NDN Network Environment: Open mHealth

Jeff Burke

December 22, 2015

Abstract

The abstract.

1 Introduction

As part of the NSF-supported NDN "Next Phase" research from 2014-2016, the NDN project team has selected two network environments, **Open mHealth** and **Enterprise Building Automation & Management**, and one application cluster, **Mobile Multimedia**, to drive our research, verify the architecture design, and ground evaluation of the next phase of our project. The two environments represent critical areas in the design space for next-generation Health IT and Cyberphysical Systems, respectively. They also extend work started in the previous NDN FIA project on participatory sensing and instrumented environments to focus on specific application ecosystems where we believe NDN can address fundamental challenges that are unmet by IP. Based on the successful initial results of previous NDN research, we have identified Mobile Multimedia as an application area of cross-cutting relevance, motivated not only by the network environments above but our team's desire to further develop NDN by using it for our everyday communication.

This technical report provides background information on the **Open mHealth** network environment including key application challenges faced using IP and describes the design for a pilot application that the NDN team is building. It serves as the primary design document for this application.

1.1 Open mHealth Background

Mobile health (mHealth) has emerged as both an important commercial market and a key area of Health IT, a national priority. The 2013 mHealth Summit will host over 4,500 participants. Recent surveys suggest there are over 13,000 health-related apps available to Apple iPhone users, and over 6000 for Android users [**?**]. The Internet's role as a critical enabler of mHealth will grow further over the next decade.

Mobile health continues our work in the first NSF-supported NDN project on participatory sensing. To explore mHealth as a network environment for NDN, our team will collaborate with the Open mHealth project [**?**] led by Deborah Estrin (Cornell) and Ida Sim (UC San Francisco). Within the many applications of mobile technology to health, Open mHealth focuses on leveraging the public's everyday mobile devices (cell phones, tablets, etc.) to extend evidence-based interventions beyond the reach of traditional care and thereby improve disease management and prevention. For example, mobile applications exist or have been proposed to manage: pre-

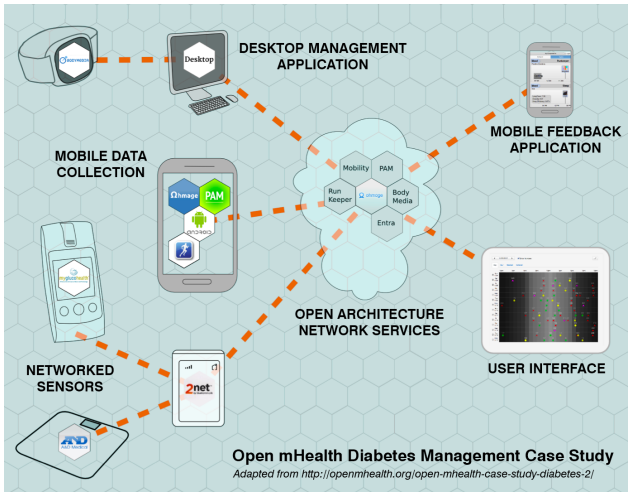


Figure 1: Networked data producers and consumers in a diabetes management case study from the Open mHealth team, who promote interoperability between mobile health components via community-standardized data exchange. [**?**]

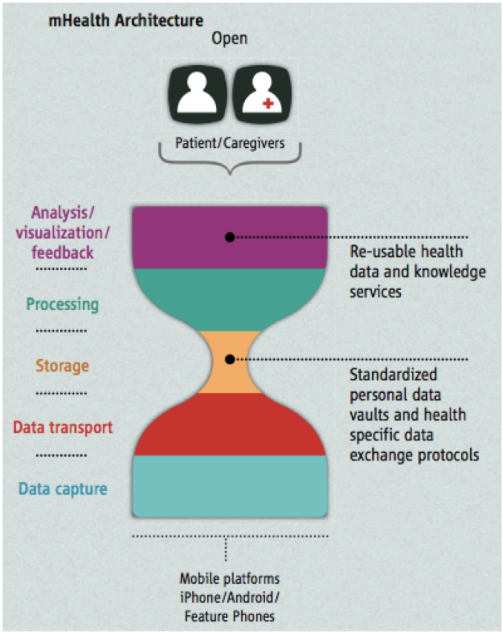


Figure 2: The Open mHealth architecture uses a data-centric hourglass model, where the interoperability layer ("thin waist") is based on standardized data exchange. [**?**]

and post-natal care of mothers [**?**]; diabetes [**?**, **?**];
everyday activity in stroke patients and others with
chronic disease [**?**]; and community exposure to environmental pollutants [**?**].

The Open mHealth team envisions that the Internet will interconnect 1) data capture, 2) secure storage, 3) modeling and analytics, and 4) user interface components to create a modular, layered sense-making framework. Such applications will use low-level state classifications of raw data (e.g., estimated activity states such as sitting, walking, driving from continuous accelerometer and location traces) to derive mid-level semantic features (e.g., total number of ambulatory minutes, number of hours spent out of house), that can be mapped to behavioral biomarkers for specific diseases (e.g., chronic pain, diabetes, multiple sclerosis, fatigue, depression, etc) [**?**].

For example, Figure 1 shows a network of open components for self-care of diabetes, which affects 25.8M people in the U.S., less than half of whom meet recommended standards, such as blood glucose index levels, for managing their own health. Type 1 diabetics continuously self-monitor blood glucose and insulin levels, and other important factors such as diet and exercise. Many developers are exploring mobile health technologies to assist self-monitoring and diabetes management, since almost all patients have access to mobile health capable technology [**?**]. But such applications often have proprietary or siloed designs that inhibit data exchange, e.g., data streams from apps for blood glucose and physical activity do not easily integrate, a missed opportunity to provide more comprehensive analysis and coaching to the patient and more complete longitudinal data to providers.

The Open mHealth team instead advocates an interoperable, Internet-inspired approach. They propose a thin waist of open data interchange standards that will enable an ecosystem of sensing, storage, analysis, and user interface components to support medical discovery and evidence-based care. In the same way that the Internet's IP layer enabled innovation and interoperability among distributed devices, they believe a common and open approach to mHealth data exchange will encourage the emergence of a market-supported, patient-centered landscape of innovative health applications. Central to this vision is patient-controlled, privacy-aware data exchange across device, component, and application boundaries. 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.

1.2 Challenges of IP

- Locating endpoints to source/sink data.
- Key pain point is OAuth 2.0: Implementation relies on this doesnt scale to the DPU model and has numerous problems. Quickly identified by

Others...

1.3 Fit of NDN to Application Domain

The following ideas about why NDN fits Open mHealth well should be validated through experimentation and directly influence the design goals found later in this document.

- Like NDN, Open mHealth focuses on **named data as the "thin waist" of interoperability**. As discussed above, the Open mHealth architecture focuses on data exchange rather than system interoperability.
- **Reduction in complexity for request-response architecture** should make interoperability across a wide ecosystem easier, and support low-capability devices directly - given that many apps run on a variety of types of devices. Using NDN rather than IP enables Open mHealth applications to be coded using data naming directly rather than having to create abstractions to translate data names to IP-based hosts providing services.
- **Distributed storage is straightforward to implement on NDN**. Could drive a new data-diffusion focused model for this application. We will explore how the application's storage architecture can be simplified and map closely to network architecture. NDN naturally supports distributed storage, which can ease the burden of fault-tolerance and load-balancing in large networks, reducing cost-of-entry and fostering innovation.
- **Data-centric security requirements of Open mHealth are a good match for NDN**, and could be a major improvement over a current pain point OAuth, in terms of ease of development and overall security. Because NDN does not rely on perimeter- or channel-based security, it can promote global health data ecosystems rather than previous walled garden approaches, if the ecosystem is designed directly. In the long term, this shift has direct relevance for opt-in epidemiological studies in public health, by enabling researchers to draw from large populations who have elected to share some data with them.
- **Intrinsic disruption tolerance and multi-path support** are a good fit for mobile devices if challenges of mobile publishing can be addressed.

1.4 Collaboration

The following team is collaborating on the 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.
- U. Arizona - forwarder support; NFD Android port.
- U. Michigan - trust and security.
- Tsinghua - repository.
- Basel - data flow processing using NFN.
- Anyang - Ohmage capture application port.

