

# ***Data Authentication in NDN***

## *Trust Schema*

---

Alex Afanasyev (FIU)

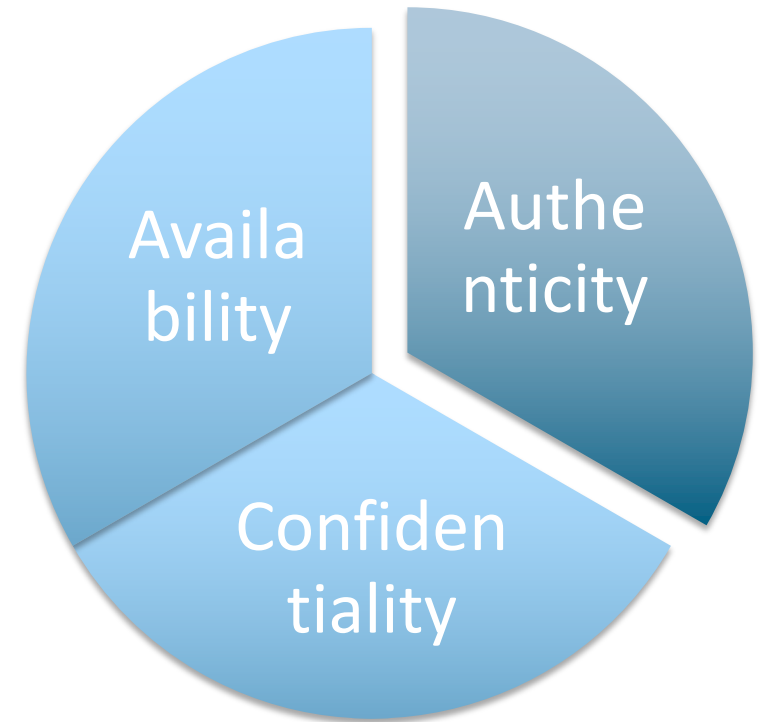
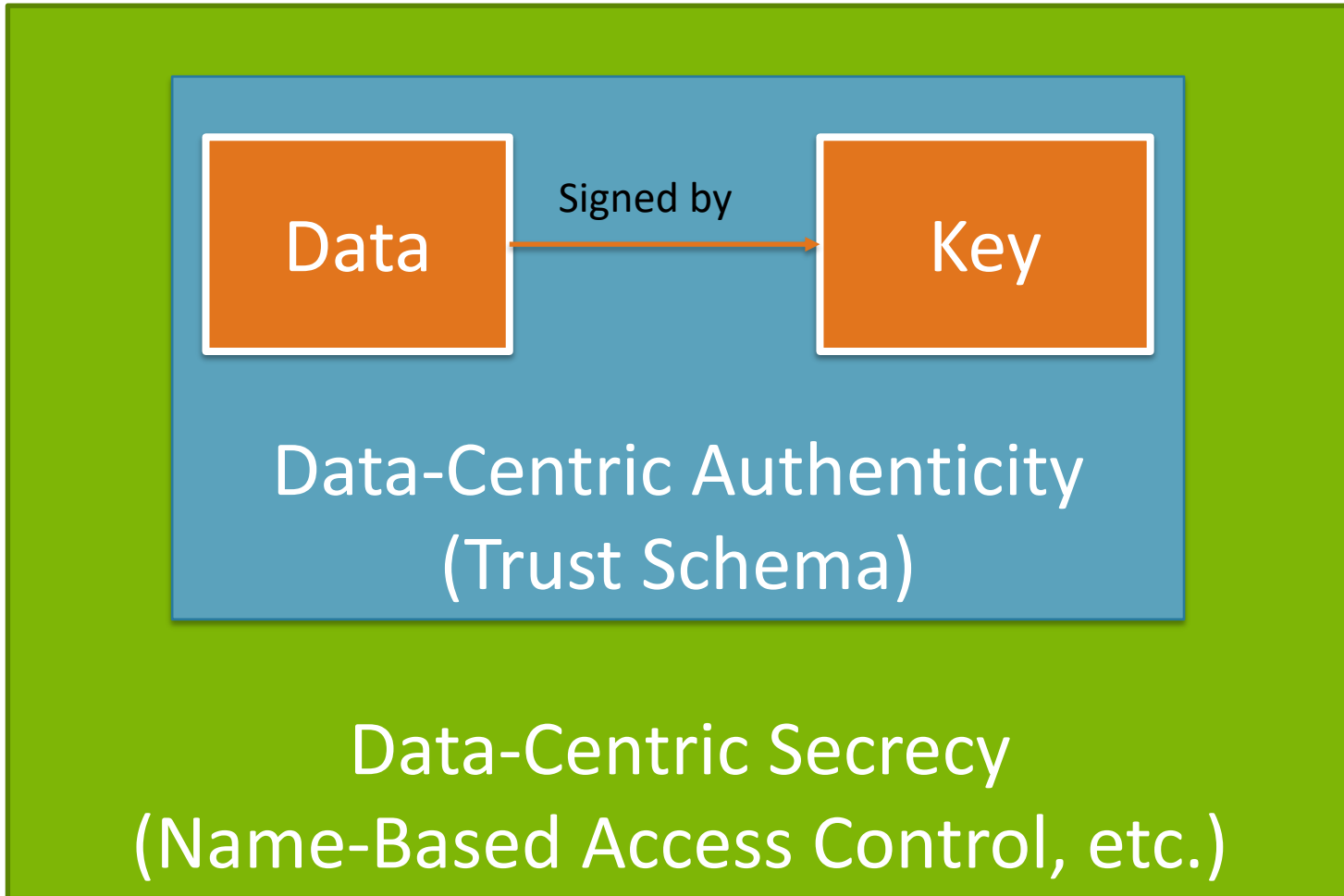
# Automating the use of crypto keys via named data

