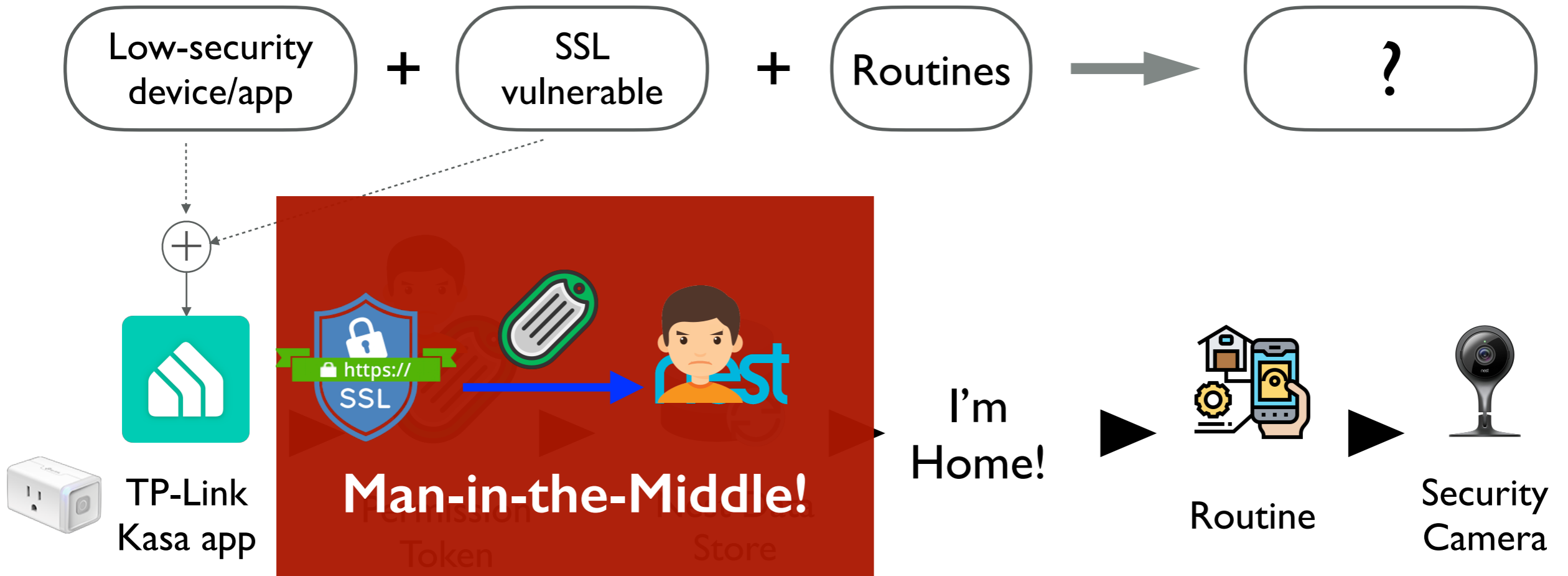


# CIS 4930: Secure IoT

**Prof. Kaushal Kafle**

Lecture 11

# LATERAL PRIVILEGE ESCALATION



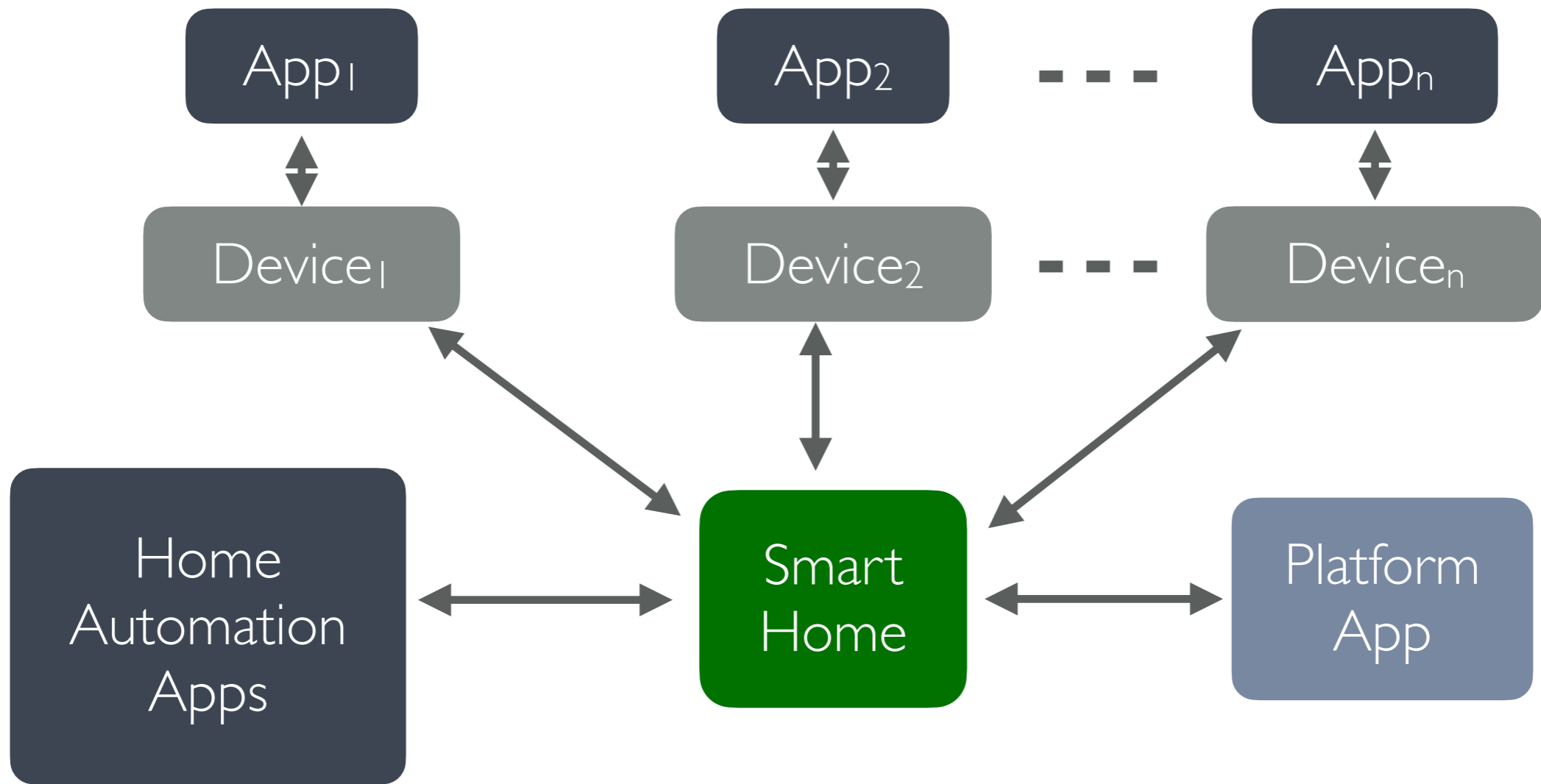
# Trust

- **Trusted:** A system or component whose failure can break the security policy.
- **Trustworthy:** A component that *will not fail*.

# Motivation



- Smart homes are *decentralized*, even if a single platform (e.g., HomeKit) is used
- Devices from heterogeneous vendors integrate via APIs



# Motivation



- Smart homes are *decentralized*, even if a single platform (e.g., HomeKit) is used
- Devices from heterogeneous vendors integrate via APIs
- Compute/Store/Expose similar *states*
  - Tracking whether the user is at a specific location
  - Tracking time
  - ...?

*Key problem: redundant computation of the same states*

# Motivation



*Key problem: redundant computation of the same states*

*Why is this a **security** problem?*

These states, or *situations*, are critical for security/privacy/safety policies

## Situational Access Control in the Internet of Things

Roei Schuster  
Tel Aviv University  
Cornell Tech  
rs864@cornell.edu

Vitaly Shmatikov  
Cornell Tech  