Additionally, we will partner with the Open mHealth team – both its leadership and developers – to understand the requirements and current state-of-the-art in this network environment, as well as limitations they face from the current Internet architecture. We will pick one or more applications (e.g., diabetes management or post-heart attack health management) that are representative of the envisioned ecosystem, and port existing software of the Open mHealth team to the NDN architecture. We will use an interactive development process, soliciting regular feedback from the Open mHealth team.

1.5 Proposed Milestones 2014-2016

The following milestones were proposed to NSF for this network environment.

- Review limitations in current IP-based architecture for Open mHealth needs. (Y1) *This review will be incorporated into this document.*
- Design namespace, repository, trust and communication model for use cases, e.g., diabetes or PTSD treatment (Y1; updated in Y2) *The design of the initial NDNFit use case will be included in this document.*
- Repository implementation providing backing storage for prototype applications. (Y1) *The initial repository design will be described in this document.*
- Integrate named data networking into the Ohmage mobile data collection framework. (Y2) *Ohmage is a reference application for the NDNFit application described in this document, and we plan to explore this interoperability in 2016.*
- Pilot user-facing application using NDN, for beta testing by Open mHealth project team. (Y2) *The primary user-facing application is NDNFit, described in this document.*

2 Related Work

2.1 Suggested reading

- Estrin, Deborah, and Ida Sim. "Open mHealth architecture: an engine for health care innovation." Science 330.6005 (2010): 759-760.
- Diabetes Case Study document from openmhealth.org
- Hicks, John, et al. ohmage: An open mobile system for activity and experience sampling. CENS Technical Reports 100: 125, 2010.
- Kang, J., Shilton, K., Estrin, D., Burke, J., Hansen, M. "Self-surveillance privacy." Iowa L. Rev. 97 (2011): 809.

See also Deborah's TEDMED talk: `https://www.youtube.com/watch?feature=player_embedded&v=lAEhSGYEHWU`

2.2 Model Applications

Several existing applications and application frameworks serve as excellent models for the Open mHealth pilot application. They are briefly reviewed below.

2.2.1 Open mHealth

Ohmage, etc.

2.2.2 Lifestreams

"Smartphones can capture diverse spatio-temporal data about an individual; including both intermittent self-report, and continuous passive data collection from onboard sensors and applications. The resulting personal data streams can support powerful inference about the user's state, behavior, well-being and environment. However making sense and acting on these multi-dimensional, heterogeneous data streams requires iterative and intensive exploration of the datasets, and development of customized analysis techniques that are appropriate for a particular health domain.

"Lifestreams is a modular and extensible open-source data analysis stack designed to facilitate the exploration and evaluation of personal data stream sense-making. Lifestreams analysis modules include: feature extraction from raw data; feature selection; pattern and trend inference; and interactive visualization. The system was iteratively designed during a 6-month pilot in which 44 young mothers used an open-source participatory mHealth platform to record both self-report and passive data about their diet, stress and exercise. Feedback as participants and the study coordinator attempted to use the Lifestreams dashboard to make sense of their data collected during this intensive study were critical inputs into the design process. In order to explore the generality and extensibility of Lifestreams pipeline, it was then applied to two additional studies with different datasets, including a continuous stream of audio data, self-report data, and mobile system analytics. In all three studies, Lifestreams' integrated analysis pipeline was able to identify key behaviors and trends in the data that were not otherwise identified by participants." [**?**]

2.2.3 HumanAPI

HumanAPI Also OAuth 2.0 Apparently similar objectives to Open mHealth

2.2.4 Ginger.io

Ginger.io Platform for predictive modeling

2.3 Previous work by the NDN team

The primary piece of past work by the NDN team is the *personal data cloud* effort. Within the participatory sensing application area of the previous NDN FIA project, we extended the concept of a host-centric "personal data vault" developed at UCLA and USC [**?**] to create a geographically distributed *personal data cloud (PDC)* using NDN.

This prototype went through three iterations reflecting increasing understanding of how to develop applications using NDN. The first version implemented data collections, key management, storage, data transfer and authentication/setup phases in a way largely analogous to TCP/IP based applications. A second revision integrated the PDC architecture into a deployed participatory sensing application at the Center for Embedded Networked Sensing, called Ohmage [**?**].

The most recent revision transitioned to use the new Sync primitive for transferring content between entities, and removes much of the session-like semantics initially present[1]. This experience will inform work with the Open mHealth team as they continue to develop pilot applications using Ohmage and similar platforms. With the emergence of the Sync primitive and our recently developed ChronoSync synchronization library [**?**], as well as lighter-weight mobile client options based on NDN.JS [**?**], we plan to explore end-to-end applications data dissemination via NDN.

3 Pilot Application: NDNFit

3.1 Introduction

Our pilot application for this network environment is **NDNFit** (NDN-Exercise) and related components required by the design approach, such as a user-facing **Identity Manager**.

[1] https://github.com/remap/PDC-SYNC

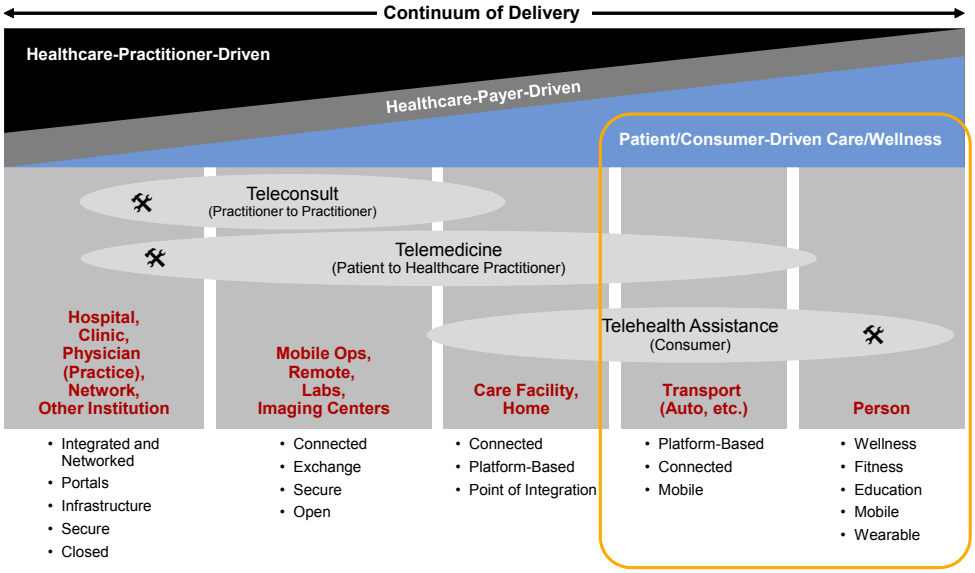


Figure 3: Focus of Open mHealth network environment shown in yellow box. Figure from Gartner, 2013.

NDNFit is a mobile physical activity monitoring application (fitness tracker) that supports location-based notifications / content push. Commercial parallels include Nike+, Fitbit, Endomondo (see Figure 4), etc. It also is a non-proprietary ecosystem for consumer physical activity data. Following the philosophy of Open mHealth, our objective is to create NDNFit, which is perceived by the user as a single application, through a **simple ecosystem of composable services** rather than siloed application. In this way, it is to act as an example of interoperating components of an Open mHealth ecosystem. This places additional requirements on the design.

Supporting physical activity is both a critical part of building healthy communities and a key retail market. Per the original proposal, we focus on a consumer-facing application, as shown in Figure 3, rather than clinical applications or other with formal connections to the healthcare system.

3.2 End-user features

NDNFit targets the following features for the end-user:

- Capture and report walking, jogging, and running activity. Capture will occur on a mobile device and reporting will occur through a mobile-friendly website.
- Calculate and report activity metrics based on GPS and accelerometer data for both automatically and self-identified rounds of exercise.
- Support comparison within ad-hoc and formal groups or teams.
- Provide location-based content push during the exercise, which can be used for health, entertainment, local, and team-related content.

