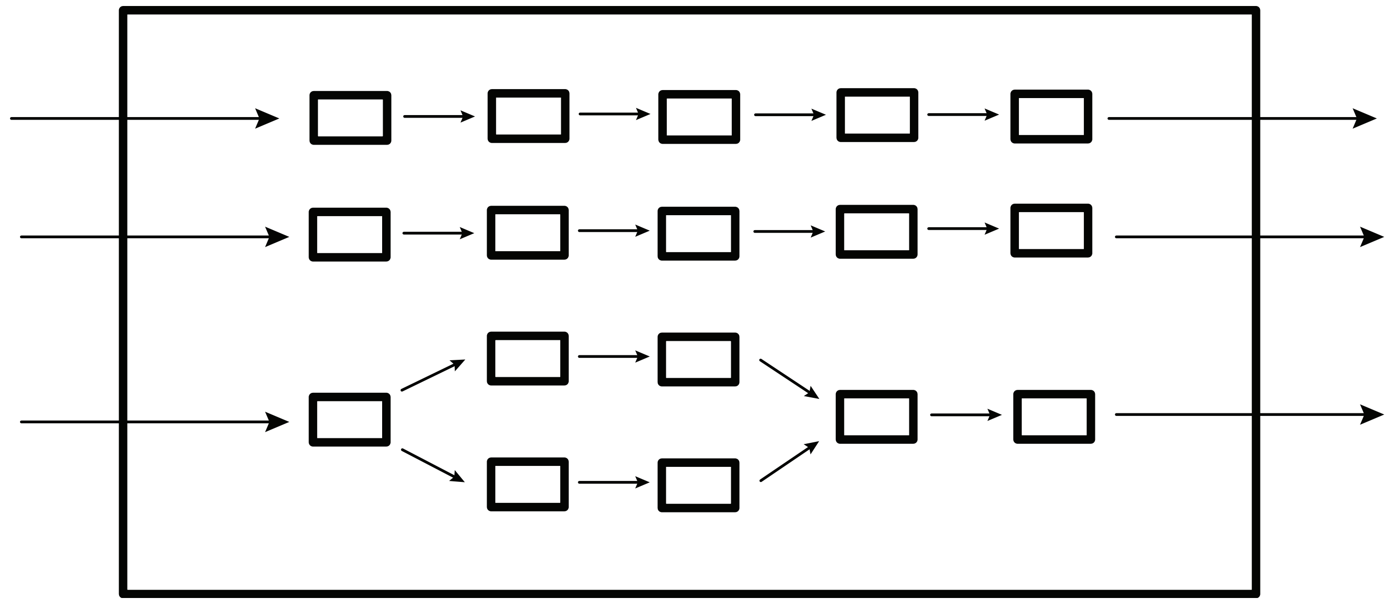
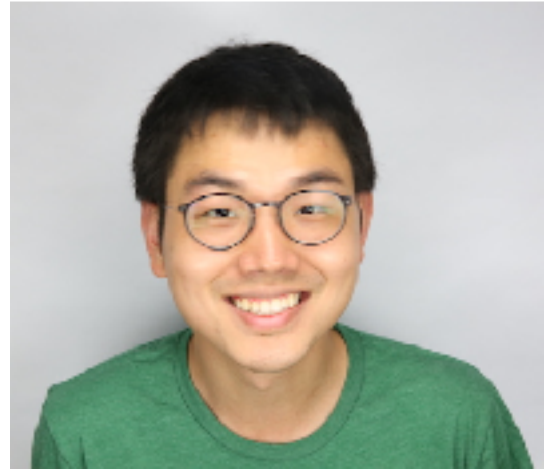
Peekaboo

A Hub-Based Approach to Enable Transparency in Data Processing within Smart Homes





Haojian Jin

Gram Liu

David Hwang

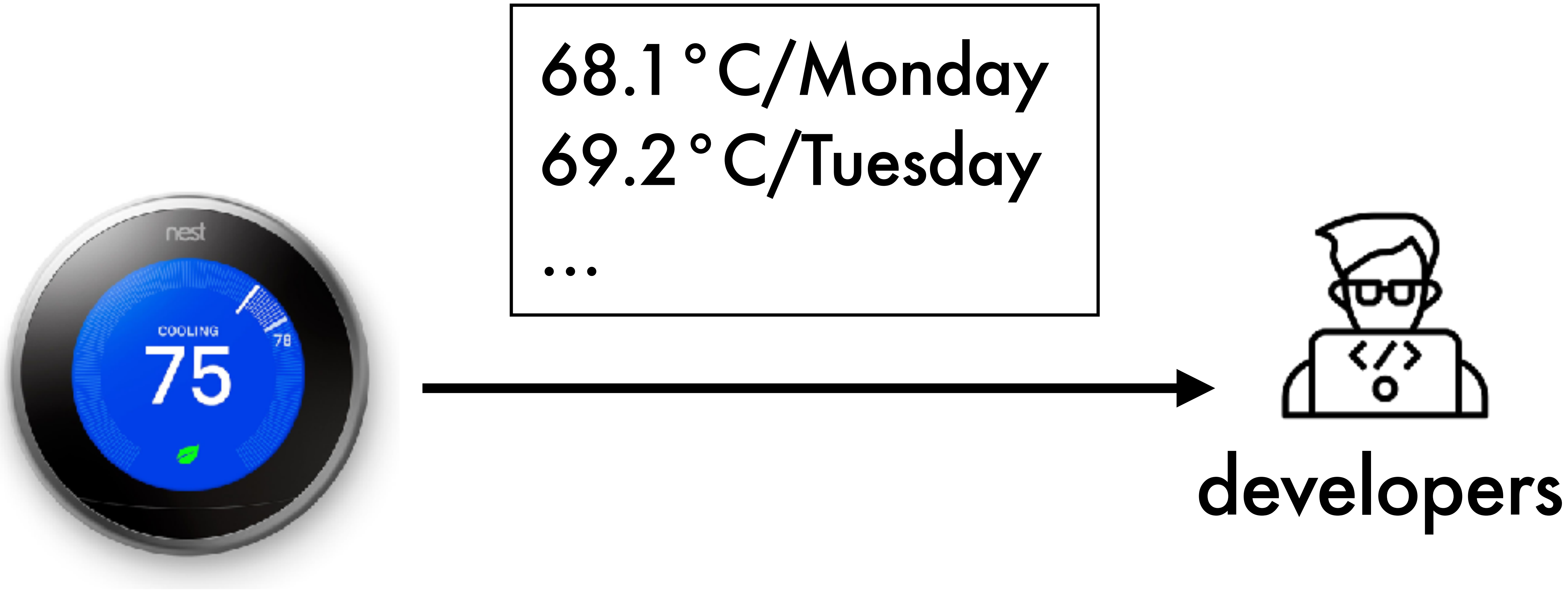
Swarun Kumar

Yuvraj Agarwal

Jason Hong

Carnegie Mellon University

How can Nest prove that they only collect aggregated data?



