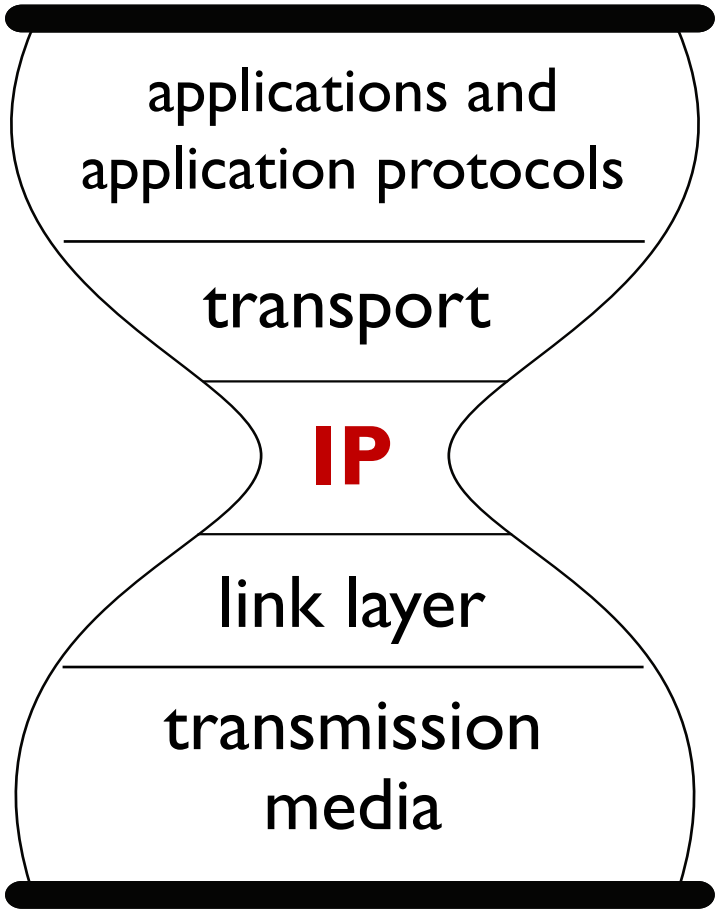
NAMED DATA NETWORKING

Named Data Networking of Things

Alex Afanasyev
Florida International University
aa@cs.fiu.edu

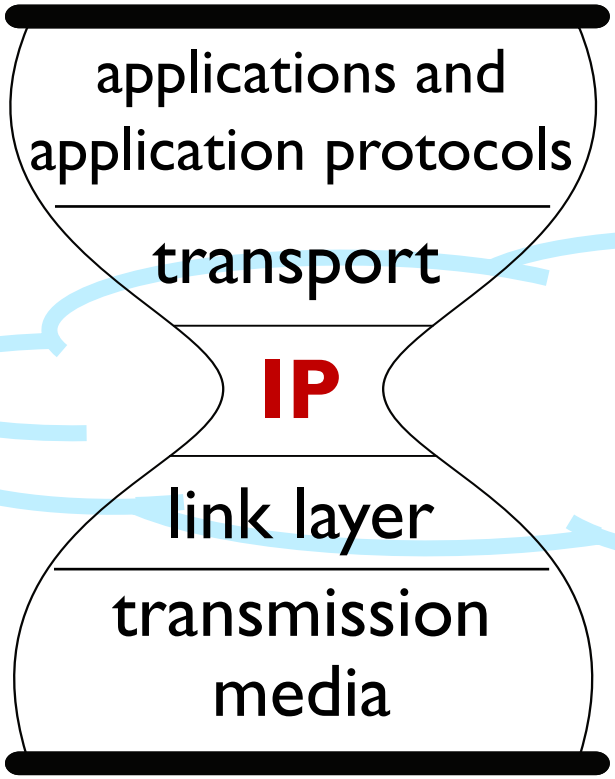
Asian Internet Engineering Conference (AINTEC 2017)
November 20, 2017
Bangkok, Thailand

Hourglass-shaped Internet Protocol Architecture



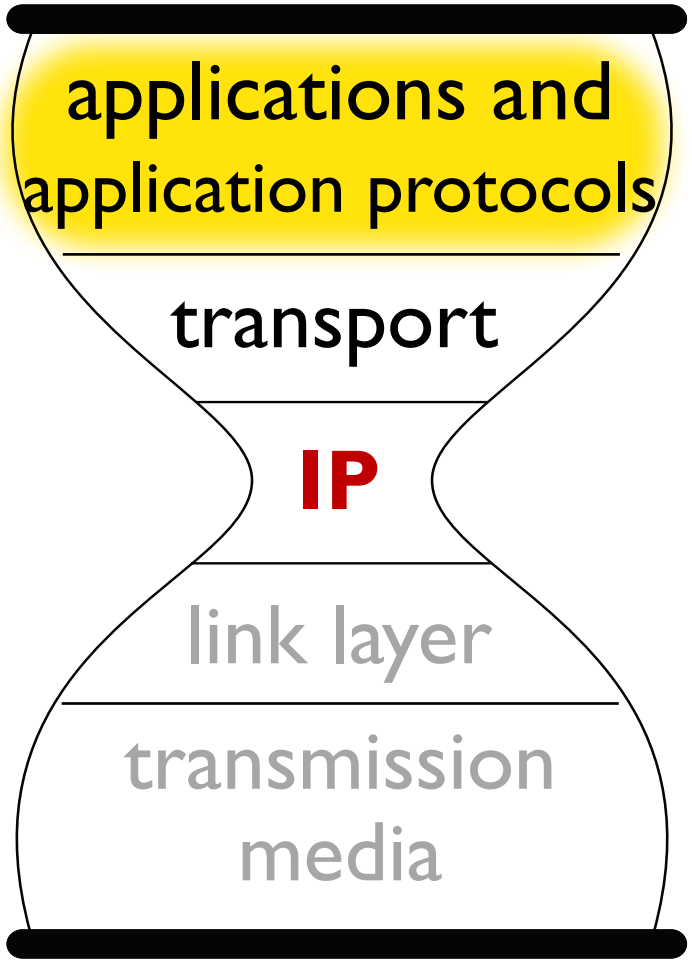
- Transport layer: only sees the two ends
  - Network as a black box
- IP: Delivering packets from any host to any other host
- Link layer: from one node to the next
  - across one-hop

The Magical Power of IP: Routing



- Transport layer: only sees the two ends
  - Network as a black box
- IP: Delivering packets from any host to any other host
  - Running global routing protocol
- Link layer: from one node to the next
  - across one-hop

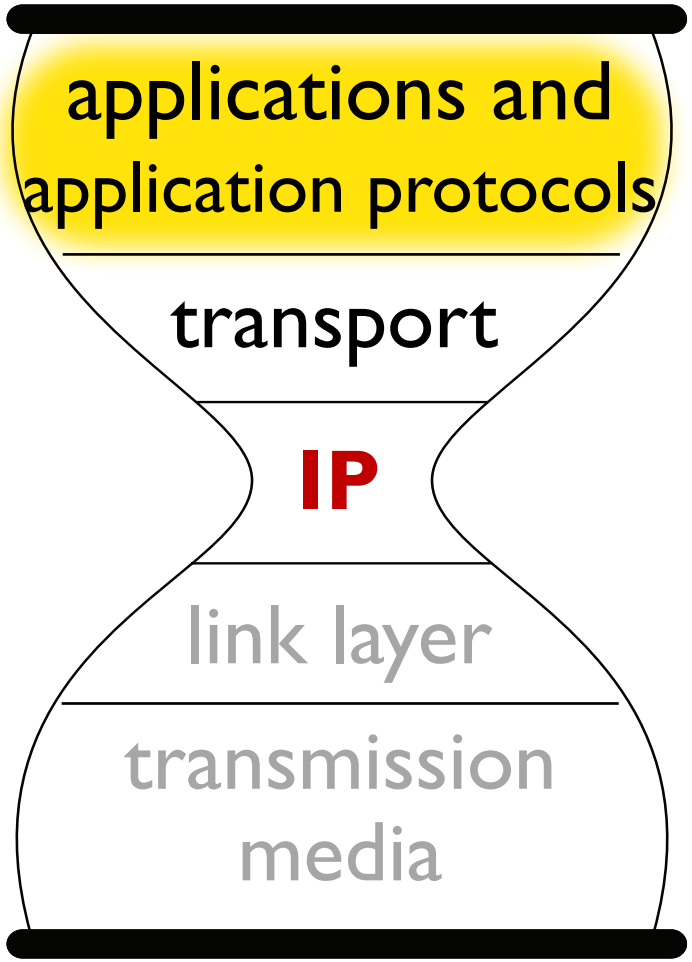
What the Hourglass Picture Doesn't Show

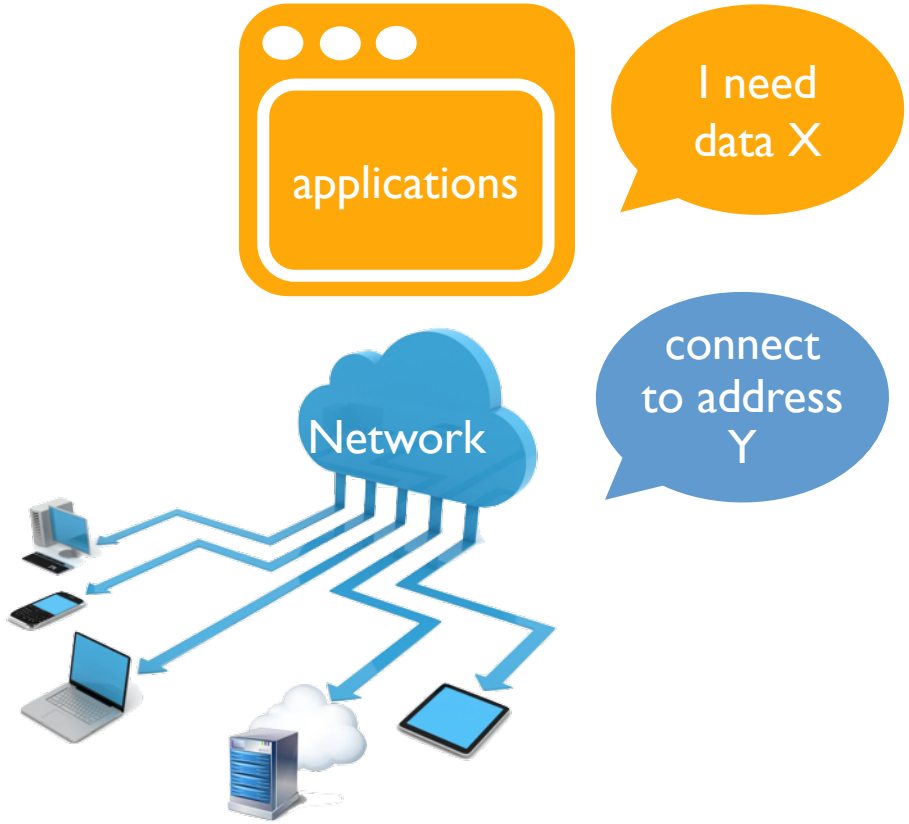


Use of different namespaces

- Applications/application protocols use **names** for *data exchange*
- IP + Transport: a virtual *pipe* between **a pair of IP addresses**
- Link layer: deliver based on MAC addresses
  - ignoring here for simplicity