4 Testbed

All components under development as part of NDNFit should connect and communicate through the NDN testbed. Some components may also support ad-hoc communication, but this is not a requirement.

An instance of TCP/IP Ohmage (client and server) is now up and running thanks to Haitao in Lixias group at UCLA.

Figure 4: Endomondo commercial fitness tracker. `https://www.endomondo.com/`

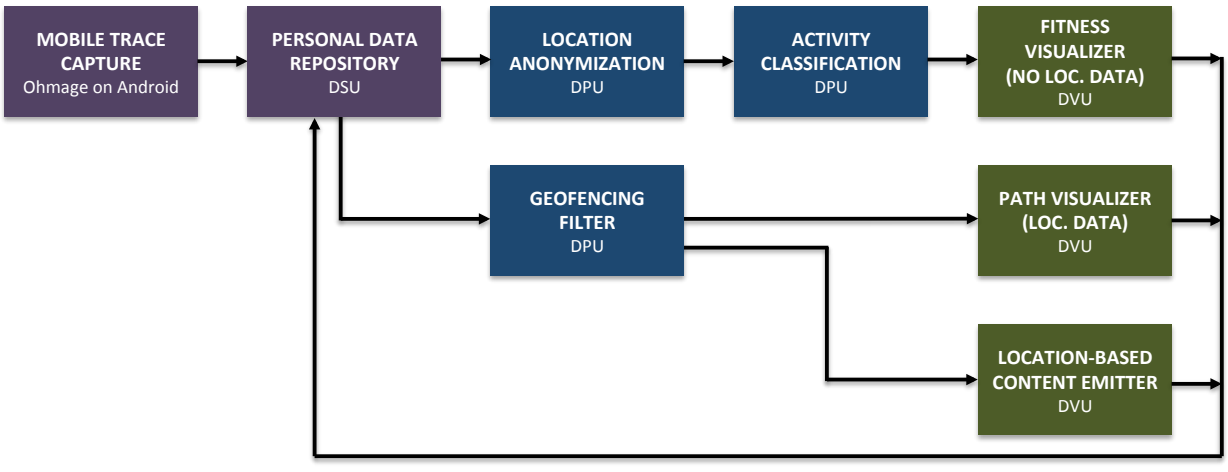


Figure 5: Conceptual block diagram showing data flow.

5 Design of NDNFit

Fundamentally, NDNFit should follow the Open mHealth paradigm, but adapting its REST-based communication model to a data dissemination approach using NDN. The application must be constructed from an ecosystem with each type of component envisioned by the Open mHealth team, including 1) data capture, 2) secure storage, 3) modeling and analytics, and 4) user interface components to create a modular, layered sense-making framework. Each component could be provided by different service providers at different stages in the processing chain, rather than a siloed application.

The design should rely as much as possible on basic NDN primitives (hierarchical naming, interest/data exchange, sync, repositories, keys-as-data, etc.) as possible, rather than designing new protocols. Given that Open mHealth already envisions a data-centric thin waist in their ecosystem, NDN provides much more relevant functionality at the network layer than IP. So basic solutions in NDN have much more direct impact on the scalability, security, and ease of development; we need not build up additional layers on IP to get near the app challenges.

The application architecture should be consistent with the discussion above and incorporate reasonable knowledge of the "cutting-edge" of participatory sensing projects (e.g., Mun et al, 2009) and related work in the commercial sector.

The basis for design for all components should include the following:

Basis for Design

- Review of **NDN research goals and related requirements** as described above.
- Review and evaluation of Ohmage reference application and open mHealth schema.
- Review high level motivation of Estrin & Sim, 2010 and other references above.
- Review case studies, including those on Ohmage website and in the Appendices.
- Review past CENS/UCLA participatory sensing research in activity classification,self-surveillance privacy, mobile phone based data collection.

Areas of concern NDNFit requires specific design of the following, as an example of the Open mHealth network environment.

- Namespace / schema
- Repository / storage
- Service composability
- Authentication / identity assurance
- Data provenance
- Access auditing
- Mobile publishing
- Legal requirements for success

Design Goals

- **Interoperable, Internet-inspired data exchange** as the backbone of the application ecosystem.
- **Thin waist of open data interchange standards** that will enable an ecosystem of sensing, storage, analysis, and user interface components to support medical discovery and evidence-based care
- **User-centric, privacy-aware data exchange** across device, component, and application boundaries

- Imported from Open mHealth:
  - Data-centric rather than service-centric interoperability. ⇒ **Focus on data namespace design.**
  - Distributed architecture of Capture, DSU, DPU, DVU. ⇒ **Implement data flow approach in NDN.**
  - End user focus (not hospitals, doctors, etc.) ⇒ **Consumer app deployment scenario.**
  - User-centric privacy approach. ⇒ **Need to inform user of choices, data flow.**
  - Encrypted communications. ⇒ **Encryption-based access control, name encryption.**
  - Mobile publishing. ⇒ **Use as our driver to solve this oft-cited challenge.**
- In contrast to Open mHealth:
  - REST/HTTP is not used. ⇒ **Move away from RPC call model** and carrying state in Interests, towards data dissemination.
  - Host-based endpoints for services. ⇒ **Focus on data dissemination model** and NFN style **distributed processing**.
  - OAuth authentication ⇒ Need new identity / authentication approach.
  - Single storage "location" ⇒ **Distributed, "personal" repositories**.
- **mHealth Reality Check**. Finally, each component developer should consider the following "reality check" questions (and their implicit design goals) for all mHealth applications.[2]
  - **Are your systems interoperable?** Estrin & Sim in Science, 2010. Open mHealth.
  - **Are you using open standards?** WHO, 2013. eHealth unit.
  - **How will you evaluate?** Greenhalgh et al. in BMC Med Res. Methodology, 2011. Realist and meta-narrative evidence synthesis.

5.1 Specific Requirements

5.1.1 Naming

Our research objective is to see how well basic NDN primitives, such as Interest-Data exchange and Sync, can support the application, proposing new primitives and/or designing new application-level approaches where needed.

- **Namespace design and data payload format should be adapted as directly and consistently as possible from the Open mHealth reference platform and schema library.**
- Basic Interest-Data exchange and Sync should be used wherever possible, for example:
  - Consumers should be able to easily access raw and processed data for a certain time period by issuing simple Interests for the appropriate names.
  - Consumers should be able to efficiently read data sequentially, also by issuing simple Interests for the appropriate names.
- The approach to distributed processing should adapt the Named Function Networking concept for distributed processing (Open mHealth Data Processing Units (DPUs).
- Data will include:
  - Raw time-location data (GPS, accelerometer) from mobiles.
  - Successive rounds of processing that, for example:

[2]From the PLOS Medicine Editors. "A reality checkpoint for mobile health: three challenges to overcome." PLoS Medicine 10.2 (2013).

    - Generate classified activity data that follows the Physical Activity JSON schema and perhaps other related schemas. (This happens at client-side in Ohmage Mobility but could happen at a DPU.)
    - Identify / segment "bouts"of physical activity or exercise.
    - Add features to a bout from DPUs to the existing store.
  - End-user configuration information
  - Identity and trust related data

5.1.2 Trust and security

Our research objective is to see how well the NDN architectural mechanisms fit into security requirements, and propose new ones where necessary.

- Critically, the application's approach to **identity and trust management** scenario should emerge from the notion of NDNFit as consisting of interoperating components in an ecosystem, not a silo'd application.
- All data payloads should be encrypted.
- Name encryption should be explored as an advanced feature in this application.
- The ecosystem must provide granular access control over various components of the data namespace in particular, raw location data.
- Doing better than Oauth2 for securing distributed processing is important.
- Use passive key publication approach (rather than active broker services) if possible, though tradeoffs between the two solutions should be explored.
- If possible, we should support different identities relative each part of the system: collection, processing, and visualization, such as:
  - Collection: User may publish data to serve multiple applications, but doesnt want them to be able to conspire / correlate that they are the same user.
  - Processing: Design should provide the minimum possible information to the processing components about user identity.
  - Visualization: Visible face of the app to the user.