68.1°C/Monday
69.2°C/Tuesday
...

developers

Open source?

Your TV watch history contains too much insights

video #	duration	name	time	...
aaa	-	-	-	-
bbb	-	-	-	-
...	...	...	...	...

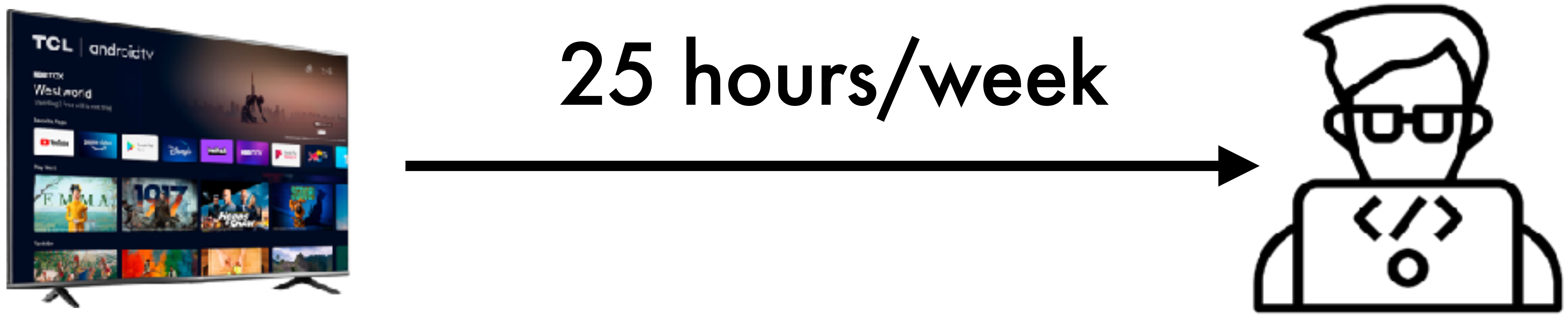
25 hours/week

How much time does the user spend on the TV?

- Is the user at home
- Activity routine
- User interests
- ....

Best practice

Only collect the necessary data for a specific purpose.

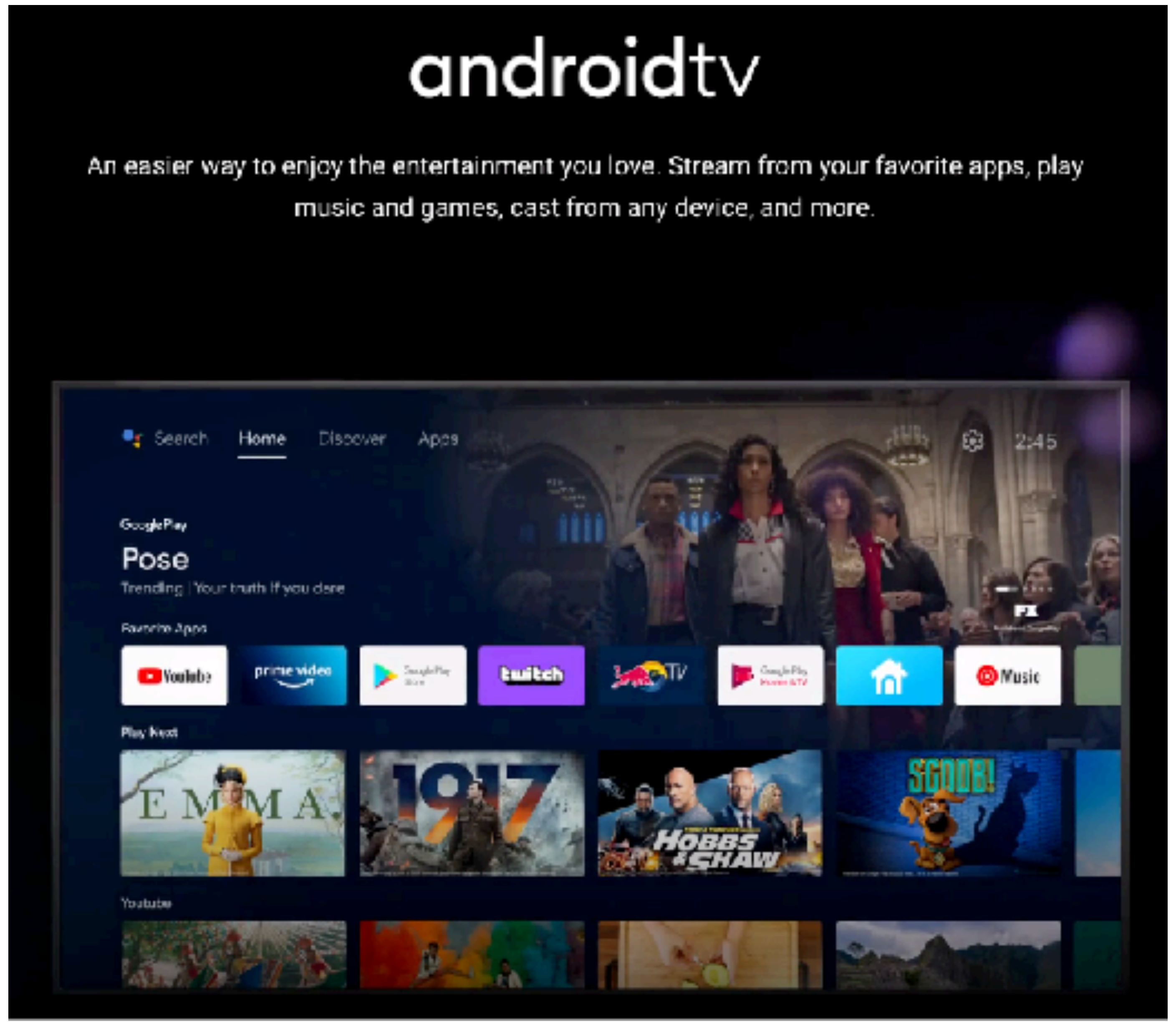


Use weekly usage data to measure device engagement.

- ~~Is the user at home~~
- ~~Activity routine~~
- ~~User interests~~
- ....

How can developers prove themselves?

