NDN NP Network Environments (really short) Recap

(Unproofed, from the morning)

Jeff Burke
jburke@ucla.edu
February 6, 2015

Goal for students

I hope that you can walk away today with an initial understanding how the network environments (and sample applications) can 1) provide context and motivation for aspects of your research and 2) incorporate your contributions and ideas.

How you can help

Wentao with BMS repo / query design.

REMAP with BMS publisher incl access control.

Haitao with Open mHealth comm model.

Dan Pei's group with omh storage.

Dustin with the identity manager and omh UX.

Christian / Basel with NFN processing for omh.

Anyang Univ with Ohmage mobile publishing.

Context: Each netenv…

- …is motivated by past work we know pretty well, have some design experience in, and have already explored a bit in NDN.
  - OmH: Participatory sensing at UCLA; NDN Personal Data Vault.
  - EBAMS: Instrumented environments; NDN light control, NDN BMS.
- …targets a critical domain / need.
  - OmH: Patient-centered health and wellness via open data exchange.
  - EBAMS: Resilient, secure and internet-connected industrial controls.
- …was selected to push on key research issues.
  - OmH: Naming personal data, mobile publishing, confidentiality, end user-centric trust, "data flow" processing, end-user experience.
  - EBAMS: Naming physical world, enterprise environment, integrity and authorization, institutionally controlled trust, reliability.

Context: Each netenv…

- …is instantiated first in a sample application, already underway.
  - OmH: "NDNEx" physical fitness application. (1 yr to prototype)
  - EBAMS: UCLA-BMS data acquisition and SQL query support (6 months).
- …has a draft namespace design (for the app) influenced by application domain.
  - OmH: Open mHealth project schema.
  - EBAMS: UCLA Deployed BMS namespace, past NDN-BMS.
- …has a deployment context / system design and scale.
  - OmH: Open internet, hundreds of service providers, millions of users.
  - EBAMS: Enterprise network, with configuration/topologies mirroring existing UCLA deployment, 150k sensors @ up to 1Hz, hundreds of building, hundreds of users.

Context: Each netenv…

- …has fairly clear trust requirements in the sample app.
  - OmH: how to trust components selected by an end user from an ecosystem of offerings; how to delegate trust to then have these components interoperate.
  - EBAMS: UCLA-BMS data acquisition and query support (6 months).
- …has an open and a closed side.
  - OmH: Some names and data private, some data very public.
  - EBAMS: Ditto.
- …has important background to read in order to contribute.
  - OmH: Estrin & Sim, 2010. Plus papers on PEIR and Ohmage.
  - EBAMS: Shang et al, 2014. Plus NIST report and UCB BOSS paper.

Each 2015-16 sample application is…

- …*for* someone.
  - OmH: Consumers with smartphones.
  - EBAMS: UCLA Facilities Management.

- …*about* something well-defined.
  - OmH: NDNEx is about personal and group physical activity fitness.
  - EBAMS: UCLA-BMS is about sensor data acquisition.

- …*not about* something else.
  - OmH: Not about hospital-doctor-patient relationship.
  - EBAMS: Not focused on control, constrained devices, or smart homes.

  (Though these are part of the netenv big picture!)