---

- ◇ Use name semantics to enable applications to reason about security, and
- ◇ Utilize NDN naming/naming conventions to automate key management in
  - Secure sign-in
  - Certificate issuance
  - Signing and verification
  - Content encryption


# Data-Centric Security in NDN

---



# Data Authenticity


/home  
/LivingRoom  
/VideoFeed/FrontView  
/mp4/\_frame=12/\_chunk=20



Signed by

/home  
.../KEY

024FG002	53D03C00
387525C1	4F553D
242434E	3D4A6
53D4553	414
0312E30	0424
CC	024E4E4E
1	09 8833B0C
33EE8EF	DF0787F



Signed by



KeyLocator: /home/.../KEY

KeyLocator: /AlexHome.com/.../KEY

# Not Just Signature, but Whose Key Signed It?

---

`/home/LivingRoom/VideoFeed`  
`/FrontView/mp4/_frame=12/_chunk=20`



`/home/Camera/KEY`



A frame from a camera  
I have installed in my  
living room

`/home/LivingRoom/VideoFeed`  
`/FrontView/mp4/_frame=42/_chunk=1`



`/Heisenberg/KEY`



A forged frame  
pretending to be an  
image of my living  
room

# Defining Trust Model for My Smart Home

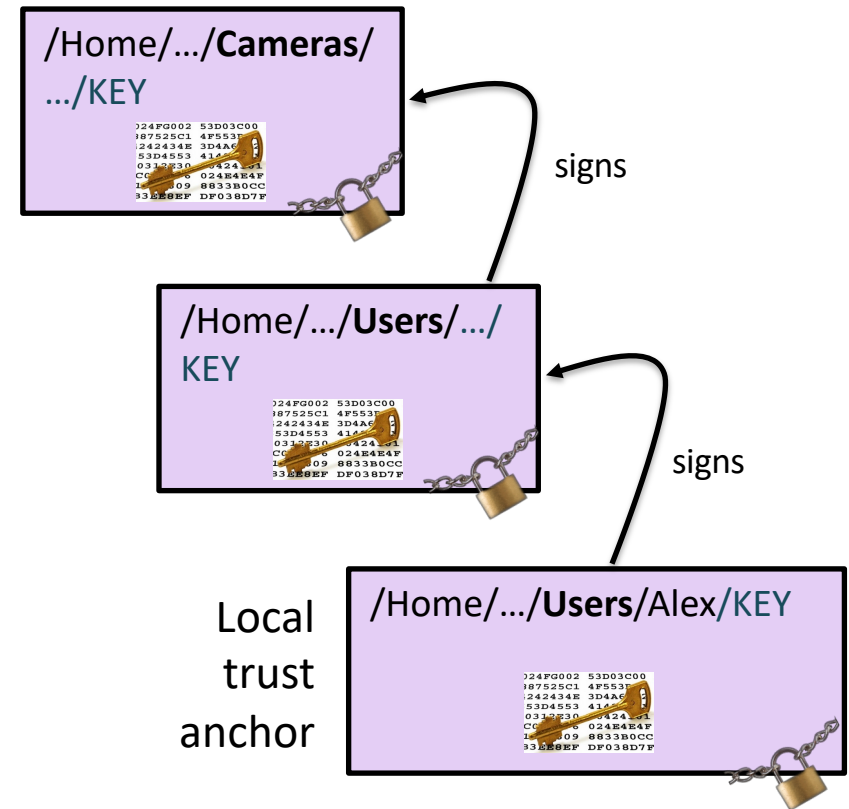
◇ Room's video feed can only come from a camera in the room