A strawman solution: fine-grained permission manifest



An easier way to enjoy the entertainment you love. Stream from your favorite apps, play music and games, cast from any device, and more.

https://www.android.com/tv/

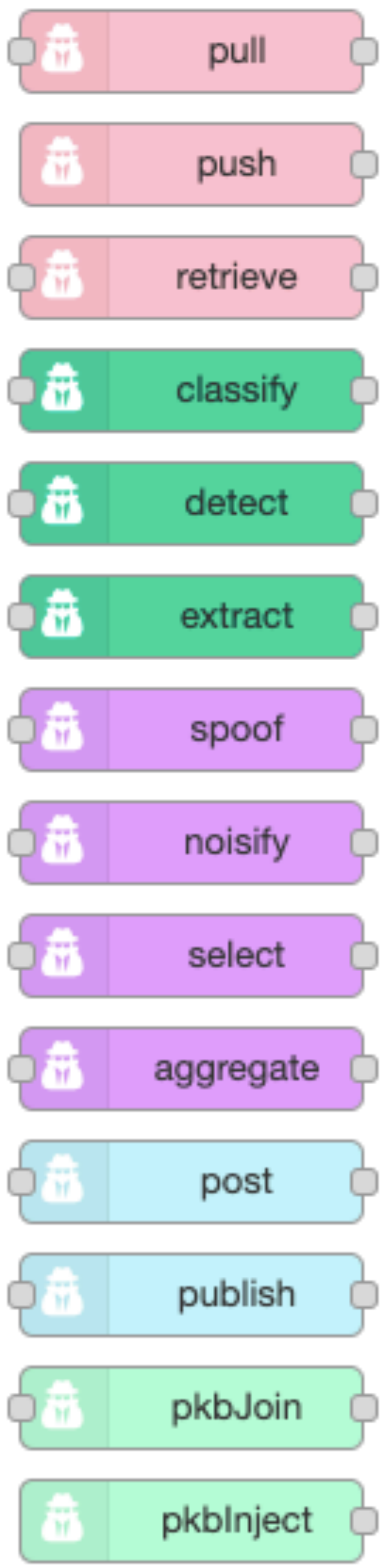
```
<manifest ...>
  <uses-permission android:name="android.permission.
       TV_AGGREGATED_DURATION_WEEKLEY" />

  <uses-permission android:name="android.permission.
       TV_AGGREGATED_DURATION_DAILY" />


  ......
</manifest>
```

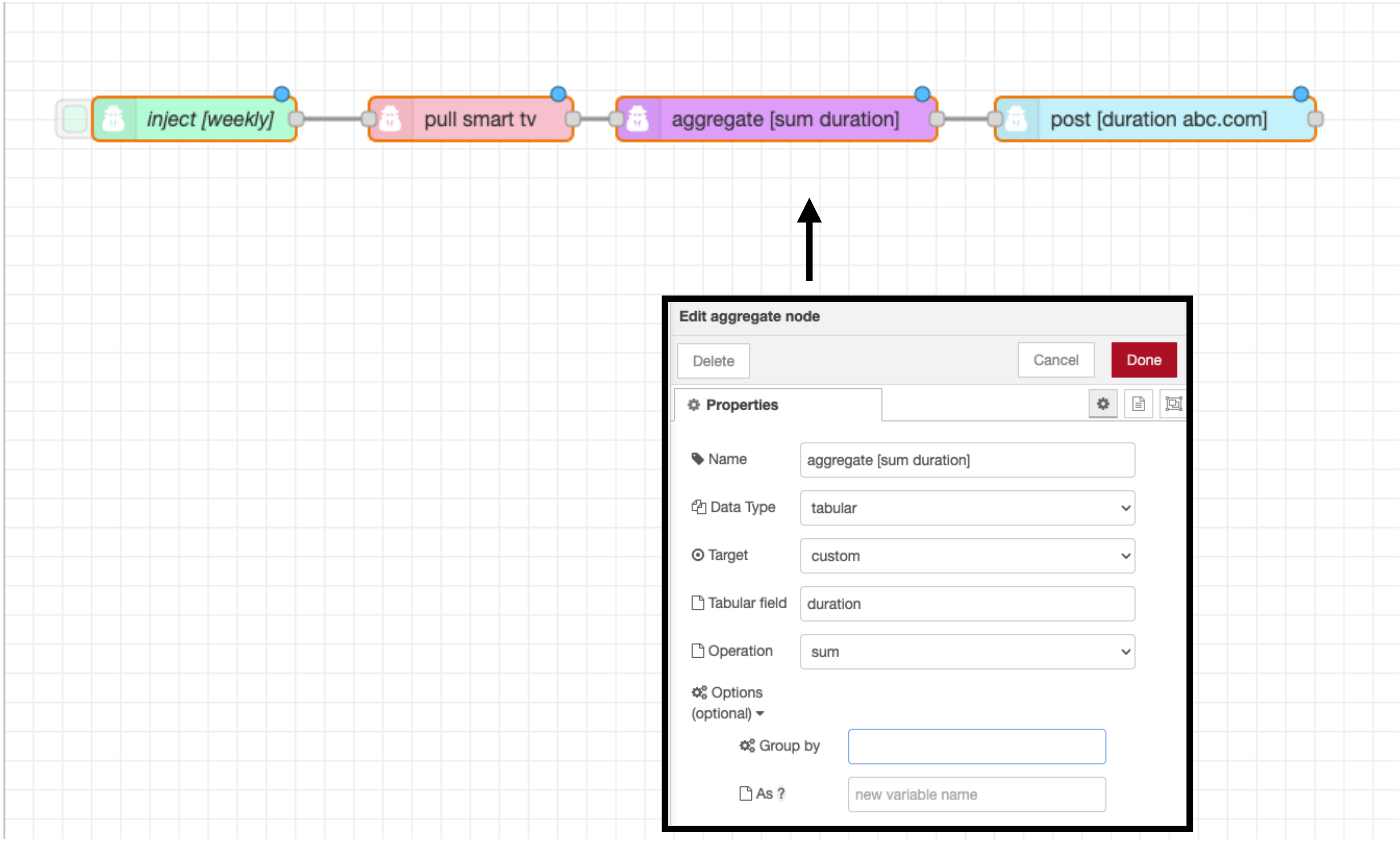
Fine-grained permission manifest

Peekaboo primitive (1)

Program pre-processing functions using chainable *operators*

A fixed set of operators



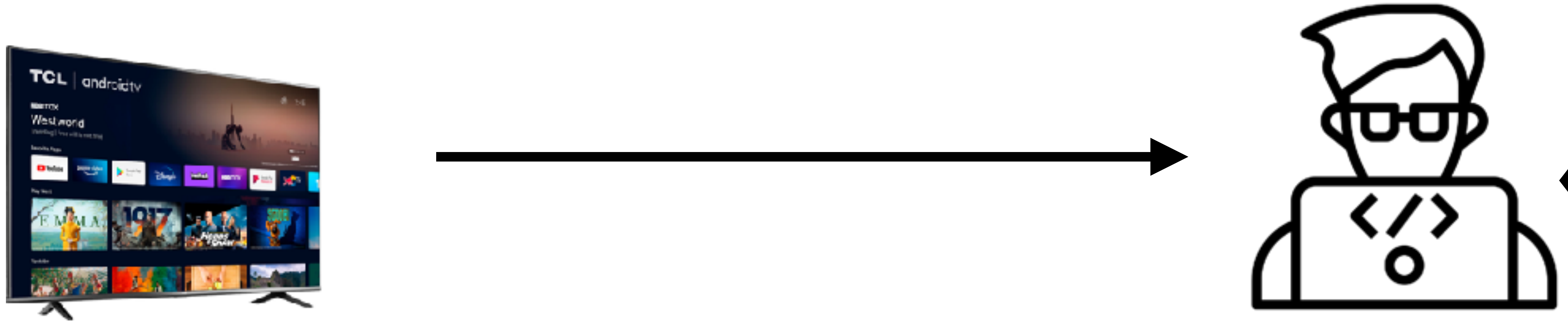


Edit aggregate node

Name	aggregate [sum duration]
Data Type	tabular
Target	custom
Tabular field	duration
Operation	sum
Options (optional)	
Group by	
As ?	new variable name