Recap of apps…

Basis for data namespace design: Existing BMS

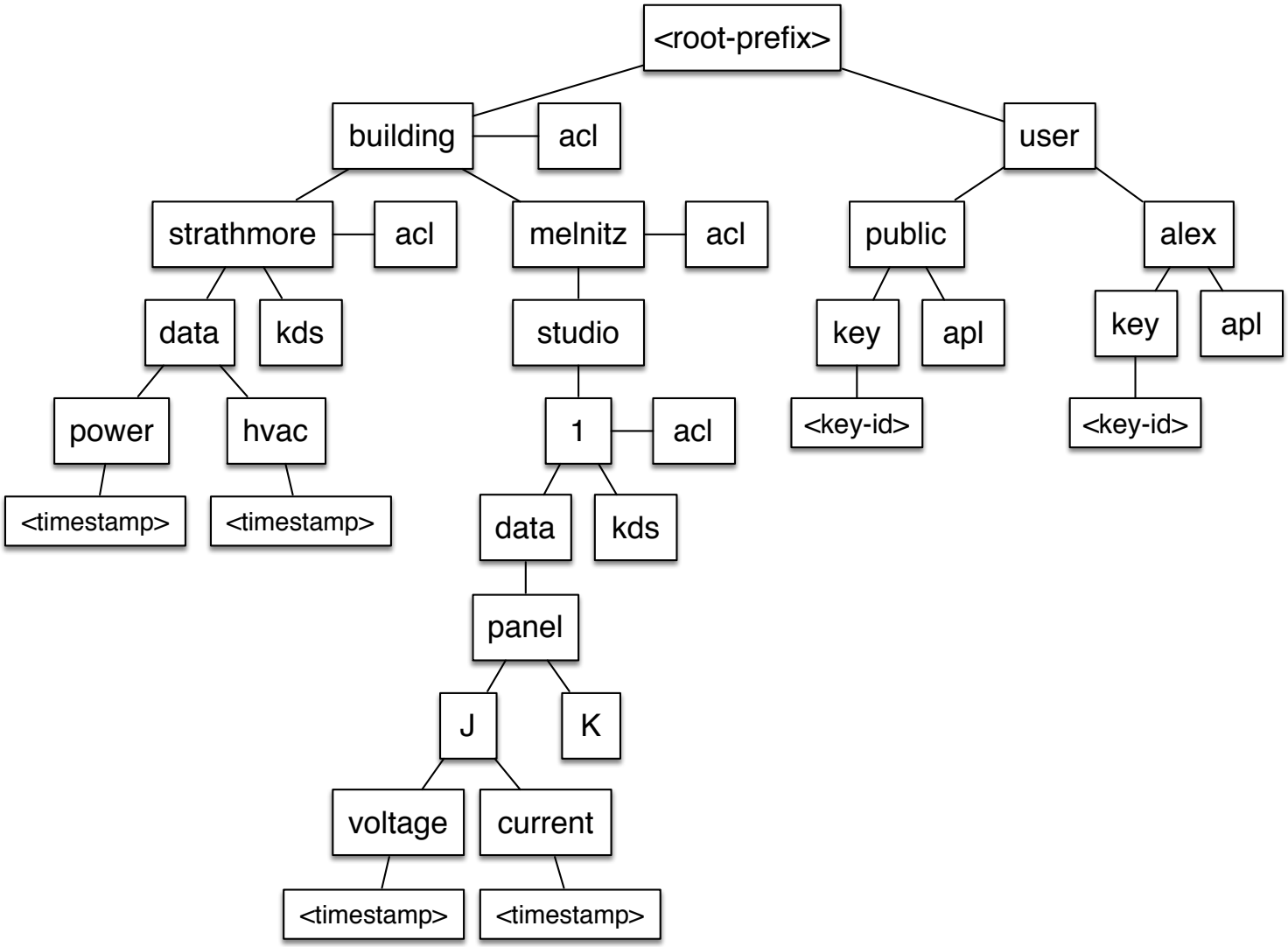
2015-02-05 00:07:32.137000 **/ndn/edu/ucla/bms/powell_lib/b80/xfmr-b/dmd/inst 1423123666.222**
{"pointname": "**UCLA:POWELL_LIB.B80.XFMR-B.DMD.INST**", "timestamp": "1423123666.222", "timestamp_str": "2015-02-05 00:07:46.221999", "locked": "0", "nanoseconds": "221999883", "unknown_1": "577", "seconds": "1423123666", "unknown_2": "192", "type": "1", "value": "213.50399780273438", "conf": "0", "security": "0"}

2015-02-05 00:07:32.341000 **/ndn/edu/ucla/bms/young_libry/stm-fins 1423123667.022**
{"pointname": "**UCLA:YOUNG_LIBRY.STM-FINS**", "timestamp": "1423123667.022", "timestamp_str": "2015-02-05 00:07:47.022000", "locked": "0", "nanoseconds": "22000074", "unknown_1": "577", "seconds": "1423123667", "unknown_2": "192", "type": "1", "value": "3170.07958984375", "conf": "0", "security": "0"}

2015-02-05 00:07:32.645000 **/ndn/edu/ucla/bms/young_hall/b215/xfmr-6/dmd/inst 1423123667.4229999**
payload: {"pointname": "**UCLA:YOUNG_HALL.B215.XFMR-6.DMD.INST**", "timestamp": "1423123667.4229999", "timestamp_str": "2015-02-05 00:07:47.422999", "locked": "0", "nanoseconds": "422999858", "unknown_1": "577", "seconds": "1423123667", "unknown_2": "192", "type": "1", "value": "14.169094085693359", "conf": "0", "security": "0"}

