] *****
changed: [43q9rkz8vk]

TASK [Copy ansible.log] *****
ok: [fvh8k|nb7y]

TASK [Clone Display v1.2.0 from github] *****
ok: [43q9rkz8vk]

TASK [Reboot at swedish midnight as cronjob] *****
changed: [fvh8k|nb7y]

TASK [Add SSH authorized key for the installer, so we can SSH into this rasp when doing ansible stuff] *****
ok: [fvh8k|nb7y]

TASK [Install the Display (npm install)] *****
ok: [43q9rkz8vk]

TASK [Enable I2C in /boot/config.txt] *****
ok: [43q9rkz8vk]

TASK [Ensure that /etc/modules contains i2c-dev] *****
ok: [43q9rkz8vk]

TASK [Make the 'pi' user owner of everything under the root dir (non-recursive)] *****
ok: [43q9rkz8vk]

TASK [Copy in the rasp's id_rsa, so it can connect to the jumpserver] *****
ok: [fvh8k|nb7y]

TASK [Make the 'pi' user owner of everything under the updater dir (recursive)] *****
changed: [43q9rkz8vk]

TASK [Make the 'pi' user owner of everything under the apps/meter dir (recursive)] *****
changed: [43q9rkz8vk]

TASK [Make the 'pi' user owner of everything under the apps/display dir (recursive)] *****
changed: [43q9rkz8vk]

TASK [linsinfile] *****
changed: [43q9rkz8vk]

TASK [Remove any existing snapshot-id, so the updater will download the right app versions] *****
changed: [43q9rkz8vk]

TASK [Start updater service] *****
changed: [43q9rkz8vk]

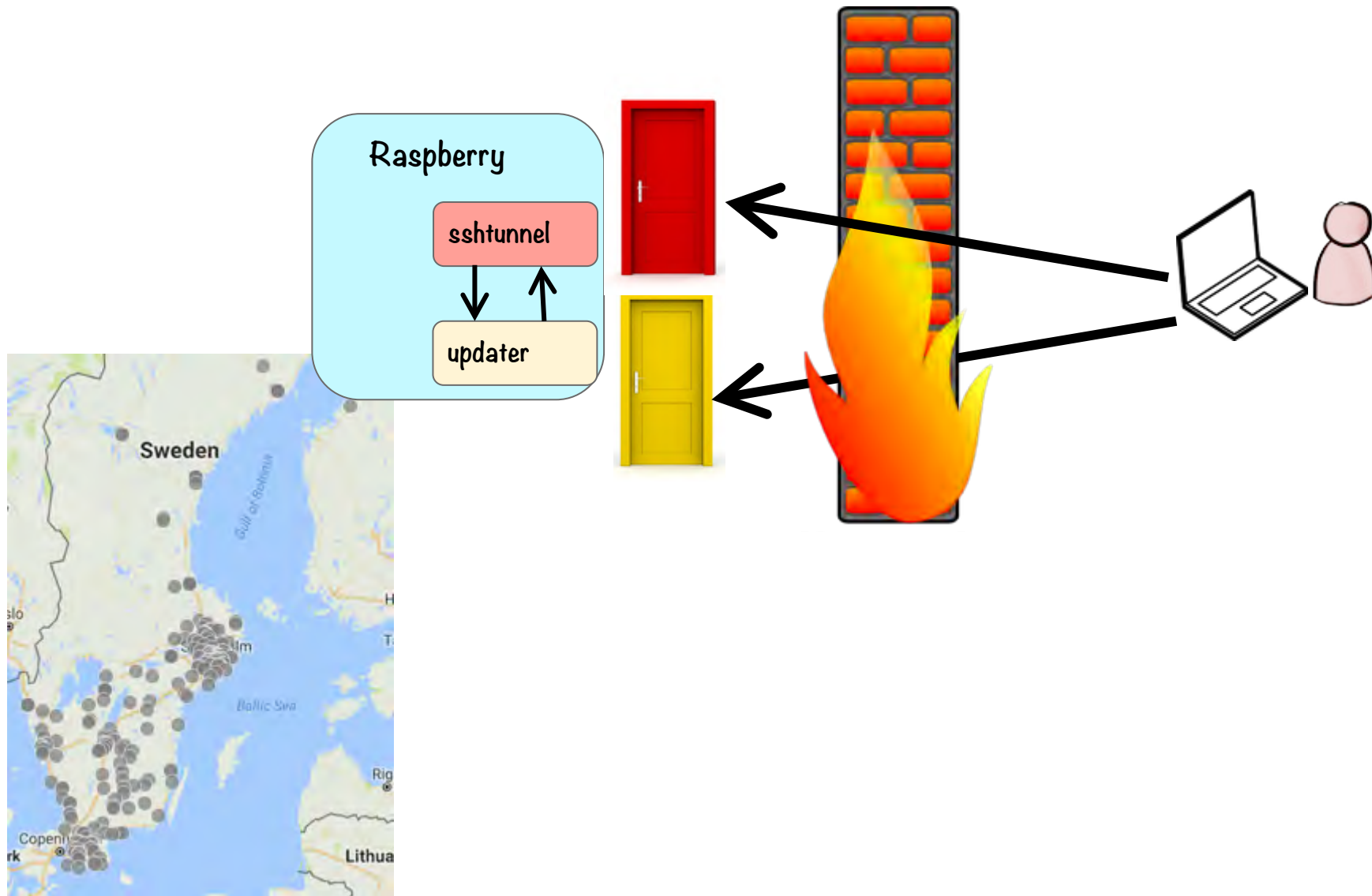
TASK [debug] *****
ok: [43q9rkz8vk] => {
  "msg": "DONE! Ready to clone! Don't reboot before cloning it!"
}