◇ Cameras in the room can be configured by someone I have authorized

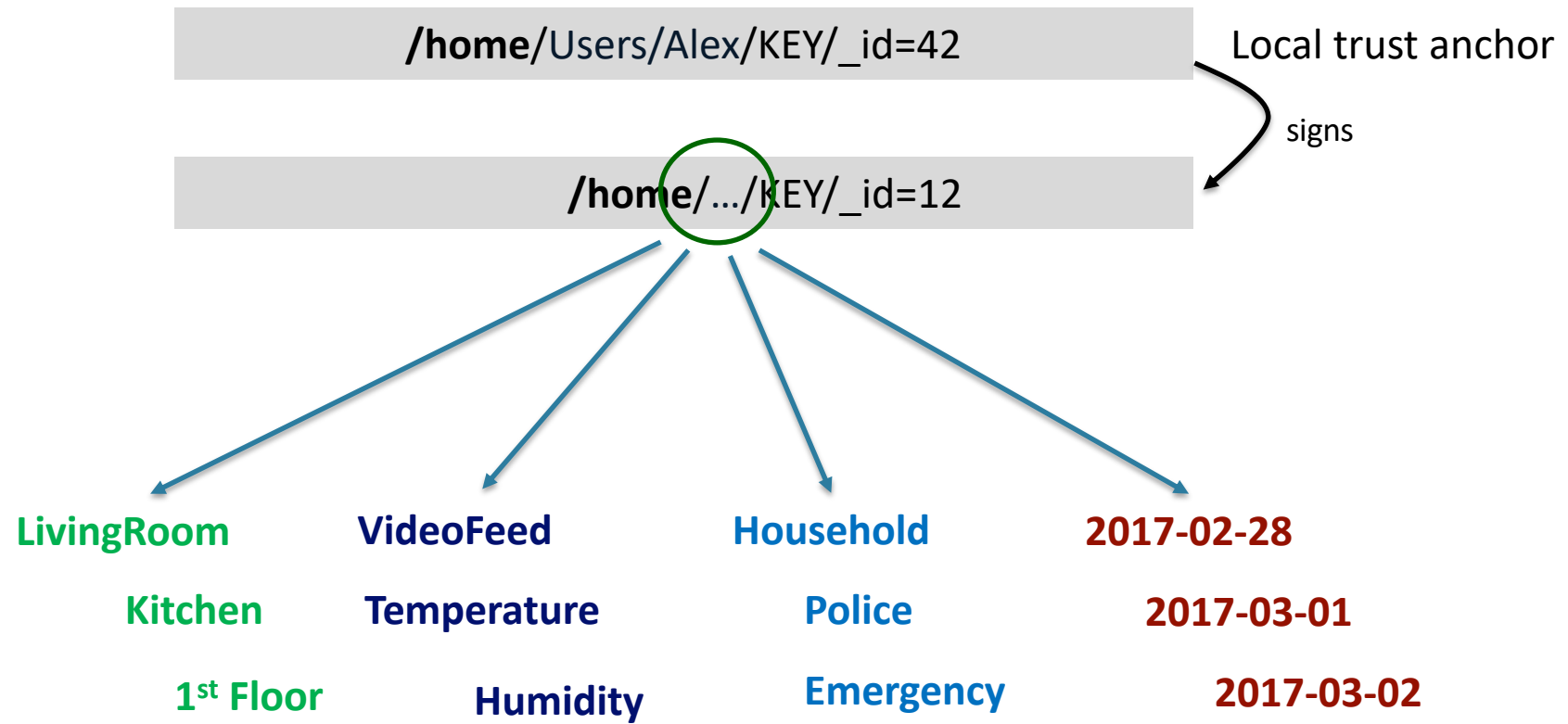


◇ Only I can authorize users to play with my cameras



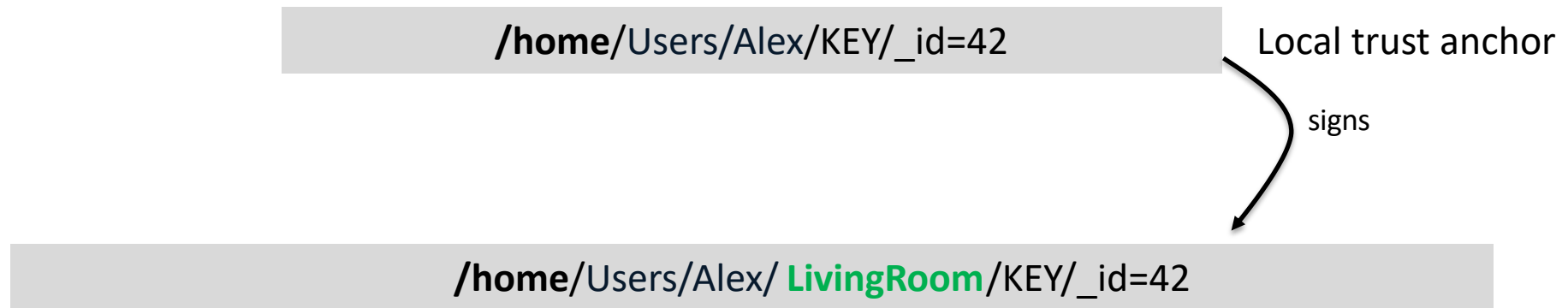
# Defining Limits via Namespace Design

---



# Restricting Power of Keys

---



The new key is now restricted to authorize data and operations within the **living room** only



# Restricting Power of Keys

---

`/home/Users/Alex/ LivingRoom /KEY/_id=42`

signs

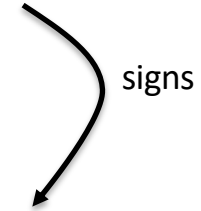
`/home/LivingRoom/Cameras/CSP750/View/FrontView/KEY/_id=1001`

The delegate key is now even more restricted: to publish "camera" data in the living room with a static frontal view