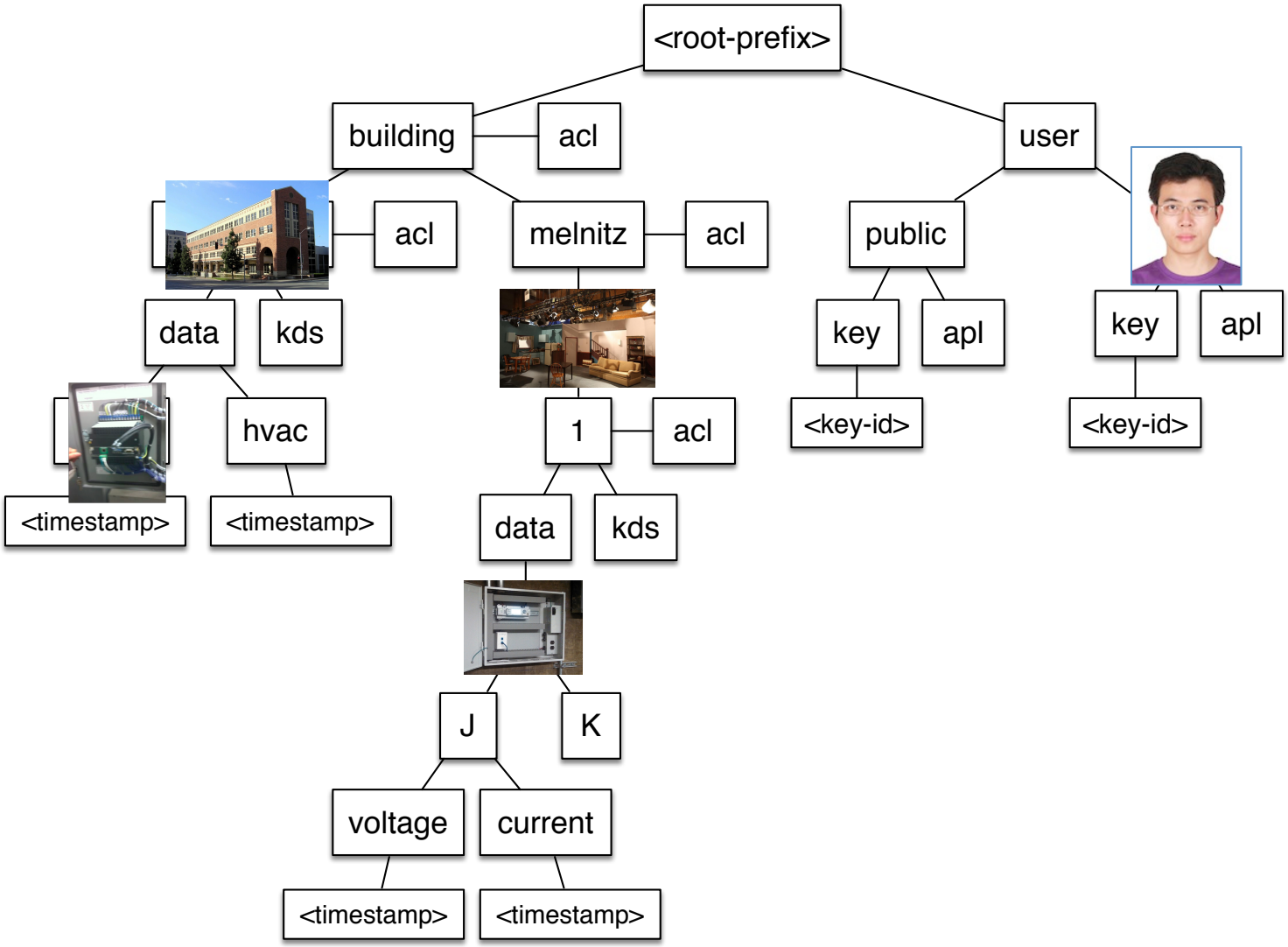
2015-02-05 00:07:32.645000 **/ndn/edu/ucla/bms/young_libry/b1716/chws/rt 1423123667.4229999**
{"pointname": "**UCLA:YOUNG_LIBRY.B1716.CHWS.RT**", "timestamp": "1423123667.4229999", "timestamp_str": "2015-02-05 00:07:47.422999", "locked": "0", "nanoseconds": "422999858", "unknown_1": "577", "seconds": "1423123667", "unknown_2": "192", "type": "1", "value": "231.6475830078125", "conf": "0", "security": "0"}