TASK [Copy in the rasp's id_rsa.pub (not sure if it actually needs to be here though)] *****
ok: [fvh8k|nb7y]

TASK [Add SSH authorized key for the jumphost, so we can SSH into this rasp from the jumphost] *****
ok: [fvh8k|nb7y]

TASK [Remove password for ssh login] *****
ok: [fvh8k|nb7y]
```


2 ways in = important lifesaver!



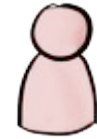
Buy vs Build

“Couldn’t we just use <insert tool/framework here>?”

Beware vendor lock-in!

Can I use your infrastructure to replace your infrastructure with something else?

Uh... no!



Who has this box?

Meter 983246430

 60 W (21 minutes ago)
Online Connected 43 minutes ago
 0 W current power  -0 grams CO2eq / hour
Total energy measured: 25.9 kWh
Last energy pulse: Feb 5, 2018, 10:01:00 PM (21 minutes ago)
Last sign of life: Feb 5, 2018, 10:06:27 PM (16 minutes ago).
[Mark meter installation done](#)

SmartBox #waa2ebc3g5

Property Designation:
Address:
Peak Power:

Customer
(none)

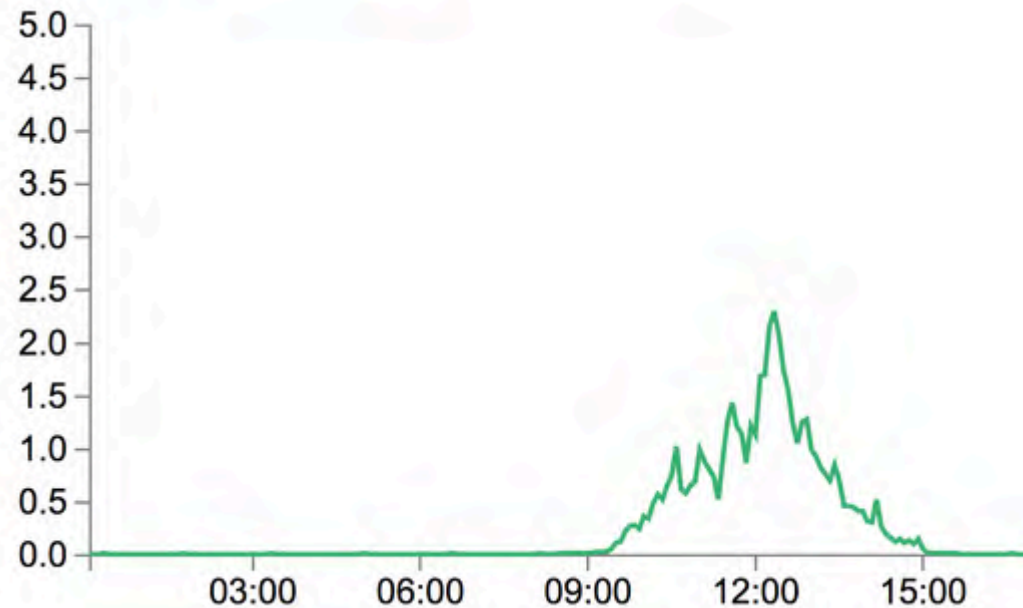
Power (kW)

Week 2

< 7 Jan 8 Jan 9 Jan 10 Jan 11 Jan 12 Jan 13 Jan >

 4.5 kWh day-total production  -896 grams CO2eq

2018-01-09



Someone in Nyköping, according to the IP address...

Challenge: Who has which box?

1. Printed ID on Raspberry Box



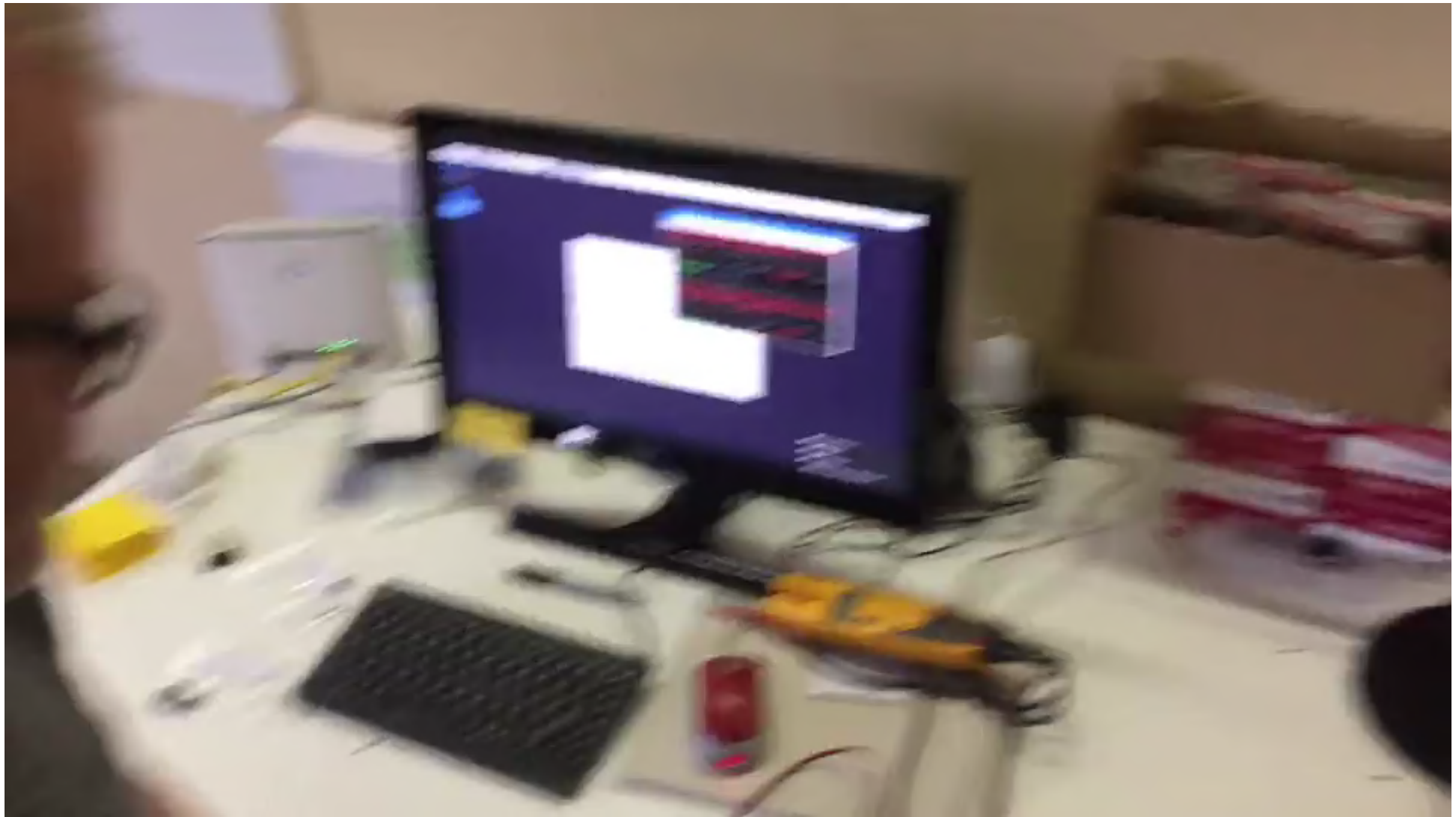
2. Display with QR code



3. Customer Name on Screen