shmat@cs.cornell.edu

Eran Tromer  
Tel Aviv University  
Columbia University  
tromer@cs.tau.ac.il

## Practical Integrity Validation in the Smart Home with HomeEndorser

Kaushal Kafle  
William & Mary  
Williamsburg, VA, USA  
kkafle@wm.edu

Kirti Jagtap  
Penn State University  
Pennsylvania, USA  
ktj35@psu.edu

Mansoor Ahmed-Rengers  
University of Cambridge  
Cambridge, UK  
mansoor.ahmed@cl.cam.ac.uk

Trent Jaeger\*  
University of California, Riverside  
Riverside, California, USA  
trentj@ucr.edu

Adwait Nadkarni  
William & Mary  
Williamsburg, VA, USA  
apnadkarni@wm.edu

*Who computes?*

*What about integrity?*

# Situational Access Control

- Recall from last class: The *risk* associated with an action may vary, based on the context
  - Risk: *probability of failure \* impact*
- A typical *access control policy* contains:
  - **Subjects**: Active entities that do things (e.g., *us, apps, devices*)
  - **Objects**: Passive entities that things are done to (e.g., *states of the home/environment, devices*)
  - **Rights**: Actions that are taken (e.g., *read, write, share*)
  - e.g., a *home security monitoring app* can *read* the *camera feed*.