# Restricting Power of Keys

---

`/home/LivingRoom/Cameras/CSP750/View/FrontView/KEY/_id=1001`



`/home/LivingRoom/VideoFeed/FrontView/mp4/_frame=1/...`

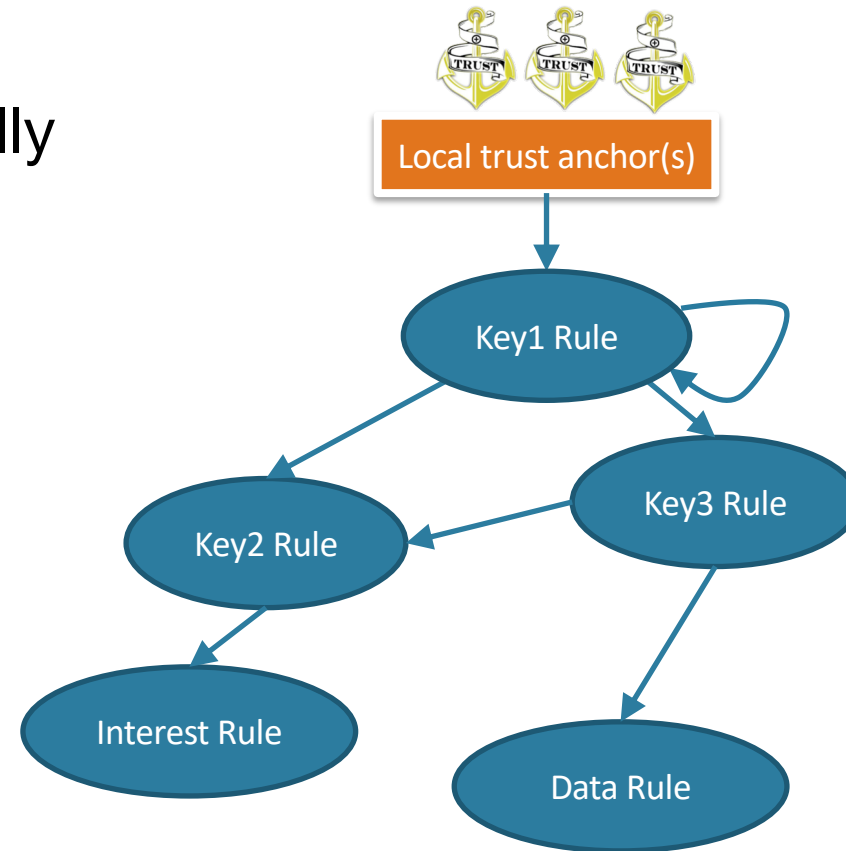
Camera's key has a very  
narrow privilege

# Trust Schema: Name-Based Definition of Trust Model

---

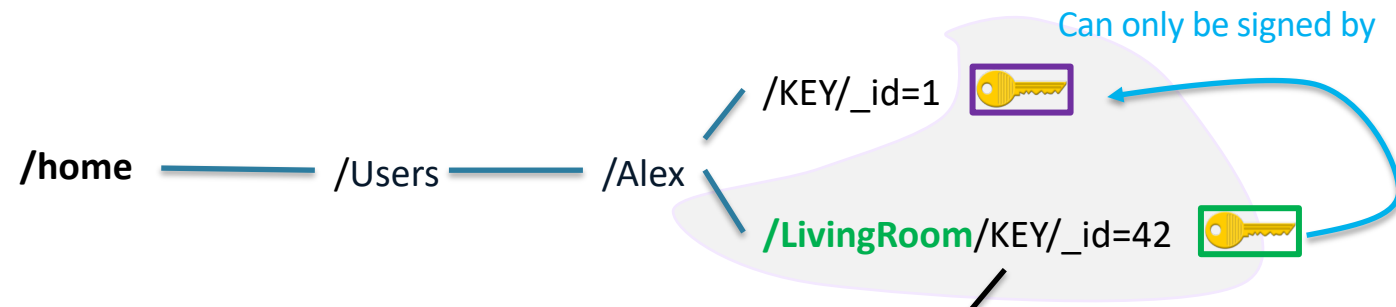
- ◇ A formal language to formally describe trust model
  - Schematize data and key name relationships

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



# Schematizing Rules: Specific Restriction

---

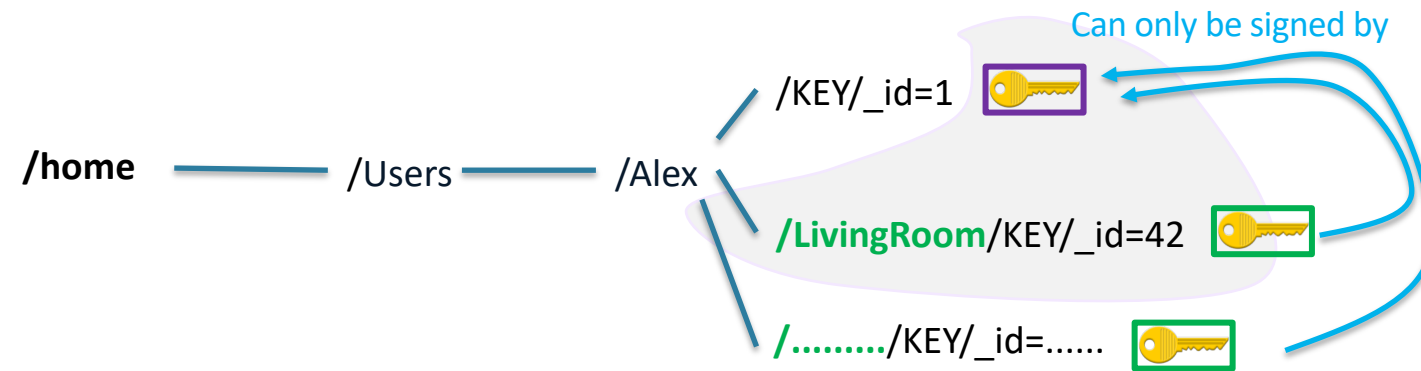


```
<home><Users>[user]<LivingRoom><KEY>[key-id]
```