Box assembly – spring 2017



Today's assembly line

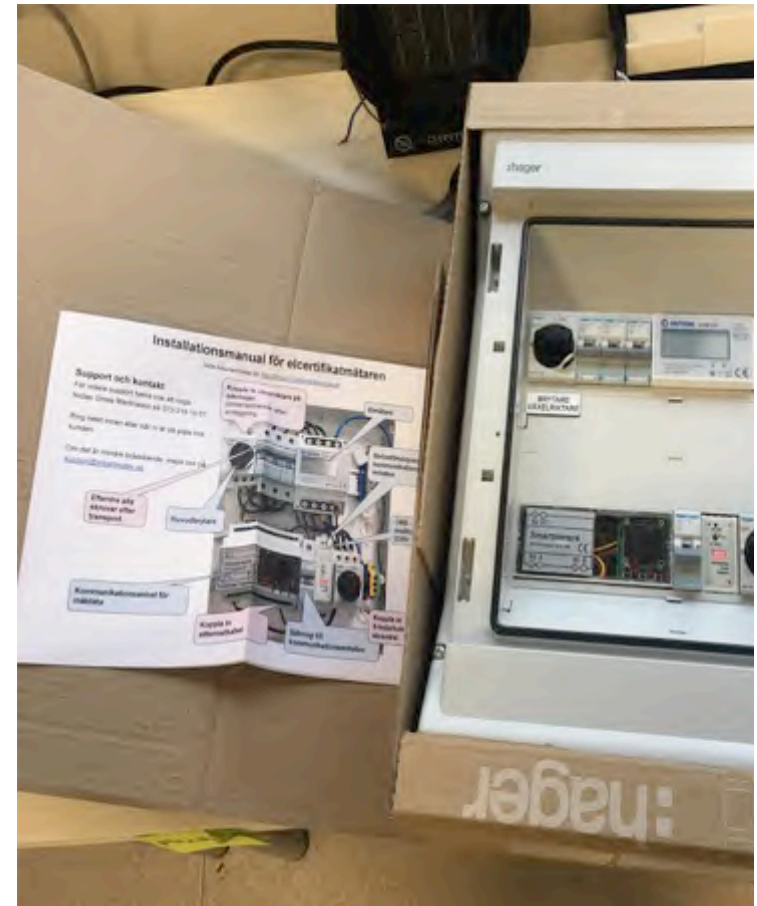
Assembly



Test



Packaging



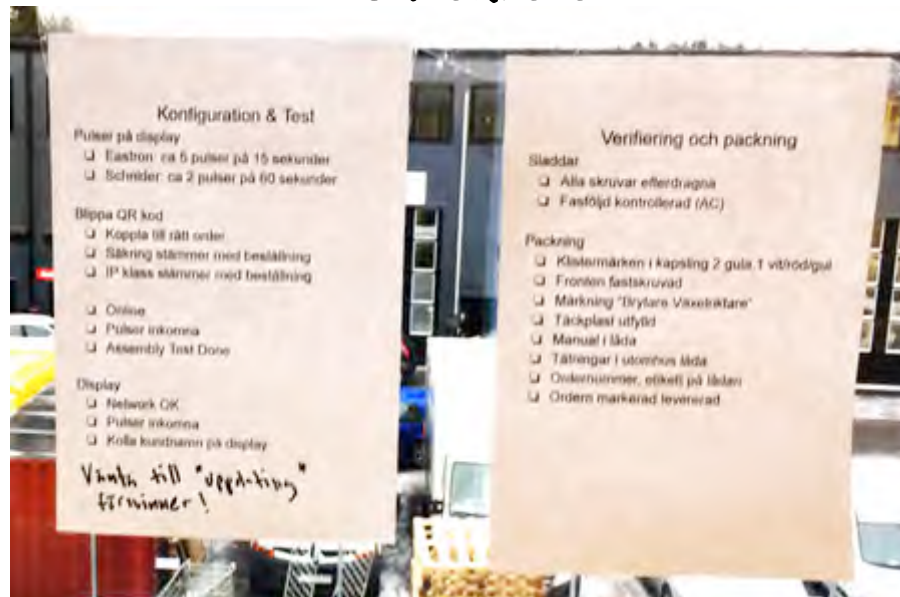
Assembly - Lean is pretty good for hardware too :)

Single piece flow.

Quality in delivery builds trust

Open orders 15 (about 48 new orders per month)					
Order #390 Svea Solar Hans Brattberg (Hasses Kebab och Parabolantennor AB) Fuses: 0 Created: 2018-02-05 Deadline: 2018-01-05 Not delivered yet	Order #391 Svea Solar Henrik Kniberg (Henriks Company AB) Fuses: 0 Created: 2018-02-05 Deadline: 2018-02-10 Not delivered yet	Order #128 Svea Solar Lars Miller Outdoors - IP65 Inverters: 3, 3 st SE27,6K växelriktare 