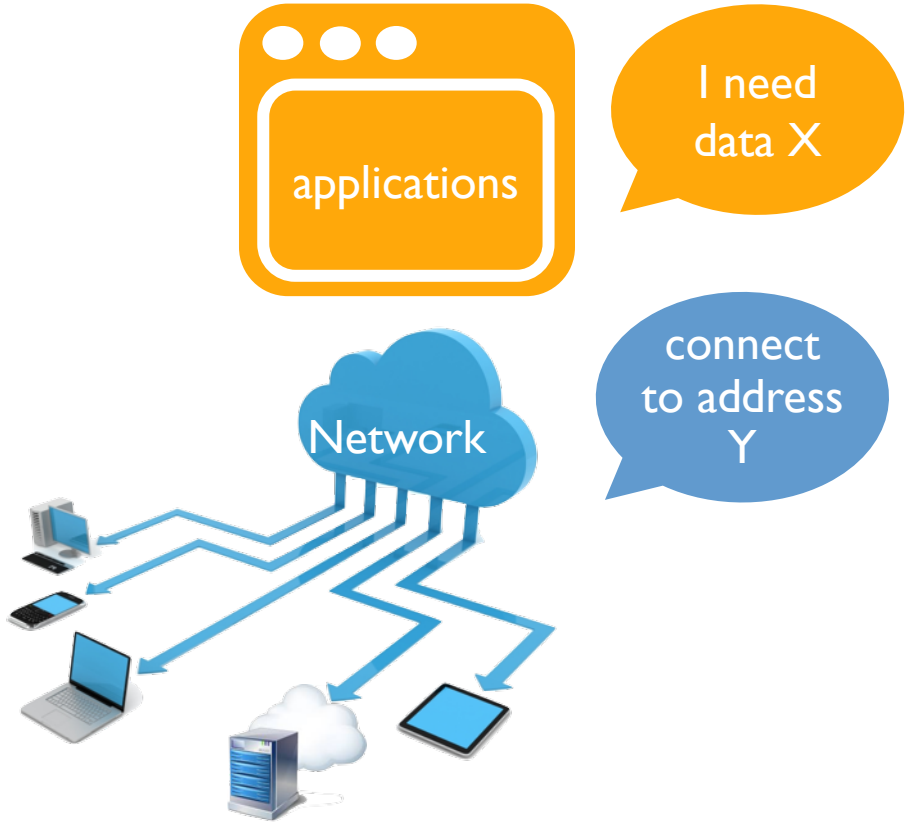
Why Using Different Namespaces Matters





IP: Dependency on infrastructure service/network stability

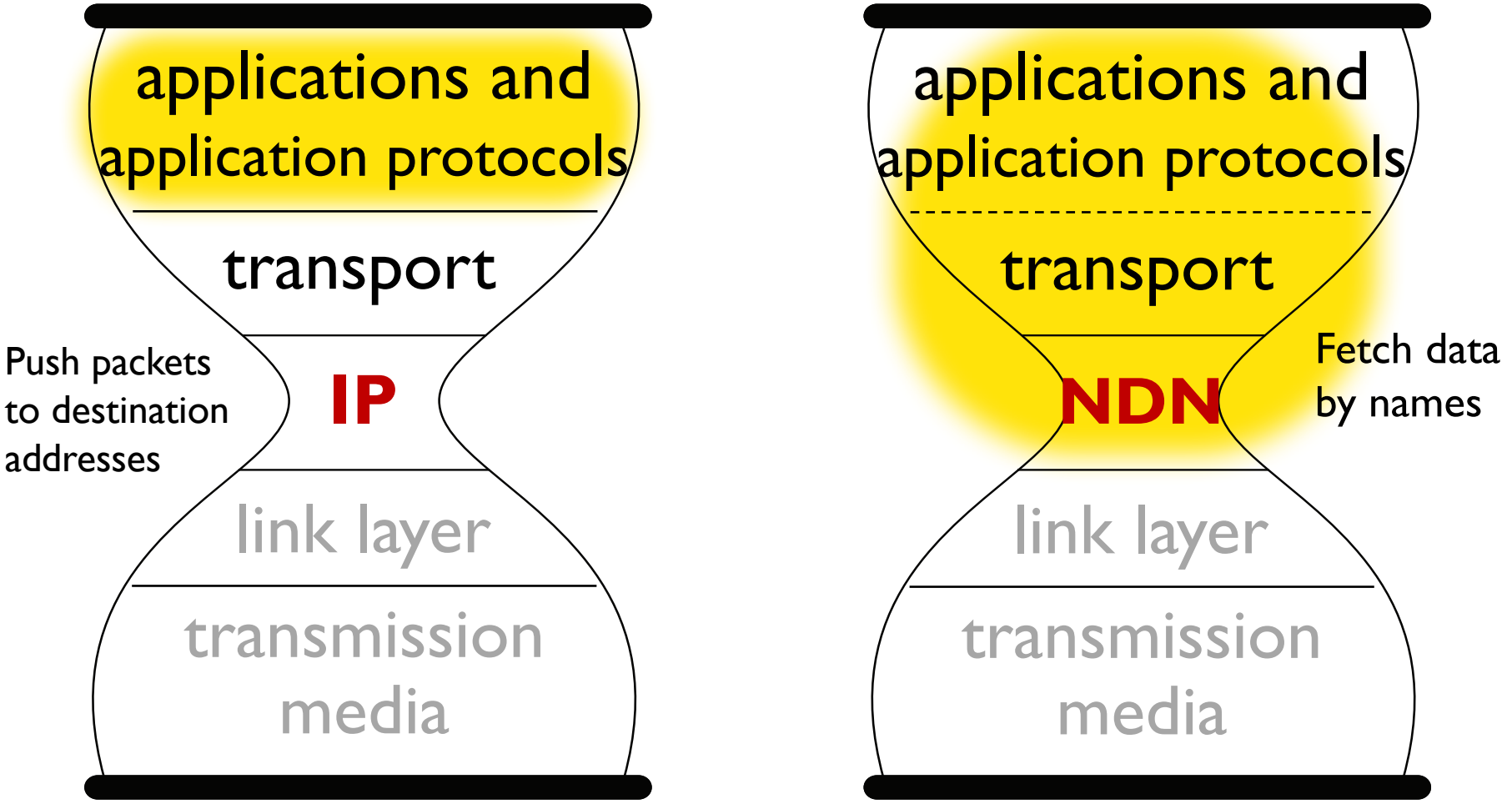
- Dependency on DHCP service:
  - A node must get an IP address before it can communicate
- Dependency on DNS service:
  - application name → IP address
  - require global connectivity (to DNS service) to run local applications
- Dependency on stable end-to-end connectivity



The new way: naming data, instead of naming locations

https://www.youtube.com/watch?v=oCZMoY3q2uM

From IP to NDN: a conceptually simple change

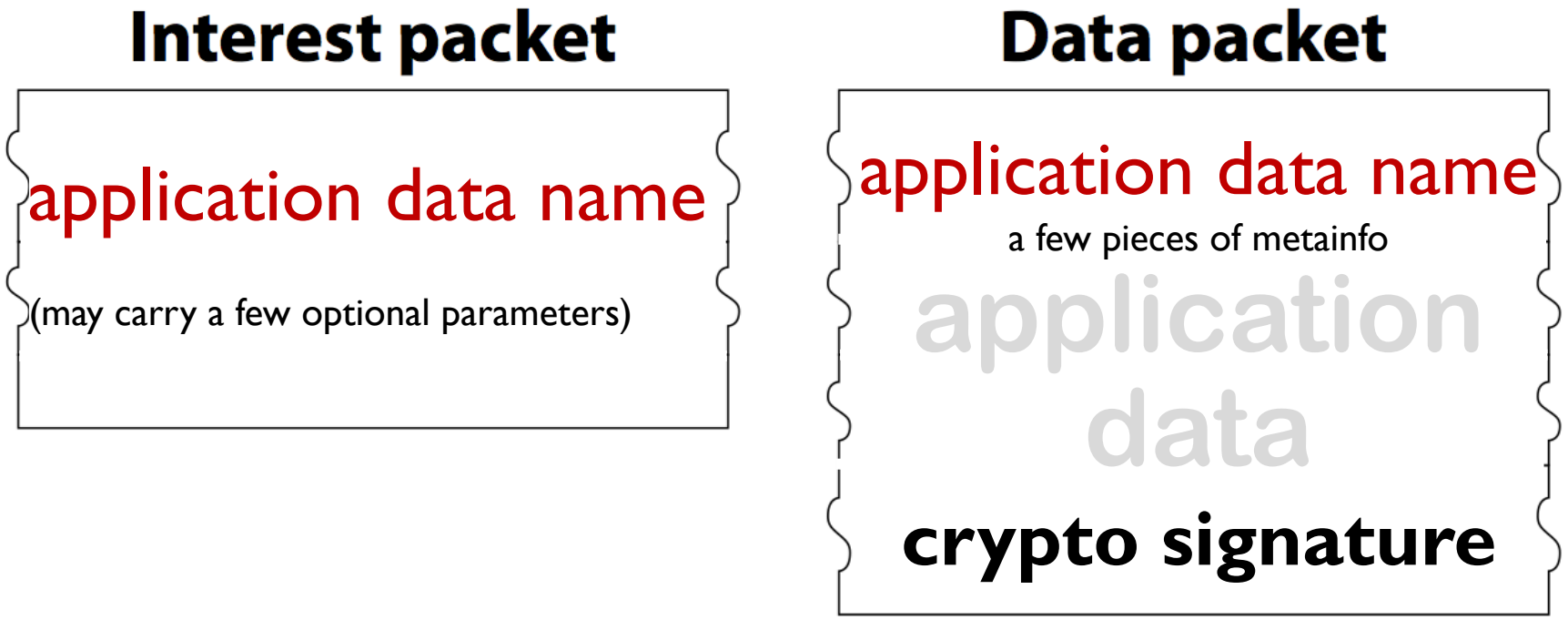


Application and network layer share the same ***data*** namespace

Changing the narrow waist → changing network layer packet semantics

It also leads to changes in semantics of NDN transport layer, making it more inline with applications – leaving that discussion out for now

NDN: 2 types network layer packets



Data consumers send Interest packets

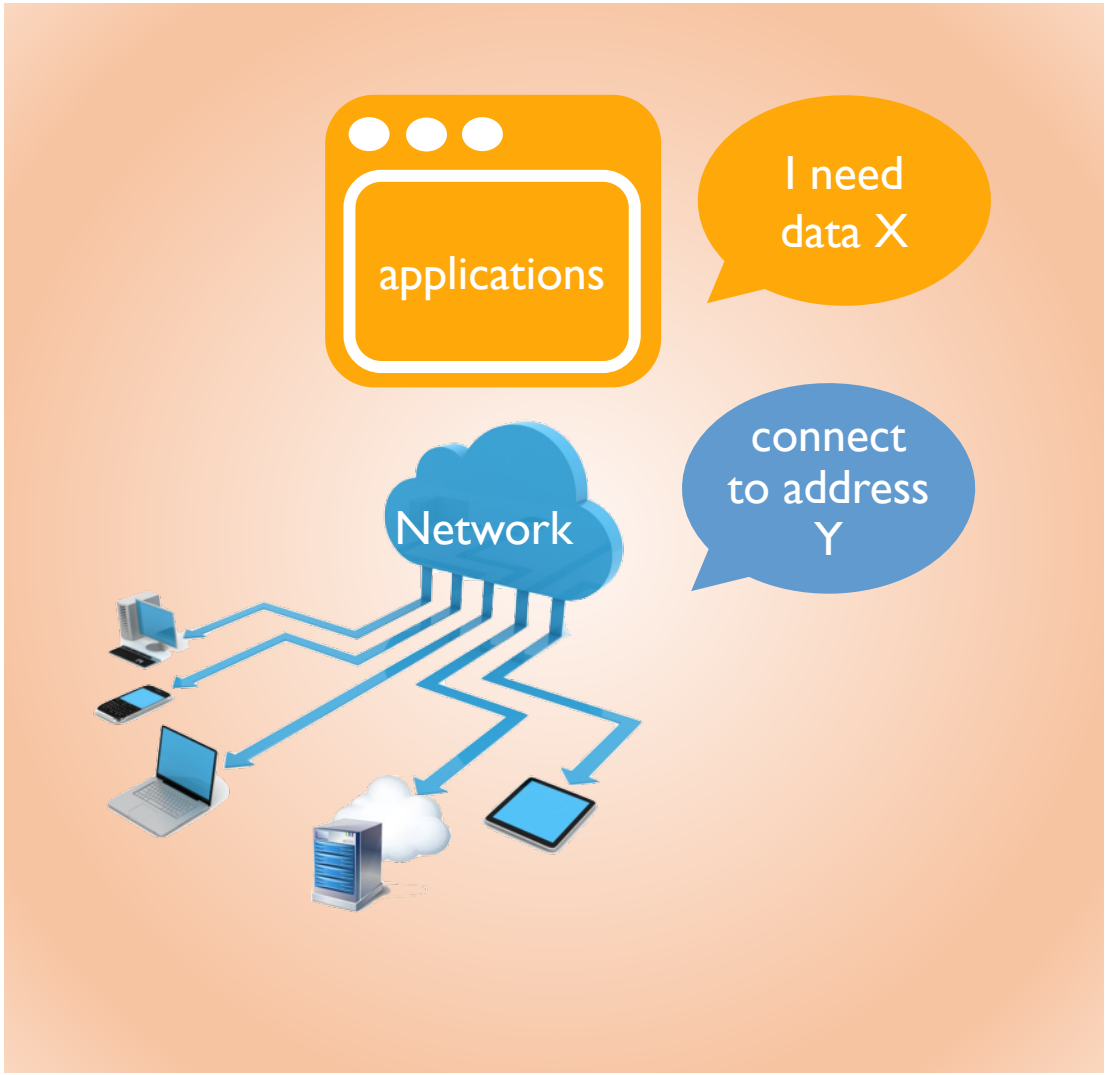
Whoever has the matching Data packet can reply

Publisher binds name to content; receivers verify

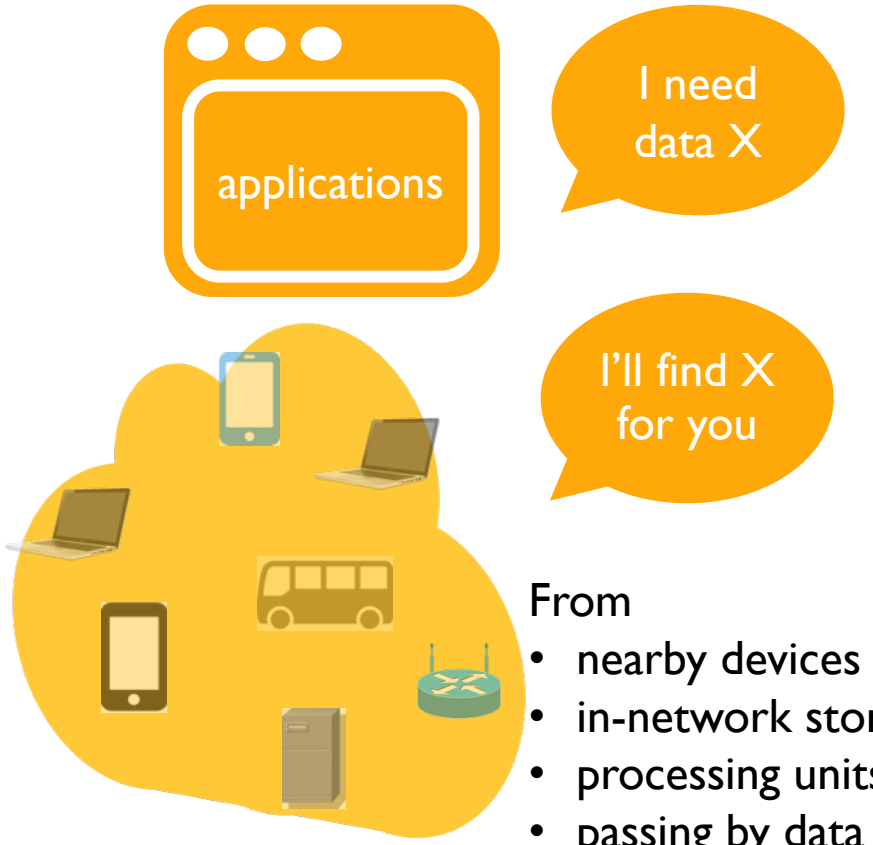
All data immutable

NDN: Independence on infrastructure or stability

Old IP way



NDN way



From
- nearby devices
- in-network storage
- processing units
- passing by data mules via ad hoc connectivity

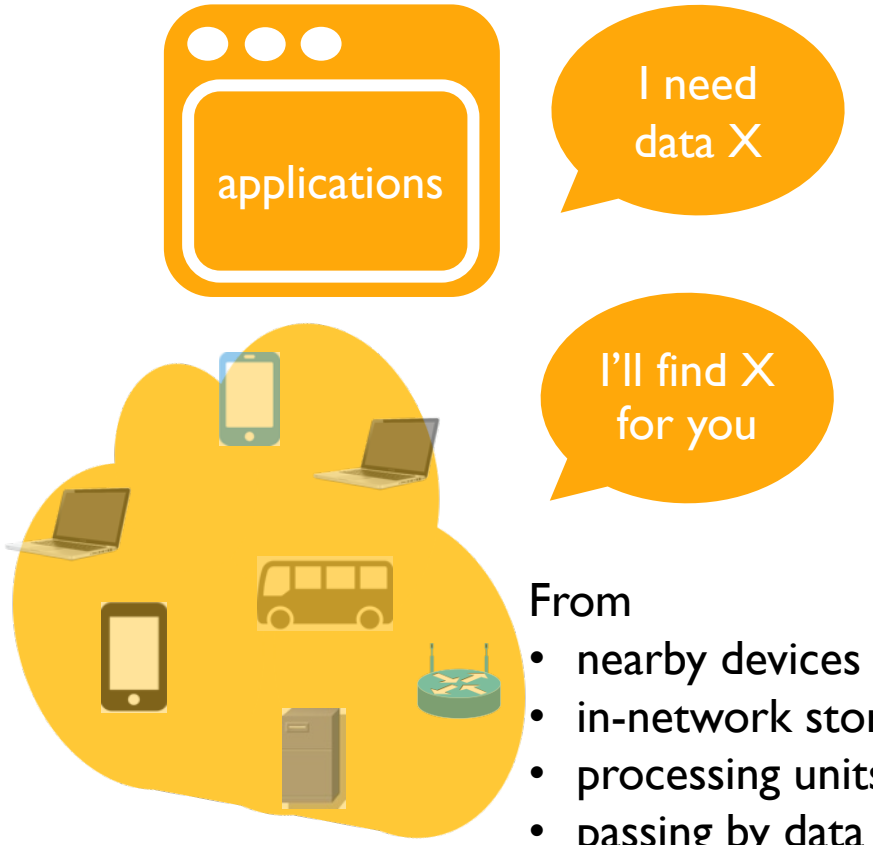
in addition to the original producer

All devices: data suppliers

NDN: Independence on infrastructure or stability

NDN way

- No need for DHCP service
- No need for DNS service
- Applications do not require stable end-to-end connectivity
  - Even when data must be fetched from the original producer: Via NDN's resilient hop-by-hop forwarding