LocalAnchor

User rule

# Schematizing Rules: Broader Restriction

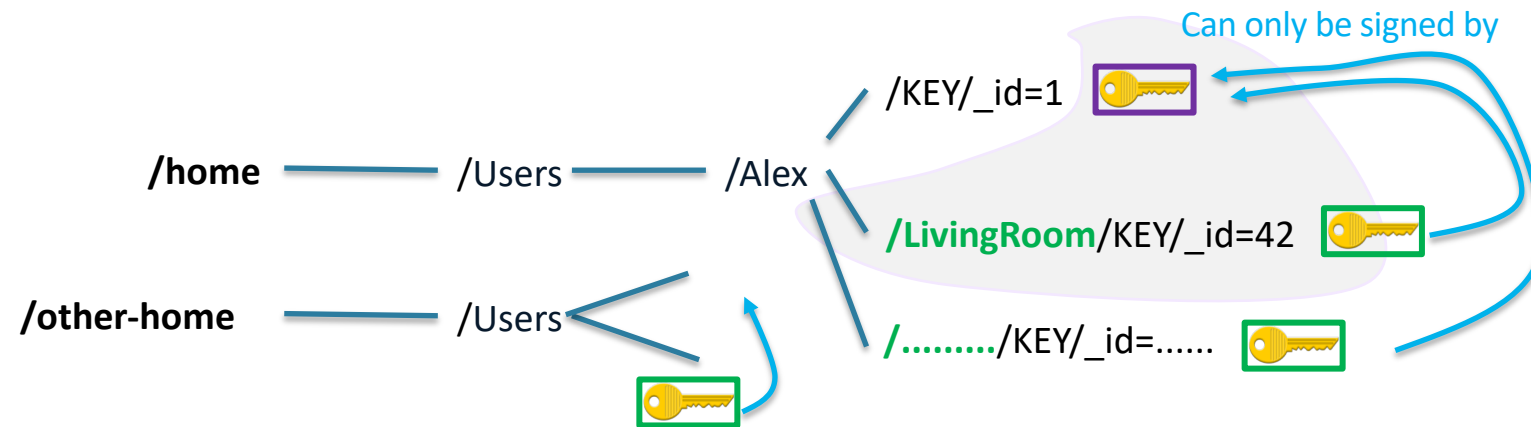


`<home><Users>[user](:Location:<>?)<KEY>[key-id]`

**LocalAnchor**

**User rule (parametrized by Location)**

# Schematizing Rules: Generalized Restriction



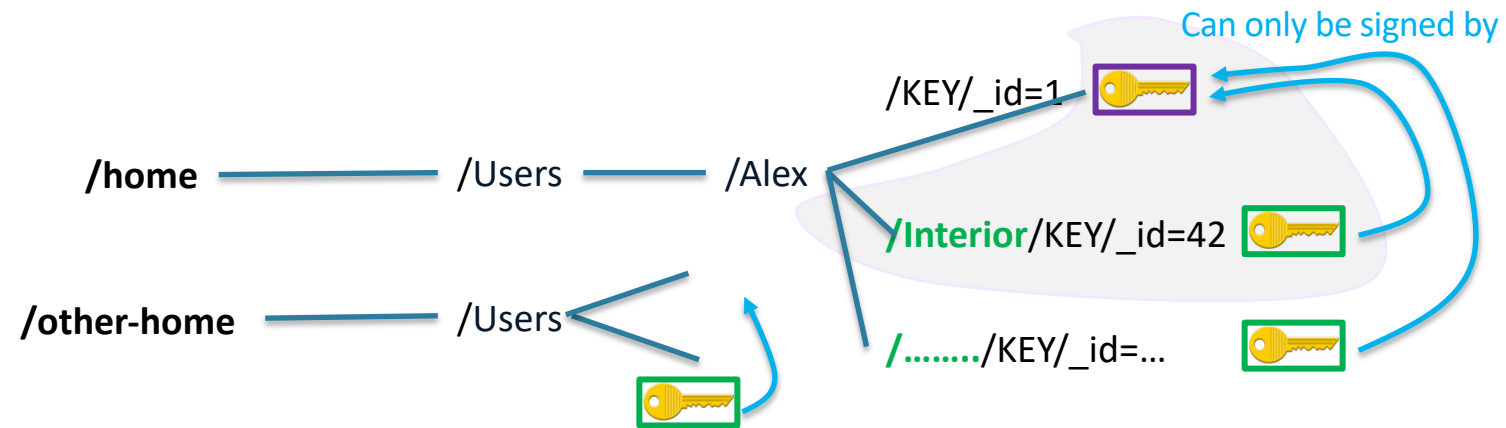
`(:Prefix:<>*)<Users>[user](:Location:<>?)<KEY>[key-id]`

