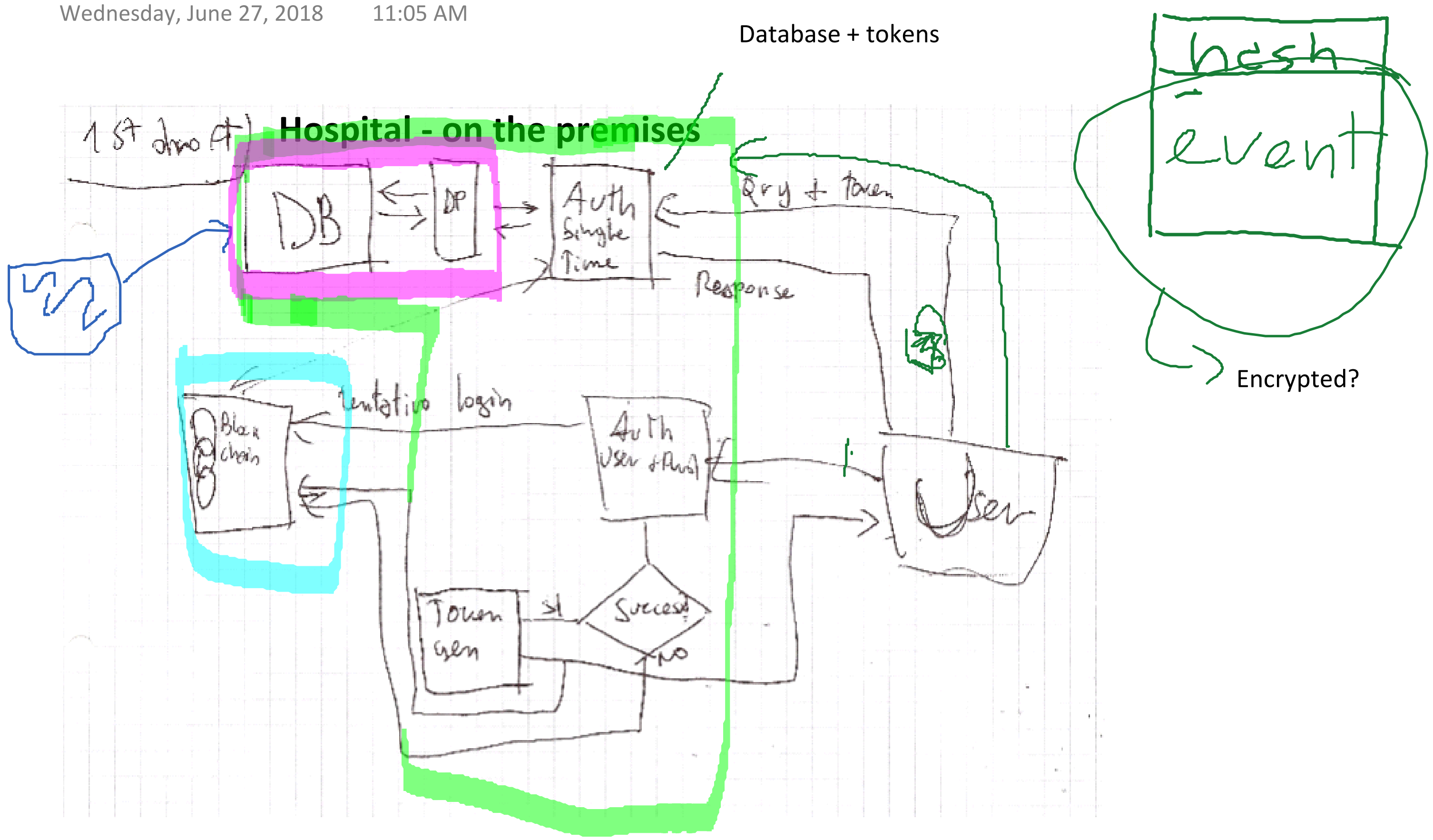
Valero Maggio the gas station

Wednesday, June 27, 2018 10:46 AM

Diagram v1

Wednesday, June 27, 2018 11:05 AM



Notes on v1

Wednesday, June 27, 2018 10:56 AM

Basically saying that once you get the token you can access the database. It is special because of the middle thing.

The general idea why we're talking about blockchain was that the blockchain could be used as a way to keep track of people accessing the data - when and why and stuff.

Let's see if we can compromise security of this diagram.

Overall security here there are issues

Rainbow table - could brute force the hash. So need salt.

Three ways to authenticate:

- Something you know
- Something you have
- Something you are

Three-factor

- Fingerprint
    - Something you are
- Phone
    - Something you have
- Password
    - Something you know

We are planning on anonymizing. How many checks and double-check to let someone in?

Data is something you know.

Where do we store the key for the blockchain?

- You need a key management system. That's another bag of worms.

If you know the hash and salt you have to reverse-engineer what the input might be.

Why would you store the token in the blockchain? If it's unencrypted, anyone who sees that in the Blockchain has 5 minutes to get it and issue as many commands as they can.

You want a cross-site request monitoring tool so people can't spoof from cookies in other places

You're not trying to defend against a giant DDOs atack by people tat are tring to take down your entire network.
You're just creating a threat model and hoping that a guy sitting in a bsement can't mess with youl

What is the data that we need to hide?

HIV status? Other illnesses? Genetic mutations?

In the database we have the attributes of whether or not something is sensitive?

We need to create a distinction between different data

Don't get too deep into the weeds

Should you be able to see the data of people who aren't your patient?
- Access restrictions
- Data access policies

Potential assumptions?
- If you're a doctor, you already have access to the patient's medical data. It's not necessary to allow access to patient medical data from this system. So you just know you put <image + data> in and you get a certain <output> but you can restrict access to the input data from your database.