Pilot UCLA BMS Namespace (last year)



W. Shang et al.

Pilot UCLA BMS Namespace (last year)

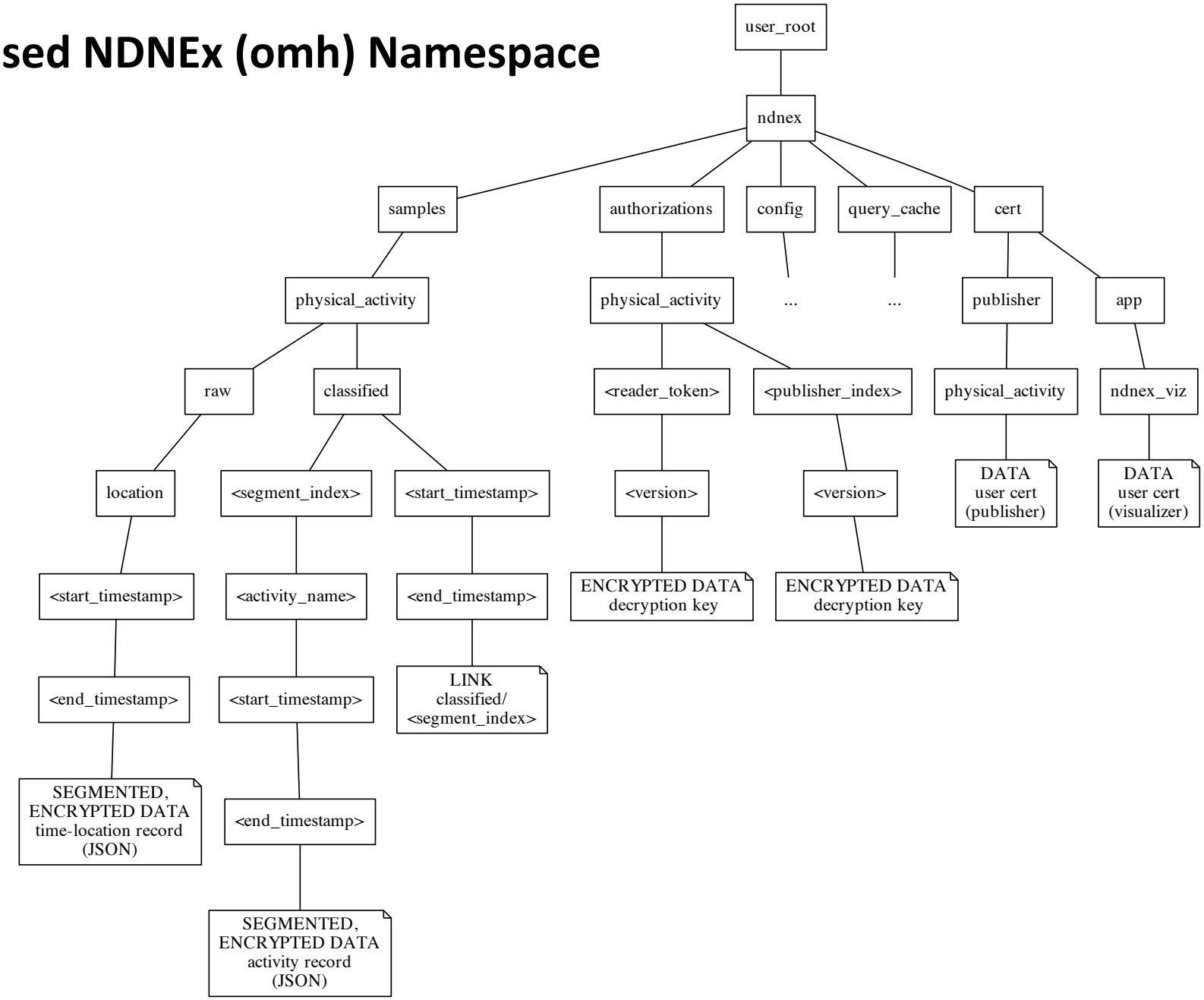


W. Shang et al.

Basis for namespace design: Open mHealth schema

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "This schema represents a single episode of physical activity.",
    "type": "object",
    "references": [
        {
            "description": "The SNOMED code represents Physical activity (observable entity)",
            "url": "http://purl.bioontology.org/ontology/SNOMEDCT/68130003"
        }
    ],
    "definitions": {
        "activity_name": { "$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"]
}
```

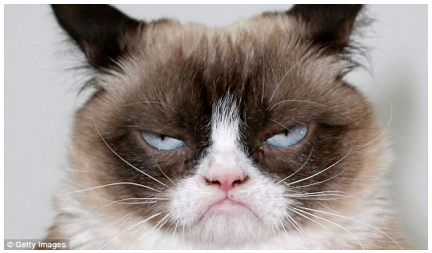
Proposed NDNEx (omh) Namespace



Needs and non-needs to move on sample apps

Care

Naming focused designs - everywhere
Trust model simple, in the namespace
Storage robust, fast, and deployed
-
Encryption available in libraries
Mobile publishing support
-
Understand strategy impact on operational apps.
More debugging tools!
Widely used autoconfig.
-
End-user experience of NDN
Non-expert developer experience of NDN



Don't care

Packet format
Which library implements what
Impl-specific security noodling
Data lifetime issues (yet)
Things about doctors (omh)
Things about IoT (ebams)

"Do not want"

Manually configured routes on clients.
Assuming "connect to the testbed" is simple.
REST-like comm (spec names, data instead)
App APIs (last resort, instead names, data)
Over- and under-worrying about performance

Proposed breakouts

For the **sample applications of the two network environments, articulate requirements** (ask clarifying questions today!), propose a **design direction**, **sketch examples**.

1. **Data authentication/integrity approach, with sample policy expressions in Y & V languages.**
2. **Data confidentiality/encryption based access control approach.**
3. **Adapting Let's Encrypt mechanism to bootstrapping trust in devices and other principals.**
4. **How to approach key storage (both systems and namespace problems.)**

Primary Output

Design sketch in some kind of tangible form. Slides, readable notes, drawings, Labanotation, etc.

Other Goals

Focus on actionable ideas for various groups to work on.

Focus on parsimonious engineering / minimum complexity with maximum applicability.

Breakout #1: **Data authentication/integrity approach, with sample policy expressions in Y & V languages.**