Can only be signed by

**LocalAnchor(Prefix)**

**User rule (parametrized by Prefix and Location)**

# Schematizing Rules: Generalized Restriction



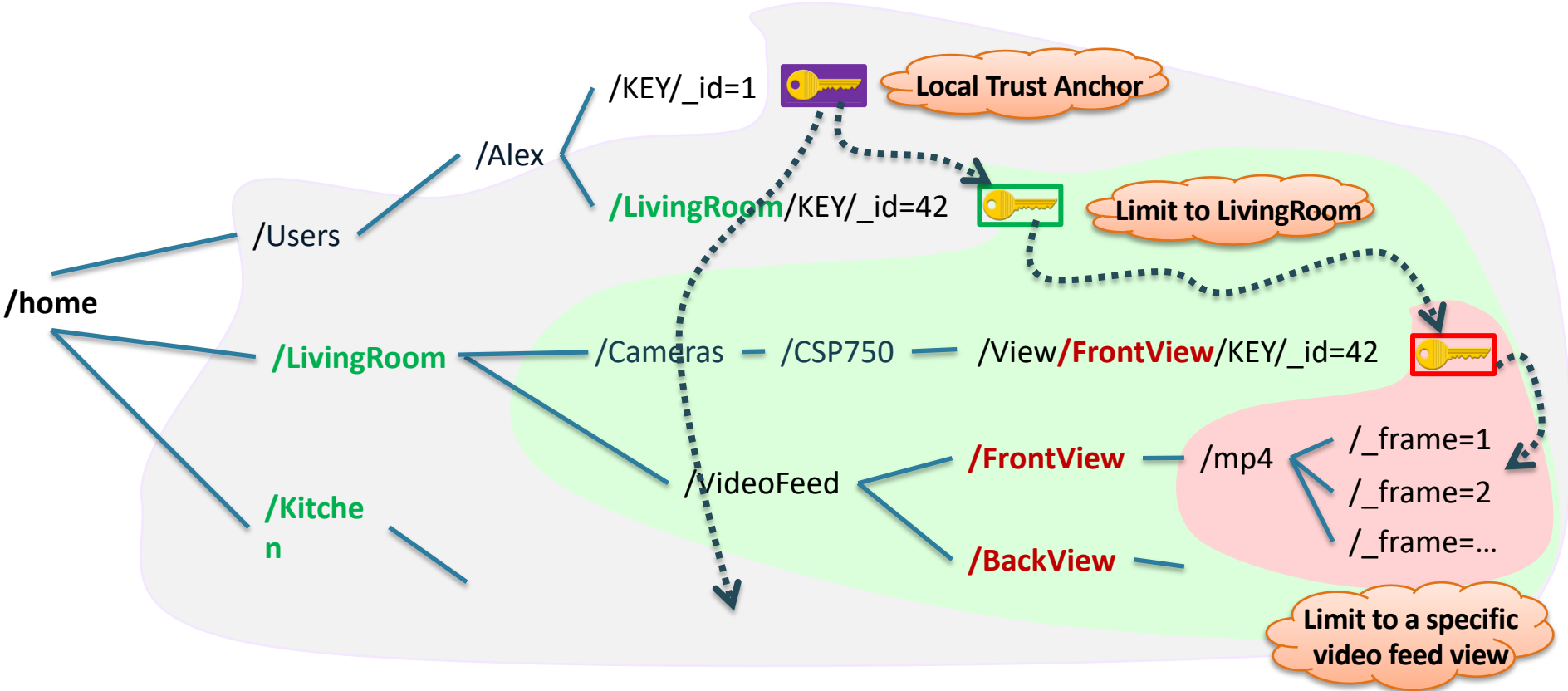
`(:Prefix:<>*)<Users>[user](:Location:<>?)<KEY>[key-id]`

Can only be signed by

**LocalAnchor(Prefix)**

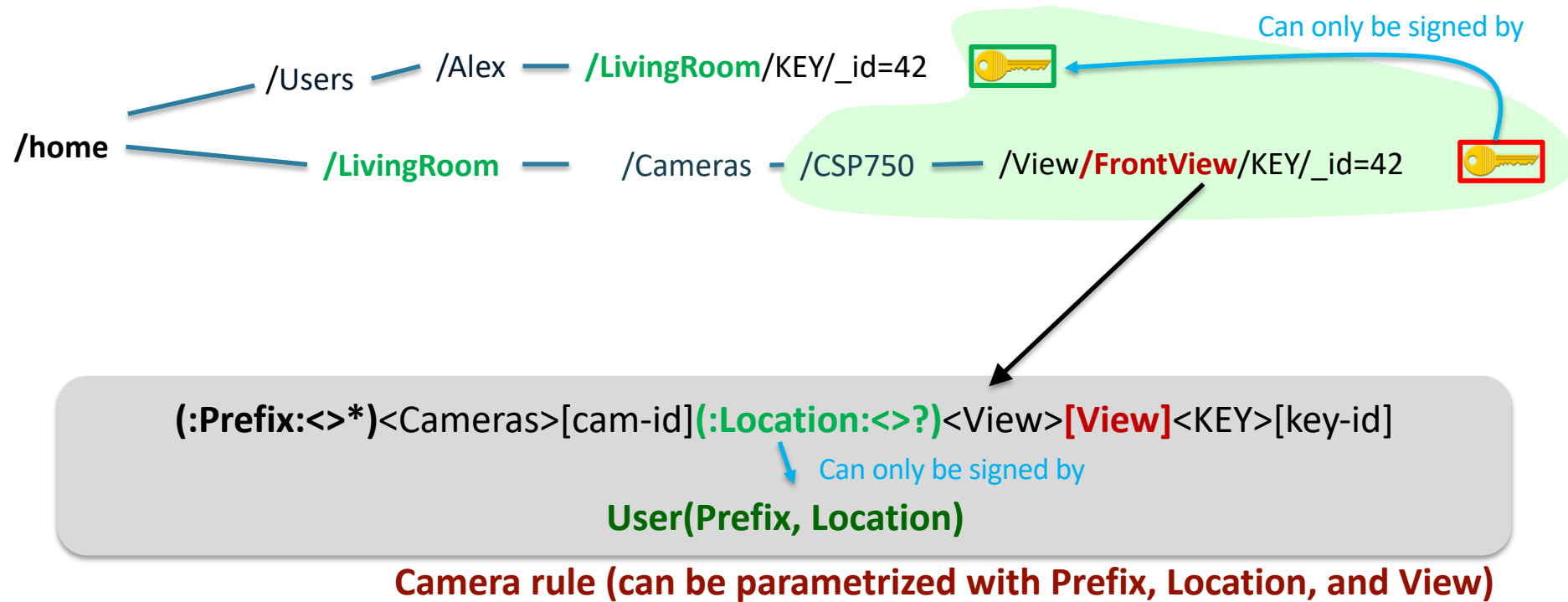
**User rule (parametrized by Prefix and Location)**