*What's missing?*

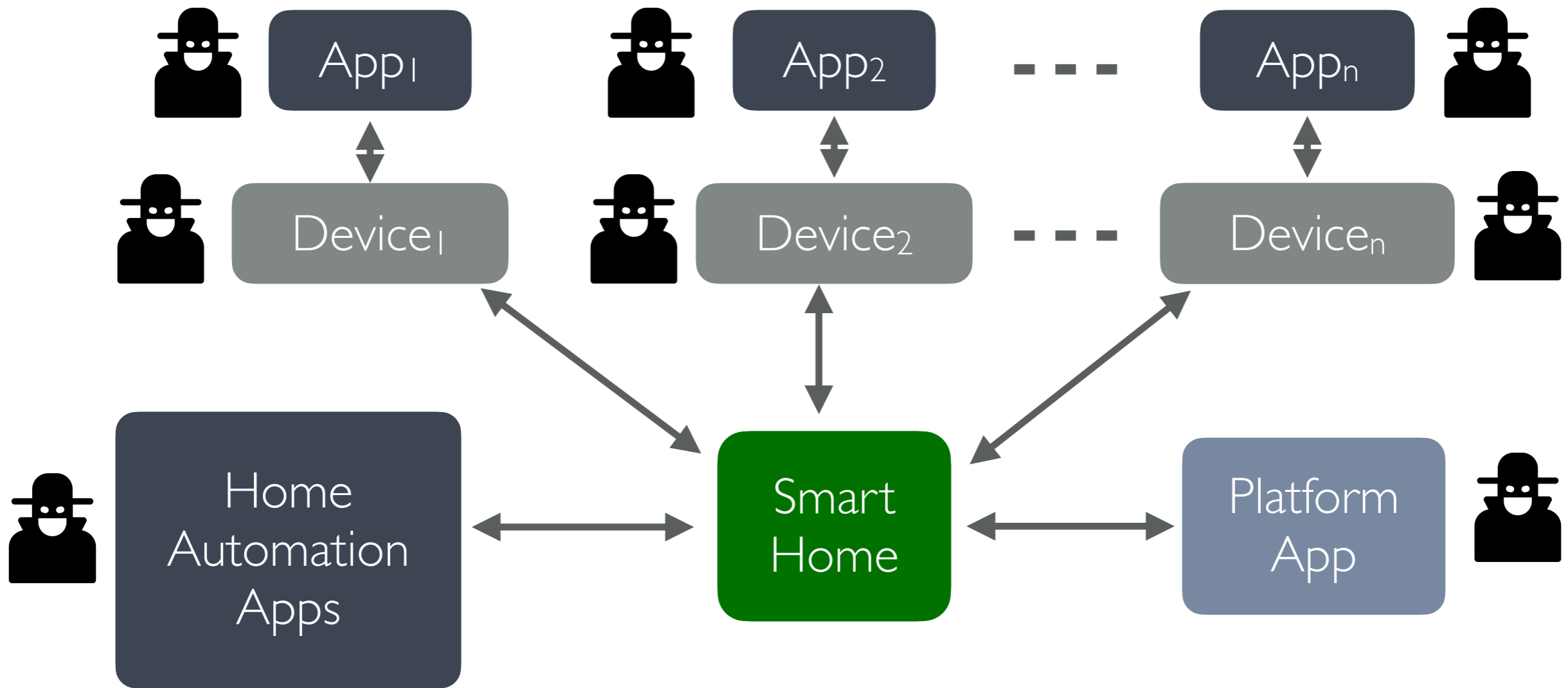


# Situational Access Control

*Whats missing?*

- Situations!
  - a *home security monitoring app* can *read* the *camera feed*, *but only when the user is away*
  - *How is ^^ implemented currently?*
    - *Turn the camera OFF when the user is at HOME*
    - *Sufficient?*

*Key problem: Enabling (1) situational access control policies without  
(2) redundant computation of the same states*

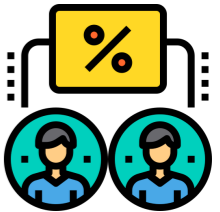


*Each of these components can compute **home/away** and write that into the smart home for all to use; why is this bad?*

