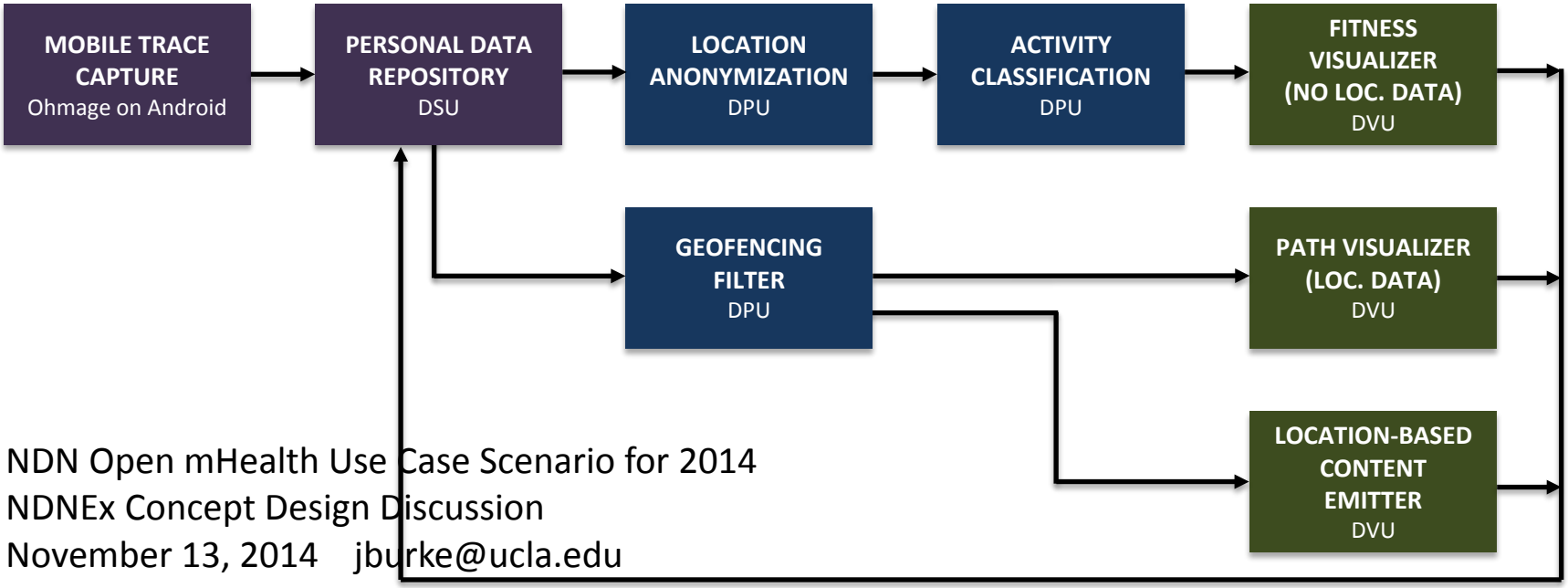
Attribute-based Encryption for NDNEx

Haitao Zhang

irl - UCLA

Background - NDNEx

NDNEx tries to transplat open mHealth to NDN network. The goal is to design a physical activity data ecosystem.
Following is the data conceptual block diagram of NDNex.



NDN Open mHealth Use Case Scenario for 2014
NDNEx Concept Design Discussion
November 13, 2014 jburke@ucla.edu

DSUs, DVUs, and DPUs are potentially run by different entities.

