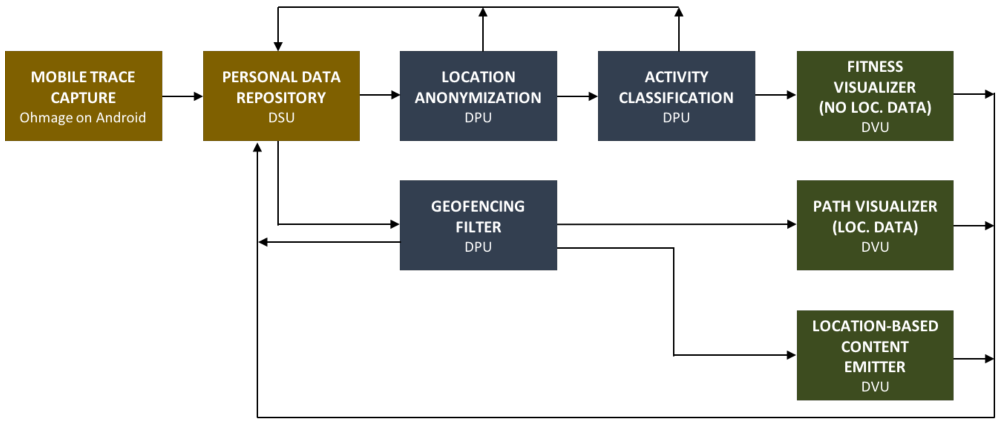

Figure 2: data flow for a single user of NDNfit

1 Open mHealth

Mobile health (mHealth) has emerged as an important market and a key area of Health IT, a national priority. Open mHealth project [1] led by Deborah Estrin (Cornell) and Ida Sim (UC San Francisco) proposes a thin waist of open data interchange standards (Figure 1) to enable an ecosystem of sensing, storage, analysis, and user interface components to support medical discovery and evidence-based care. Specifically, the Open mHealth architecture is proposed have standardized personal data vaults and health specific data exchange as the narrow waist, which provides health-specific syntactic and semantic data standards, patient identity standards, core data processing functions such as feature extraction and analytics and data stores that allow for selective, patient-controlled sharing.

The focus on data exchange as the backbone of the application ecosystem makes Open mHealth an excellent network environment to both drive and evaluate NDN. Since NDN takes data exchanging as the narrow waist of network architecture, provides a standard way to get identity and manage trust relationship, provides provenance via the signature, and secures data at generation, data owners (different from data producers) directly controls data sharing.

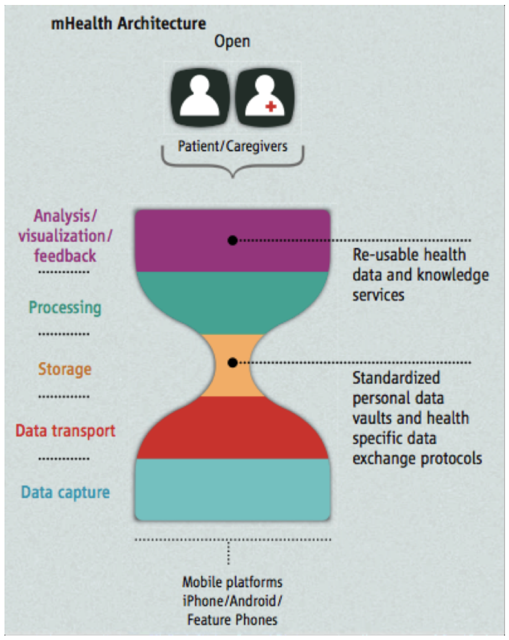
mHealth Architecture
Open
Patient/Caregivers
Analysis/ visualization/ feedback
Processing
Storage
Data transport
Data capture
Re-usable health data and knowledge services
Standardized personal data vaults and health specific data exchange protocols
Mobile platforms iPhone/Android/ Feature Phones

Figure 1: The Open mHealth architecture uses a data-centric hourglass model, where the interoperability layer ("thin waist") is based on standardized data exchange. [1]

Our high level though is to build a pilot NDN-based fitness appication and system, which is compatible with Open mHealth paradigm, to capture, process and visualize users' time-location data. Figure 2 shows the data flow for a single user who uses NDNFit to get: 1) fitness/activity metrics, 2) walking or running path visualizations, and 3) location-based content during exercise - all through the same ecosystem, but from different providers.