*We need to reduce the **attack surface***

# Approach

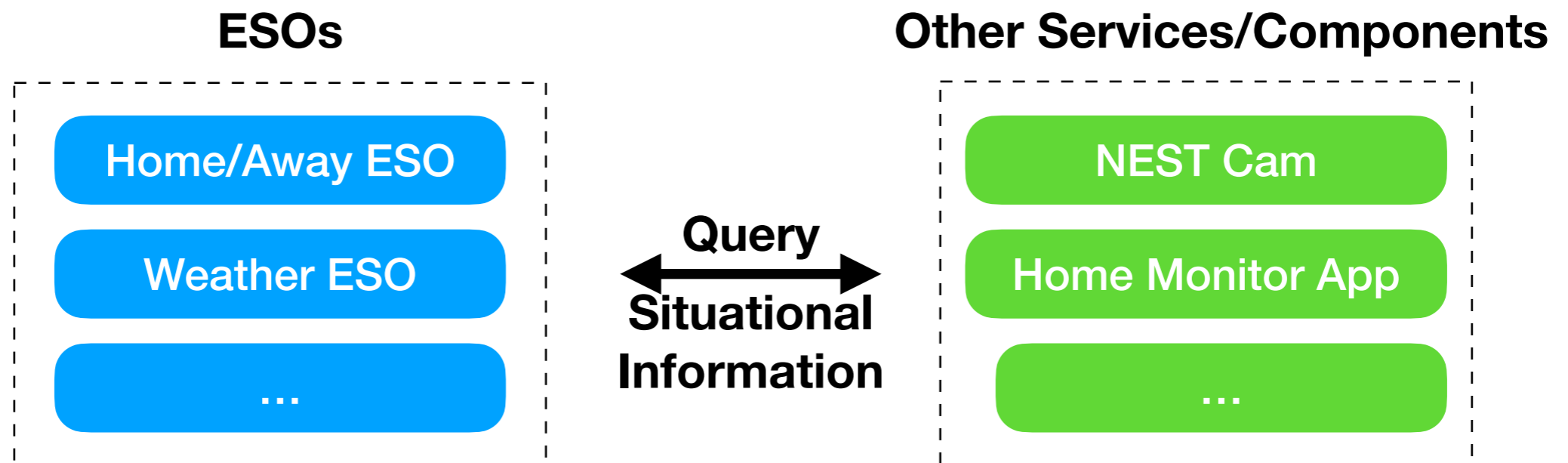
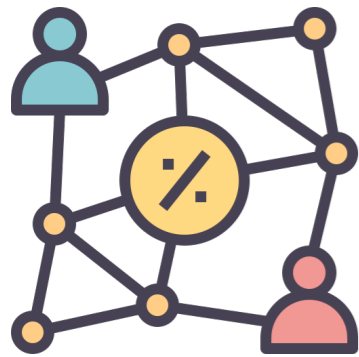
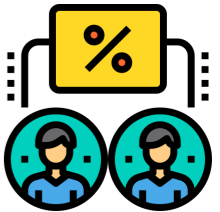
- *Decoupling* situation retrieval and provisioning from platform(s).
- **A unified interface:** Environmental Situational Oracles (ESOs)
  - Services responsible (and dedicated to) specific situational variables.



*Have you seen this elsewhere?*

# Approach

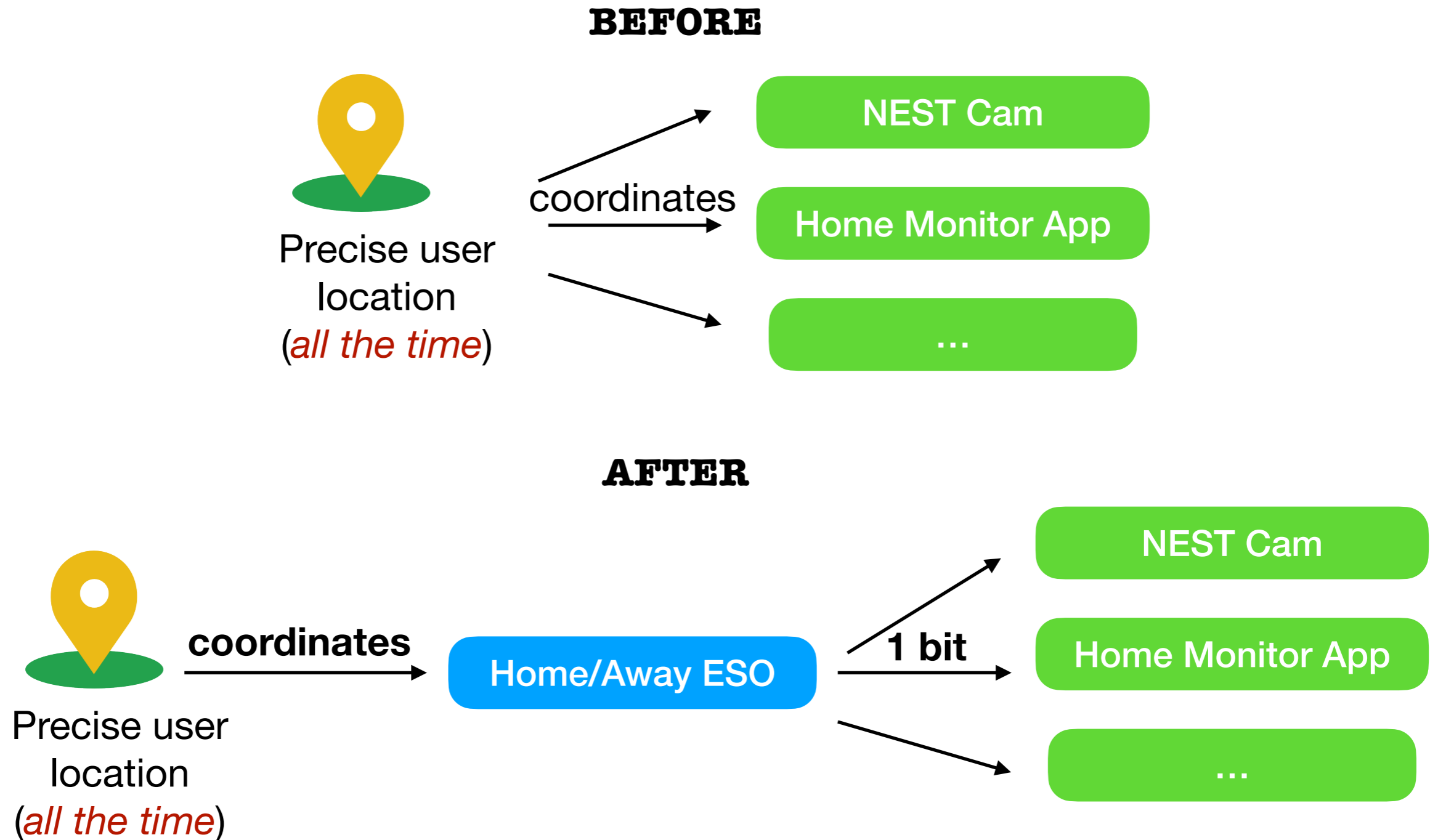
- *Decoupling* situation retrieval and provisioning from platform(s).
- **A unified interface:** Environmental Situational Oracles (ESOs)
  - Services responsible (and dedicated to) specific situational variables.