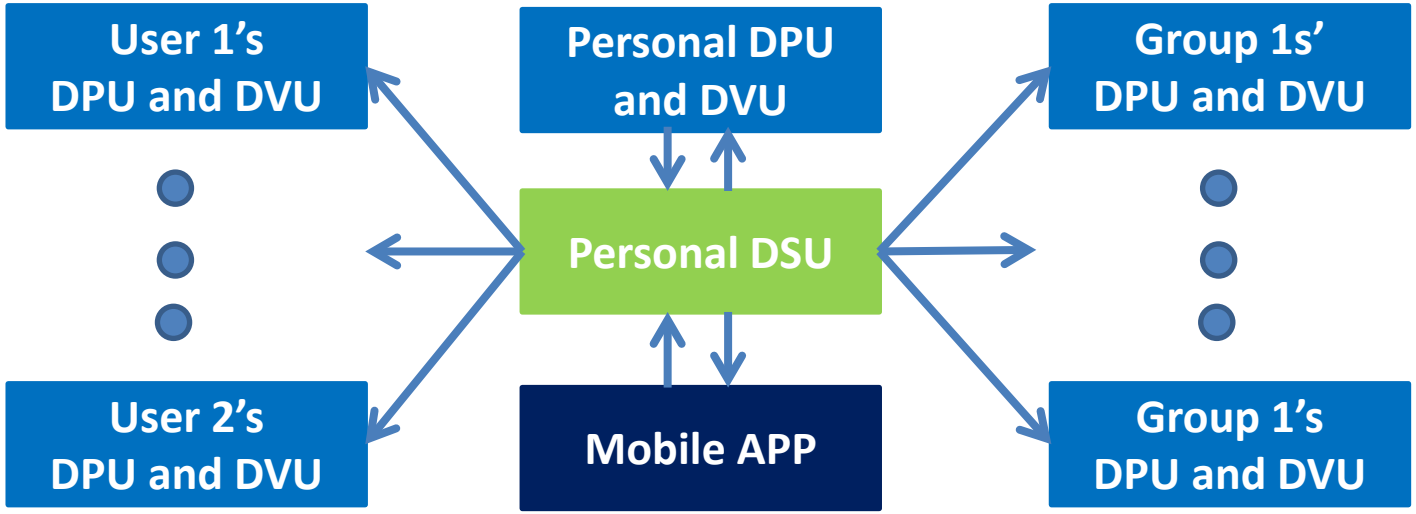
(Personal) DSU is used to store a user's data. The user's DSU (thus the user) should have fully control of her data.

Data flow

There are there kinds of data flows:

1. A user's (Personal) DSU pulls data from the user's mobile app and (personal) DPU+DVU; mobile app and personal DPU+DVU pull data from DSU.
2. A user may authorize other users' DPUs and DVUs to read **part** of her data from her DSU.
3. The user may join some groups, in which case the user should authorize these groups' DPUs and DVUs to read **part** of her data from her DSU.

Besides these, the user wants no one else get access to her data.

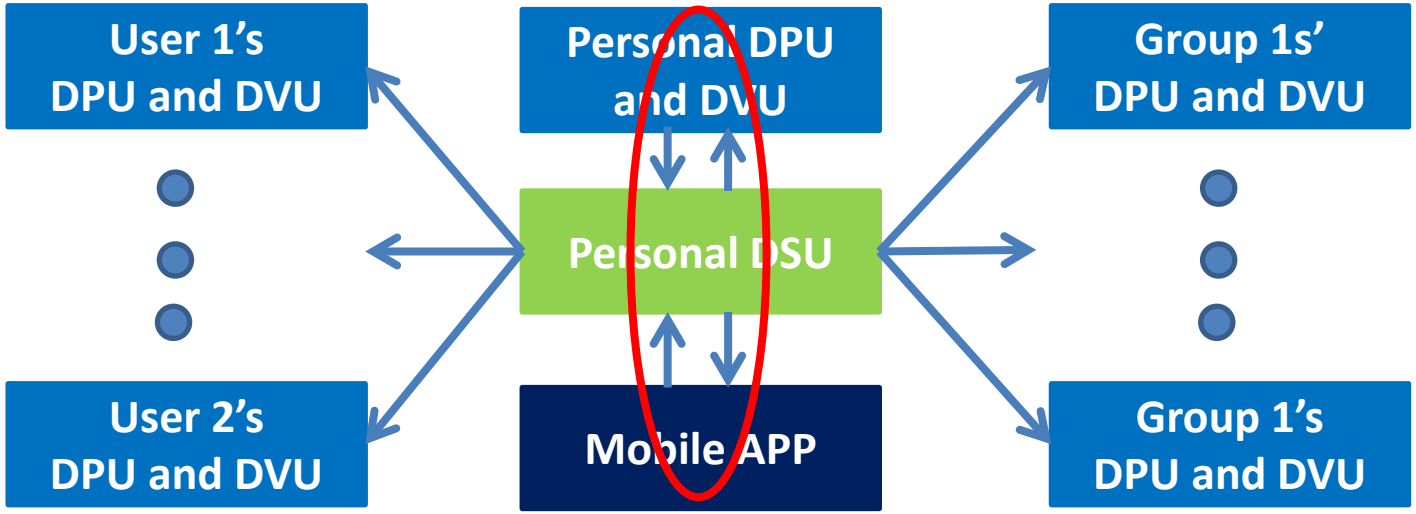


Data flow

There are there kinds of data flows:

1. A user's (Personal) DSU pulls data from the user's mobile app and (personal) DPU+DVU; mobile app and personal DPU+DVU pull data from DSU.
2. A user may authorize other users' DPUs and DVUs to read **part** of her data from her DSU.
3. The user may join some groups, in which case the user should authorize these groups' DPUs and DVUs to read **part** of her data from her DSU.

Besides these, the user wants no one else get access to her data.