Basically saying our target user - user we're designing the system for - is a researcher. They want the data for a research project.
We have images and metadata and our user workflow is the user querying the images and metadata. What are the questions they are going to ask? Then we can look at how those might be exploited.

"I want to look at all medulloblastoma <images? Metadata?> of males between the ages of 10 and 12"

Looking at aggregate data in the set. What are the simple ways given that you can make any query like that, that you can comprise privacy?

We can make the assumption that the images are useless in compromising privacy but the metadata definitely can.

The question is, are we returning metadata?

If we can make these queries, we clearly have the metadata and are handing it out. Do we explicitly include the metadata in our return queries?
<Image 1, sex: male, age: 12, survival: false>

The OPBG data is already anonymized; i.e. components like age technically shouldn't be explicitly stated

Summary of 15 minutes: GUID would be quite suboptimal

Just think about the threat model?

Two different use cases and threat models:
1. Untrusted researchers who want to securely access subsets of the data. They should be resistant to deanonymization.
2. Trusted researchers with access to deanonymized data and won't try to deanonymize it but they could put machine learning models contingent on that data and then give that out to people who might be adversarial.
   The only thing is securing the model (model = bag) and is it possible to extract private data from a model? The answer is yes.

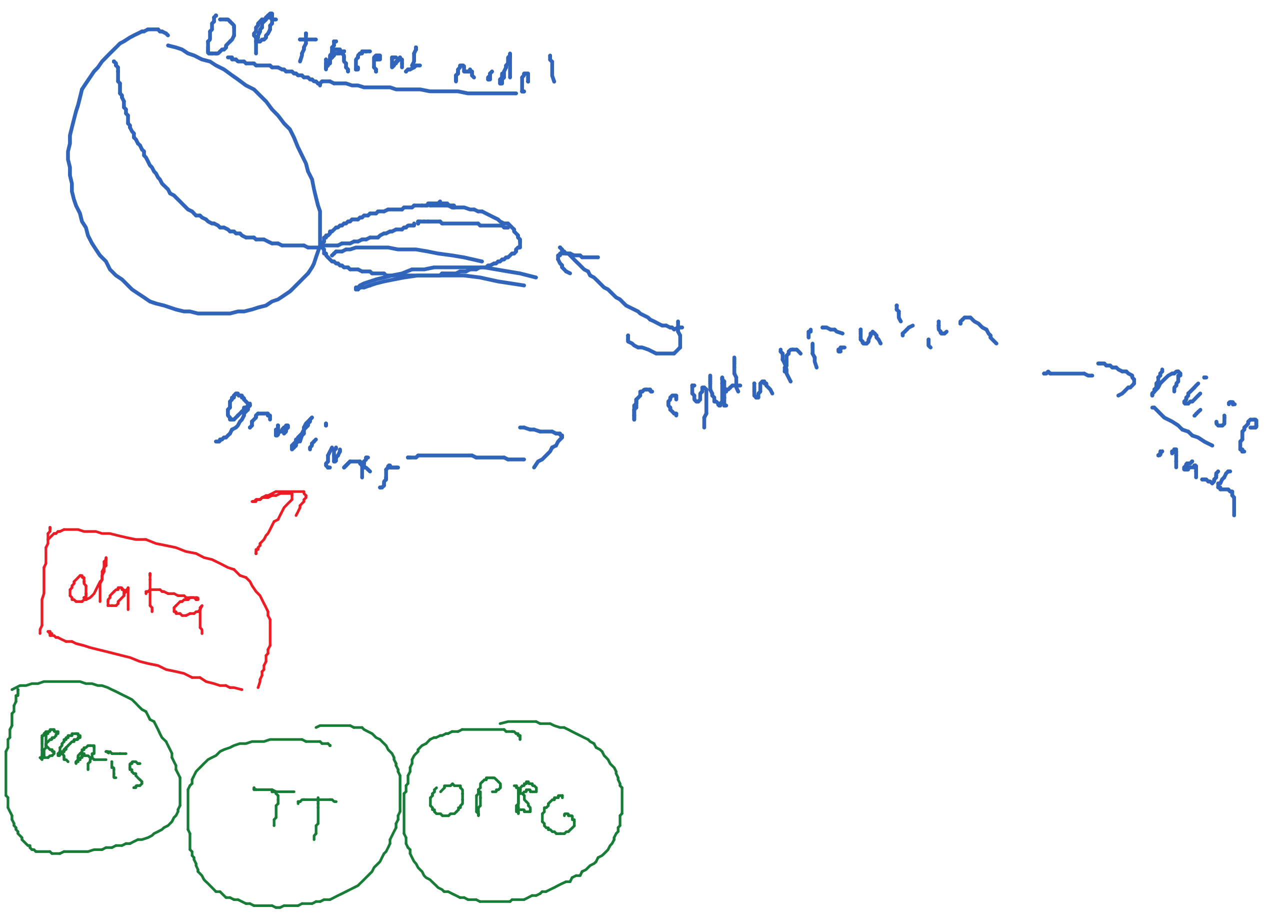
With deep learning there's a lot of recent work where you can add noise and the model achieves epsilon-delta differential privacy where epsilon and delta change depending on the amount of noise you add.

There is the best way to get started with that (this is just math). It becomes what can I prove in terms of when I add this noise to my training dataset I'm guaranteed this level of privacy

I can give you a bunch of pointers towards machine learning models or deep learning there's a lot of code for differentially private optimizers that can do this noise addition and guarantee - the best one right now is in TensorFlow ad it'd be a good idea since you're working in PyTorch to port it over.

Data threat model

Wednesday, June 27, 2018 12:20 PM



Flowchart v2

Wednesday, June 27, 2018 12:31 PM