*How does this approach impact the decentralized nature of the smart home?*

# Benefits of ESOs

## 1. Reducing Overprivilege

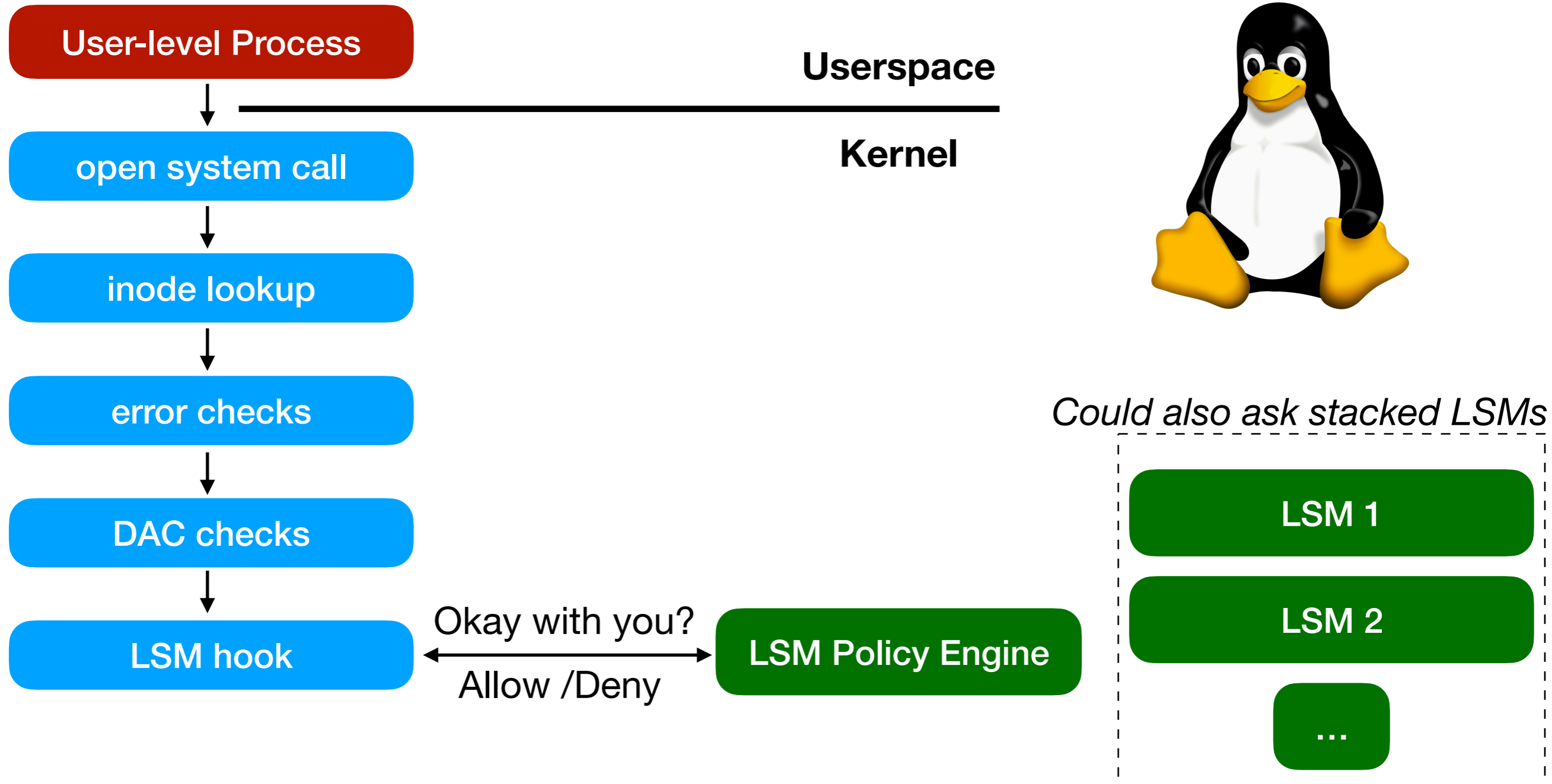


# Benefits of ESOs

2. **Reduce Errors** (i.e., abstract away the details)
  - i.e., a few *dedicated* sources of situational information are better than retrieving it yourself
  - Analogy: use a vetted *SSL library* instead of implementing one in each app!
3. **Reduce the attack surface** (*not discussed in the paper*)
  - *ESOs are fewer than apps*
4. **Implement platform-independent situational policies**
  - No need to rely on the platform's ability to provide the situational variable

# OS Security Extensibility

- An analogy: [the Linux Security Modules \(LSM\) Framework](#)



# OS Security Extensibility

- Similar efforts on Android: the Android Security Modules (ASM) Framework [1]
- Hook into the various *managers*
  - *Function-specific System services*
  - E.g., Telephony Manager, Location Manager
- Various ASMs can register for hooks and get callbacks

*How are ESOs related to these managers?*



# Recall: Trust

- **Trusted:** A system or component whose failure can break the security policy.
- **Trustworthy:** A component that *will not fail*.

*Which category do ESOs belong to?*

- **Trusted third party:** *Trusted by all parties* for some set of actions

# Trusted Third Parties elsewhere

Class 3 Public Primary Certification Authority

- VeriSign Class 3 Public Primary Certification Authority - G5
  - VeriSign Class 3 International Server CA - G3
    - www.chase.com

**www.chase.com**  
Issued by: VeriSign Class 3 International Server CA - G3  
Expires: Thursday, August 16, 2012 7:59:59 PM ET  
This certificate is valid

**Details**

Subject Name

Country US

State/Province New Jersey

Locality Jersey City

Organization JPMorgan Chase

Organizational Unit CIG

Common Name www.chase.com