# Privilege Separation Through Naming

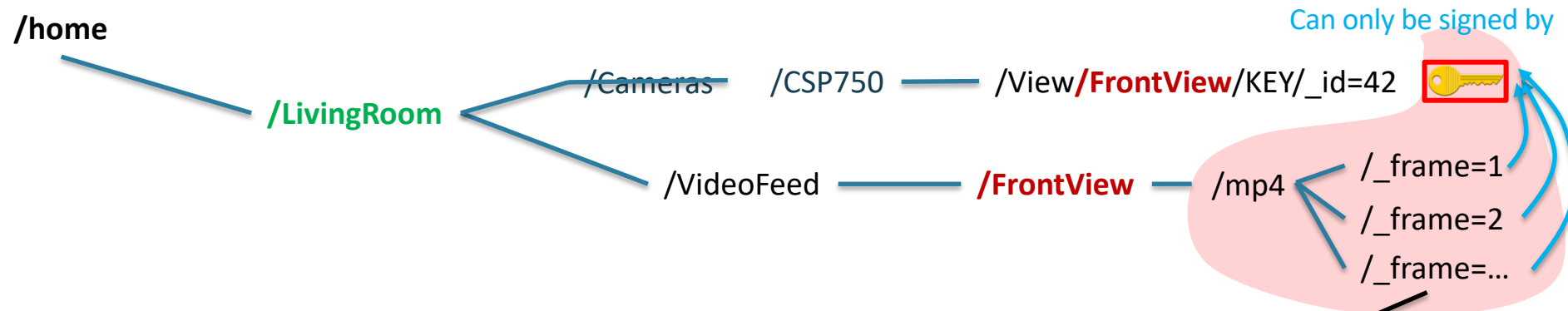




# Schematizing Key-Key Naming Rule: Camera



# Schematizing Data-Key Naming Rule: VideoFeed



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

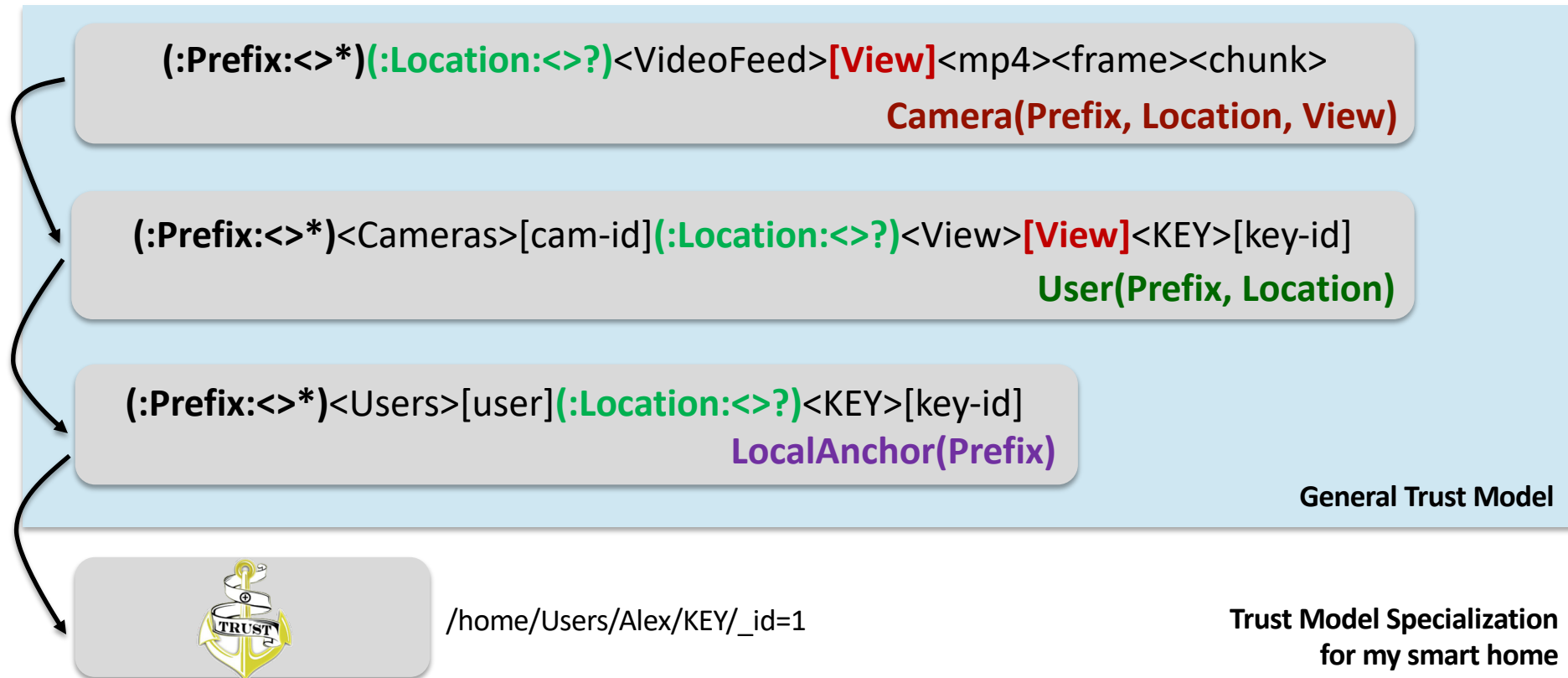
Must be signed by

**Camera(Prefix, Location, View)**

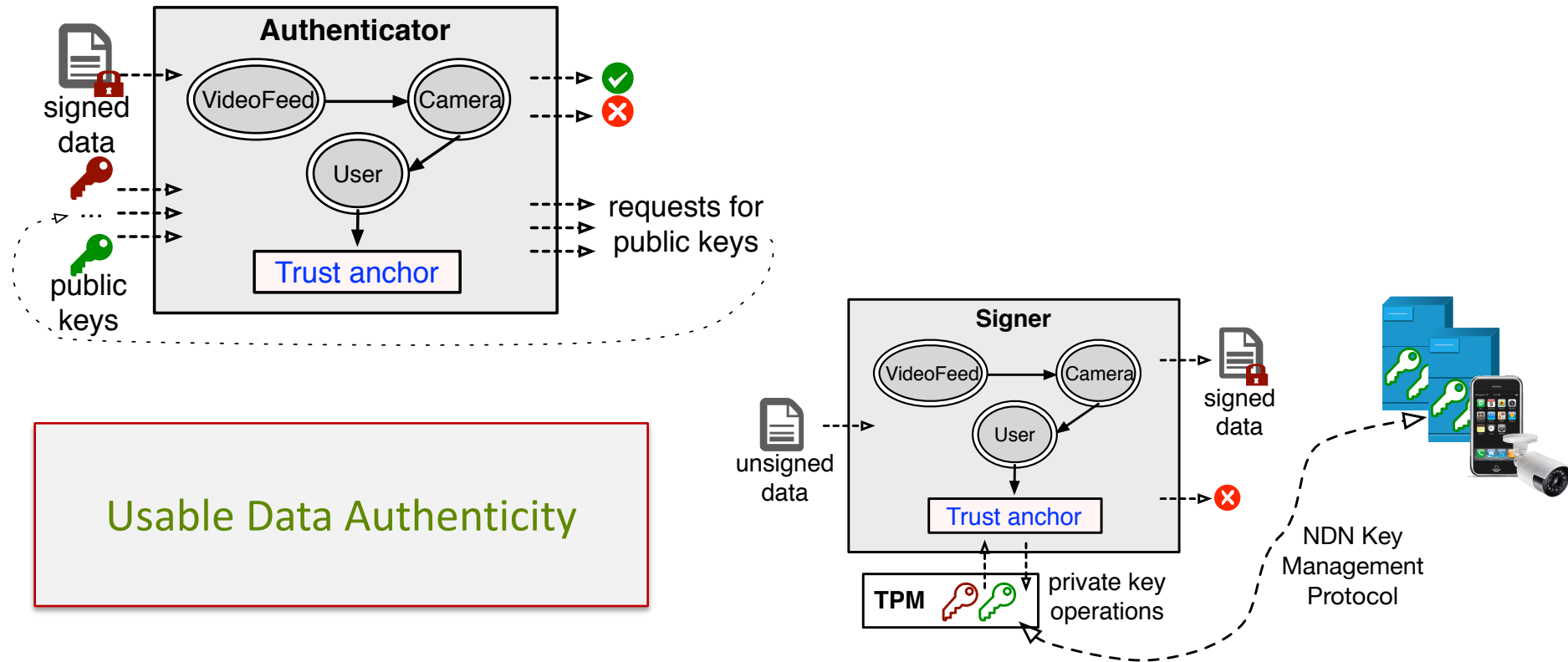
**VideoFeed rule (can be parametrized with Prefix, Location, and View)**

# Complete Example of Smart Home Trust Schema

---



# Trust Schema as an Automation Tool



# Trust Schema Summary

---

- ◇ Hierarchical data/key name relations embed real power
  - Differentiated levels of security and separate privileges
- ◇ Trust schema influence the application namespace design and is influenced by the namespace design
- ◇ Enables automation for data validation and signing
- ◇ Enables automation of NDN certificate management

# Demo

---

- ◇ Example of simple trust schema in ValidatorConfig (“old”) format