

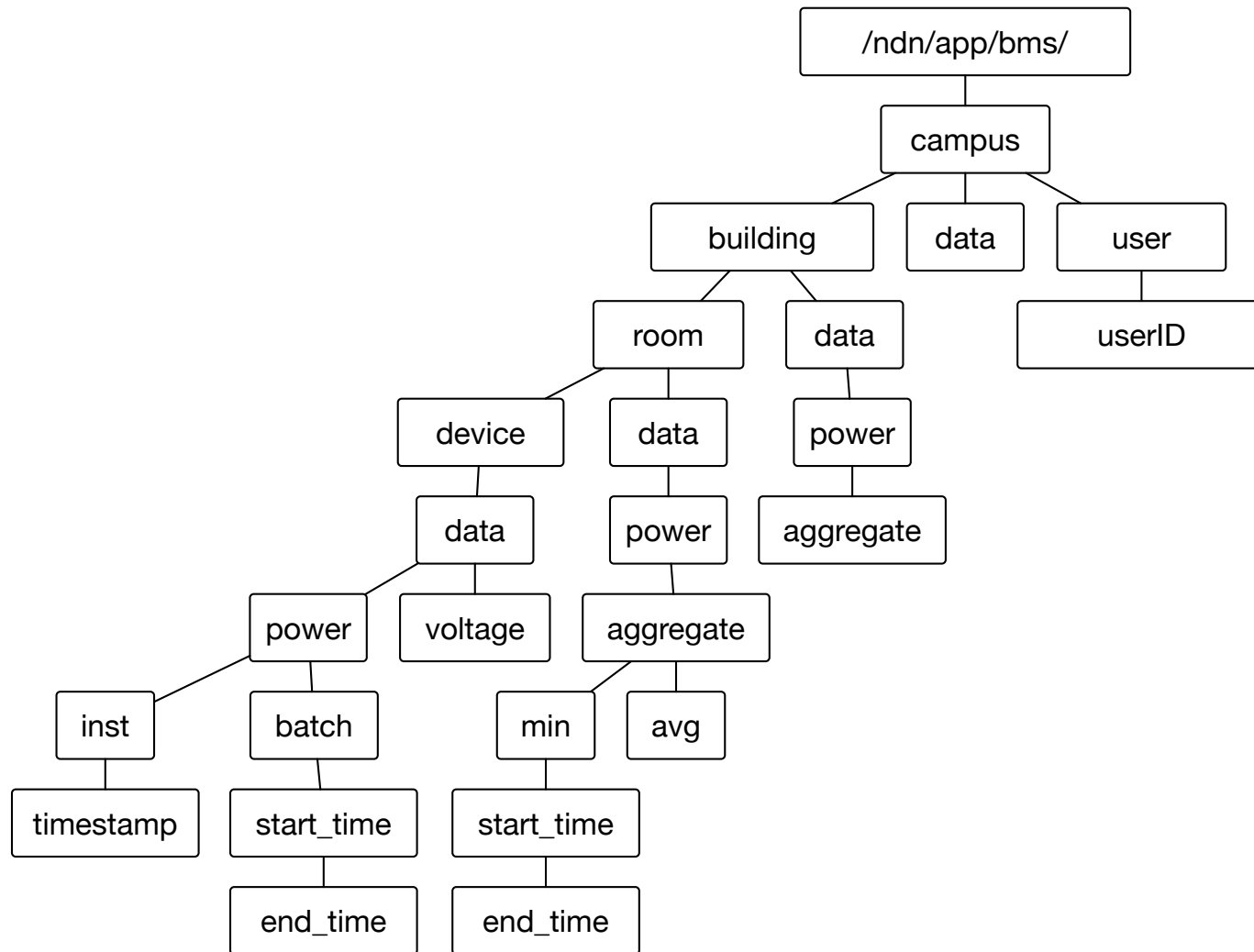
BMS Application Design

Zhehao Wang

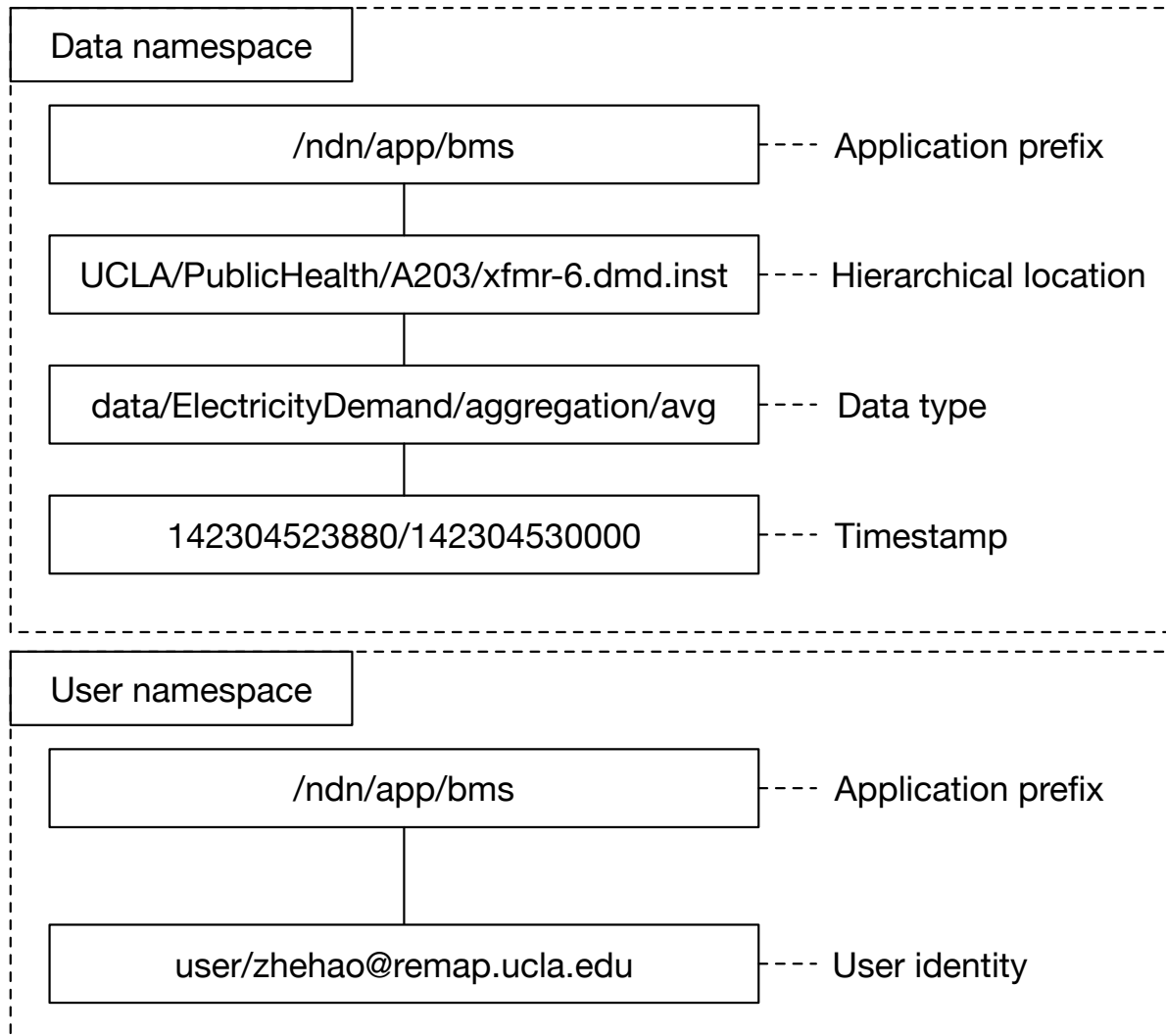
zhehao@remap.ucla.edu

Jul 28, 2015

Namespace