Issuer Name

Country US

Organization VeriSign, Inc.

Organizational Unit VeriSign Trust Network

Organizational Unit Terms of use at <https://www.verisign.com/rpa> (c)10

Common Name VeriSign Class 3 International Server CA - G3

Serial Number 61 5C 33 29 65 09 08 60 A4 E6 82 50 00 F6 22 F0

Version 3

Signature Algorithm SHA-1 with RSA Encryption ( 1 2 840 113549 1 1 5 )

Parameters none

Not Valid Before Tuesday, August 16, 2011 8:00:00 PM ET

Not Valid After Thursday, August 16, 2012 7:59:59 PM ET

OK

Certificate Manager

Your Certificates | People | Servers | **Authorities** | Others

You have certificates on file that identify these certificate authorities:

Certificate Name	Security Device
▼ The Go Daddy Group, Inc.	
Go Daddy Secure Certification Authority	Software Security Device
Go Daddy Class 2 CA	Builtin Object Token
▼ The USERTRUST Network	
Network Solutions Certificate Authority	Software Security Device
Register.com CA SSL Services (OV)	Software Security Device
UTN-USERFirst-Hardware	Builtin Object Token
UTN - DATACorp SGC	Builtin Object Token
UTN-USERFirst-Network Applications	Builtin Object Token
UTN-USERFirst-Client Authentication and Email	Builtin Object Token
UTN-USERFirst-Object	Builtin Object Token
▼ Türkiye Bilimsel ve Teknolojik Araştırma Kurumu...	
TÜBİTAK UEKAE Kök Sertifika Hizmet Sağlayıcı...	Builtin Object Token
▼ TÜRKTRUST Bilgi İletişim ve Bilişim Güvenliği Hiz...	
TÜRKTRUST Elektronik Sertifika Hizmet Sağlay...	Builtin Object Token
▼ University of Pennsylvania	
DSL CA Authority	Software Security Device
▼ Unizeto Sp. z o.o.	
Certum CA	Builtin Object Token
▼ ValiCert, Inc.	
RSA Public Root CA v1	Software Security Device
<a href="http://www.valicert.com/">http://www.valicert.com/</a>	Builtin Object Token
<a href="http://www.valicert.com/">http://www.valicert.com/</a>	Builtin Object Token

View... Edit... Import... Export... Delete...