Nine sites are collaborating on design and development of this application:

- UCLA REMAP - application design; library support; web-based visualization; values in design.

- UCLA IRL - architecture implications; repository; library and forwarder support; trust and security
- University of Arizona - forwarder support.
- University of Michigan - trust and security.
- Tsinghua University - repository.
- University of Basel - data flow processing using NFN (Named Function Networking).
- Anyang - Ohmage capture application port.
- WUSTL - testbed support.
- UCSD - project management support.

Last year, we developed an overall application architecture (DO YOU HAVE THE FIGURE LISTED ON PAGE 12 OF LAST YEAR'S REPORT?), including a trust model and encryption-based access control design, an approach to publisher mobility support, and an initial direction for distributed computation based on the University of Basel's Named Function Networking research [2]. We also began to define autoconfiguration support, the NDN equivalent of DHCP needed for data publishers.

This year, we revised the namespace design, implement the NDNFit Android mobile application, designed and implemented the identity management Android application, designed and implemented data transportation protocols, designed and implemented NFN integration mechanism, and implemented name-based access control (NAC) [3]. We also started to revise the previous autoconfiguration support mechanism. The most exiting part is that we have built a demo system running on the testbed which has a single user, a single DSU and two DPUs (one NFN DPU and one native NDN DPU).

1.1 Namespace

Figure 3 shows the second version of the proposed data namespace. In the namespace, group and DPU, DVU branched are listed there for future expansion. User data brach is designed in detail.

Since name-based access control is adopted in NDNfit, the namespace should be designed following its requirement. According to NAC, a user should publish data under data brach and publish keys under read branch. The first implementation of NDNFit captures and processes time-location data, and in most of the real use cases, data is fetched by time range, so timestamp is designed to be the last component of data names.

1.2 NDNFit Android Application

Based on the previous analysis, Anyang University and UCLA IRL have built a NDNFit Android data capture application shown by Figure 4.

This application captures time-location data, produces Data packets and temporarily stored these Data locally. It uses the data transportation designed follwing to "upload" data to the DSU.

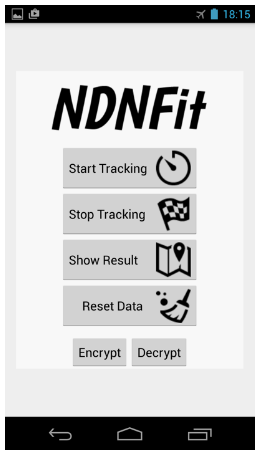

Figure 4: NDNFit Application UI

1.3 Identity manager

ZHEHAO, CAN YOU PUT CONTENTS IN THIS SUBSECTION?

1.4 Data transportation protocol

The most difficult part of data transportation is from mobile to DSU, data transportation between other components can reuse (part of) the protocol designed for it.

There are three problems need to be solved when design the data transportation protocol for mobile to DSU: (1) Mobile device are not always