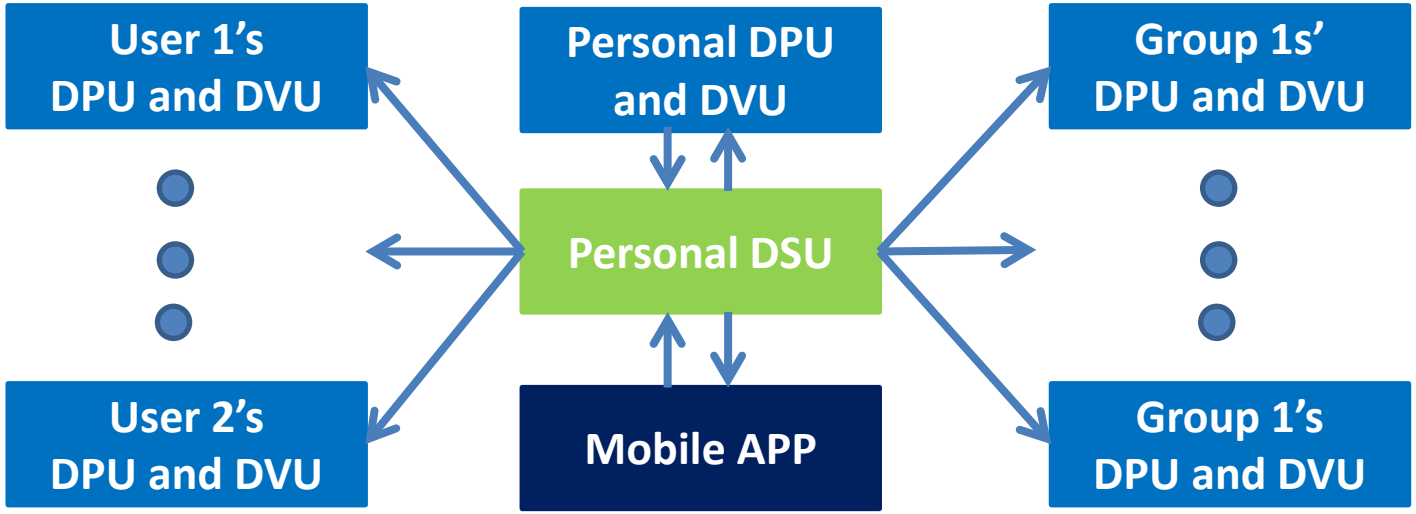


Data flow

There are there kinds of data flows:

1. A user's (Personal) DSU pulls data from the user's mobile app and (personal) DPU+DVU; mobile app and personal DPU+DVU pull data from DSU.
2. A user may authorize other users' DPUs and DVUs to read **part** of her data from her DSU.
3. The user may join some groups, in which case the user should authorize these groups' DPUs and DVUs to read **part** of her data from her DSU.

Besides these, the user wants no one else get access to her data.

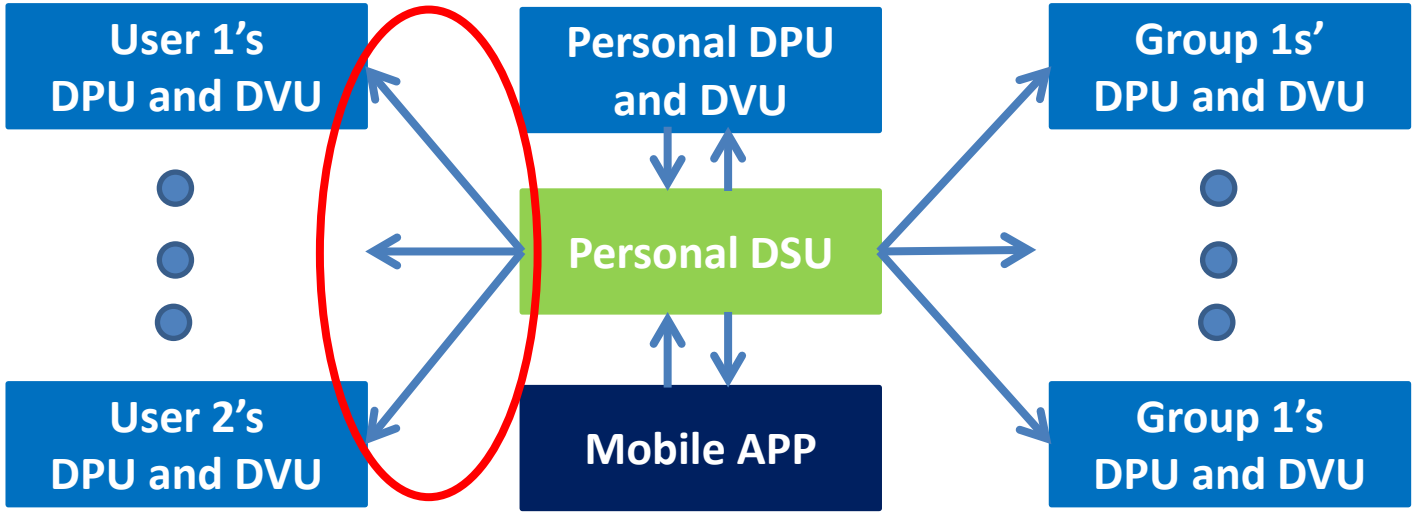


Data flow

There are there kinds of data flows:

1. A user's (Personal) DSU pulls data from the user's mobile app and (personal) DPU+DVU; mobile app and personal DPU+DVU pull data from DSU.
2. A user may authorize other users' DPUs and DVUs to read **part** of her data from her DSU.
3. The user may join some groups, in which case the user should authorize these groups' DPUs and DVUs to read **part** of her data from her DSU.

Besides these, the user wants no one else get access to her data.

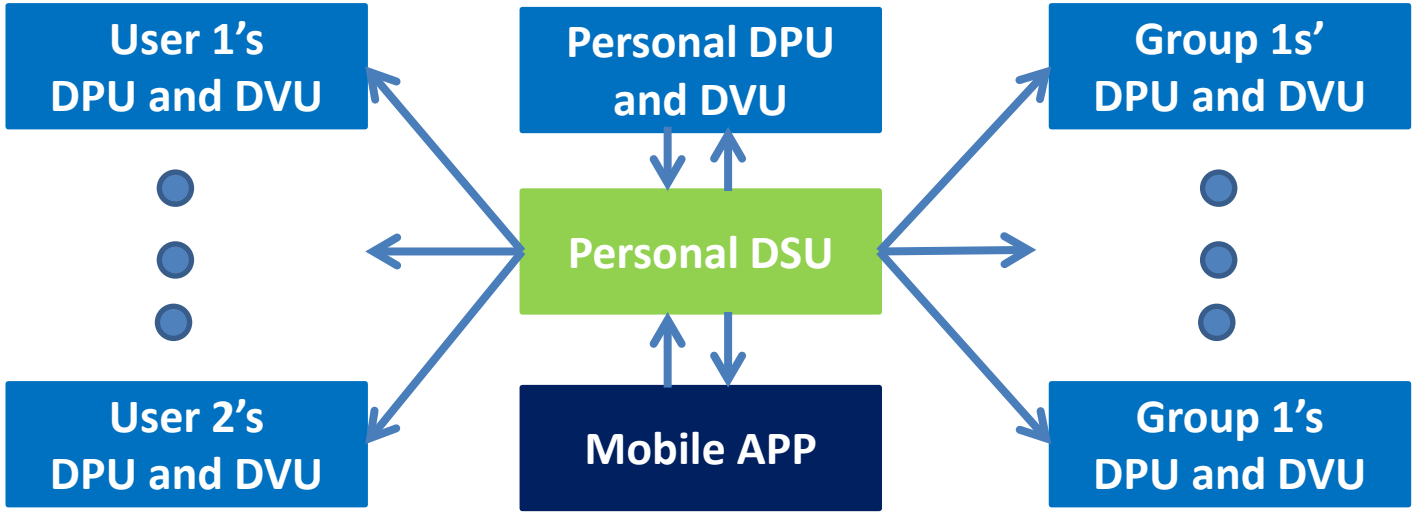


Data flow

There are there kinds of data flows:

1. A user's (Personal) DSU pulls data from the user's mobile app and (personal) DPU+DVU; mobile app and personal DPU+DVU pull data from DSU.
2. A user may authorize other users' DPUs and DVUs to read **part** of her data from her DSU.
3. The user may join some groups, in which case the user should authorize these groups' DPUs and DVUs to read **part** of her data from her DSU.

Besides these, the user wants no one else get access to her data.