Peekaboo primitive (2)

A text-based whitelist *manifest* (i.e., program representation)

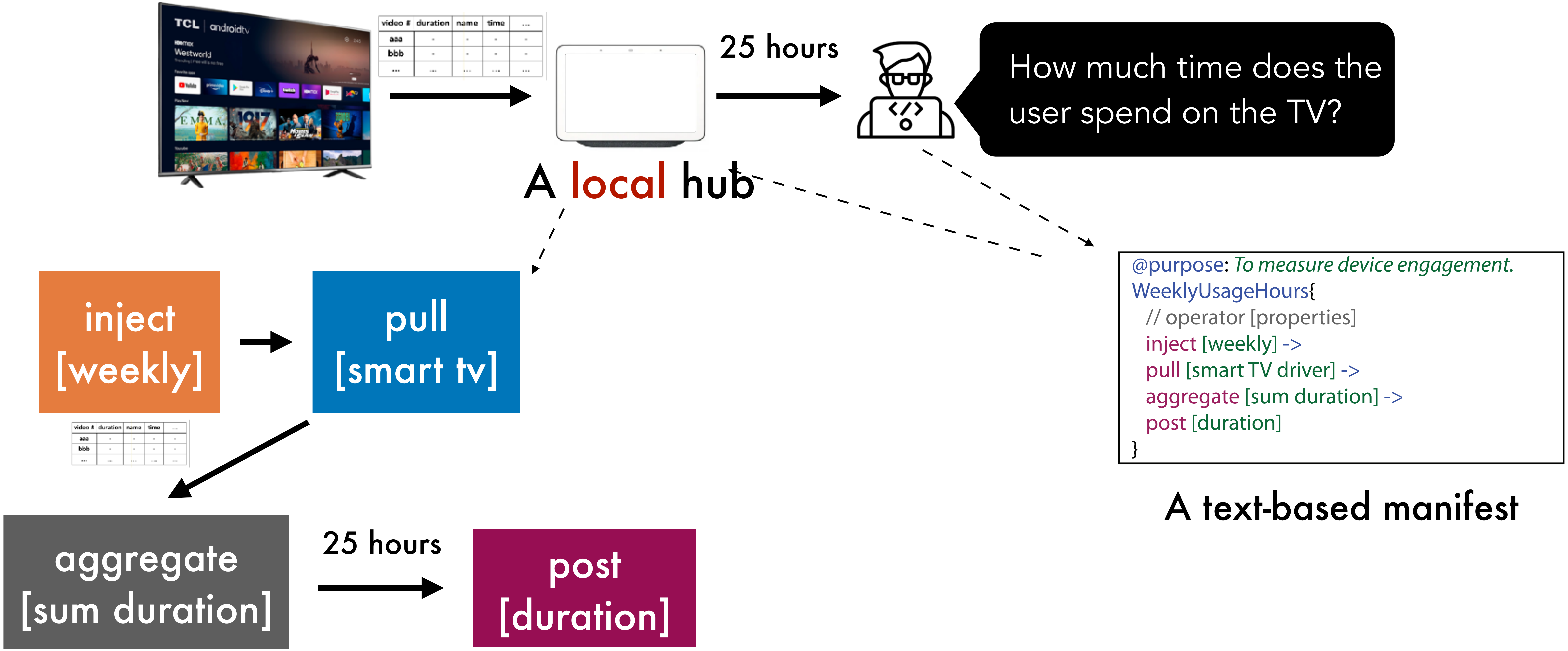


How much time does the user spend on the TV?

```
@purpose: To measure device engagement.
WeeklyUsageHours{
  // operator [properties]
  inject [weekly] ->
  pull [smart TV driver] ->
  aggregate [sum duration] ->
  post [duration]
}
```

Peekaboo primitive (3)

A trusted runtime with pre-loaded implementations



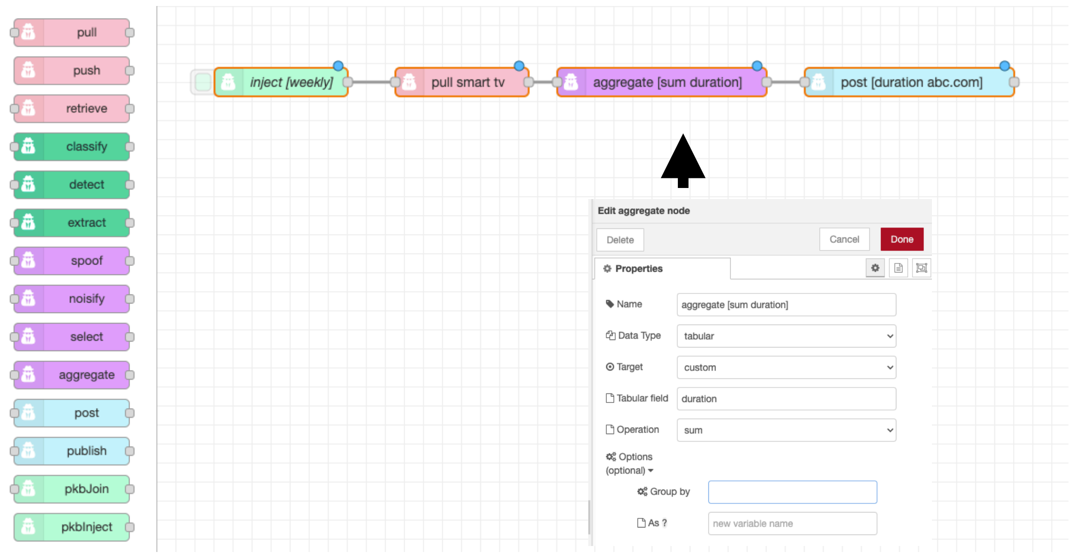
video #	duration	name	time	...
aaa	-	-	-	-
bbb	-	-	-	-
...	...	...	...	...

```
@purpose: To measure device engagement.
WeeklyUsageHours{
  // operator [properties]
  inject [weekly] ->
  pull [smart TV driver] ->
  aggregate [sum duration] ->
  post [duration]
}
```