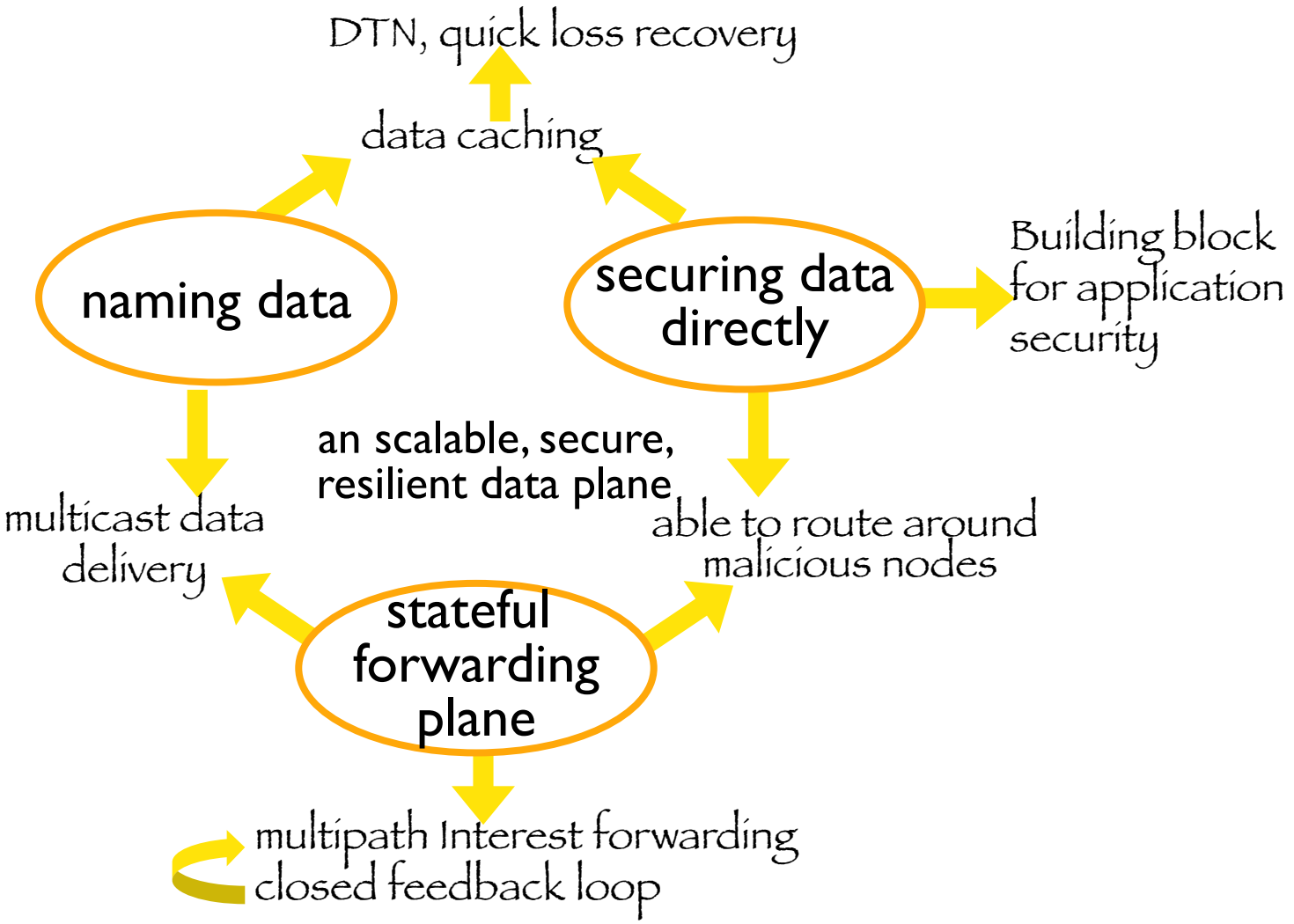


From
- nearby devices
- in-network storage
- processing units
- passing by data mules via ad hoc connectivity

in addition to the original producer

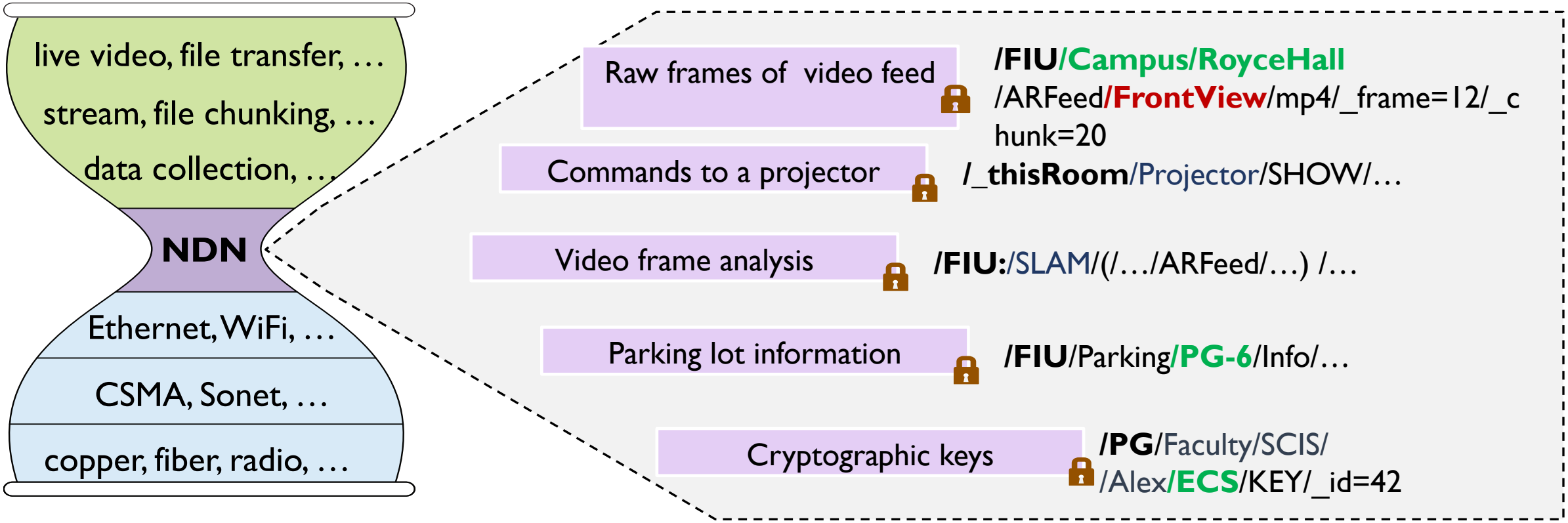
All devices: data suppliers

A Quick Summary on NDN: 3 simple ideas

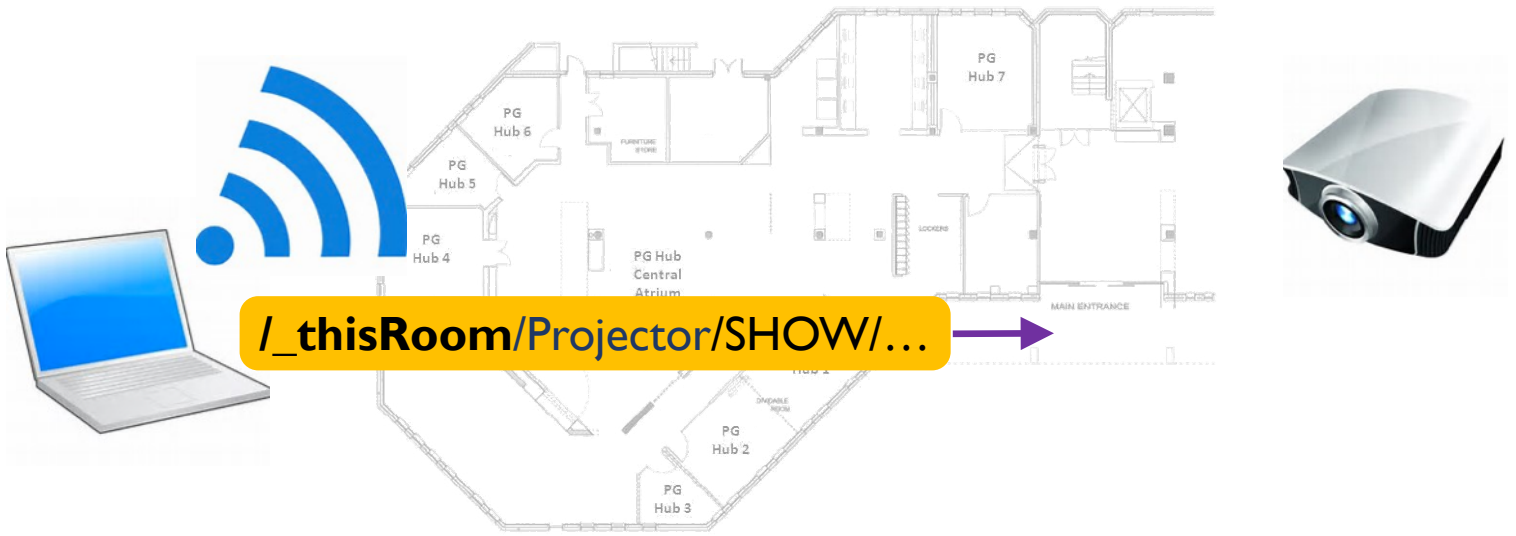


It's all about the *names*

- **Interest and data packets carry names,** no address or port.
  - That's makes all the difference, bringing benefits and challenges
- Names are hierarchical
  - Facilitate name aggregation
  - Preserve application context for data consumption
  - facilitate data authentication, confidentiality support
- Use of naming conventions to facilitate communication

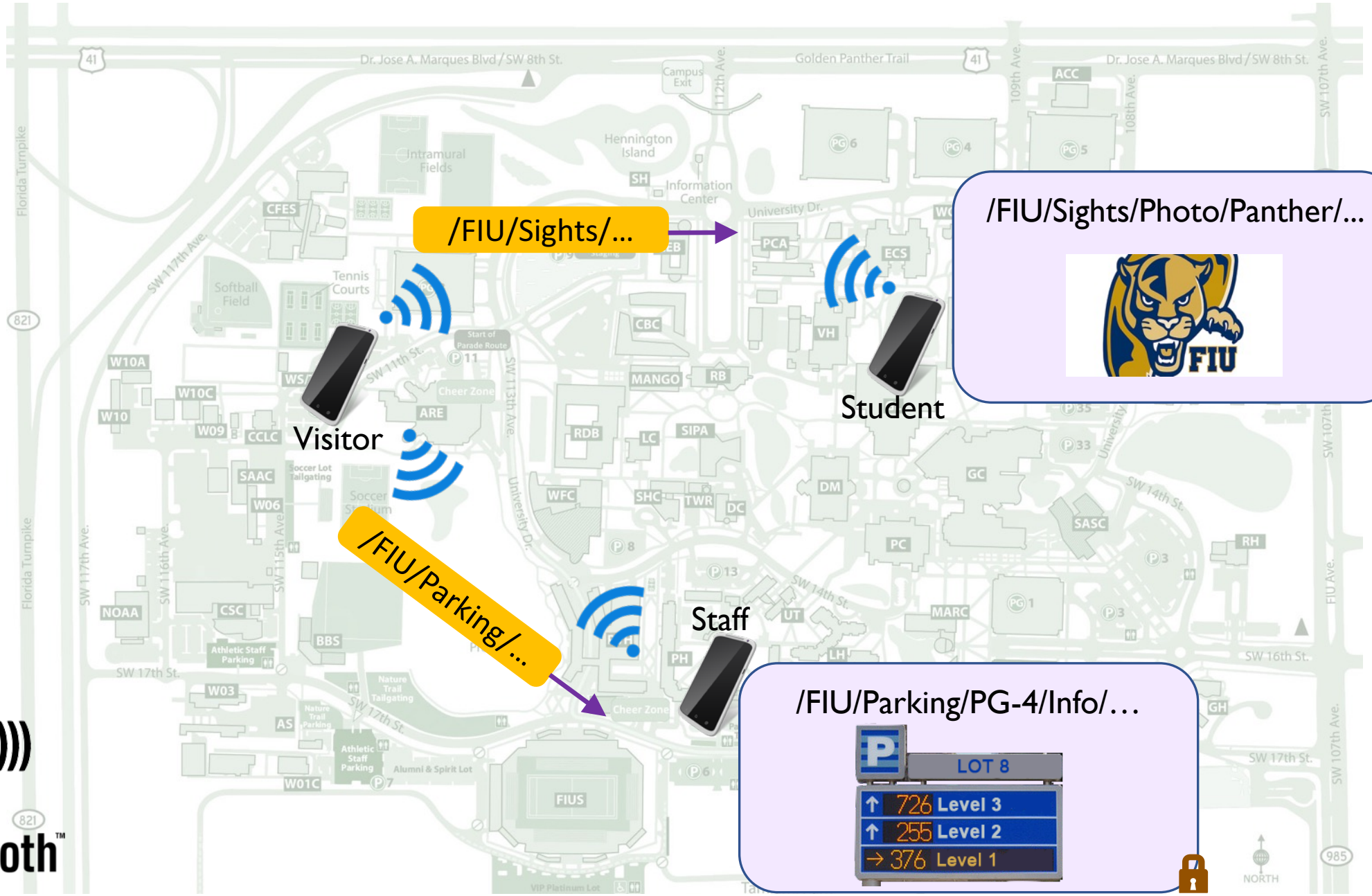


Zero Configuration and Auto Discovery



- Utilizing well defined naming conventions
  - **"/_thisRoom"**: Interest carrying this prefix travels within local one room environment (e.g., one hop)
    - local: WiFi, Ethernet, etc; no long distance like LTE
  - "/Projector": identifies type of the device for which the interest is intended
    - Once projector located, may have further exchange on model/parameter details