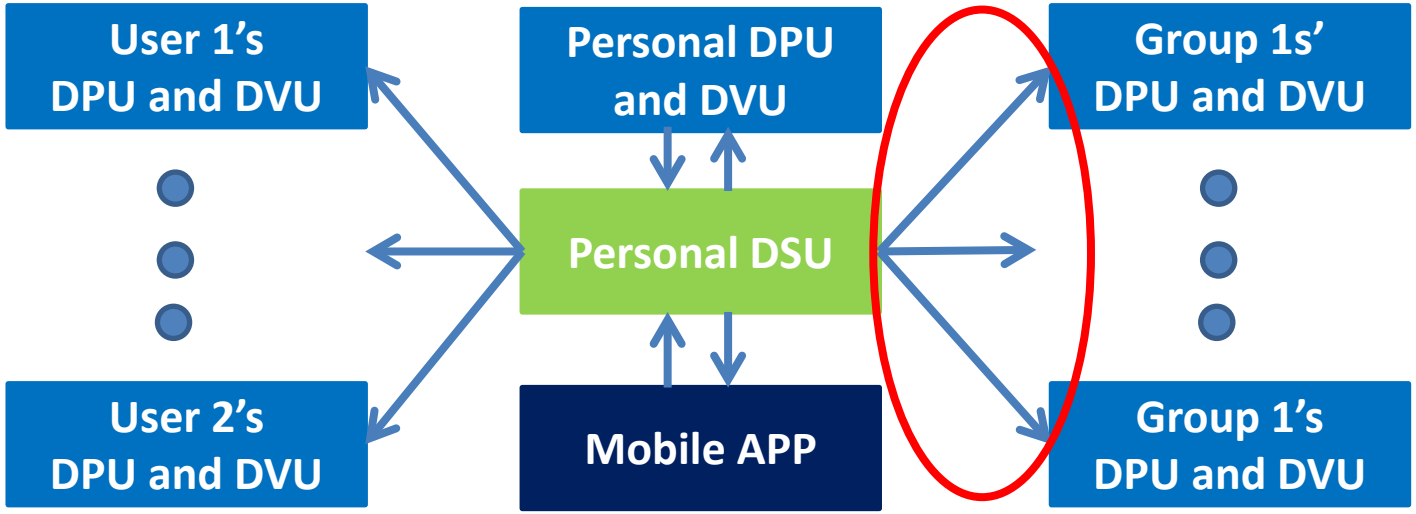


Data flow

There are there kinds of data flows:

1. A user's (Personal) DSU pulls data from the user's mobile app and (personal) DPU+DVU; mobile app and personal DPU+DVU pull data from DSU.
2. A user may authorize other users' DPUs and DVUs to read **part** of her data from her DSU.
3. The user may join some groups, in which case the user should authorize these groups' DPUs and DVUs to read **part** of her data from her DSU.

Besides these, the user wants no one else get access to her data.

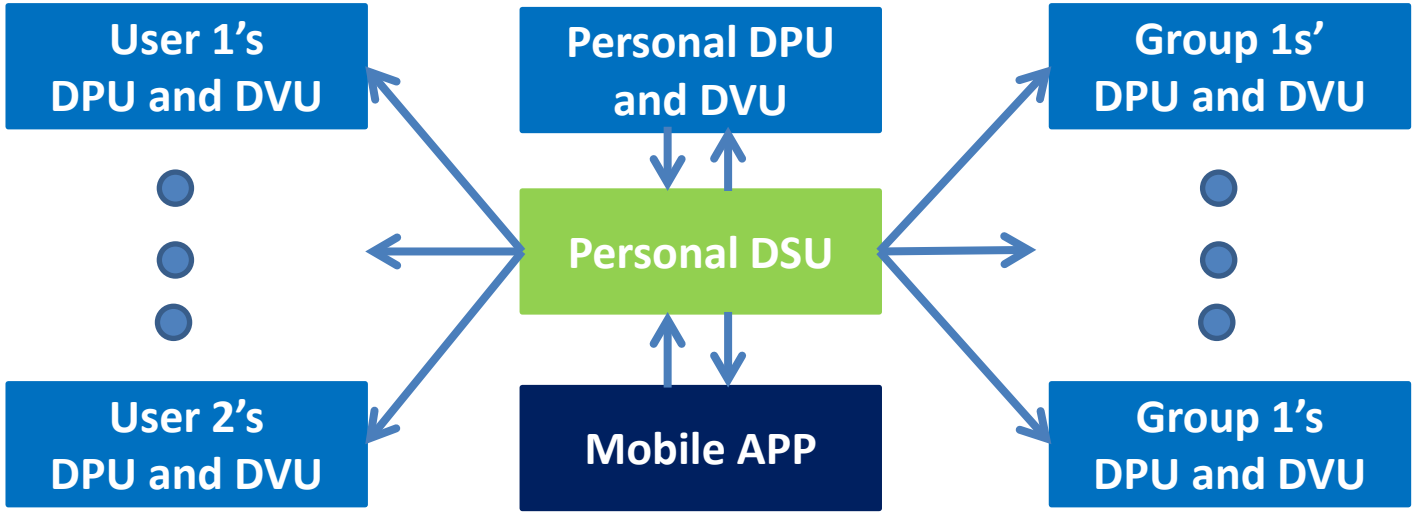


Data flow

There are there kinds of data flows:

1. A user's (Personal) DSU pulls data from the user's mobile app and (personal) DPU+DVU; mobile app and personal DPU+DVU pull data from DSU.
2. A user may authorize other users' DPUs and DVUs to read **part** of her data from her DSU.
3. The user may join some groups, in which case the user should authorize these groups' DPUs and DVUs to read **part** of her data from her DSU.