Namespace – examples

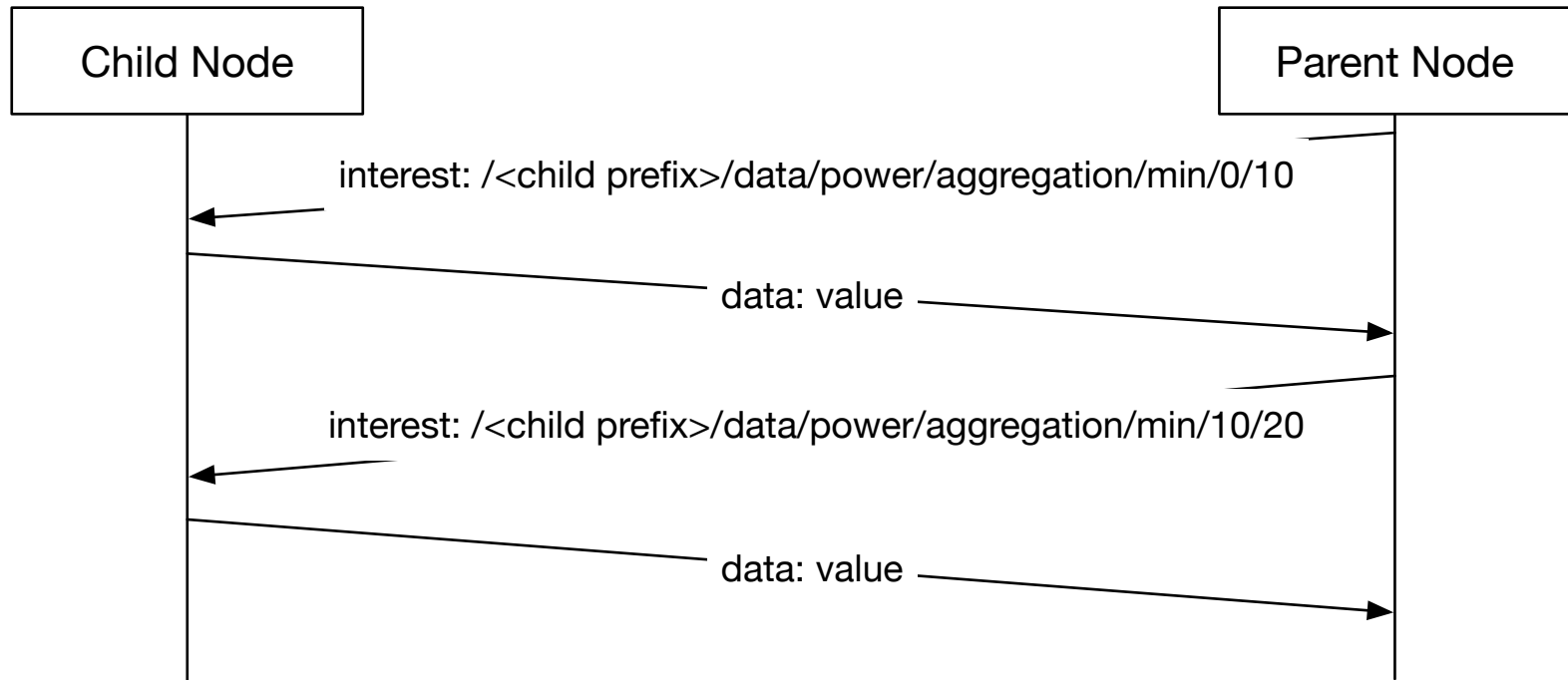


Hierarchical storage

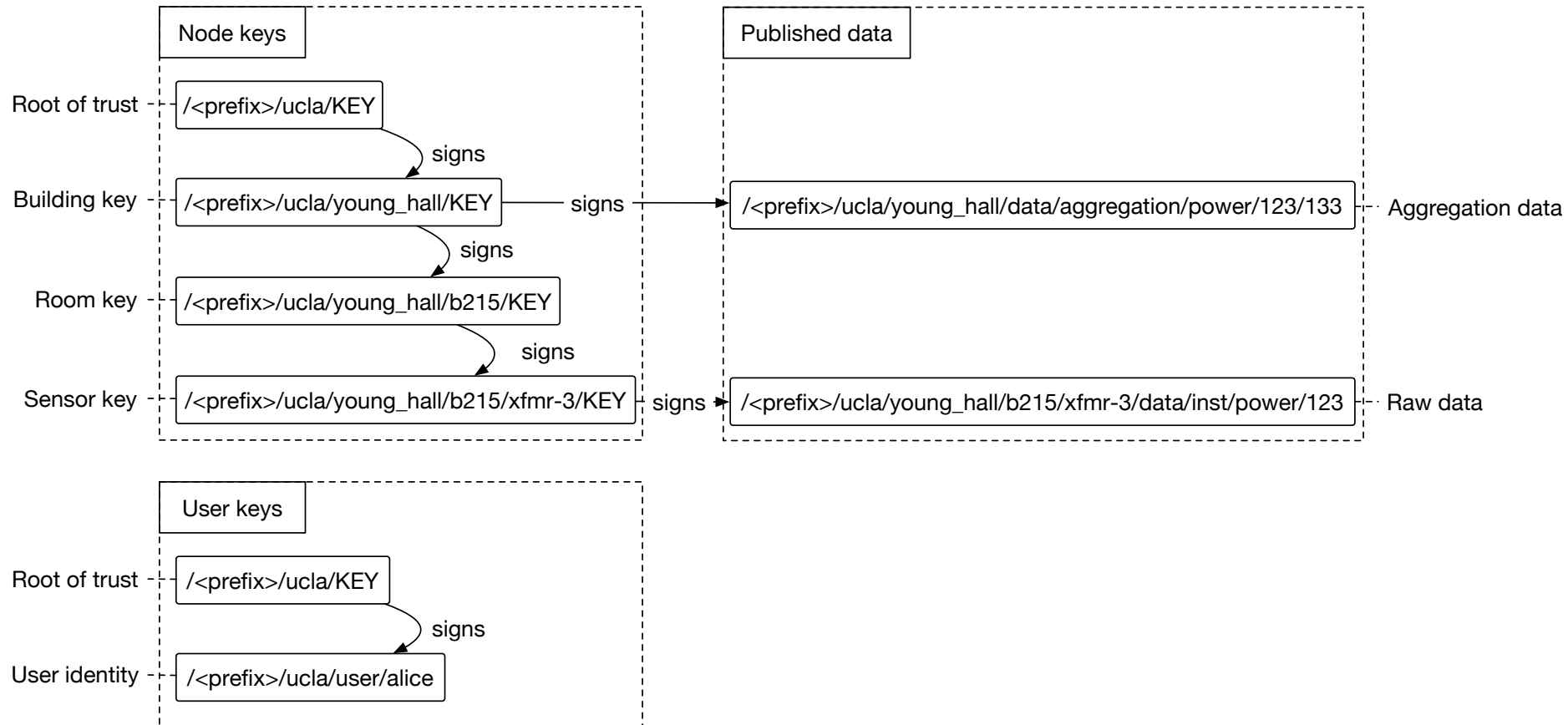
- Raw data collected, kept, and batched only at the leaf nodes.
- Leaf nodes publish aggregated (min, sum, avg, etc) data at fixed time window.
- Non-leaf nodes fetch the aggregated data from all of its children, and aggregate the data after all children respond.
- Non-leaf nodes can publish aggregates with the same time window T ; or $n * T$, $n = 1, 2, 3...$