A text-based manifest

Smart home app store

App developers



Edit aggregate node

Name: aggregate [sum duration]

Data Type: tabular

Target: custom

Tabular field: duration

Operation: sum

Options (optional)

Group by:

As ?: new variable name

Programming environment with operators

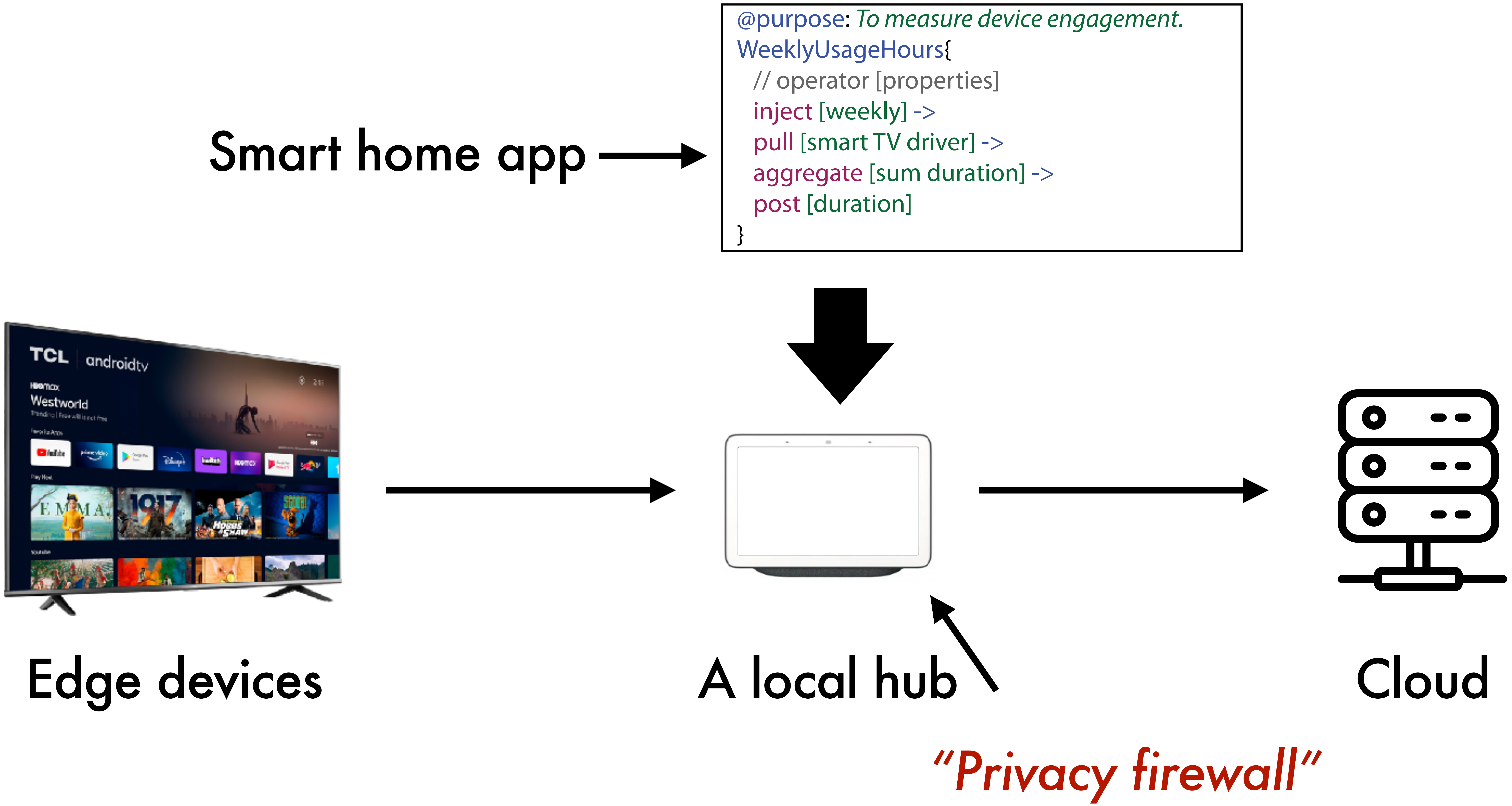
Runtime with preloaded implementations

```
@purpose: To measure device engagement.
WeeklyUsageHours{
  // operator [properties]
  inject [weekly] ->
  pull [smart TV driver] ->
  aggregate [sum duration] ->
  post [duration]
}
```

Manifest

Peekaboo adoption

Smart home app store



Smart home app

```
@purpose: To measure device engagement.
WeeklyUsageHours{
  // operator [properties]
  inject [weekly] ->
  pull [smart TV driver] ->
  aggregate [sum duration] ->
  post [duration]
}
```