OK

# Trusted Third Parties elsewhere

**Things can go horribly wrong when we trust *all* CAs equally**

**Class 3 Public Primary Certification Authority**  
VeriSign Class 3 Public Primary Certification Authority - G5  
VeriSign Class 3 International Server CA - G3  
www.chase.com

**www.chase.com**  
Issued by: VeriSign Class 3 International Server CA - G3  
Expiration: Thursday, August 16, 2012 7:59:59 PM ET

Country US  
State/Province New Jersey  
Locality Jersey City  
Organization JPMorgan Chase  
Organizational Unit CIG  
Common Name www.chase.com

Issuer Name  
Country US  
Organization VeriSign, Inc.  
Organizational Unit VeriSign Trust Network  
Organizational Unit Terms of use at https://www.verisign.com/terms-of-use  
Common Name VeriSign Class 3 International Server CA - G3  
Serial Number 61 5C 33 29 65 09 08 60  
Version 3  
Signature Algorithm SHA-1 with RSA Encryption  
Parameters none  
Not Valid Before Tuesday, August 16, 2011  
Not Valid After Thursday, August 16, 2012

**Home DigiNotar, Internet Trust Provider**  
www.diginotar.com

**DigiNotar®**  
A VASCO COMPANY

HOME ANNOUNCEMENTS PRODUCTS BRANCH SOLUTIONS ABOUT DIGINOTAR PARTNERS PROJECTS

Know for sure with whom you have an agreement  
How do you check the identity of someone who's doing business online?

EV SSL | Contact | FAQ

**Go to ...**  
Managed PKI  
SSL Certificates  
SIM-ID  
Signing Service  
DocProof

**DigiNotar®, Internet Trust Provider**  
As independent Internet Trust Service Provider DigiNotar focuses on ensuring the integrity of information flow, and legal guarantees for all online information exchange. More information >>

**Announcements**  
> **Publication report Fox-IT**  
Click here for the Interim report of Fox-IT  
> **Cooperation Dutch government**  
Read the press release >>  
> **DigiNotar reports security incident**  
Read the press release >>

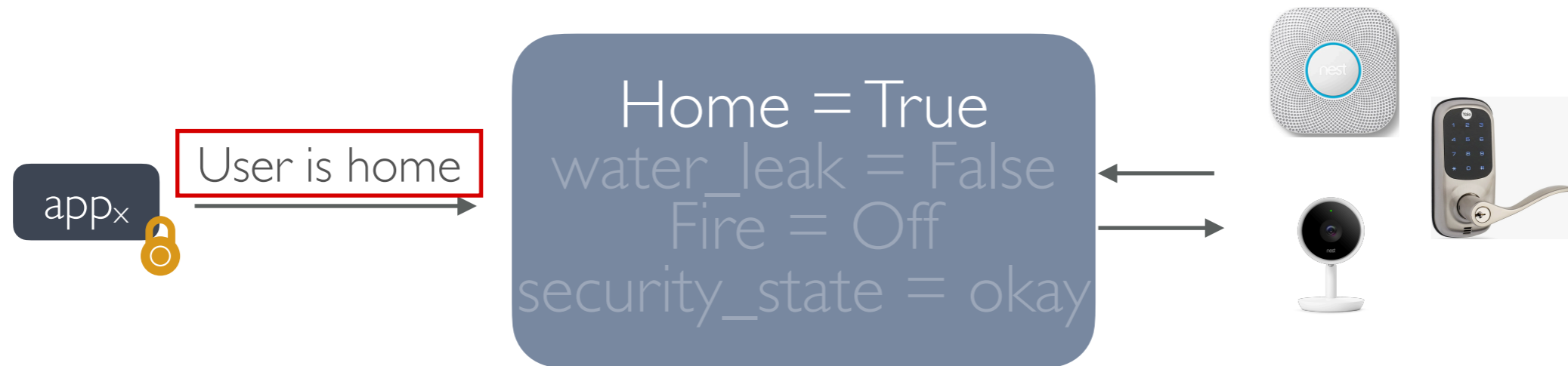
**GO** EV SSL  
**BUY NOW**

**VASCO**  
A VASCO COMPANY

# Takeaway

- Decoupling the retrieval of situational variables has its advantages
  - Reducing errors, the attack surface, overprivileged
- However, there may be severe practical challenges, such as,
  - Single point of failure?
  - ...

# PROBLEM & SCALE



Crypto-API misuse  
Analysis of IoT  
apps<sup>1</sup>



917 apps with  
over 1M  
downloads



94.11% with at least  
1 crypto issue

1. Jin, Xin et. al. "Understanding IoT Security from a Market-Scale Perspective" *Proceedings of the 29th ACM Conference on Computer and Communications Security (CCS)*, 2022

# PRIOR SOLUTIONS

Remove all access to  
AHOs?