5.1.3 Storage in the network

Our research objective is to design one or more repository implementations that support application-specific requirements while preserving as many basic NDN conventions (e.g., versioning, segmenting, etc.) as possible.

- NDN-enabled "repos" should be used wherever persistent storage is needed in the application, including: 1) storage of sensed data on the mobile capture device; 2) a "cloud-based" (or home) personal data repository; 3) temporary storage for processing blocks.
- Each user may choose a different storage provider, though we may only have one option in the initial implementation
- Repos must support encryption-based access control.
- New legal / economic relationship between the players

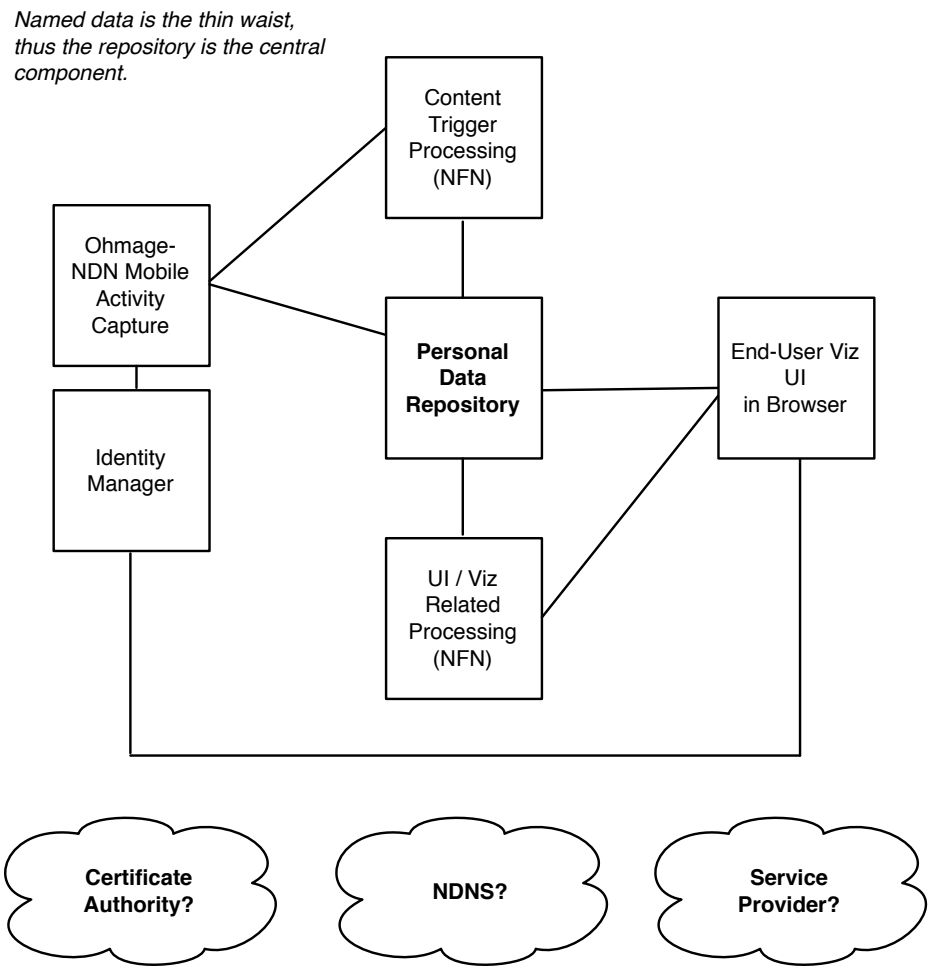


Figure 6: Draft NDNFit ecosystem design.

5.2 NDNFit Ecosystem components

First, we introduce the components of the NDNFit application ecosystem, then we discuss overall design strategies for trust, naming, etc.

The NDNFit user interacts with at least four pieces:

- Mobile capture application
- Website for visualization / review
- Personal data repository service
- Identity manager (newly envisioned mobile component)

5.2.1 Ohmage-NDN Mobile Activity Capture

Responsible team members Anyang University.

Function Capture location and activity information, publish.

Approach Initially, we will only support Android. Port NFD, tools. Provide NDN-CCL and NFD support for Android.

Mobile capture application is one of two primary user interfaces.

Initial analysis of Ohmage mobile client communication completed by Prof. Euihyun Jungs group from Anyang Univ.

From Basel: Today, we had an internal presentation from the Psychology Dept which explained their use of the Ohmage system that is running in Basel. Although NDNFit comes too late for being applied in ongoing projects, these projects are a good source of insights about user requirements for our future NDNFit system, especially the aspect of data sovereignty.

Building Block: Ohmage & Mobility `http://ohmage.org/` We are going to use the Ohmage Android client (including the Mobility module) with only the in-app storage and communication changed to NDN. See (Tangmunarunkit, H., et al., 2014) and also the papers on the Ohmage website. Note that Mobility generates activity classified data that may not require the Activity Classification DPU in the initial version.

ohmage is an open-source participatory sensing technology platform. It supports: 1) expressive project authoring; 2) mobile phone-based data capture through both inquiry-based surveys and automated data capture as well as temporally and/or spatially triggered reminders, 3) data visualization and real-time feedback; privacy respecting data management; and 4) extensible data exploration.

Tangmunarunkit, H., et al. "Ohmage: A General and Extensible End-to-End Participatory Sensing Platform, ACM Trans. on Intelligent Systems and Technology (in submission), UCLA CS Technical Report 140015. (Used in 20 projects.) `http://web.ohmage.org/~hongsudt/pub/ohmage_ucla_140015.pdf`

5.2.2 Personal Data Repository

Responsible team members Design led by Jianxun in Dan Peis group at Tsinghua.

Function Provide Open mHealth Data Storage Unit (DSU) functionality. (Also, "personal data vault".) Personal in that it is controlled by the end user. NOT necessarily a home repository. More realistically, a storage service with fiduciary responsibility to protect the data of the end-user per Kang et al.

Approach One or more new NDN repo designs supporting a hierarchy of storage needs: mobile device, user private repository, temporary processing storage. Storage at:

- The mobile device itself.
- A personal data repository (which may be distributed).
- Temporary storage for processing and visualization components.

Current plan: implement in Java using jNDN, for Android support.

Reference: Open mHealth Data Storage Unit (DSU) Design `https://github.com/openmhealth/developer/wiki/DSU-Overview`

The Open mHealth DSU (Data Storage Unit) API Specification is an open specification for unified information sharing across disparate data streams. The idea is simple: create an easy-to-understand set of APIs that allow siloed data stores to share information. Third-party applications that understand this API specification can then create a single set of tools to access data across any of the servers.

Building Block: Personal Data Vault Reference Derek's work here?

Mun, Min, et al. "Personal data vaults: a locus of control for personal data streams." Proceedings of the 6th International Conference. ACM, 2010.http://remap.ucla.edu/jburke/publications/Mun-et-al-2010-Personal-Data-Vaults.pdf

Kang, J., Shilton, K., Estrin, D., Burke, J. "Self-surveillance privacy." Iowa L. Rev. 97 (2011): 809.http://escholarship.org/

5.2.3 Distributed Processing Blocks

Responsible team members Basel

Function Goal here is to have a few representative components implemented using NFN-style approach, not exactly the processing blocks listed above, necessarily. Start with GeoFencing, as activity classification is currently handled in the Mobility portion of Ohmage. But, could also consider other application-specific processing ideas.

Initially, focusing on location-based triggers (geofencing) - to trigger location-based content.

Approach Ideally, provide composable data flow inspired by Google Cloud Dataflow, Apache Spark, etc.

Web-based front end using NDN-JS with access to geofenced location information, providing location-specific content back to the mobile user.

Related to vehicular networking work.

Reference: Open mHealth Data Processing Unit (DPU) Design `https://github.com/openmhealth/developer/wiki/Open-mHealth-and-Data-Processing`

DPUs are stateless modules that input and output data. They are designed to be embedded in other software or called remotely. They do not produce anything directly visible, but are the brains and muscles of an application. The concept of a DPU is inspired by the Unix Philosophy of creating small functional tools that can be chained and reused, rather than a single large application.