Moving data across levels

- Long lived interests for fixed (configured) time window



Signing/Verification



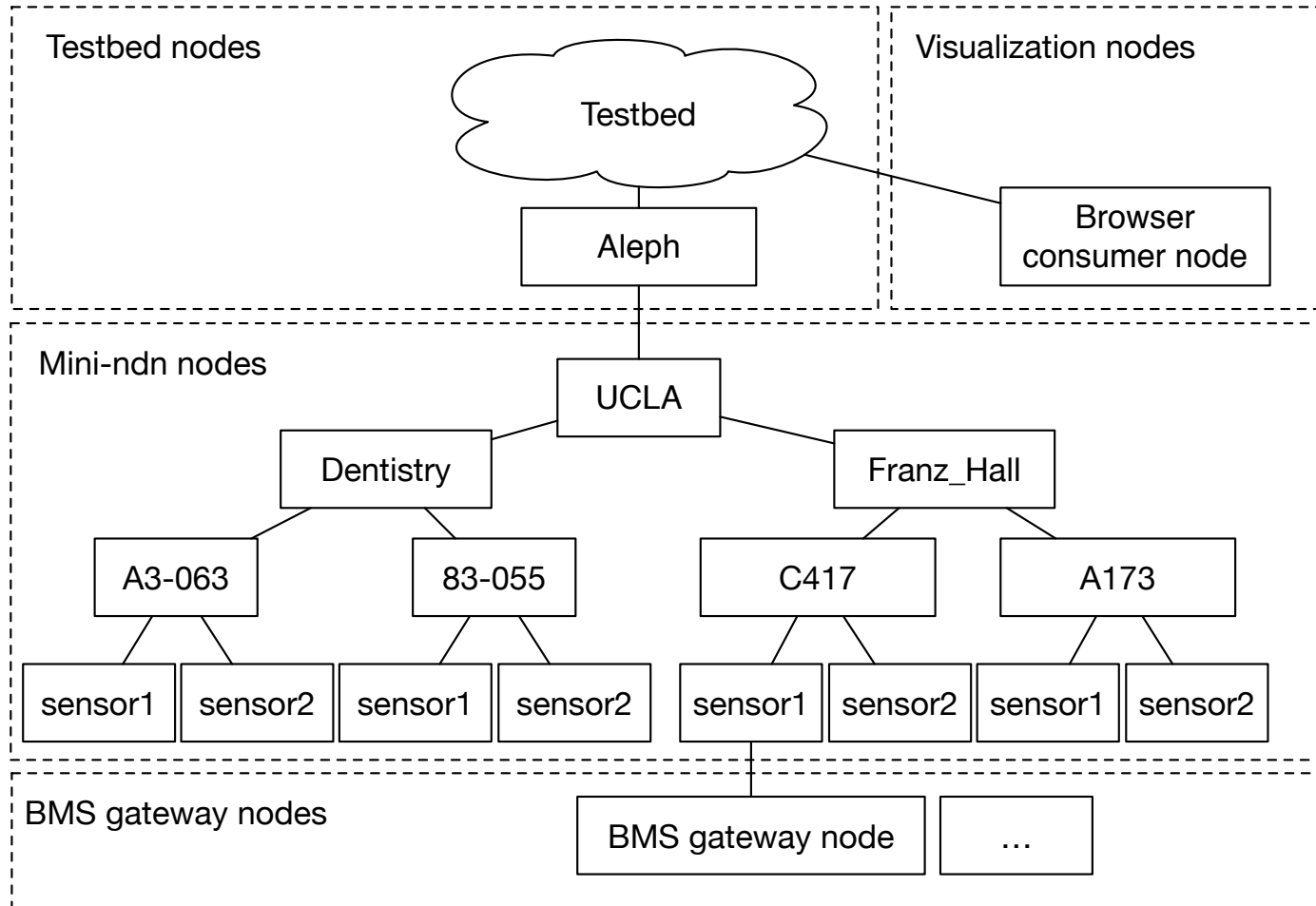
Encryption/Decryption

- Group based access control
- Each BMS node has a producer namespace and multiple data types
 - /ndn/app/bms/ucla/data/; produces: electricity, steam
 - /ndn/app/bms/ucla/melnitz/data/; producer: electricity
- For each producer namespace and data type, we have a group manager; Namespace:
 - /ndn/app/bms/ucla/data/read/electricity
 - /ndn/app/bms/ucla/data/read/steam
 - /ndn/app/bms/ucla/melnitz/data/read/electricity

Encryption/Decryption - cont

- Each group manager keeps the members that can read the corresponding data
 - Identities are in /ucla/read/electricity's group does not have access to /ucla/melnitz/read/electricity, unless explicitly added to the latter's group
- Adding identities to a group
 - Upon receiving group command data (signed by a building manager who's authorized to grant access)
 - Group generates missing E/D-Keys based on the schedule indicated

Implementation Components



(Each line is a face from parent node to child node, with child's prefix registered)

Implementation Components - cont

- BMS nodes
 - PyNDN applications running on mini-ndn
 - Three levels of BMS nodes (building, room, device); each node has its own nfd and BMS node processes; routes are statically configured
 - Panel-level nodes collect data from sensor gateway nodes
- Browser interface for data visualization: NDN-JS

Current Progress

- Current implementation has
 - Gateway publisher (working on real time data)
 - BMS nodes running in mini-ndn
- Todo
 - Group based access control

Thanks

Design Addons

- Parent nodes could keep subsampled raw data. (Express interest with certain timestamp range excluded)
- Parent nodes could keep a full copy of recent data.
- If needed, parent nodes could be configured to calculate aggregates from specific nodes in its subtree, and publish the data under
/`<prefix>`/data/power/aggregation/
`<encoded_list_of_node_names>`/min/0/10
- **Command data verification (node commands, group commands):** verify that a command comes from an identity authorized to give this command