Week 5 exercise: Technical privacy aspects and making designs better

This week we take the week 4 exercise, perform additional analysis, and apply some design patterns to make the design (slightly) better.

Making the design better
In the discussion that followed, you managed to convince the company to make some changes to the architecture.

First, you think that the co-location of the web service (Nginx and Node.js) and the PostgreSQL database on the same machine is not very optimal state of affairs. If there is a security vulnerability in Nginx, or if the Node.js application has an SQL injection vulnerability, an attacker could potentially compromise the whole database.

Second, the use of the (publicly displayed) meter serial number as an encryption key doesn’t feel exactly right. The engineers proposed putting the serial number sticker inside the meter box, so that the user would have to break the seal in order to see it, and to remove the meter number request from the web account creation stage.

Your task and to return (1/2):
- Think: Which components would you separate by adding new security domains? The Node.js backend and PostgreSQL were already identified as a separation candidate, but are there any others? Describe, in a few lines, which components you would separate, and how this would be done.
- Think: Why moving the meter number sticker inside the electric meter box is not a complete solution? Describe, in a few lines, what risks would still remain. Can you think of a better (but perhaps more challenging) re-engineering option? What would that be?
- What is your professional opinion about this system? Is it good to go after these changes? Or should it be burned in fire? Or something in between? Write a couple of lines expressing your candid opinion.

Privacy analysis
Irrespective of your professional opinion about this system, above, the company for whom you performed threat modelling last week pushes forward with it and is intending to hop on the PaaS (Platform as a Service) cloud bandwagon. Essentially, they’re going to ditch their own servers, which are an additional maintenance burden, and are switching to Heroku. The engineers have drawn a corrected data flow diagram that shows the Heroku option. (You are glad to see that it actually implements additional security domains, and segregates the SQL database from the web front-end.)
Heroku ([https://www.heroku.com/](https://www.heroku.com/)) is a PaaS platform that runs on Amazon cloud services ([https://aws.amazon.com/](https://aws.amazon.com/)). Essentially, Heroku provides a managed web application framework - in week 4 exercise, the Node.js and PostgreSQL installations are provided by Heroku, and the developers don’t have to maintain them any more.

However, the customers of the company are in Finland, and the company aims to run the system from Heroku servers residing in Amazon’s United States based availability zones (in plain English, this means that the servers are going to physically run in a server farm in the United States).

As you recall, it may not be straightforward for a European Union based outfit to export EU customers’ personal data into the United States. The company’s lawyers have identified this as a potential risk; you have been contracted to determine technical privacy aspects of the design.

**Your task:**
We provide a reference data flow diagram for this exercise right after the deadline for week 4 exercise has closed. You can use your own diagram from week 4, or wait for the reference diagram to become available. It will be available only after the deadline because we will try to discourage copying.

On top of this diagram (i.e., either your own or the reference one we provide), draw the “privacy domains” - meaning, which components are *physically* within which privacy jurisdiction. The jurisdictions here are Finland (=EU) and the United States. Ensure that it is clear which data flows cross the *physical* EU/US boundary.


Remind yourself about the “TRIM” considerations (see lecture notes) that you can perform for personal data.

**To return (2/2):**
- A data flow diagram that has the privacy domains drawn on top of it. As with the week 4 exercise, you should return the picture electronically.
- A list of:
  - *Personal data* assets that can be found in the system. Specify what data it is, and which components store or process it. Personal data is defined in Section 3 of the Personal Data Act.
  - *Sensitive personal data* assets that can be found in the system. Specify what data it is, and which components store or process it. Sensitive personal data is defined in Section 11 of the Personal Data Act.
Answer the TRIM considerations. For each of these, respond in no more than a few lines of text:

○ What data is being transferred over a legislative boundary (in this case, from EU to the US)? Can we legally do this (see Personal Data Act, Sections 22 and 23)? Under what circumstances are we allowed to do this?

○ In which places do we retain (sensitive) personal data? What sort of retention policy would you recommend for the company (i.e., when should personal data be destroyed)?

○ It seems to be pretty clear that we should inform the user about (sensitive) personal data collection, and ask the user’s consent. When would you recommend we do this, and how?

○ Can we minimise some data transfer or storage? Could we replace some of the (sensitive) personal data flows or storage with something else that would not be personal data, thus freeing us from the requirements of Personal Data Act? Or could we use “normal” personal data instead of sensitive personal data somewhere? If we could, what sort of changes would that entail?