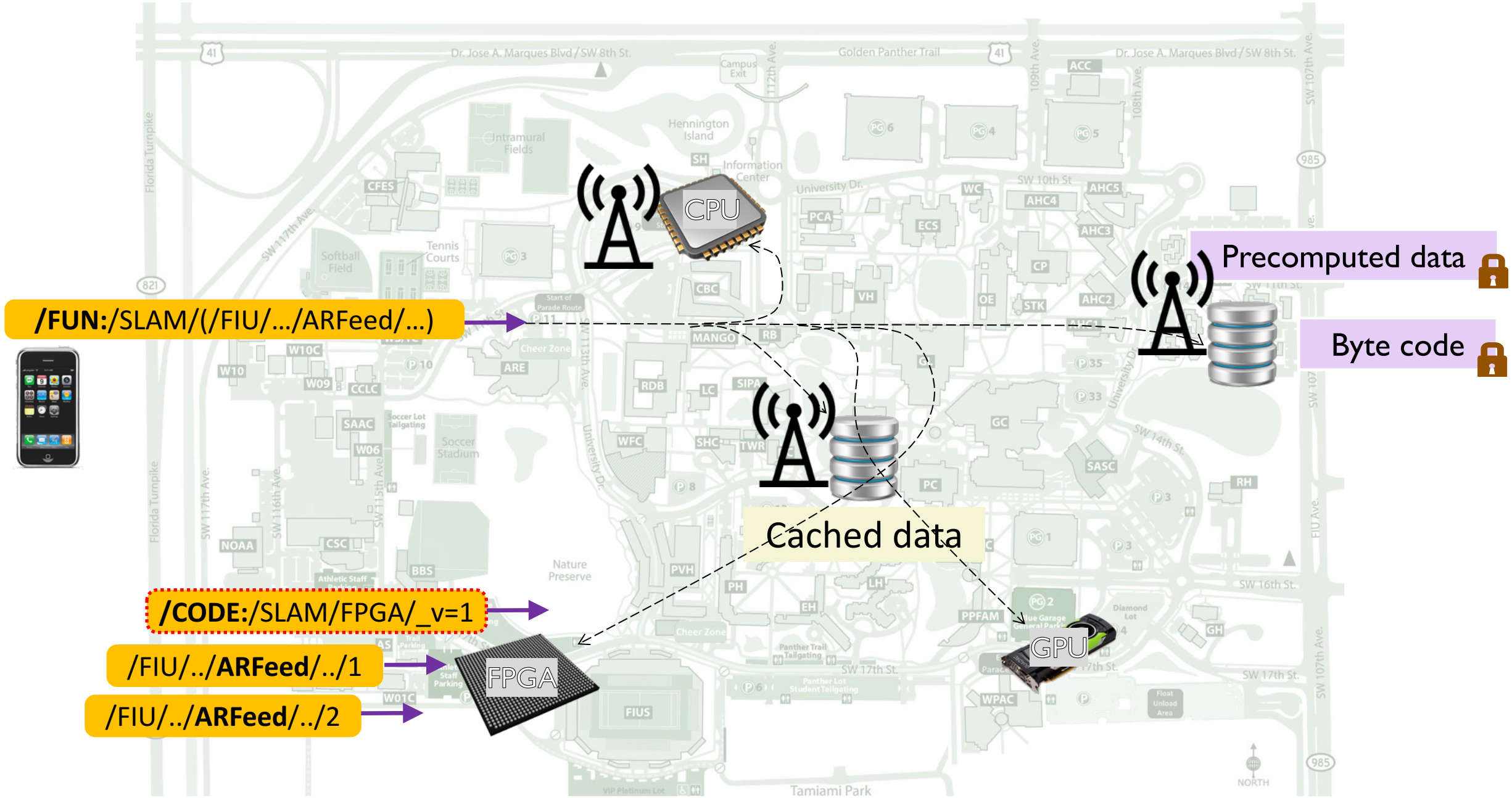
Seamless Ad Hoc Communication



Wi Fi DIRECT

Bluetooth

Integration of Networking, Storage & Computation

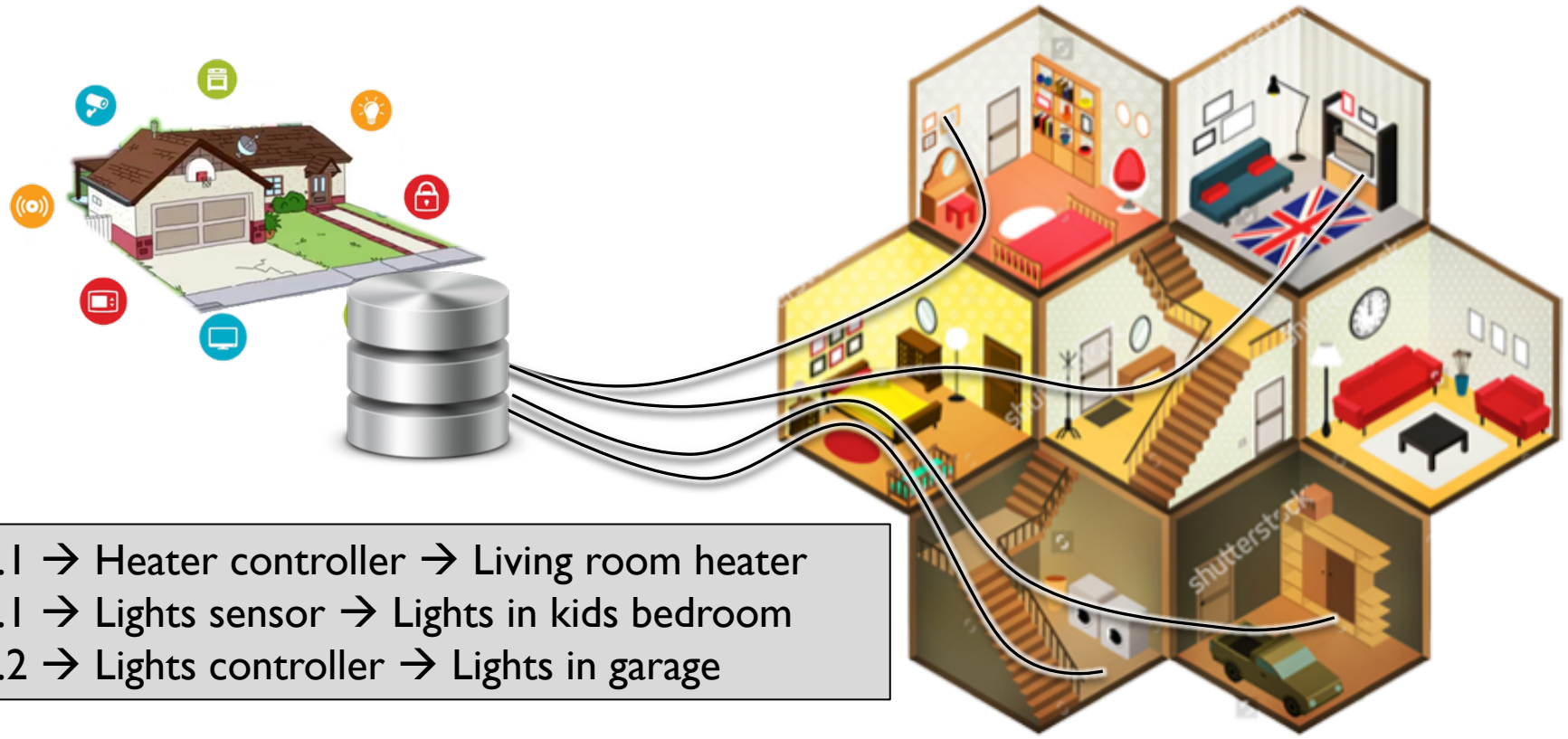


Naming data → Use of Multiple Interfaces

- Requesting data by name, independent from which interface packets may go in or go out
- Can seamlessly use any/all available interfaces/resources

Today's IoT over TCP/IP

- Point-to-point communication model
- Cloud dependency
- Focus on devices that are associated with a "things", not "things" themselves

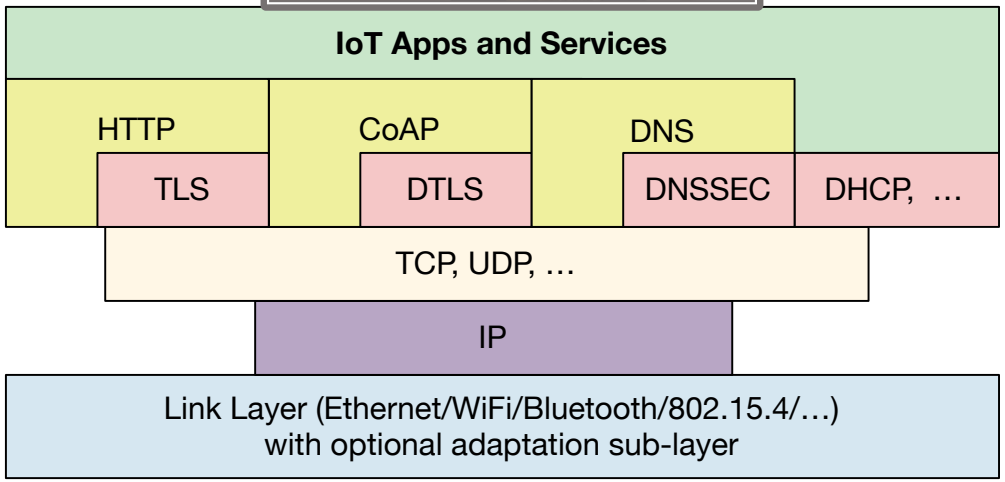


1.1.1.1 → Heater controller → Living room heater
1.1.2.1 → Lights sensor → Lights in kids bedroom
1.1.3.2 → Lights controller → Lights in garage

amazon web services
Microsoft Azure
androidthings
Google Cloud Platform
Weave
iCloud
SAMSUNG SmartThings

Complexity and Semantic Mismatch for IP/IoT

- **App: "Living room frontal view feed"**
- Network:
  - Request stream (HTTP/CoAP)
  - Connect to camera (TCP/IP)
- +
  - Lookup mapping "Living room" -> camera URI
  - Connect to AlexHome.com (cloud?) service
  - DNS lookup IP of AlexHome.com service
  - DHCP to assign IP addresses to all devices



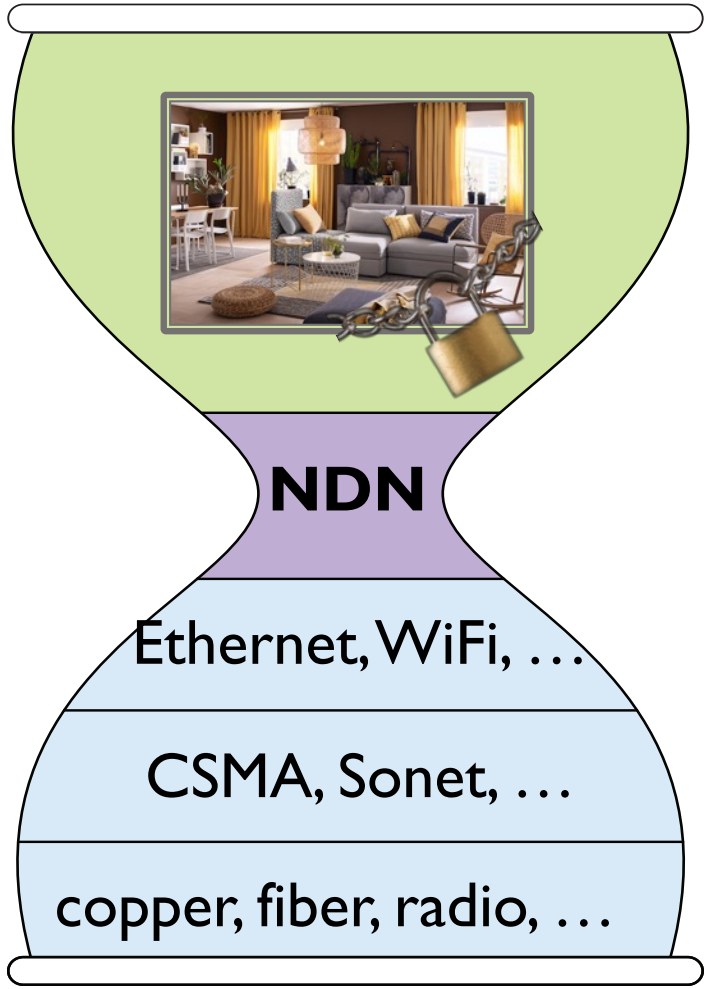
NDN "Edge" for IoT

- Bring IoT semantics to the network layer
- Name the "things" and operations on "things"
  - "Living room frontal view feed", "CO level in kitchen"
  - "blood pressure", "body temperature"
  - "max/min/avg pH of soil in specific point of US soil grid"
- Focus on data associated with things, not devices
- Secure data directly

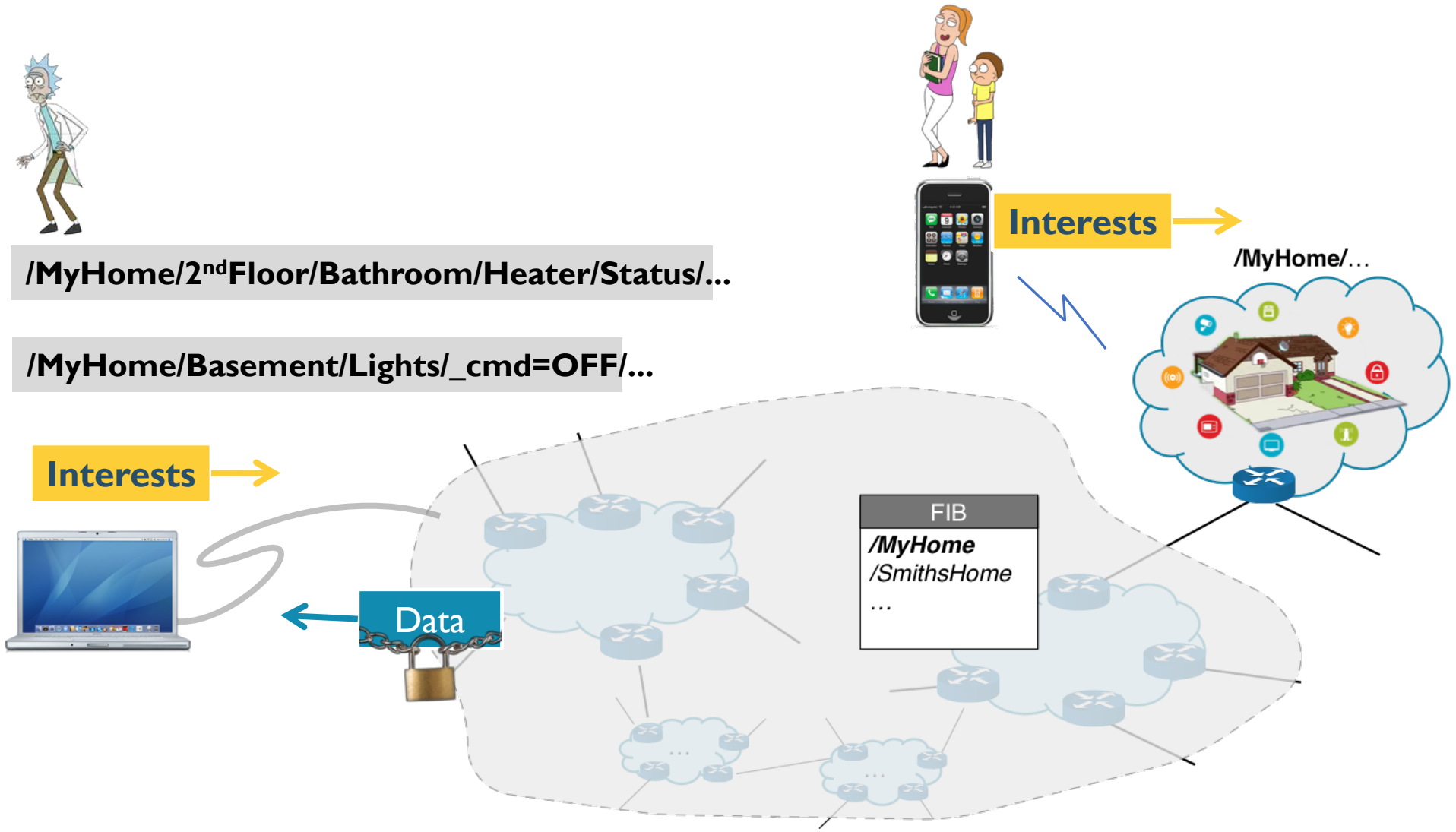
W. Shang, Z. Wang, A. Afanasyev, J. Burke, and L. Zhang, "Breaking out of the Cloud: Local Trust Management and Rendezvous in Named Data Networking of Things," in *Proceedings of IoTDI*, 2017.