125 huvudbrytare Created: 2018-01-26 Deadline: 2018-02-15 Not delivered yet	Order #356 Svea Solar Bengt Engdahl Outdoors - IP55 Inverters: 4, 4st 27,6K Created: 2017-12-18 Deadline: 2018-02-16 Not delivered yet	Order #389 Jönköpings Energi Jonas Hallin Indoors - IP21 Fuses: 16 Created: 2018-02-01 Deadline: 2018-02-23 Not delivered yet	Order #388 Jönköpings Energi Thord Bivred Indoors - IP21 Fuses: 16 Created: 2018-02-01 Deadline: 2018-02-23 Not delivered yet

Checklists



Standardize work



Scaling the production - Creating SD cards

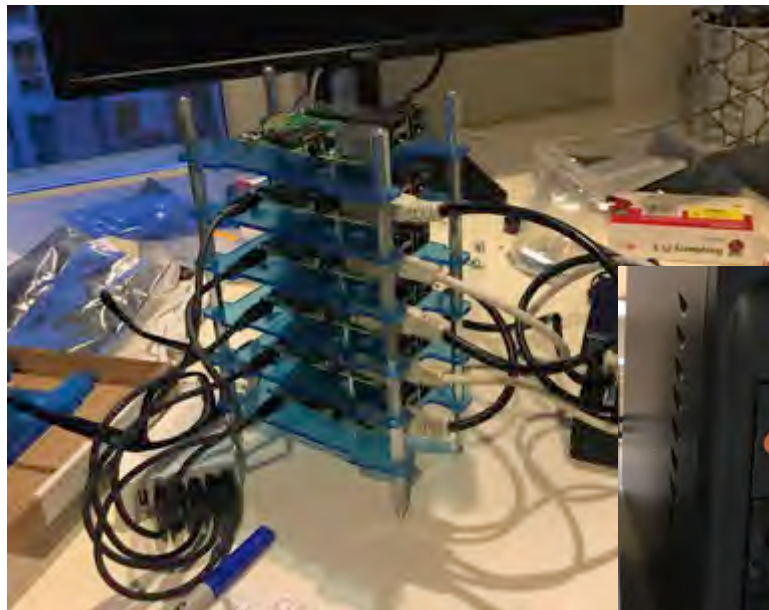


Solution 1: Single Card

Run Ansible on a single Raspberry Pi to create SD cards, one at a time