Building Block: Named Function Networking `http://www.named-function.net/`

Names serve to access and invoke functions, which incidentally can produce passive content once it is needed. New questions arise from this point of view, namely how the network organizes the flow of functions, which brings us squarely into active networking turf.

Content Source: Trails Database `http://archinect.com/news/article/111897927/tour-los-angeles-history-w`

The LASHP Trails Mobile Website gives residents and visitors to Northeast Downtown Los Angeles site-specific access to a dynamic combination of historic information and health-related activities along urban trails starting and ending at the Los Angeles State Historic Park.

5.2.4 Visualization Interface

Responsible team members UCLA REMAP

Function Provide basic visualization of fitness data.

Approach Start with Ohmage web front end NDN-JS and D3 Web-based front end using NDN-JS with access to geofenced location information, providing (for example) running trail visualization. Perhaps use many GPX format visualizers. E.g., `http://flowingdata.com/2014/02/05/where-people-run/`

Reference: Lifestreams Dashboard [?]

Building Block: Analytics / Presentation: Ohmage Front-end for Mobilize `https://wiki.mobilizingcs.org/app/web`

And Lifestreams? Web-based front end using NDN-JS to access derived data without location information. Examples: http://quantifiedself.com/fitbit/

The web frontend (powered by the ohmage project) is used to provide students secure access to their data. It supports secure login, campaign management, data management and basic campaign monitoring and visualization. The students can review and share their data to the growing data set collected by their class. The web frontend can also be used to discover the answers to basic statistical inferences in real-time as data is being collected. When data collection is complete, the web frontend allows for easy exporting of the data to a more thorough statistical analysis tool.

5.2.5 Identity Manager

Responsible team members UCLA IRL (Yingdi)? and REMAP (Dustin)?

5.3 Naming

Responsible team members UCLA REMAP, UCLA IRL

5.3.1 Data

Personal health data (and metadata) namespace and repository design focusing on support for physical activity data in the first round. What schema?

Basis of design: Open mHealth Physical Activity data schema[3], as well as other schemas from Open mHealth as needed.

The proposed data namespace is shown in Figure 7. Its components are discussed briefly here (The black part is what we have reached agreement on, and the gray part is what needs more discussions and feedbacks):

- Trust anchor

  The root `/org/openmhealth` is the trust anchor of NDNFit, users get identities and certificates from the trust anchor.

- DPU and DVU

  NDNFit-provided DPU and DVU have identities of the format `/org/openmhealth/<service-id>`, they get identities and certificates from the trust anchor. Notice that non-NDNFit-provided DPU and DVU can have identities of other format, such as `/edu/ucla/cs/irl/speed-caculator`, `/edu/ucla/remap/distance-caculator` and so on.

- User's data

  Each user has an identity `/org/openmhealth/<user-id>`. Users get identities and certificates from the trust anchor.

  A user can possess multiple mobile devices, to differentiate them, the user can further assign identity `/org/openmhealth/<user-id>/<device-id>` and issue certificate to each of them.

  - Data - data branch

    The original Data has 3 levels of types. In the example (so the first implementation), they are `fitness`(first level), `physical_activity`(second level) and `time_location`(third level). Under `time_locationnode`, there are 3 branches. The leftmost branch stands for captured data. Each data packet can have multiple samples (time location points) in it, in which case the timestamp component in its name is the start timestamp of all the samples. The middle branch is catalog. Catalogs are generated every 10 minutes (or other more appropriate time interval), the timestamp components in their names must be `YYYY-MM-DDTHH:MM:00Z`. Each catalog data packet contains all the names of captured data whose timestamp falls into the time span (this catalog's timestamp next catalog's timestamp). The rightmost branch is C-KEY which is used for data access control, we will discuss this in data access control section.

    The DPU-processed data are published in the bout branch. Similar to the original data branch, the leftmost branch stands for processed data, the middle branch stands for catalog, and the rightmost branch is C-KEY.

  - Data access control - read branch

    Name-based access control is used here. Take captured data as an example to illustrate how it works here in NDNFit.

    Every hour, a new C-KEY (symmetric key) is generated and used to encrypt all the captured data in this hour. The C-KEY is encrypted by the E-KEY of each access group who has access to the

[3] `http://www.openmhealth.org/developers/schemas/#physical-activity` and `http://bioportal.bioontology.org/ontologies/SNOMEDCT?p=classes&conceptid=68130003`

corresponding captured data, and is named `/<prefix>/C-KEY/<start_timestamp_hour>/<end_timestamp_hour>/<E-KEYname>`. The `start_timestamp_hour` and `end_timestamp_hour` are the start and end hour points when the C-KEY is valid. For each access group, the E-KEY and D-KEY (asymmetric key pair) are also generated periodically (the period is not fixed to 1 hour). The E-KEY is named `/<prefix>/E-KEY/<start_timestamp_hour>/<end_timestamp_hour>/`; and the D-KEY is encrypted for each member (data consumer) of this access group and named `/<prefix>/D-KEY/<start_timestamp_hour>/<end_timestamp_hour>/for/<consumer-id>`. Notice that in this case, all the C-KEYs, E-KEYs and D-KEYs are generated by the user.

For the DPU-processed data (the bout branch), it's a little complicated. In this case, the C-KEY should be generated by the DPU, and then it should be sent back to the user (the C-KEY should be encrypted only for the user) to be encrypted by some access groups? E-KEYs. All the other steps are the same as the previous.

– Data publish control - write branch

When a DPU processes user's data, it should use some valid signing key to sign the processed results. The signing key pair is generated by the DPU, the private key is kept by the DPU and the public key is sent back to the user. Then the user checks whether the public key is named properly (so data consumers can validate the signature of processed results according to trust schema), if it is, the user can sign and publish this public key. The name structure of this public key is similar to that of D-KEYs.

- User group

Some users can form groups. Groups have identities of the format `/org/openmhealth/<group-id>`. This is not part of first implementation. Details will be discussed in the future.

5.3.2 Certificates

5.3.3 Processing

Borrow ideas from Named Function Networking concept for distributed processing

5.4 Trust and security

Responsible team members U. Michigan, UCLA IRL, UCLA REMAP

5.4.1 Trust model

In this application, the user should be the root of trust for their own data, though this may be enabled by various service providers.

If possible, we would like to leverage the existing PKI/SPKI support in NDN, and take advantage of the existing infrastructure. *However*, certificates used should not leak real world identity information except by the choice of the user.

One entity (here, the " user") maintains multiple publishers whose data are consumed by many services with varying levels of access based on the: type of data (as expressed in the name), level of granularity, and date/time when the data was produced. ABE seems a good fit here, is it viable for practical experimentation or do we need a directory-of-symmetric-keys or some other approach? If not ABE, where should we begin for our approach? Are there existing best practices?

5.4.2 Identity

Users may have different identities per service (or at least per flow).

We are exploring the idea of an "identity manager", an application manages the certificates (identities) that an individual uses to interact with the various services involved in this application. Are there good examples of *user interfaces* for identity management already? In fact, pointers to state-of-the-art in end-user interfaces for security decision making would be helpful. Alex doesn't think there are many.