Named Data Networking of Things

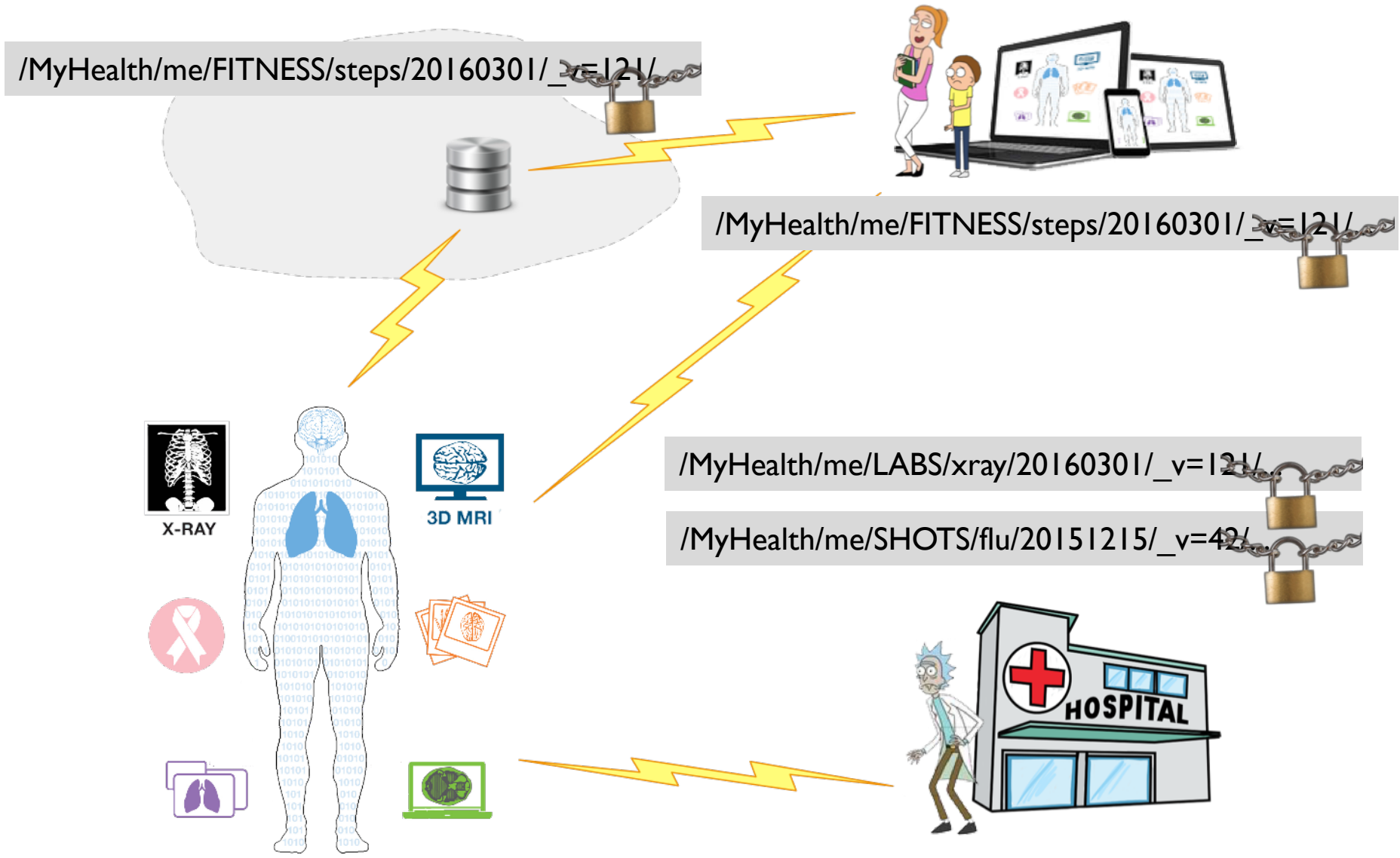
- **App: "Living room frontal view feed"**
  - /AlexHome.com/LivingRoom/VideoFeed/FrontView/mp4/_frame=12/_chunk=20
- Network:
  - Use the name to send request to my camera responsible for Living's room front view
  - OR retrieve data from caches
- +
  - Cameras provision with "identity name" that defines what they are and what data names they produce
  - Can announce name prefixes or respond to local broadcasts



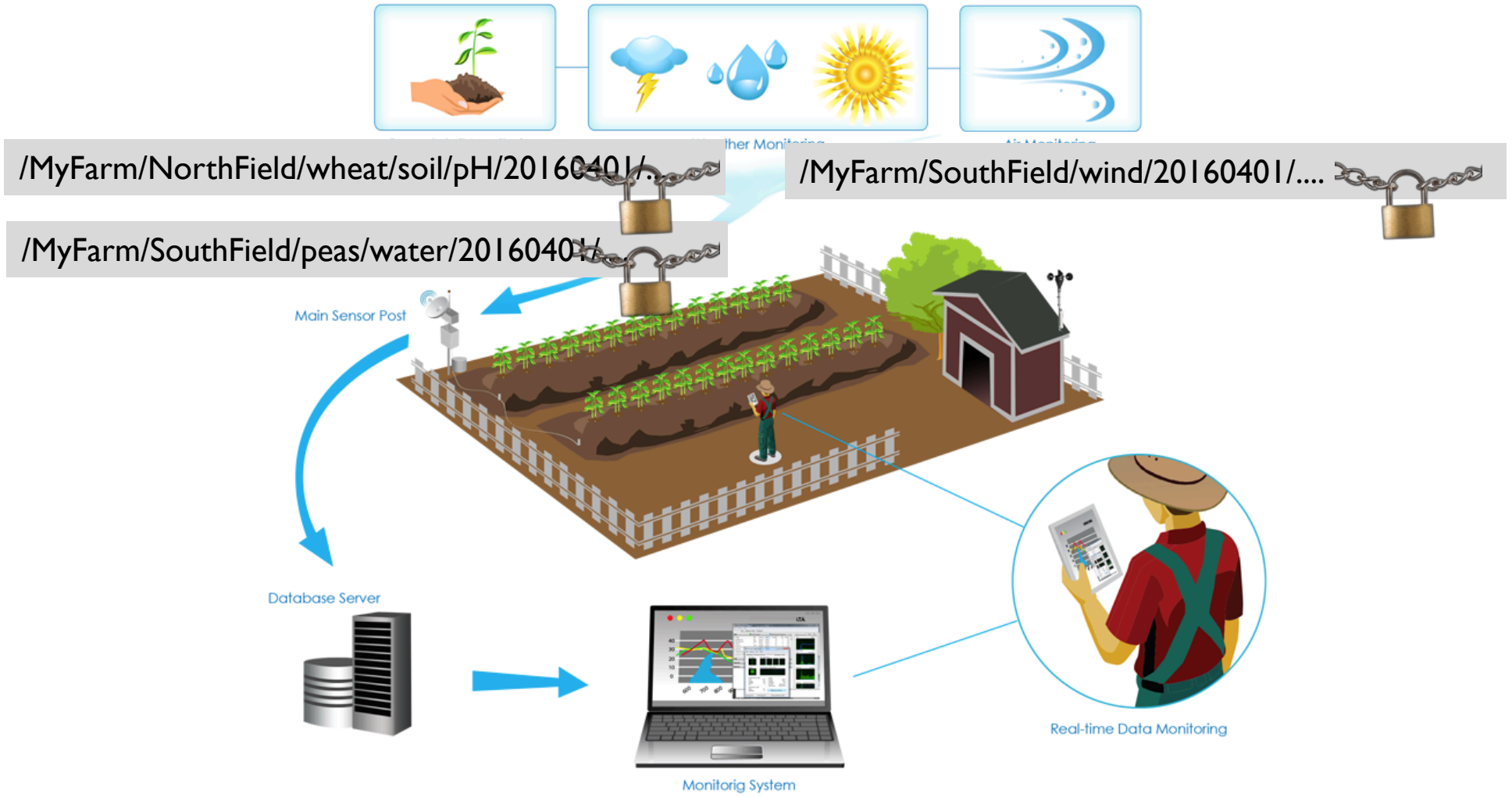
Accessing Smart Home



NDN Personal Health

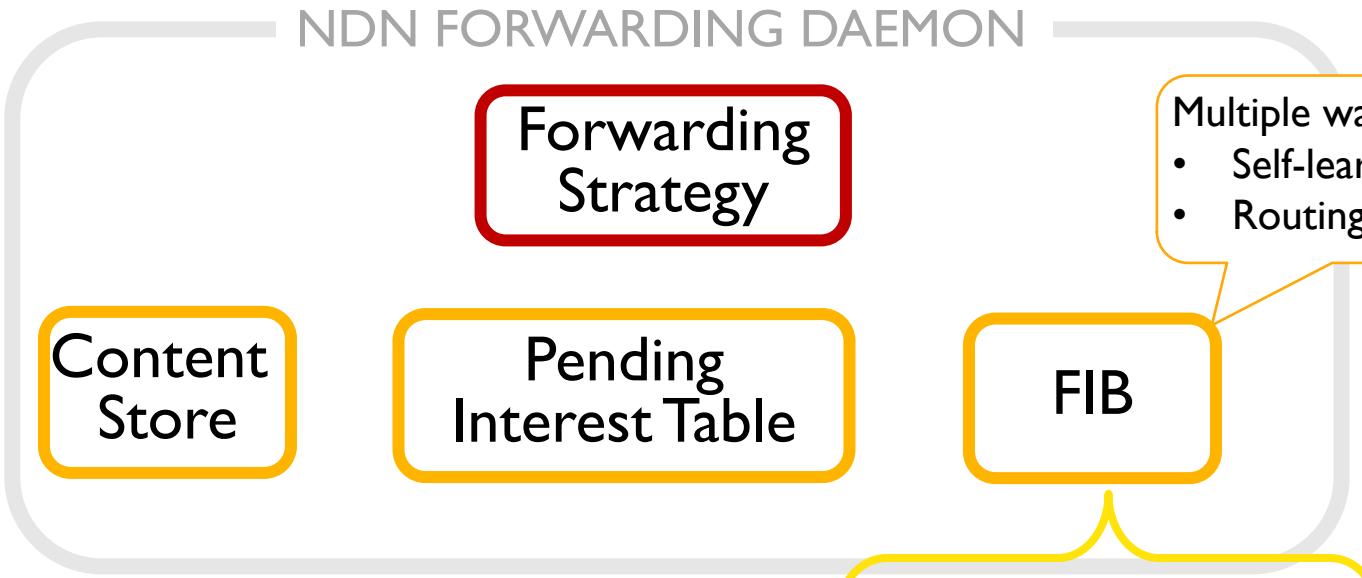


NDN Precision Agriculture



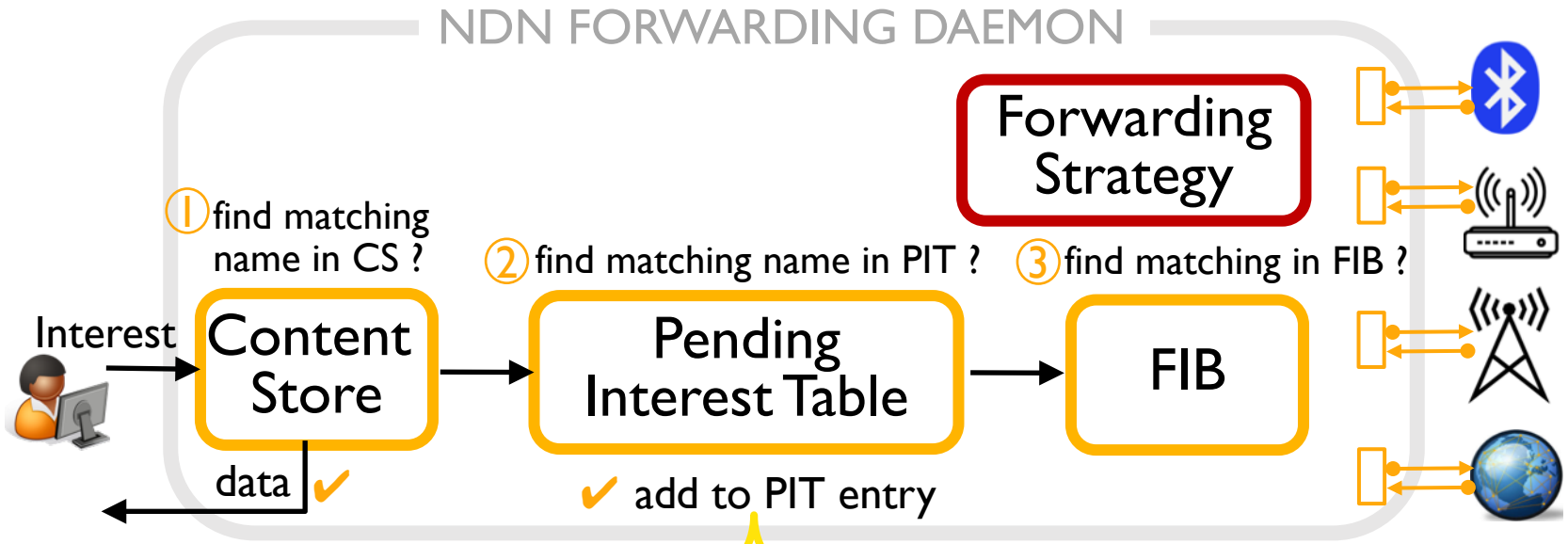
Stateful Forwarding

NDN's node model



NFD module resides in every NDN node

NDN Interest Forwarding: 3 steps

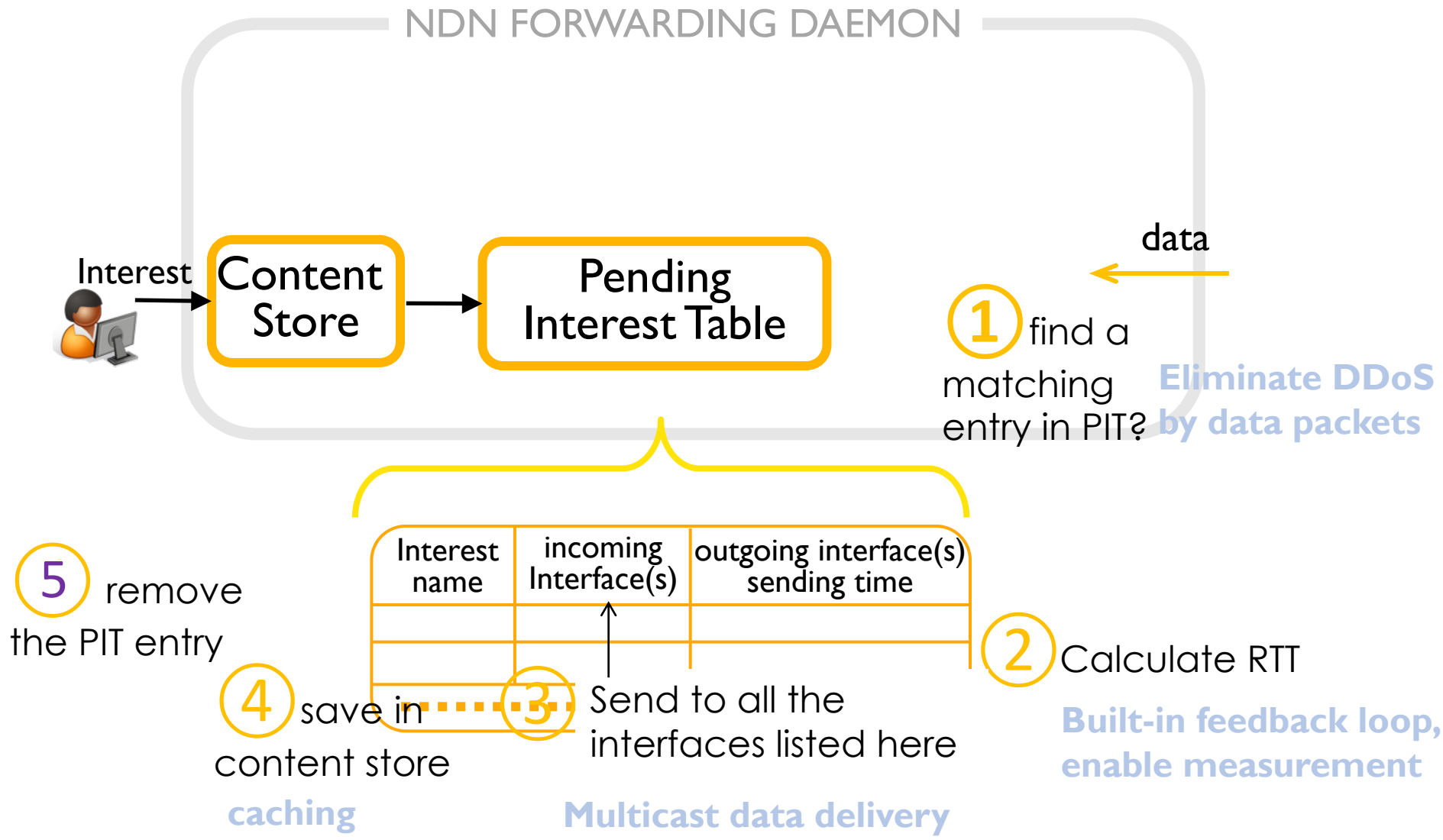


For each PIT entry:

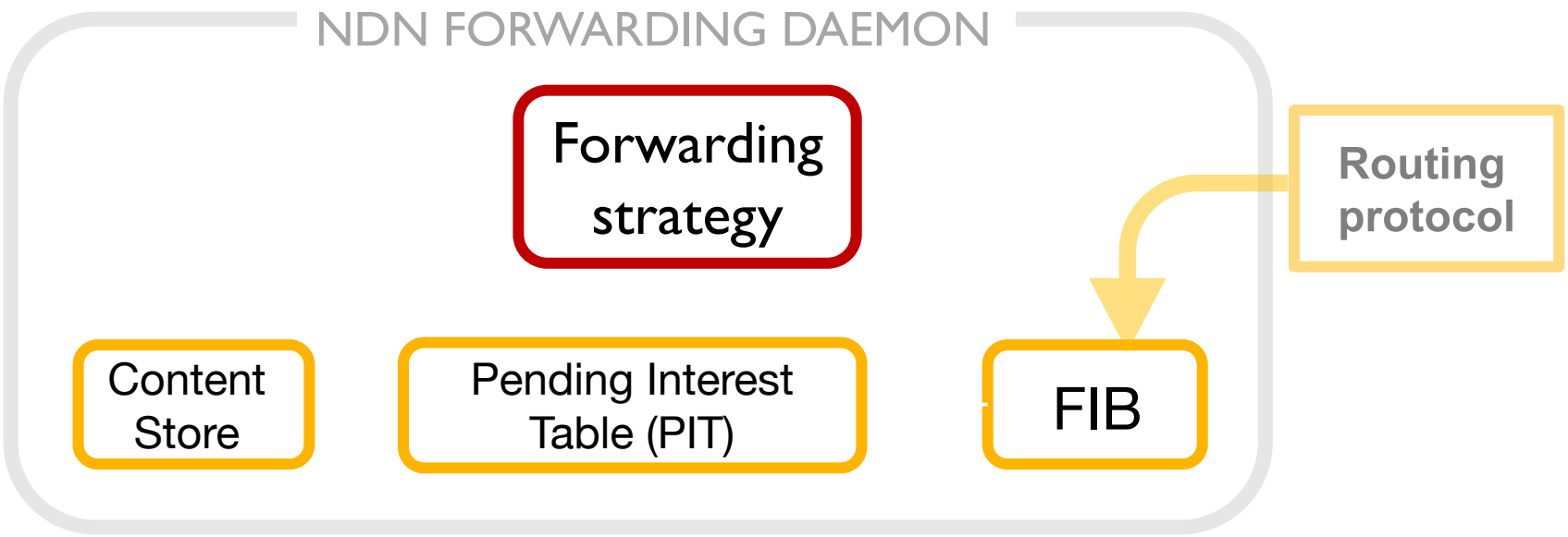
Interest name	incoming Interface(s)	outgoing interface(s) sending time
▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪		

May forward an interest packet out through one or more interfaces

NDN Data Packet Return



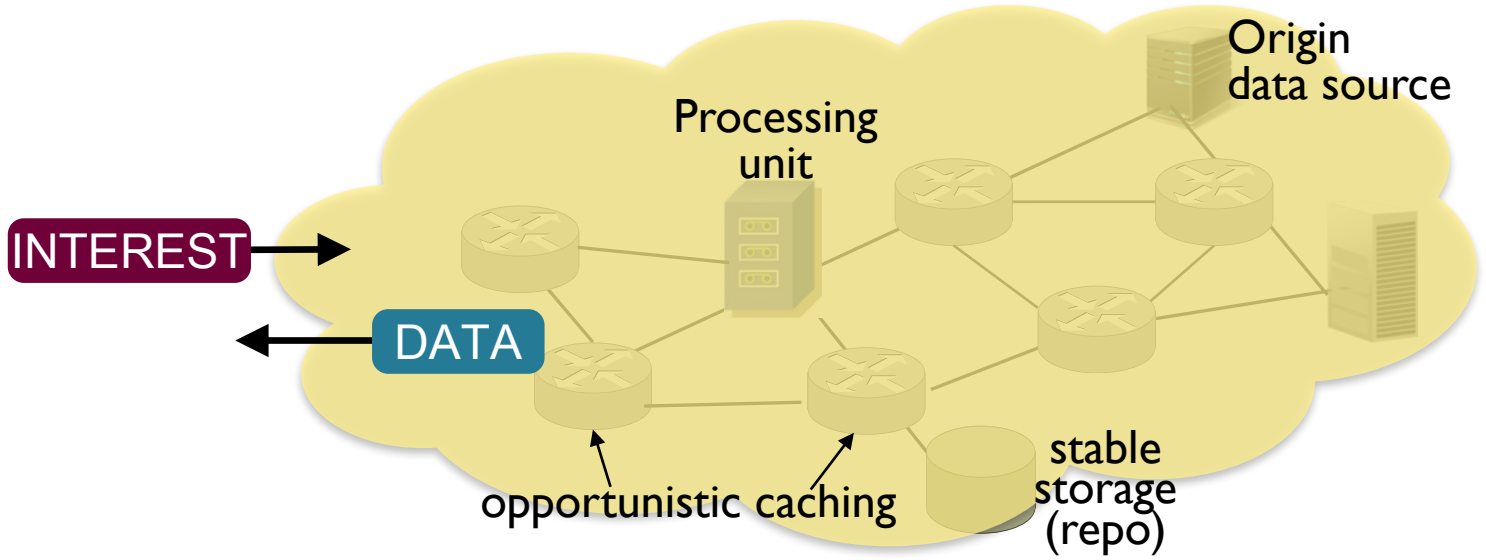
Forwarding Strategy



- Forwarding Strategy makes interest forwarding decisions by taking input from
  - FIB
  - measurement from Interest-data exchange (and any other local resource information)
  - *Per-namespace forwarding policies*
- a hook to control plane

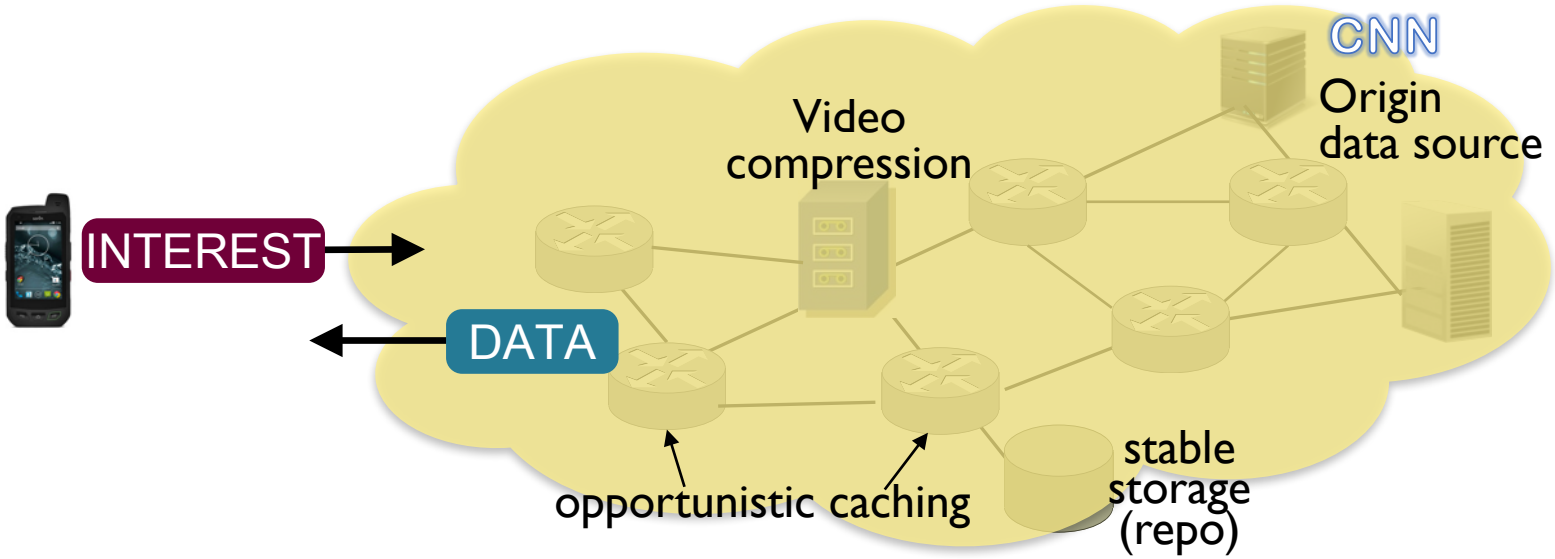
Steering interest packets toward matching data

- Data source, storage, and processing units can all supply requested data
- With no prior knowledge: may try broadcast discovery, or even random walk
- Building knowledge of directions to reach data: routing announcement

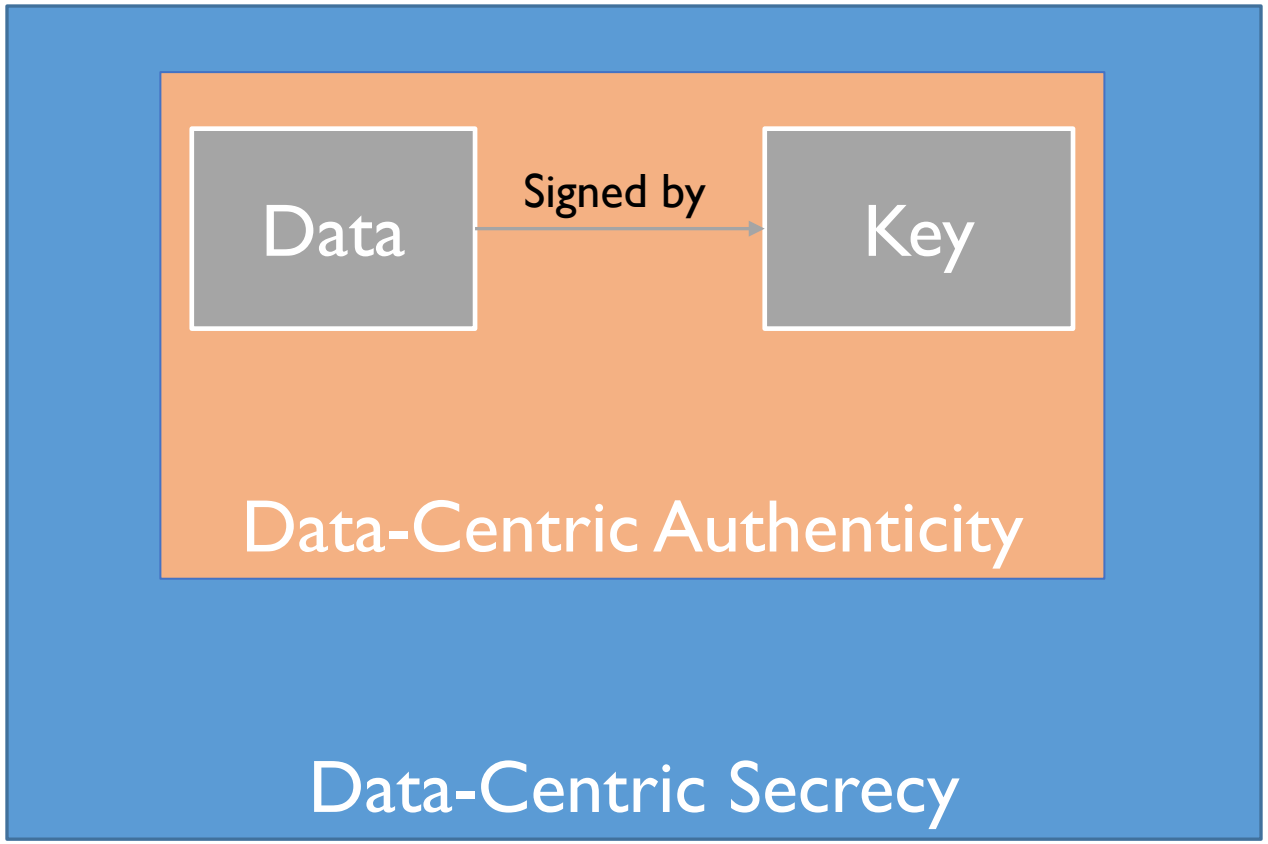


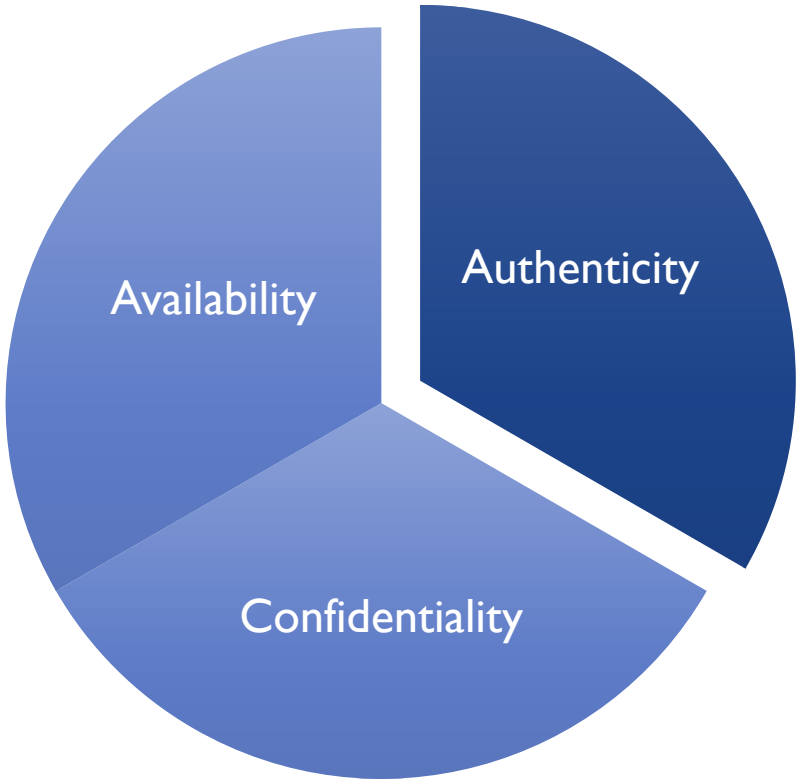
Steering interests toward data via routing announcement