Besides these, the user wants no one else get access to her data.



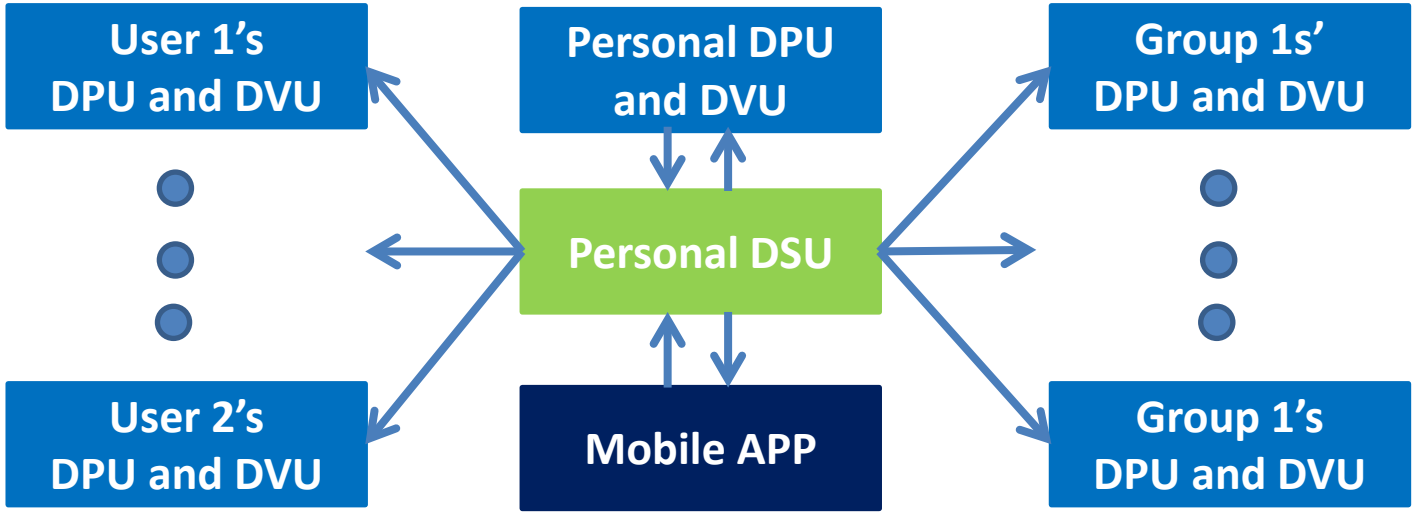
Data flow

There are there kinds of data flows:

1. A user's (Personal) DSU pulls data from the user's mobile app and (personal) DPU+DVU; mobile app and personal DPU+DVU pull data from DSU.
2. A user may authorize other users' DPUs and DVUs to read **part** of her data from her DSU.
3. The user may join some groups, in which case the user should authorize these groups' DPUs and DVUs to read **part** of her data from her DSU.

Besides these, the user wants no one else get access to her data.

Data Access Control – Encryption & Decryption



Data access control requirement

Requirement - Fine-Grained Access Control

Different data consumers should read different parts of the user's data.

But how?

1. Encrypt data for every consumer, respectively?
   If there are many consumers, the workload of encrypting is really heavy.
2. Encrypt different data chucks with different keys, and distribute these keys to consumers according to access control list?
   It's hard to decide when to update the key. Every five hours, Or every second? When I go back home from campus, should I change my key?
   How to authorize new consumers to read historical data? The data generated between 7am and 8am are encrypted with the same key, but we want to authorize new consumers to read the data generated between 7am and 7:30am.

ABE – Attribute-Based Encryption

Encrypt data only once. Each data consumer owns one specific decryption key which is generated according to her access privilege. Data consumers having different decryption keys could decrypt different data chucks.

ABE - Attribute-Based Encryption

Two kinds of ABEs: key-policy ABE and ciphertext-policy ABE

1. key-policy ABE (*Vipul Goyal, Omkant Pandey, Amit Sahai, Brent Waters*)

Data are tagged with their relevant attributes. To grant access to a portion of data, the data owner creates specific keys that embed the policy formulae determining which part of data may be accessed.

$$((10am\ -11am) \cap irl) \cup running$$

Notice: these are the attributers of the data

The consumers can decrypt the data whose attributes satisfy the formula .