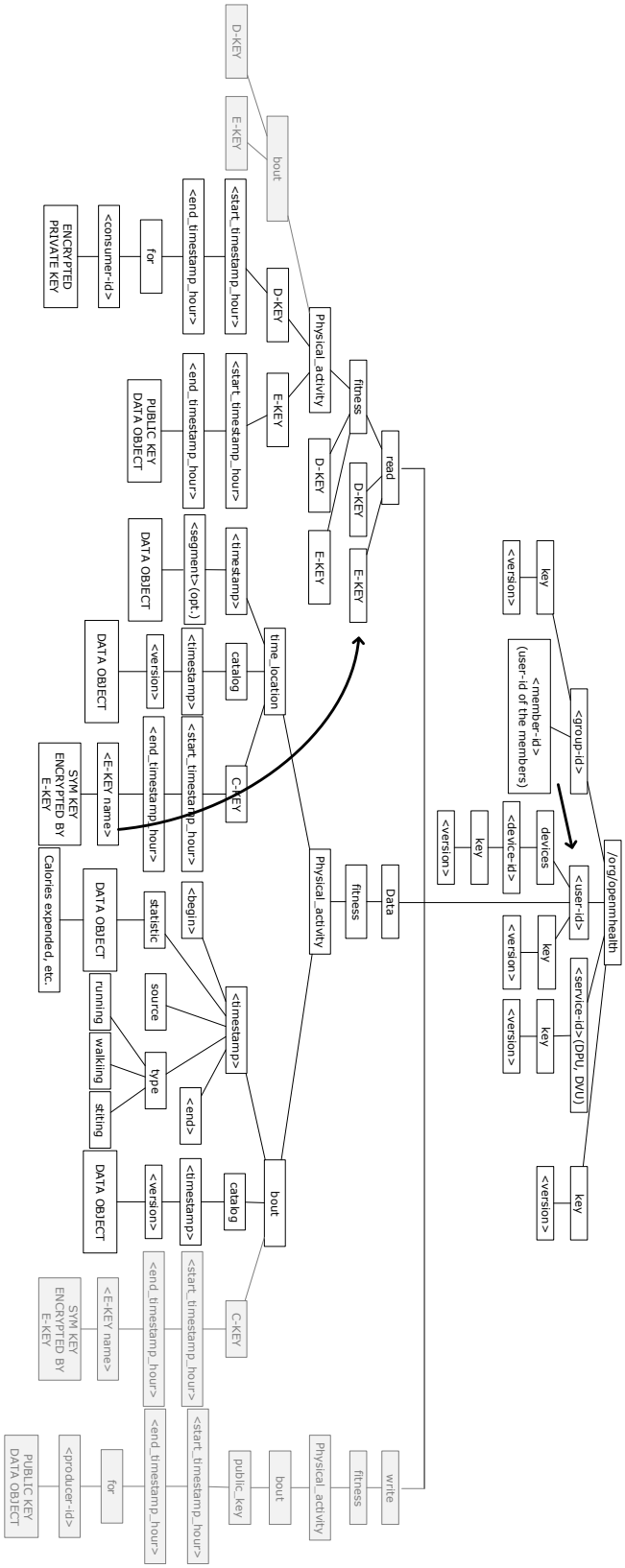
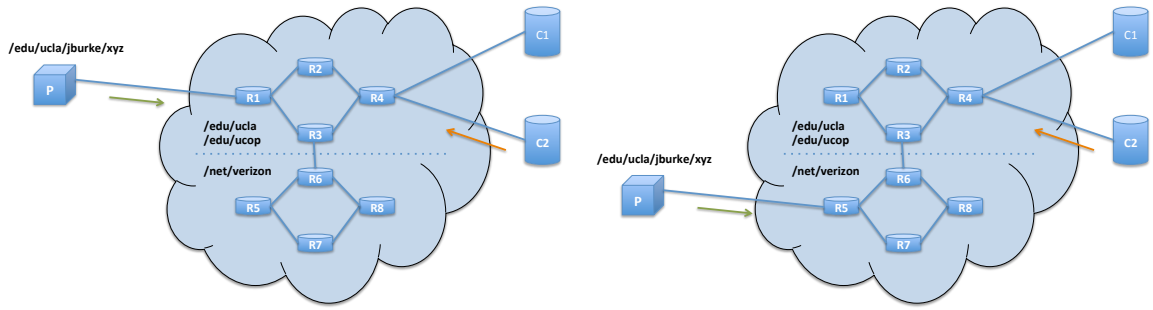

/org/openmhealth
<group-id>
<member-id>
(user-id of the members)
<user-id>
<service-id> (DPU, DVU)
key
<version>
devices
<device-id>
read
write
fitness
Physical_activity
Data
bout
time_location
<timestamp>
<begin>
<end>
catalog
C-KEY
D-KEY
E-KEY
DATA OBJECT
statistic
source
type
running
walking
sitting
Calories expended, etc.
SYM KEY ENCRYPTED BY E-KEY
<E-KEY name>
ENCRYPTED PRIVATE KEY
PUBLIC KEY DATA OBJECT
<consumer-id>
<producer-id>
for
public_key
 (opt.)
<start_timestamp_hour>
<end_timestamp_hour>

Figure 7: NDNFit Namespace, version 2.



(a) Mobile publisher connected to "home" hub, which directly routes its publishing prefix.

(b) Mobile publisher connected to another hub, which does not directly route its prefix.

Figure 8: Mobile publisher scenario.

Each step in the data processing model tends to generate derived data that must also be stored and may or may not be associated with the original identity.

5.4.3 Integrity

5.4.4 Confidentiality

Principal of minimum information: services should only request / consume the minimum amount of information needed to complete their application.

We will need to come up with an approach for name encryption for this environment, in a way that still enables applications to operate on the namespace–perhaps without having to be concerned with decryption–once it has been decrypted and/or de-encapsulated.

Public access should be considered but is primarily for future access. Extensions to support epidemiological studies incorporating semi-anonymized opt-in data across large populations.

5.4.5 Data flow model support

Must consider access control for services in "data flow" model for communication between processing components, replacing Oauth.

We imagine a data flow like model for this system: [Publisher]-¿[Processing]-¿[Processing]-¿[Visualization], with each [] block being owned by a different entity and an objective to leak the minimum amount of context to the processing components. I'm not sure we understand how to handle authentication / access control of the intermediate processing blocks, to each other and to the source/sink of the data. Where can we look for best practices for security in current data flow architectures?

5.5 Storage

Responsible team members UCLA IRL, Tsinghua, UCLA REMAP

A distributed network of repos replaces the ecosystem of DSUs envisioned in the Open mHealth TCP/IP architecture.

(Hierarchical network of repositories, similar to BAS/BMS, including both personal repositories, service provider backups for personal data, and aggregated "anonymized" stores).

Need to provide write access control

A mechanism for delegating authority to publish into a repository is necessary.

5.6 Routing & Forwarding

Responsible team members UCLA IRL

Mobile publishing support. NDNS?

6 Evaluation

At this early stage, we evaluate the pilot application design by discussing whether the affordances of the NDN architecture, libraries, and deployment scenarios make it easier (or possible at all) to meet the requirements of the application. To do so, we employ Green & Petre's *cognitive dimensions* framework [**?**] to compare a possible IP-based approach with the approach taken here. These dimensions are "descriptions of the artifact-user relationship, intended to raise the level of discourse." [**?**] They do not provide a comprehensive evaluation framework, merely a starting point for discussion.

Dimensions of evaluation that we will consider are:

- Abstraction gradient
- Closeness of mapping
- Consistency
- Diffuseness
- Error-proneness
- Hidden dependencies
- Premature commitment
- Progressive evaluation
- Role expressivenesss
- Secondary notation
- Viscosity
- Visibility

We also separate discussion of the prototype application from the deployed system of NDN (libraries, testbed, forwarder, etc.) and the architecture itself, following John Wroclawski's suggestion at the 2013 NSF FIA PI Meeting. [4]

6.1 Architecture

Fundamental capabilities of the NDN architecture vs. the IP architecture.

6.2 System

Codebase and testbed.

6.3 Prototype

Pilot application.

7 Open Challenges

Efficient read-audit functionality in NDN.

Student red team attacks.

[4] "All hat, no answers: Some issues related to the evaluation of architecture." John Wroclawski, NSF FIA PI Meeting, March, 2013. `http://www.nets-fia.net/Meetings/Spring13/FIA-Arch-Eval-JTW.pptx`

8 Conclusion

Acknowledgment

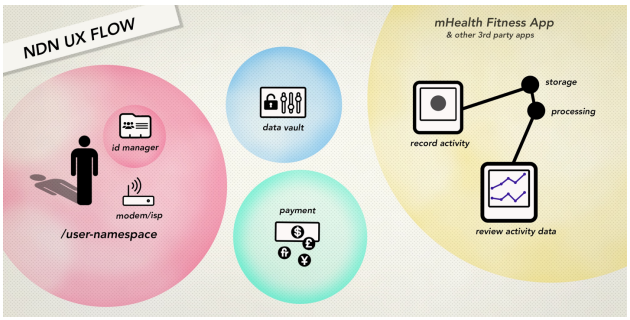


Figure 9: This sketch illustrates the user's spheres of experience and the conceptual-emotional proximity of how systems relate to one another from the user's perspective. Data is stored, published, and managed, by the data vault. Intimate to the user's navigation of third party applications is the "id manager" which gives the user a fast and easy way to specify which identities to use with specific applications. Third party applications exchange structured data within a data-dissemination based paradigm, giving users greater access to a modular understanding of systems.