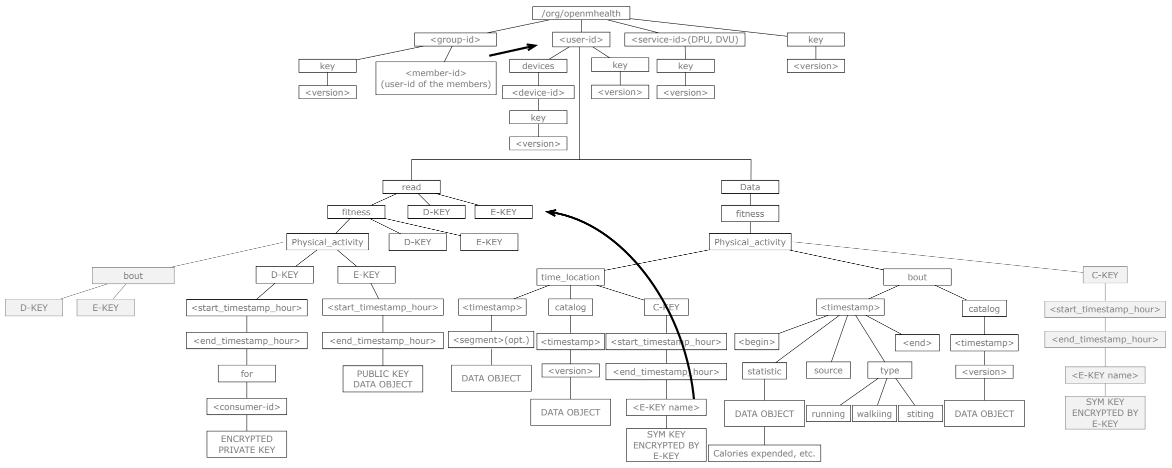

Figure 3: NDNFit Namespace, version 2.

online, so DSUs should know when to fetch data; (2) Mobile capture data are named with a random timestamp component, DSUs should know the exact names before fetching data; (3) Mobile devices have limited resource such as storage and power, so they should know when to delete locally stored data.

To solve the first problem, a mobile device needs to send signal Interest to a DSU whenever it reconnects the Internet, then the DSU can start to fetch data from the mobile device. One level of direction - catalog - used by DSUs to learn the names of captured data from mobile devices is introduced to solve the second problem. A mobile device packages captured data names into a catalog every 10 minutes, which is to say, a catalog contains the names of data captured in the past 10 minutes; and catalogs are named with some deterministic timestamp as the last component. To confirm that a DSU have fetched all the data whose names are packaged in the same catalog, a mobile device sends a confirmation request Interest right after it sends out the catalog. After the DSU gets all the data, it will send confirmation Data back to the mobile device.

## 1.5 NFN integration mechanism

## 1.6 Name-based access control (NAC)

This will be covered by other section?

## 1.7 Progress towards milestones

We summarize progress toward our proposed milestones for this specific environment:

- *Review limitations in current IP-based architecture for Open mHealth needs. (Y1)* Done. Included conversations with Open mHealth architects and developers, code review and porting of existing applications, and literature review.

- *Design namespace, repository, trust and communication model for use cases, e.g., diabetes or PTSD treatment (Y1; updated in Y2)* Updated the initial design.

- *Repository implementation providing backing storage for prototype applications. (Y1)* Done.

- *Integrate named data networking into the Ohmage mobile data collection framework. (Y2)* Done.
- *Pilot user-facing application using NDN, for testing by Open mHealth team. (Y2)* Built an initial demo system.

1.8 Findings about network architecture

- NFD autoconfiguration, prefix registration and validation, and identity management are the keys to support mobility.
- Name confidentiality is very important for protecting users' privacy. However, name confidentiality seems to be confilict with data caching, and it's hard to find a perferct solution.
- How to encrypt data while share data effectively among a ad-hoc group is still a problem worth careful considering.

1.9 Citations

NDNComm 2015 Poster Session: Haitao Zhang (UCLA), NDNFit architecture and progress

References

[1] Deborah Estrin and Ida Sim. Open mhealth architecture: an engine for health care innovation. *Science*, 330(6005):759–760, 2010.

[2] Christian Tschudin and Manolis Sifalakis. Named function networking.

[3] Yingdi Yu, Alexander Afanasyev, and Lixia Zhang. Name-based access control. *Named Data Networking Project, Technical Report NDN-0034*, 2015.