Edge devices

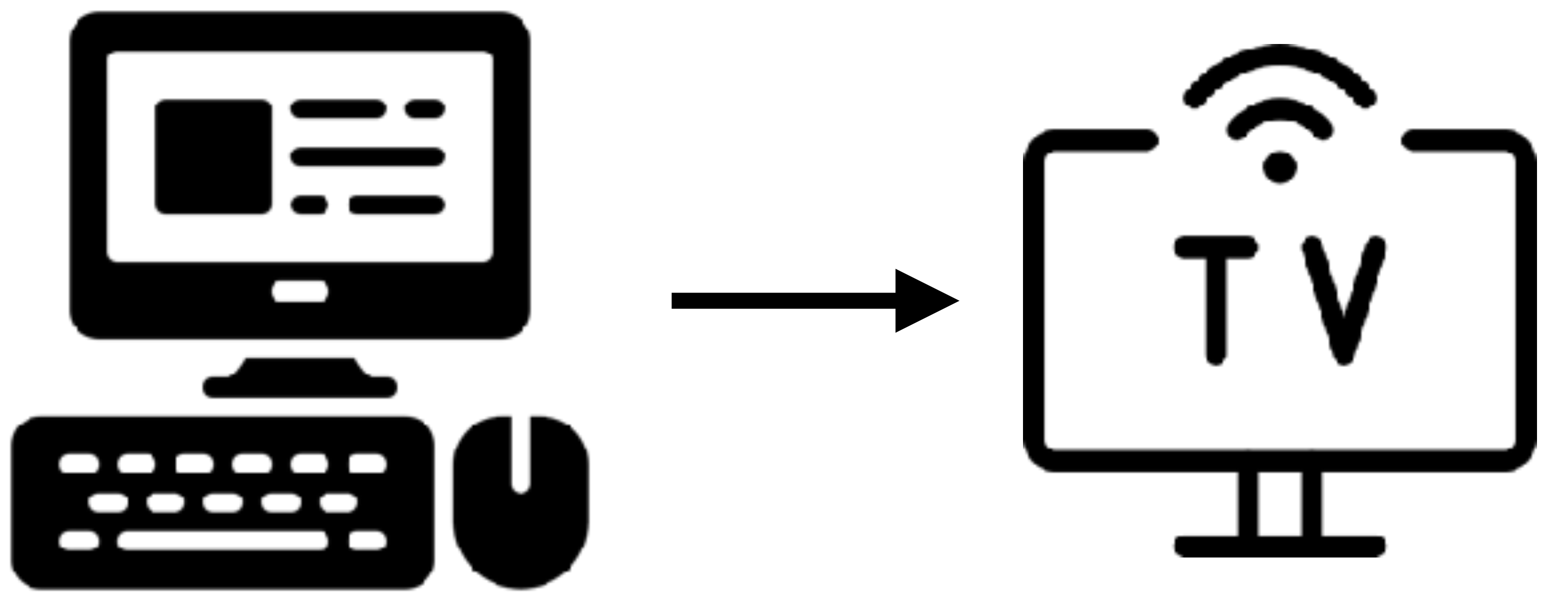
A local hub

Cloud

"Privacy firewall"

Peekaboo v.s. Firewall (1)

Developers declare purposes explicitly.



Simpler functionality

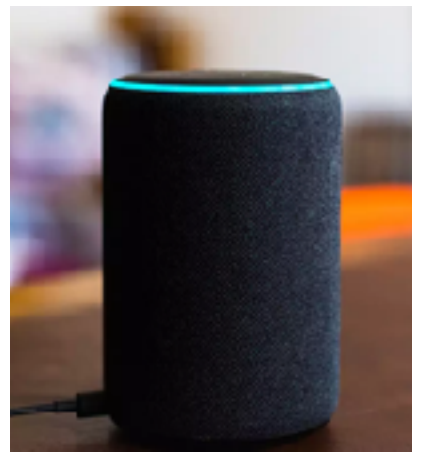
Whitelist-only

```
@purpose: To measure device engagement.
WeeklyUsageHours{
```

Developer-in-the-loop

Insight #2

77% Apps do not need raw data.

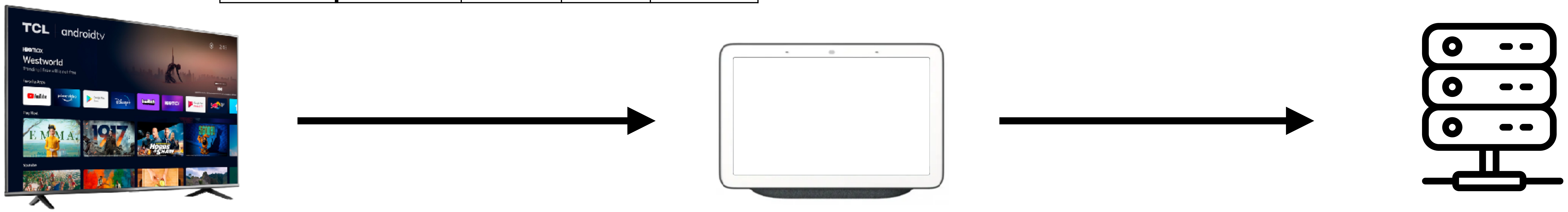
	Sensor	Raw	Needed data
Hello visitor			
Noise level			55 db

Peekaboo v.s. Firewall #2

Pre-process users’ data to mitigate data overaccess.

video #	duration	name	time	...
aaa	-	-	-	-
bbb	-	-	-	-
...	...	...	...	...