9 Appendix: Use Case Scenario

by Dustin O'Hara and Jeff Burke

9.1 Motivation

This appendix describes recent UX research within the context of the NDN (Named Data Networking) project. The UX design has primarily been focused on a mobile fitness application. The broad goal of the fitness app is to develop a non-trivial application to test the NDN network architecture. As an extension of these layered goals, the UX design process has been focused on both the mobile fitness application, the management of personal data (id manager), and the process of joining the NDN internet. The UX research/design has primarily taken the form of speculative concept sketches, user narratives and scenarios, to help think through possible futures involving the NDN internet. Within the context of the NDN project the hope is that UX design can function as an entry point for discussing critical "values in design" issues surrounding security and identity.

Consider non-technical knowledge of computing and the internet. A large percentage of internet users don't know what an internet browser is, with the difference between their operating system or "desktop" and the browser being fuzzy at best. By extension, understandings of domains and urls are limited, to marketing campaigns from the first .com boom. Many younger internet users have little concept of an internet outside of their popular social platforms. At the same time the concept of an "app" has become widely known. In the most simple, and almost mystic of ways, the app has become a magical button that offers services of amusement or utility. While the app has its internal functionality, the ontological foundation of the user experience of the "app" is the on/off, in/out experience of the phone. This "in/out" experience reproduces a silo like concept of the application as a discrete thing, something you put in your pocket, rather than a series of interrelated systems.

For the NDN internet user, the interoperability of structured data, will translate into a user experience that reorients the popular understanding of apps and urls. With this shift will be a greater awareness of how one's apps and devices are speaking to each other. The on/off quality of the discrete app will be replaced with an and/or, intuitively modular, way of understanding applications and the flow of data and systems.

9.2 Steps in Application Use

1. User Joins the NDN Network. The user unpacks their new modem. On their computer/device the user runs an application from the internet provider that walks them through process of connecting to

the NDN internet. The user gets connectivity to the NDN Internet. The application then walks the user through the process of claiming a namespace and authenticating connections with personal devices.

2. Signs Up for Personal Data Vault / Identity Manager. (This could happen as part of application signup.)

Once the user has finished claiming a name space and connecting devices, the application presents the user with a menu of recommended "personal data guardians" or data vaults. As a professional service provider the data guardian functions as a confidant for archiving and managing self generated personal data. The data vault provides the user with automated data management, granular data preferences, and programatic controls. While the data vault is technically a separate application from the identity manager, from the user's perspective they are closely connected. The identity manager allows fast on-the-go authentication and data management.

The user selects a data vault that comes with technical and legal support in the event of a problem, and a collection of additional identity management and data visualization applications for specific aspects of the user's life. The user is taken through a process of selecting key preferences, and authenticating a connection with their various devices. Devices have default data exchange preferences, with brief explanations of what they are.

The user is given a simple faceted search tool to narrow their focus to data types, specific data exchanges, devices, and 3rd party client applications. While the user can edit the preferences of a single exchange, the user can also create programmatic changes, that effect all data exchanges or certain data types.

- Preferences for local device exchanges.
- Preferences for device to manufacture exchanges.
- Preferences for new applications?

This raises several issues for us to consider:

- How does the user know that they are actually talking to the right PDV?
- How does the user know what data belongs to what?
- What does the confirmation of a secure connection look and feel like across devices?

2b. Payment. Once complete the user submits their payment information. Bitcoin, Apple Pay, payment specific devices and objects, etc...

3. Identity Manager. Once signed into the Identity Manager, the user can see a summary of the various data exchanges they have. The authentication process, or signing in process, will in many cases be through some form of biometrics, finger print, voice, etc. But there will have to be an option for those that refuse biometrics. As mentioned above, the identity manager allows fast on-the-go authentication and data management through a high-level notion of "identities" that have associated preferences. An identity is about which apps speak to each other, and how they speak to each other.

4. Fitness Application. The user downloads/bookmarks the fitness application. Confirmation of authentic connection. When initially launching the application, the ID manager gives the user options for which "identity" to associate with the application (when and where). The personal data vault request permissions to provide the application with location (etc) data, with associated default preferences, with recommendation for how the data should be formatted/summarized/edited before sharing. Highlights from the default preferences include opting to contribute anonymized data to California State Park user research effort and sharing data with fitness data visualization app / fitness with friends app. While the fitness activity recorder and fitness visualization apps are branded under a single California State Park identity and UX experience, they function as separate applications, one publishing data for the other. The user begins using fitness application. After running or walking a complete loop around the park a short audio clip is pushed to the user. The audio clip gives the user a short historic piece of information about the park and surrounding area. For every loop the user is give a short audio piece.

4b. Payment.

5. Fitness Visualization & Fitness with Friends App . The user goes to the visualization app and explores their fitness data through various visualization tools.

6. Content push Location-based content push

7. Identity Manager. Returning to the ID Manager, the user can now see their fitness applications preferences have been added. A friend sends the user an invitation to join their "Fitness with Friends Group," which the user accepts, linking their fitness data to the application. The ID manager, asks the user if they would like to send the fitness with friends app a statistically averaged (or "edited") version of their data. Third Party Applications? Flow between devices / vault / client app?

8. Usage...

N. Exiting Process. How to exit?

- Exit from third party online systems?
- Exit from the data vault?
- Exit from data manager?
- Exit from the name space?

9.3 User scenarios

9.3.1 Anna & the Salon

Anna lives in east Hollywood, and runs a small hair salon in Silver Lake. At home Anna signs up for a name space and personal data vault / manager. At work, Anna's partner Karen signs her up as a co-owner of a data vault / manager for their salon. Data that is specific to Anna and Karen synchronizes across their shared data vault, and their individual fault.

Karen eventually decides to sell her share of the salon to Anna, and in doing so, signs over complete ownership to the data vault to Anna. Some time later, Anna decides to sell the salon. As part of the sale, an edited version of the data vault / manager is transferred to the new owner.

9.3.2 Mark's Fitness

Mark lives in Lincoln Heights, and regularly goes to the Los Angeles State Historic Park to run around the park track. Mark downloads the fitness application, and starts using it in the park. While running Mark befriends Sarah who also runs in the park. After seeing each other several times, the two exchange contact information, and begin comparing running data. The two use different fitness tracking apps, but use a common app for sharing and comparing their running data.

Since Mark is using the LASHP fitness app, every time he completes a lap around the park, he receives a short audio piece about Los Angeles History. Knowing his father enjoys LA history and needs to get out more often, Mark shares the audio piece with his father, and invites him to come along for a walk in the park. Wanting the historic information, Mark's father downloads the fitness app, and starts walking with his son on a weekly basis.

Because Mark and his father are using the LASHP fitness app, and agreed to the default settings, they are contributing an anonymized version of their data to LASHP for research purposes. After visiting the park regularly for two months, LASHP forwards them a request to fill out a short survey. Mark fills out the survey, and when finished is asked if he wants to invite any of his running partners to contribute to LASHP research. He decides to forward the invitation to Sarah. Sarah receives the invitation, but instead of having to download the LASHP fitness app, she is able to authorize a data exchange with one click, giving LASHP access to an anonymized version of her data.

9.4 Open questions

- How do users join the NDN network (for the first time, or on the go as a mobile guest)?
- How do users understand and manage their data and online identities?
- How are encryption keys managed, and/ how is "authentication" understood by users?What is the user experience of the fitness application?
- How can the user experience of the fitness application be emblematic of the data-dissemination paradigm?