18 minutes

Solution 2: Raspberry Cluster

A Raspberry Pi cluster of 6
Run with Ansible to create 6 SD Cards in 18 minutes



Solution 3: SD copier

A SD card copier, with a master to copy from
2 minutes



Cross-functional team & onsite customer!



Henrik Kniberg & Hans Brattberg

Approved and
reporting 206



Henrik Kniberg & Hans Brattberg

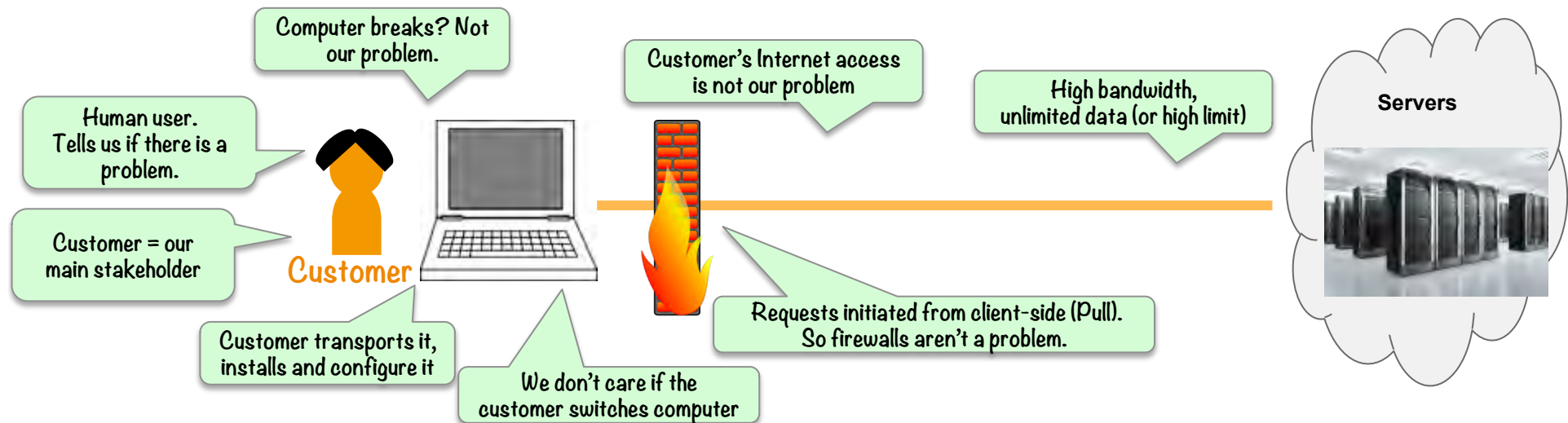
Cloud dependency...



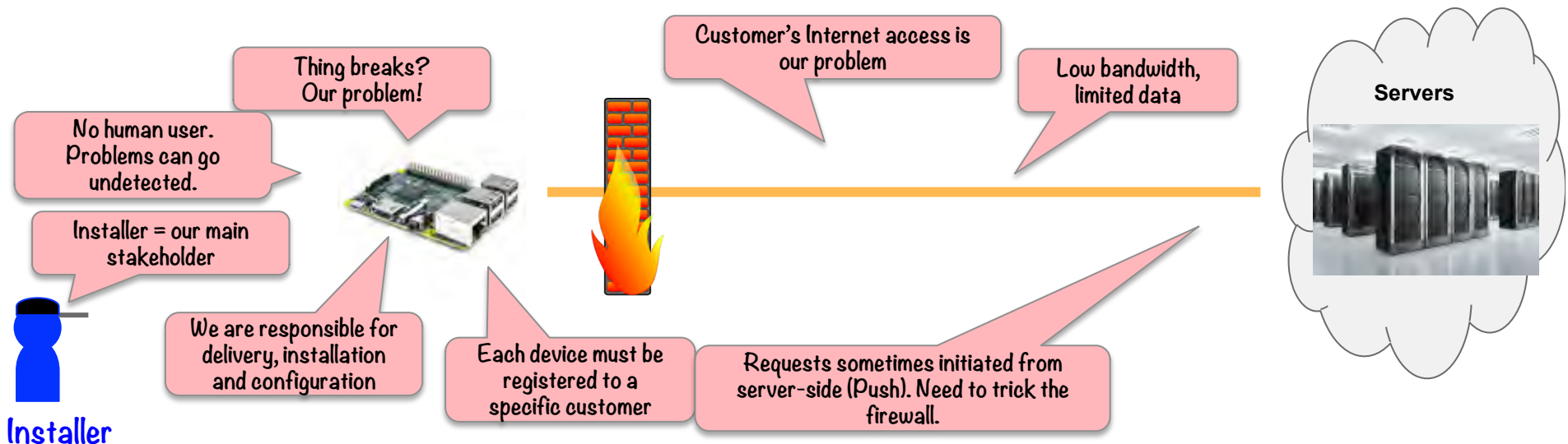
IoT = ?



Not IoT = the comfy world we came from



IoT = wild west!



What's next?

Take-aways?

Talk to us about

- Data scaling
- Saving the world from climate change
- Getting solar panels
- ...

Don't try this at home....