25 hours/week

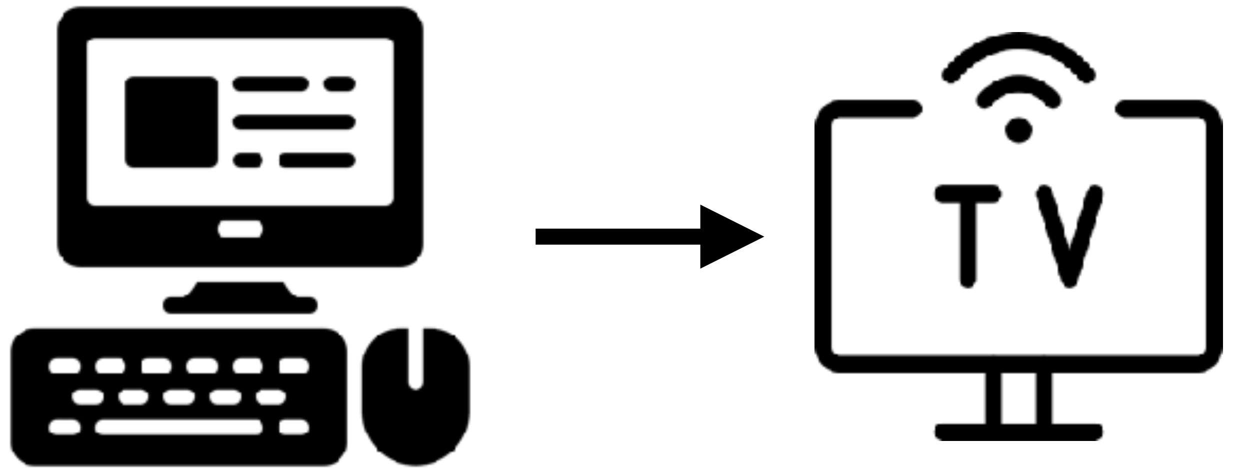


Edge devices

Hub

Cloud

Recap: Peekaboo v.s. Firewall



Simpler functionality

Whitelist-only

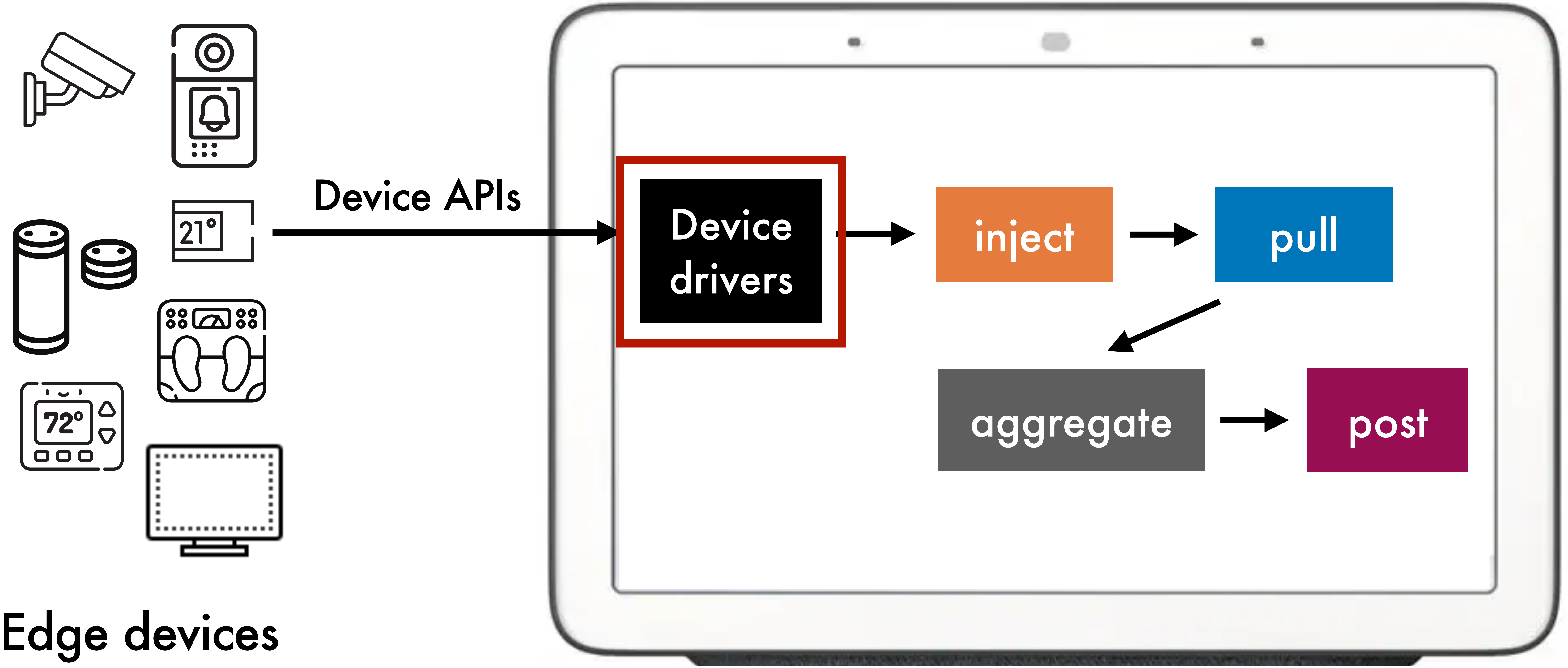
Developer-in-the-loop

77% Apps do not need raw data.