Analyze apps?

Enforce *Least Privilege*?

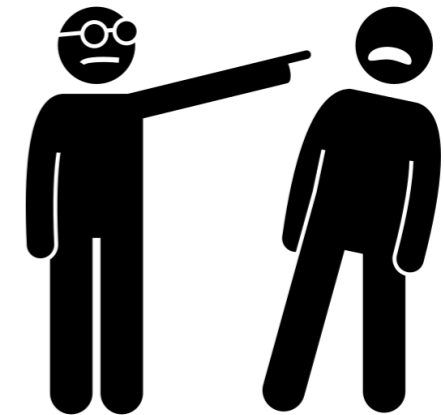
# PRIOR SOLUTIONS

Remove all access to Abstract Objects?

Critical for 3rd-party integrations



Removes user flexibility!



Google reverses course on cutting off Works with Nest connections

GOOGLE NEST

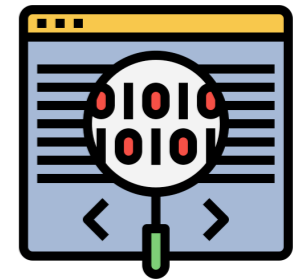
We hear you: updates to Works with Nest

# PRIOR SOLUTIONS

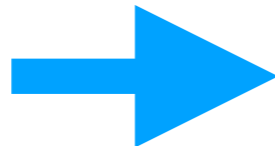
Analyze apps?

*Extract app behavior from source code*

*Look for malicious or vulnerable code*



Platforms becoming  
API-centric



*E.g. SmartThings V2 to V3,  
HomeAssistant*

*V2 - Apps hosted  
in SmartThings  
Cloud*

*V3 - Apps  
communicate via  
API-endpoints*



App source code no longer accessible  
for analysis!



# PRIOR SOLUTIONS

Enforce *Least Privilege*?

*Give apps/services only the permissions they need*

Legitimate  
permissions to  
Apps/Services can  
still be  
compromised and  
misused!

*E.g. TP-Link Kasa app in our  
previous example*



# ADAPTING IFC



Traditional Information Flow Control?



*Biba Integrity Model*

 Home, Time  
Apps, Services 



High-integrity objects   
Low-integrity objects 

A “guard” that *endorses* access from low-integrity objects to high-integrity objects

Typically, by *trusted processes* e.g. admins

# ADAPTING IFC

Traditional Information Flow Control?



*Biba Integrity Model*

⊗ Home, Time  
Apps, Services



High-integrity objects ⊗  
Low-integrity objects

A “guard” that *endorses* access from low-integrity objects to high-integrity objects

Typically, by *trusted processes* e.g. admins

Can we use users?

→ Unaware of interdependencies among devices and AHOs

→ Process would be manual

What can we rely on to serve as ‘trusted guards’ in the smart home?

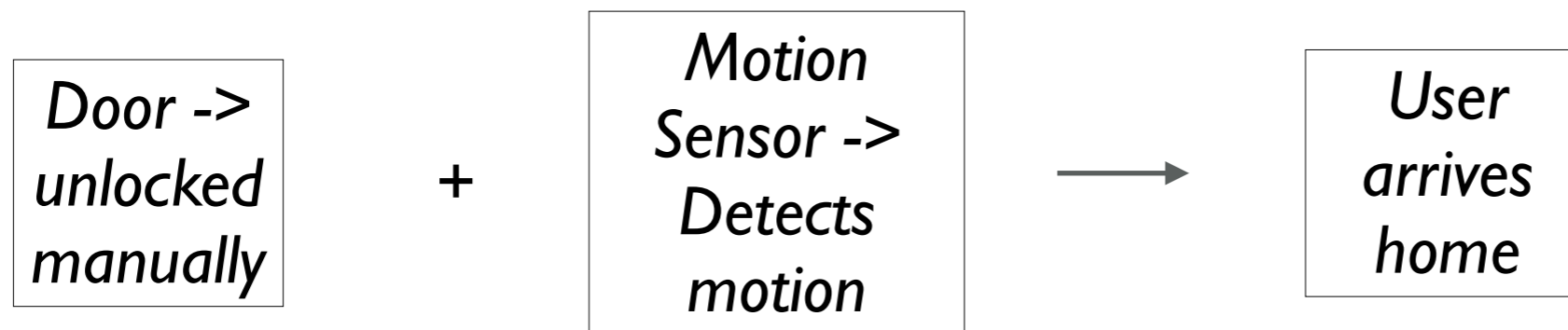
# LEVERAGING THE SMART HOME

*Home  
Devices*



*Have real-time local insight into homes!*

*Example:*



# POLICY ENFORCEMENT USING DEVICES

*Endorse an AHO update request from API using device insights!*