10 Appendix: Subschema reference

Schema referred to in the physical activity schema, Listing 1.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",

    "description": "This schema represents a single episode of physical activity.",
    "type": "object",
    "references": [
        {
            "description": "The SNOMED code represents Physical activity (observable entity)",
            "url": "http://purl.bioontology.org/ontology/SNOMEDCT/68130003"
        }
    ],
    "definitions": {
        "activity_name": {
            "$ref": "activity-name-1.0.json"
        },
        "length_unit_value": {
            "$ref": "../generic/length-unit-value-1.0.json"
        },
        "time_frame": {
            "$ref": "../generic/time-frame-1.0.json"
        }
    },

    "properties": {
        "activity_name": {
            "$ref": "#/definitions/activity_name"
        },
        "effective_time_frame": {
            "$ref": "#/definitions/time_frame"
        },
        "distance": {
            "description": "The distance covered, if applicable.",
            "$ref": "#/definitions/length_unit_value"
        },
        "reported_activity_intensity": {
            "description": "Self-reported intensity of the activity performed.",
            "type": "string",
            "enum": ["light", "moderate", "vigorous"]
        }
    },

    "required": ["activity_name"]
}
```

Listing 1: Open mHealth Physical Activity Schema, retrieved December 28, 2014. See appendix for sub-schema.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",

    "description": "The name(s) of the physical activity(ies) in which the person is engaged. It is recommended that the activ

    "references": [
        {
            "description": "The SNOMED code represents Activity",
            "url": "http://purl.bioontology.org/ontology/SNOMEDCT/257733005"
        },
        {
             "description": "CDC guidelines on standard energy expenditure values (METs).",
             "url": "http://www.startwalkingnow.org/documents/PA_Intensity_table_2_1.pdf"
        }
    ],
    "type": "string"
}
```


Listing 2: Activity Name schema, retrieved December 28, 2014.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",

    "description": "This schema represents a length or a distance.",
    "type": "object",

    "references": [
        {
            "description": "The SNOMED code represents Length",
            "url": "http://purl.bioontology.org/ontology/SNOMEDCT/410668003"
        }
    ],

    "allOf": [
        {
            "$ref": "unit-value-1.0.json"
        },
        {
            "properties": {
                "unit": {
                    "references": [
                        {
                            "description": "The unit of measure of the element. Basic unit is meter (m) [ http://unitsofmeasur
                            "url": "http://www.hl7.de/download/documents/ucum/ucumdata.html"
                        }
                    ],
                    "enum": [
                        "fm",
                        "pm",
                        "nm",
                        "um",
                        "mm",
                        "cm",
                        "m",
                        "km",
                        "in",
                        "ft",
                        "yd",
                        "mi"
                    ]
                }
            }
        }
    ]
}
```

Listing 3: Length Unit Value schema, retrieved December 28, 2014.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",

    "description": "This schema describes a time frame as a point in time or a time interval.",

    "type": "object",
    "references": [
        {
            "description": "The SNOMED codes represent Time frame (qualifier value).",
            "url": "http://purl.bioontology.org/ontology/SNOMEDCT/7389001"
        }
    ],

    "definitions": {
        "date_time": {
            "$ref": "date-time-1.0.json"
        },

        "time_interval": {
            "$ref": "time-interval-1.0.json"
        }
    },

    "oneOf": [
        {
            "properties": {
                "date_time": {
                    "$ref": "#/definitions/date_time"
                }
            },
            "required": [ "date_time"]
        },
        {
            "properties": {
                "time_interval": {
                    "$ref": "#/definitions/time_interval"
                }
            },
            "required": [ "time_interval"]
        }
    ]
}
```

Listing 4: Time Frame schema, retrieved December 28, 2014.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",

    "description": "This schema describes an interval of time. In the absence of a precise start and/or end time, the time int
    "type": "object",

    "references": [
        {
            "description": "The NCIT codes represent Timespan (synonym: time interval).",
            "url": "http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C68594"
        }
    ],

    "definitions": {
        "date_time": {
            "$ref": "date-time-1.0.json"
        },
        "duration-unit-value": {
            "$ref": "duration-unit-value-1.0.json"
        },
        "full_date": {
            "type": "string",
            "references": [
                {
                    "description": "This schema represents a date. See RFC 3339 5.6 for details.",
                    "url": "http://tools.ietf.org/html/rfc3339"
                }
            ],
            "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}$"
        },
        "part_of_day": {
            "$ref": "part-of-day-1.0.json"
        }
    },

    "oneOf": [
        {
            "properties": {
                "start_date_time": {
                    "$ref": "#/definitions/date_time"
                },
                "duration": {
                    "$ref": "#/definitions/duration-unit-value"
                }
            },
            "required": ["start_date_time", "duration"]
        },
        {
            "properties": {
                "end_date_time": {
                    "$ref": "#/definitions/date_time"
                },
                "duration": {
                    "$ref": "#/definitions/duration-unit-value"
                }
            },
            "required": ["end_date_time", "duration"]
        },
        {
            "properties": {
                "start_date_time": {
                    "$ref": "#/definitions/date_time"
                },
                "end_date_time": {
                    "$ref": "#/definitions/date_time"
                }
            },
            "required": ["start_date_time", "end_date_time"]
        },
        {
            "properties": {
                "date": {
                    "$ref": "#/definitions/full_date"
                },
                "part_of_day": {
```

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",

    "type": "string",
    "references": [
        {
            "description": "This schema represents a point in time (ISO8601). If a timezone is not included, the timezone is a
            "url": "http://tools.ietf.org/html/rfc3339#section-5.6"
        },
        {
            "description": "The SNOMED codes represent Single point in time (qualifier value).",
            "url": "http://purl.bioontology.org/ontology/SNOMEDCT/123029007"
        }
    ],
    "format": "date-time"
}
```

Listing 6: Date Time schema, retrieved December 28, 2014.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",

    "description": "This schema represents a duration or length of time.",

    "type": "object",

    "references": [
        {
            "description": "The SNOMED code represents Duration (qualifier value)",
            "url": "http://purl.bioontology.org/ontology/SNOMEDCT/103335007"
        }
    ],

    "allOf": [
        {
            "$ref": "unit-value-1.0.json"
        },
        {
            "properties": {
                "unit": {
                    "references": [
                        {
                            "description": "The unit of measure of the element. Basic unit is second (s). Allowed values are c
                            "url": "http://www.hl7.de/download/documents/ucum/ucumdata.html"
                        }
                    ],
                    "enum": [
                        "ps",
                        "ns",
                        "us",
                        "ms",
                        "sec",
                        "min",
                        "h",
                        "d",
                        "wk",
                        "Mo",
                        "yr"
                    ]
                }
            }
        }
    ]
}
```

Listing 7: Duration Unit Value schema, retrieved December 28, 2014.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",

    "description": "The period of time in which a day is commonly divided.",

    "type": "string",

    "references": [
        {
            "value": "morning",
            "description": "The SNOMED code represents morning (temporal qualifier)",
            "url": "http://purl.bioontology.org/ontology/SNOMEDCT/73775008"
        },
        {
            "value": "afternoon",
            "description": "The SNOMED code represents afternoon (temporal qualifier)",
            "url": "http://purl.bioontology.org/ontology/SNOMEDCT/422133006"
        },
        {
            "value": "evening",
            "description": "The SNOMED code represents evening (temporal qualifier)",
            "url": "http://purl.bioontology.org/ontology/SNOMEDCT/3157002"
        },
        {
            "value": "night",
            "description": "The SNOMED code represents night time (temporal qualifier)",
            "url": "http://purl.bioontology.org/ontology/SNOMEDCT/2546009"
        }
    ],
    "enum": ["morning", "afternoon", "evening", "night"]
}
```

Listing 8: Part of Day schema, retrieved December 28, 2014.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",

    "description": "This schema represents a numerical value with a unit of measure.",

    "type": "object",

    "properties": {
        "value": {
            "description": "The numeric value of the element.",
            "type": "number"
        },
        "unit": {
            "references": [
                {
                    "description": "The unit of measure of the element. Allowed values are drawn from the Common synonyms (nor
                    "url": "http://www.hl7.de/download/documents/ucum/ucumdata.html"
                }
            ],
            "type": "string"
        }
    },
    "required": [
        "value",
        "unit"
    ]
}
```

Listing 9: Unit Value schema, retrieved December 28, 2014.