Need to express, not reinvent the trust models. BMS is hierarchical in two namespaces: data and users/principles. Open mHealth users each assemble a collection of components from an "app-style ecosystem" (what model there?) and trust each other in a social network style ecosystem, but with granular sharing.

What is the appropriate relationship between data and key namespaces for each network environment?

How should trust and security models impact namespace design in terms of tree organization, data naming granularity, etc.?

What are critical semantics of each network environment, especially in terms of trust, that should be expressed in the names?

How do we express trust models at the app level (now) for moving on these sample apps?

Breakout #2: **Data confidentiality/encryption based access control approach.**

Granular and expressive approach to confidentiality is important, without overcomplicating things. Multiple spheres of selective access seem important in both apps – based on *data source/type*, *temporal range, consumer group membership*.

Eventually need to solve M2M (data flow) authentication, not always human in the loop

How should we encrypt payloads? Can all payloads be encrypted? What are the implications of payload encryption for other NDN goals (e.g., efficient caching)?

How can we encrypt portions of the namespace to prevent the names themselves from leaking information?

What are the tradeoffs of confidentiality protection in terms of complexity, performance, etc.? How can we best support advanced forms of crypto (e.g., ABE) for applications that benefit from them?

Breakout #3: **Adapting Let's Encrypt mechanism to bootstrapping trust in devices and other principals.**

For EBAMS, focus on *actual BMS deployment context*, not generic IoT or Smart Home context. (That's important but not our target in the netenv yet.) For Open mHealth, focus *on user-initiated, user-centric model*s per the use case in the appendix.

What can be completely automated? When should the human be in the loop (and how?)

How is the process/policy for bootstrapping articulated (whether in names +conventions or policies)?

How can we create visibility into the establishment of trust when needed?

Breakout #4: **How to approach key storage (both systems and namespace problems.)**

Again, focus on *actual deployment context and scale* for BMS. For Open mHealth, consider the nature of the ecosystem that's proposed, then design for the sample apps, which include just a few components.

Ran out of time

People working on the environments.