- Data producer: announce name prefixes
  - Opportunistic caching: supply data as interests come along
- Managed repos: may announce name prefixes
- Process unit: announce service name (e.g. /video-compression)
  - request: /video-compression/cnn/20170927/



Data-Centric Security of NDN

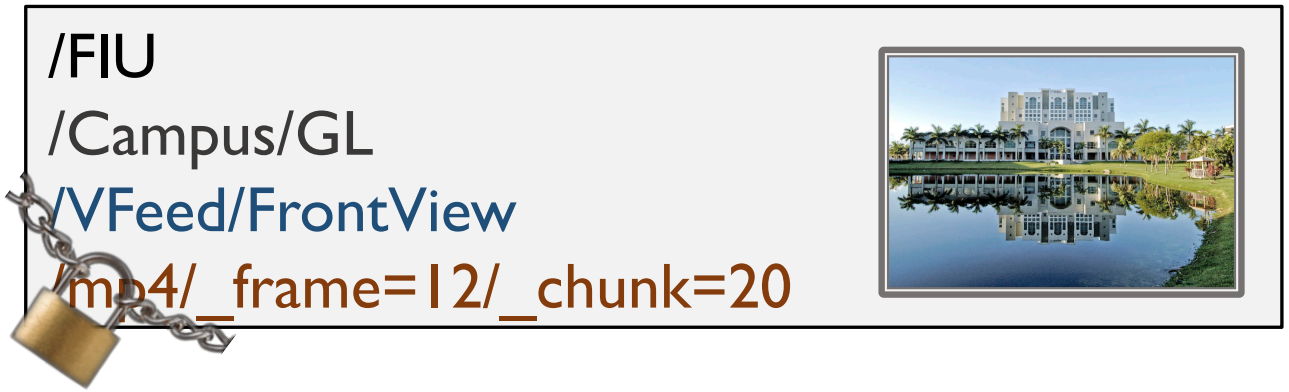




NDN Security Support

- Producer signs data right after production
- The signature cryptographically binds the (semantically meaningful) name to the content
- Eliminating security dependency on lower layers or intermediaries
- Truly end-to-end security for applications
  - From producer to consumer

- **Information is secured in motion and rest**



Authentication of NDN Data

/FIU
/Campus/GL
/VFeed/FrontView
/mp4/_frame=12/_chunk=20

Signed by

/FIU/Campus/GL/Manager/
KEY

Signed by

KeyLocator: /FIU/Campus/GL/Manager/KEY

KeyLocator: /FIU/Campus/Manager/KEY

Defining Trust Model for My Smart Home

- GL video feed can only come from a camera in pointing to GL
- Cameras pointing to GL can be configured only by GL manager
- Only campus manager can authorize GL managers to configure cameras for GL

/FIU/…/**VFeed**/…/KEY

signs

/FIU/Campus/GL/Manager/KEY

signs

Local trust anchor

/FIU/Campus/Manager/KEY

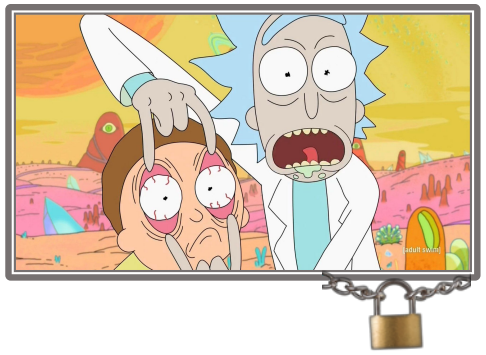
Key Privilege Separation

/FIU/Campus/GL/VFeed/FrontView
/mp4/_frame=12/_chunk=20

/UCLA/Camera/…/Campus
/RoyceHall/Camera/KEY

A frame from a camera installed in the Royce Hall

/FIU/Campus/GL/ARFeed/FrontView
/mp4/_frame=12/_chunk=20

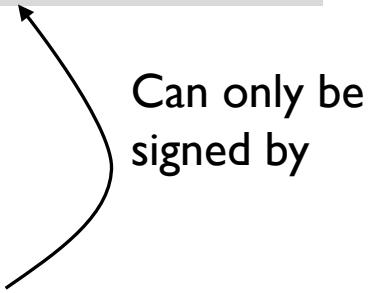


/Samsung/TV/KEY

A forged frame

Name-Based Limit of Key Power

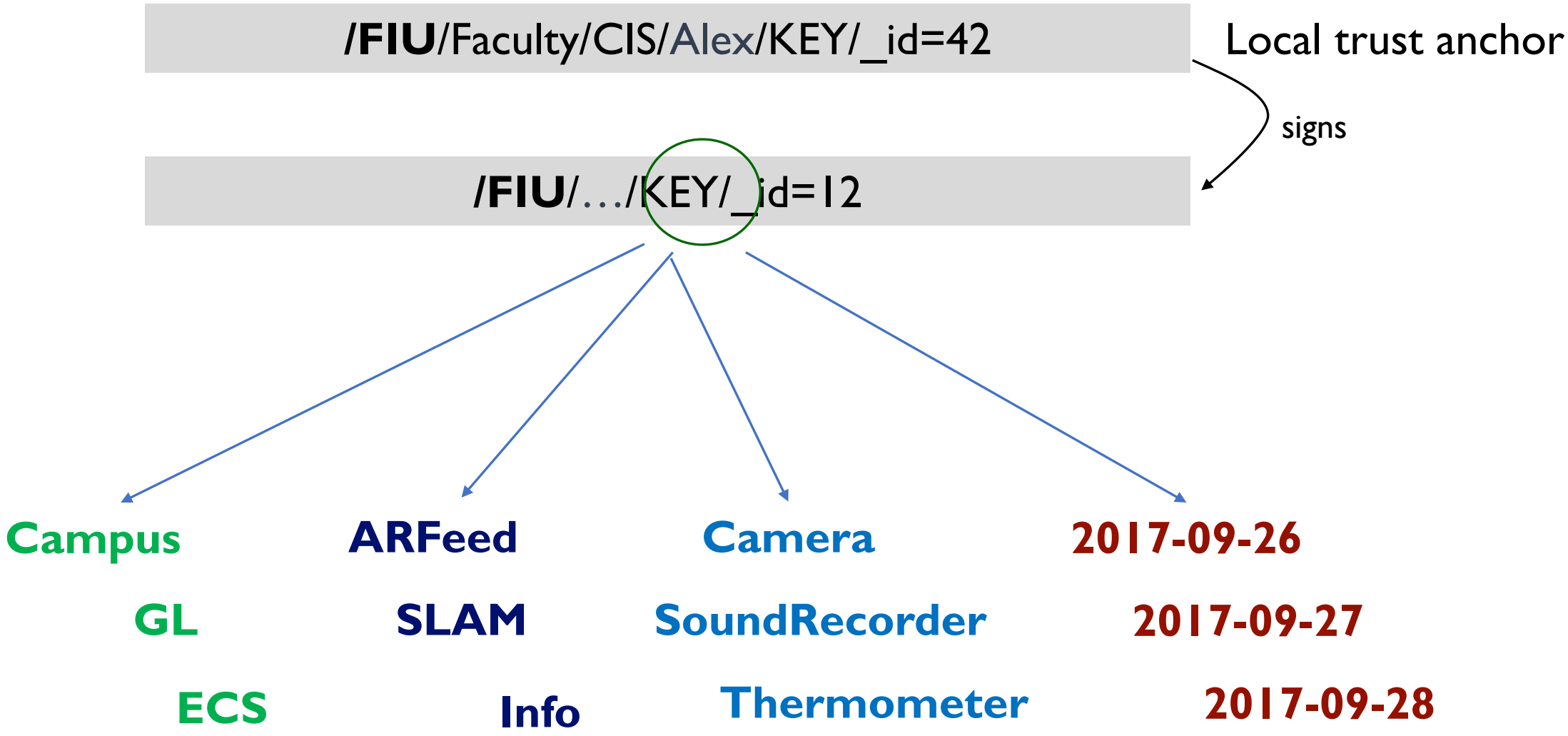
/FIU/Campus/GL**/VFeed**/…/mp4/_f=…/_s=…



/FIU/Cameras/_id=…/GL/…/KEY/_id=…

VFeed data to be valid, must be signed with a "Camera" key under the same name hierarchy

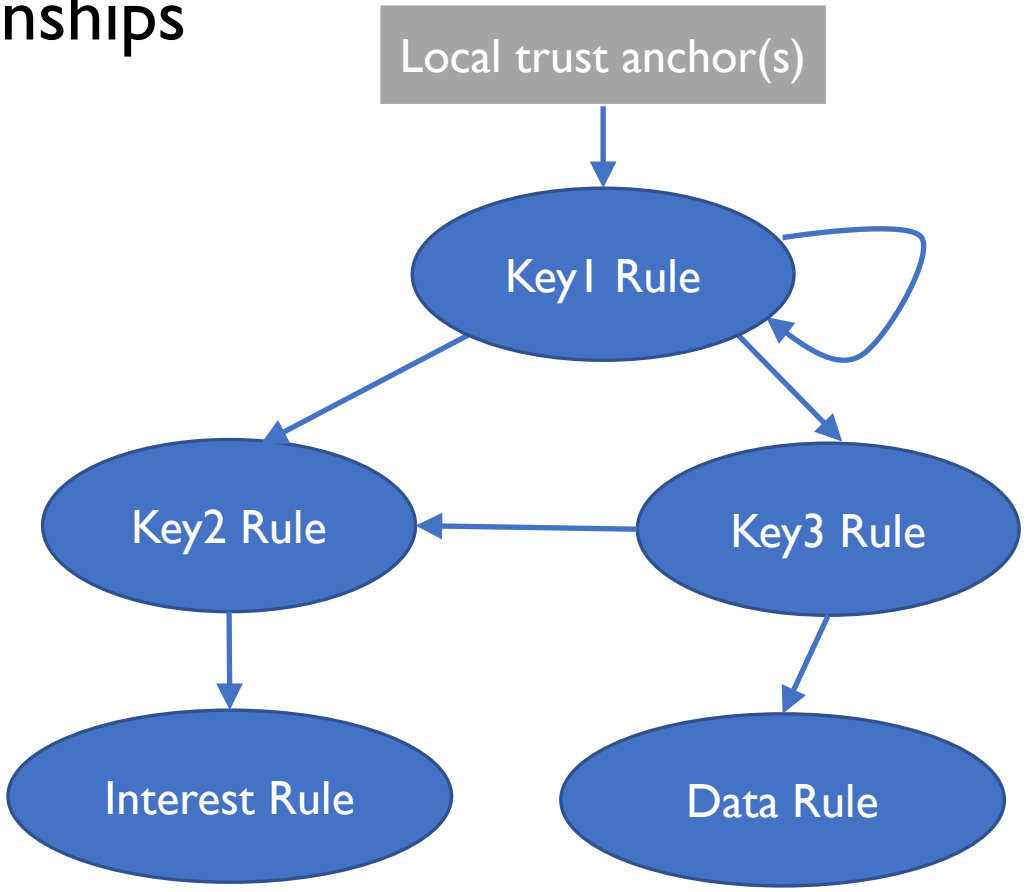
Flexible Restrictions through Namespace Design



Trust Schema: Name-Based Definition of Trust Model

- A formal language to formally describe trust model
  - Schematize data and key name relationships

```
<>          <CONST>
token*      token?
[func]
(:group:token)
```



"Schematizing Trust in Named Data Networking", ACM ICN 2015

An Example of Trust Schema for Smart Campus

(:Prefix:<>*)(:Location:<>?)<ARFeed>**[View]**<mp4><frame><chunk>

Camera(Prefix, Location, View)

(:Prefix:<>*)<Cameras>[cam-id]**(:Location:<>?)**<View>**[View]**<KEY>[key-id]

Faculty(Prefix, Location)

(:Prefix:<>*)<Faculty>[user]**(:Location:<>?)**<KEY>[key-id]

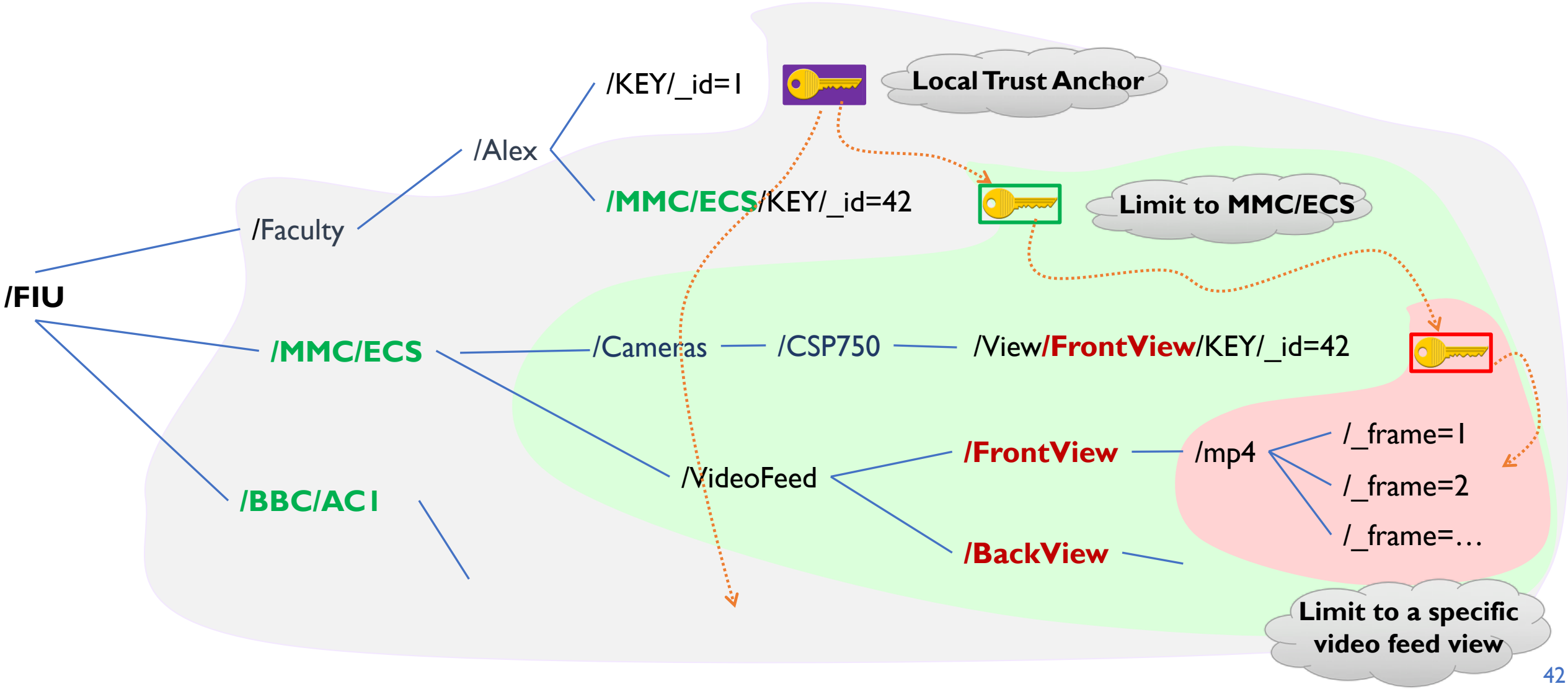
LocalAnchor(Prefix)