2. ciphertext-policy ABE (*John Bethencourt, Amit Sahai, Brent Waters*)

Each ciphertext is bound together with a policy describing who is entitled to decrypt it. These policies are typically expressed as boolean formulae referencing a list of attributes that are embedded into the producer's secret key. The consumers' keys are generated based the their attribute.

$$(UCLA \cap (irl \cup remap))\ \cup (proffessors\ \cap NDN\ developer)$$

Notice: these are the attributes of the consumers

The consumers whose attributes satisfy the formula can decrypt the data.

Improvement

1. The combination of key-policy ABE and ciphertext-policy ABE (*Joseph A. Akinyele, Christoph U. Lehmanny, Matthew D. Green, Matthew W. Pagano, Zachary N. J. Petersonz, Aviel D. Rubin*)

They provide a design and implementation of self-protecting electronic medical records (EMRs) using the combination of these two kinds of attribute-based encryption.

Key-policy ABE is used to grant a limited access to the data.

Ciphertext-policy ABE is used for individuals whose access privileges change infrequently.

See their code: *http://code.google.com/p/libfenc/*

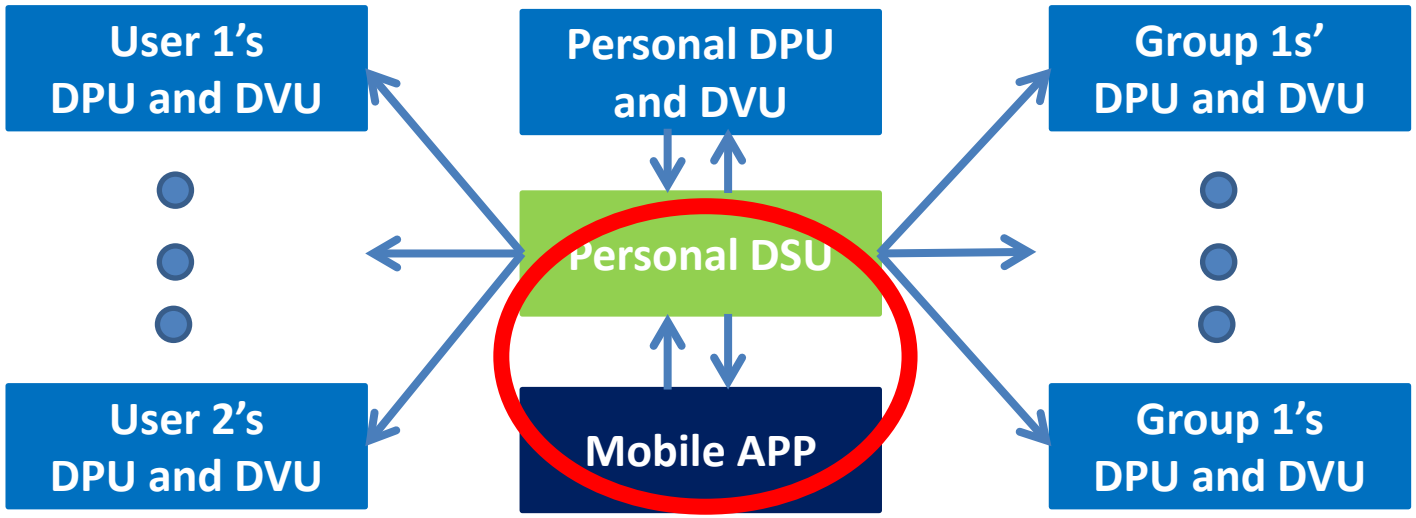
Improvement

2. Use different encryption algorithms for different data flows

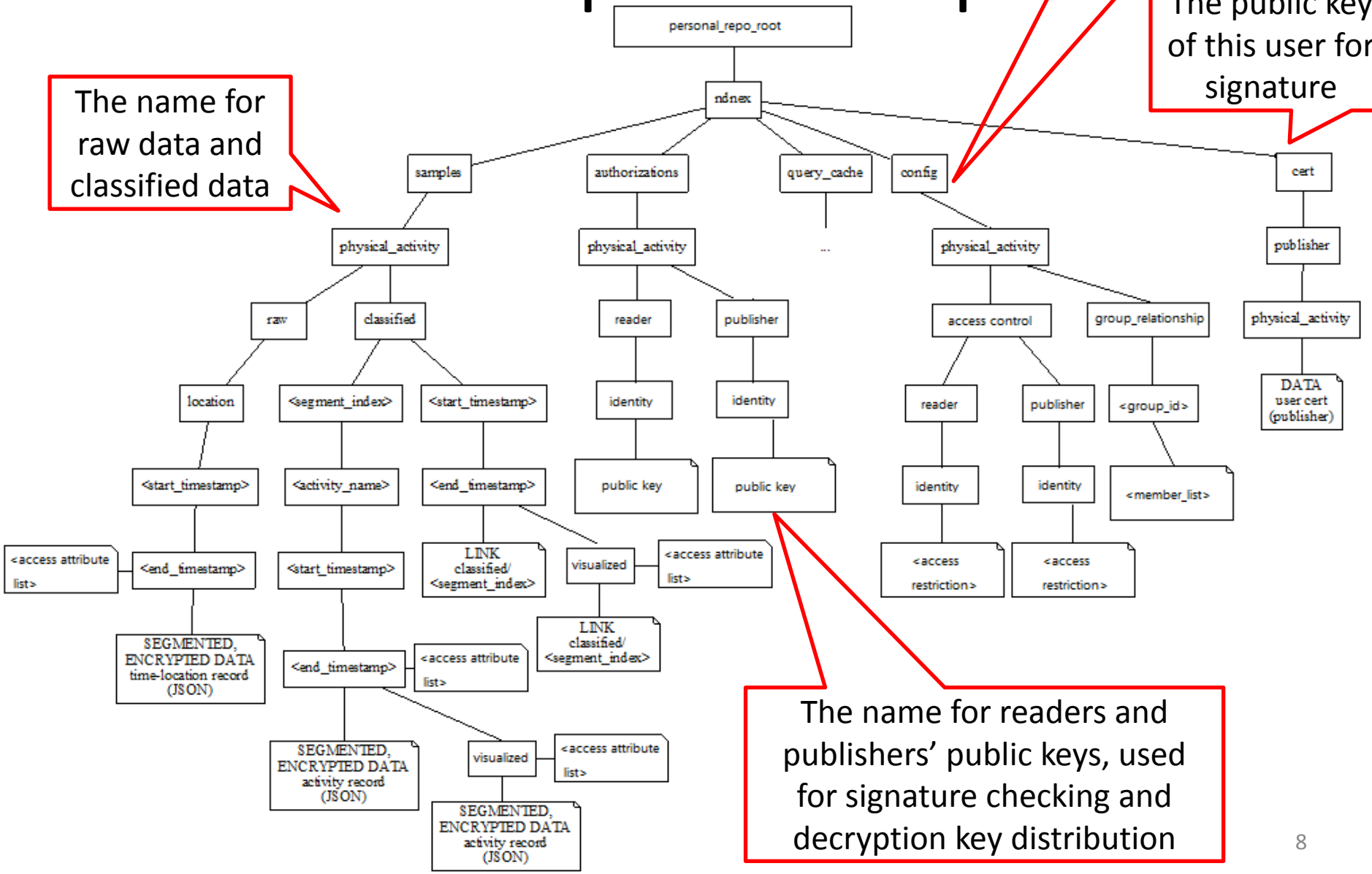
Based on the result of *Joseph A. Akinyele, etc.* the encryption and decryption processing of ABE are quite time consuming, especially for ARM CPU on mobile device. So

(1) Traditional symmetric encryption algorithm is used for data flow between personal DSU and mobile device.

(2) ABE is used to for other data flows.



Namespace for repo



Namespace and ABE