General Trust Model

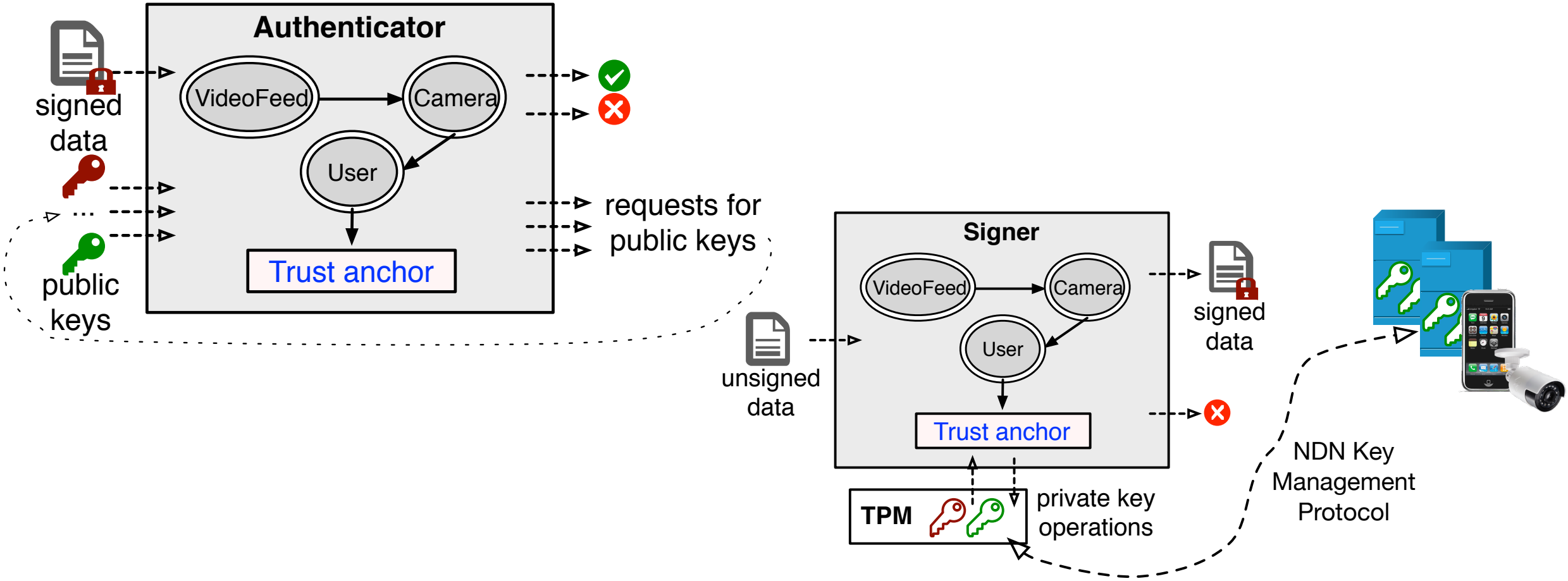
/FIU/KEY/_id=1

Trust Model Specialization for FIU campus

Privilege Separation Through Naming



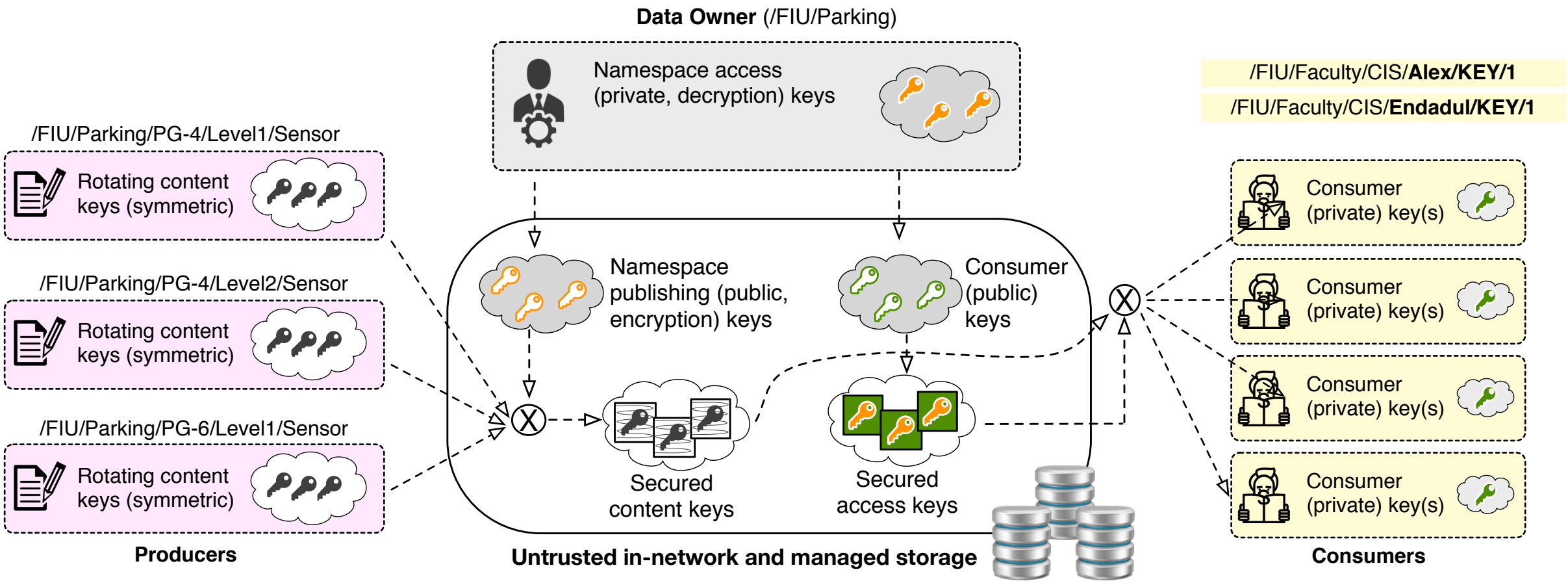
Trust Schema as an Automation Tool



Confidentiality and Access Control Requirements

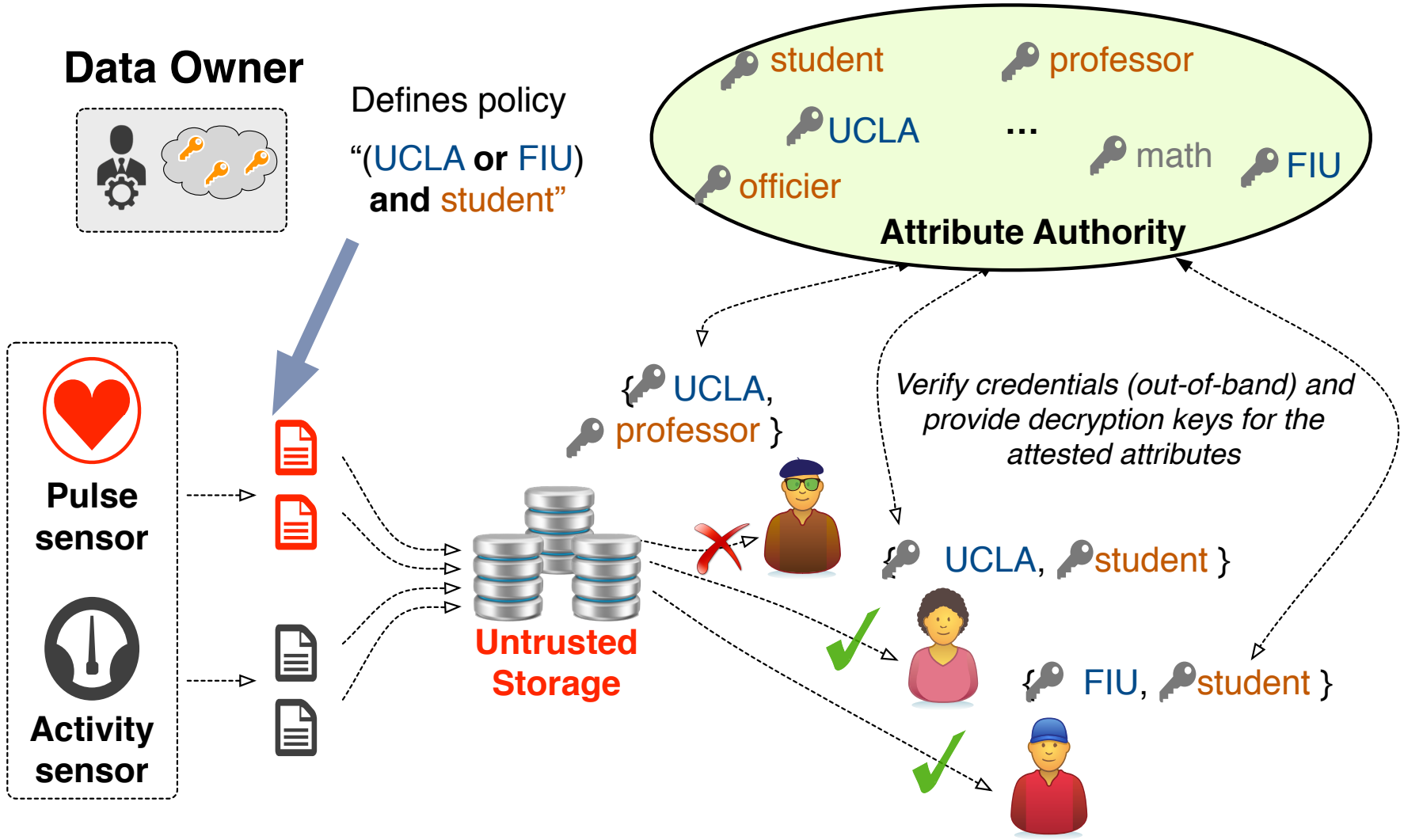
- Data-centricity
  - Confidential "end-to-end" (app-to-app), in motion or at rest
- Flexible controls
  - Granting access to publish/read at fine granularities
  - Changeable policies at any time
- Asynchrony
  - No tight coupling between distributed data production and access granting
- Scalability
  - Manageable number of encryption/decryption keys
- Multi-party
  - Seamless coordination of control among distributed data producers and consumers

Name-Based Access Control (NAC)



"NAC: name-based access control in named data networking", ACM ICN 2017

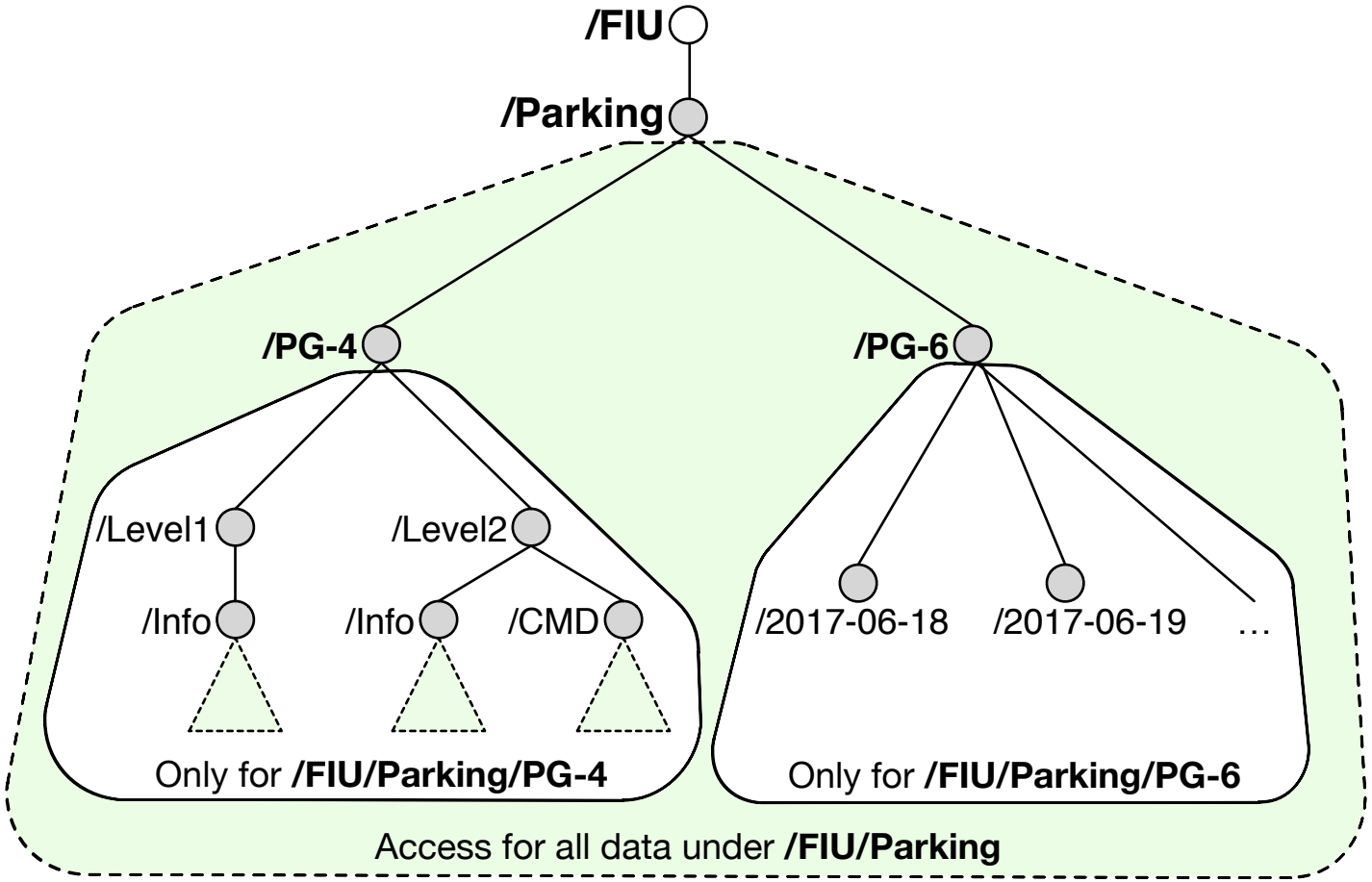
NAC with Attribute-Based Encryption



Control Granularity

- Naming conventions to leverage hierarchical scopes for read and write access

- Based on data type
  - PG-4 vs PG-6
  - Level1 vs Level2
- Based on data attributes
  - Time
  - Location



Summary: NDN in layman's words

- In cyberspace: everything is made of bags of bits
- IP: naming locations
  - Applications know what they want
  - how to get it: a complex, brittle process due to the name/semantics mismatch between apps and network layer
- NDN: directly use application data names to fetch the data
  - A name serves for both *"what it is"* and *"how to get it"*
- NDN secures data directly, removing security dependency on lower layer or intermediaries
- **Working with, or without, infrastructure**
  - **Ad hoc networking: applications (and their names) pre-exist; only the connectivity is ad hoc**
  - **Delay tolerant networking: any node with storage can carry named, secured bags of bits**

For more information

http://www.named-data.net