1. Encryption

Data's name prefix is the combination of data's attributes, which is used for key-policy ABE.

…/haitao/ndnex/sample/physical_activity/ucla/irl/020320151000/020320151005/…

➔ Attributes list: ucla, irl,, Feb 3 2015 10:00am – 10:05am

There are also access attributes list in the namespace specifying the authorized consumers' attributes, which is used for ciphertext-policy ABE.

…/haitao/ndnex/sample/physical_activity/ucla/irl/020320151000/020320151005/ access attributes list/$UCLA$ and (irl or $remap$)) …

➔ Attributes list: $UCLA$ and (irl or $remap$)

Namespace and ABE

2. Decryption key generation

The …/haitao/ndnex/config/physical_activity/access control/reader… branch specifies each consumer's reading privileges. The decryption key for the user is generated according to the <access restriction> (for key-policy ABE) or the consumer's attributes (embed in the user's name prefix, for ciphertex-policy ABE).

3. Decryption key distribution

When trust relationship is established, the decryption key can be encrypted by the consumer's public key, then distributed to that consumer. Consumers' public key is stored under name prefix …/haitao/ndnex/authentications/physical_activity/reader/...

Reference

- Akinyele et al. "Self-protecting electronic medical records using attribute-based encryption." (2010).
- Bethencourt et al. "Ciphertext-policy attribute-based encryption." *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007.
- Goyal et al. "Attribute-based encryption for fine-grained access control of encrypted data." *Proceedings of the 13th ACM conference on Computer and communications security*. Acm, 2006.

Thank you!