Pre-process users' data

How Peekaboo works

Handle heterogeneous hardware with device drivers

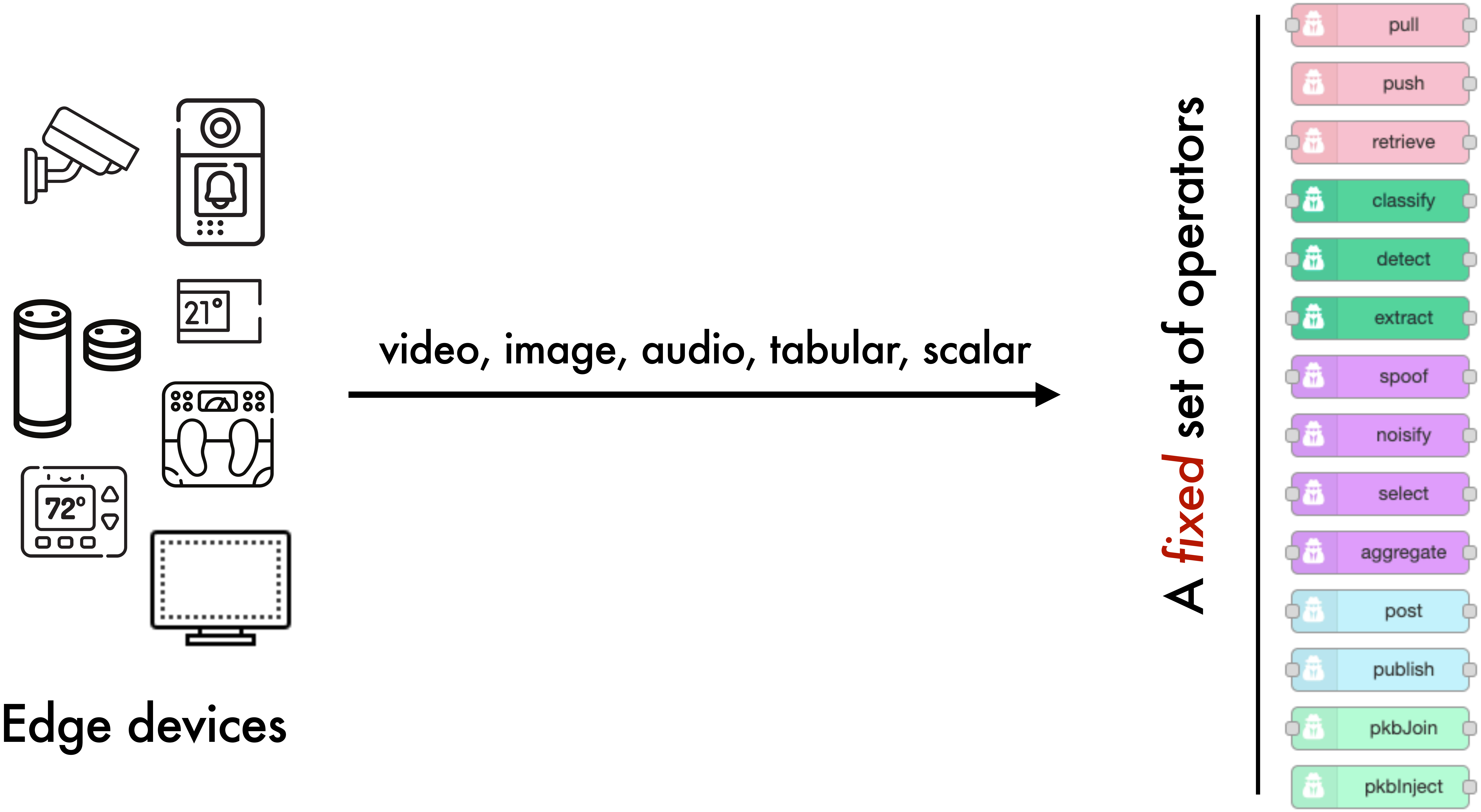


Edge devices

HomeAssistant, https://www.home-assistant.io/

How Peekaboo works

A fixed set of operators



Edge devices

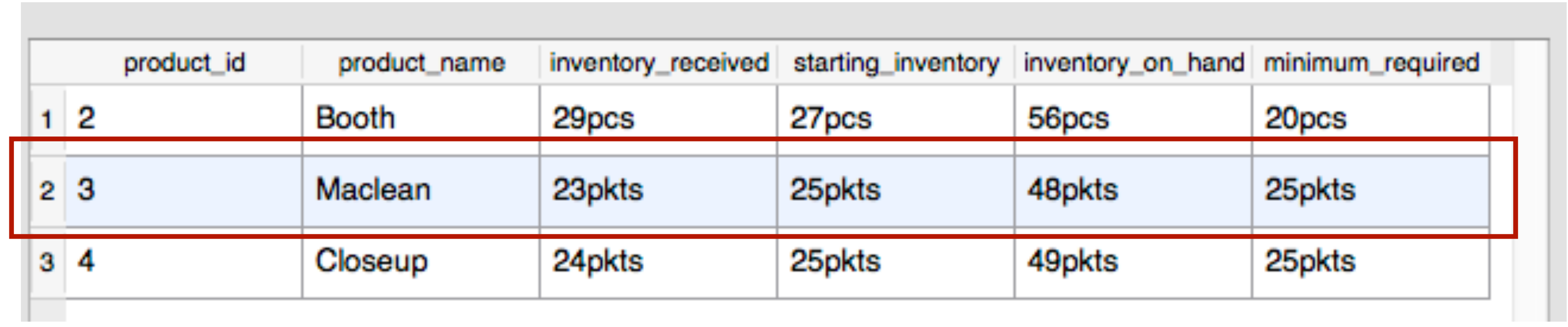
video, image, audio, tabular, scalar

A *fixed* set of operators

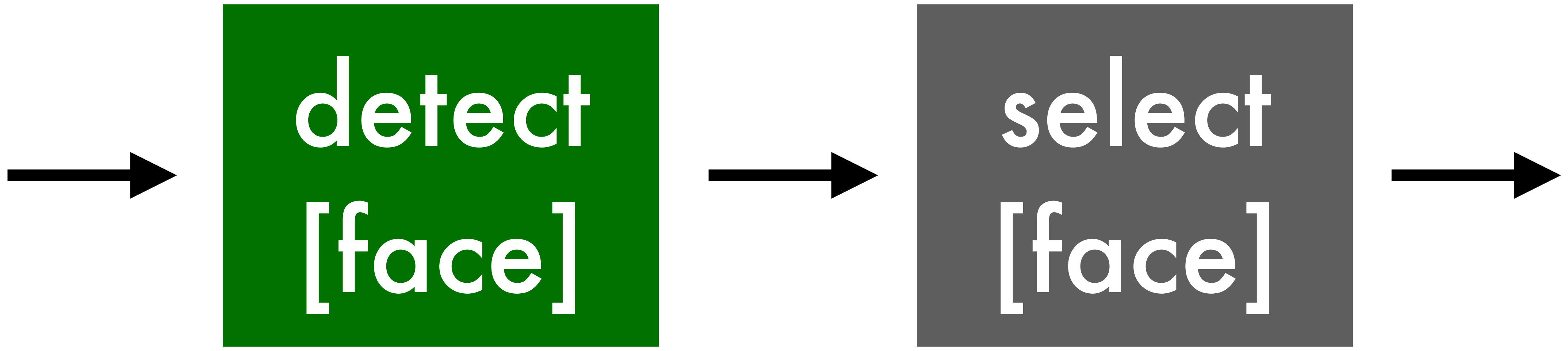
How Peekaboo works

An operator = A verb keyword

select
[row]



	product_id	product_name	inventory_received	starting_inventory	inventory_on_hand	minimum_required
1	2	Booth	29pcs	27pcs	56pcs	20pcs
2	3	Maclean	23pkts	25pkts	48pkts	25pkts
3	4	Closeup	24pkts	25pkts	49pkts	25pkts

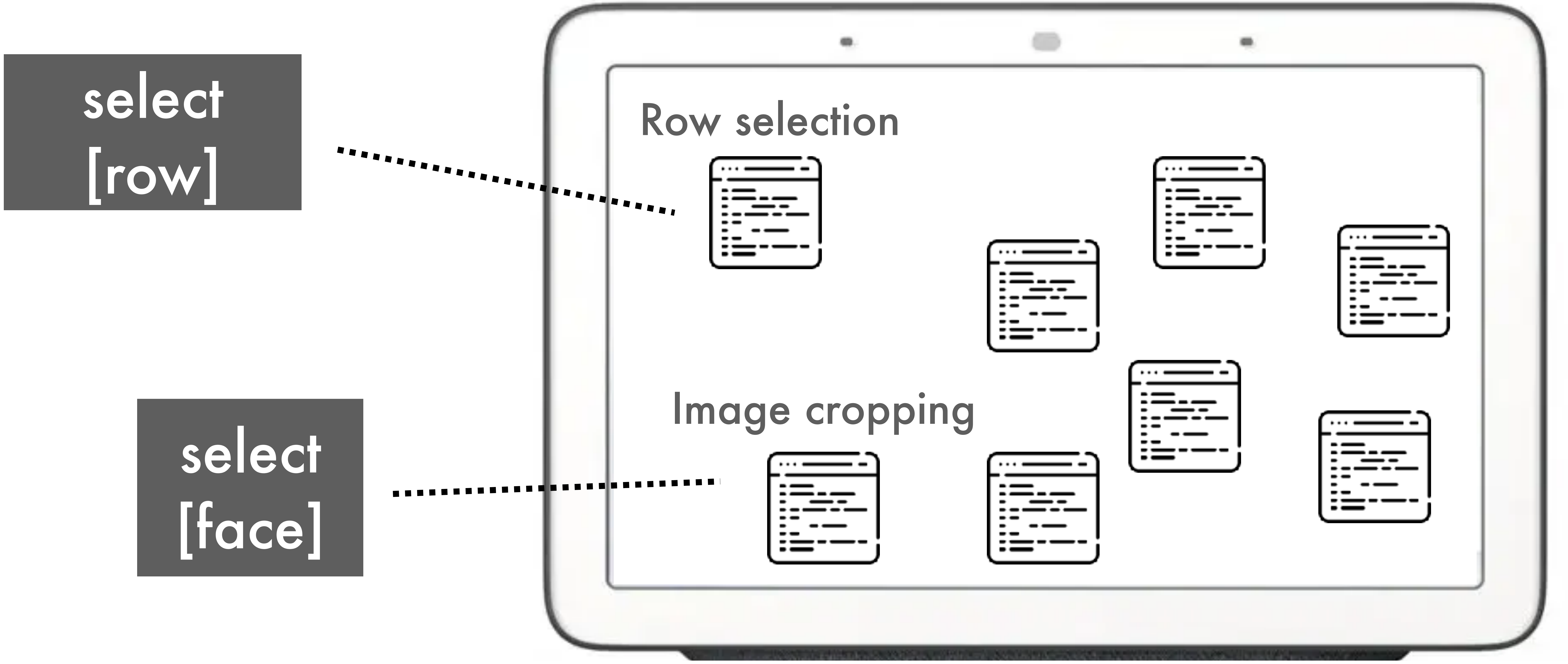


detect
[face]

select
[face]

How Peekaboo works

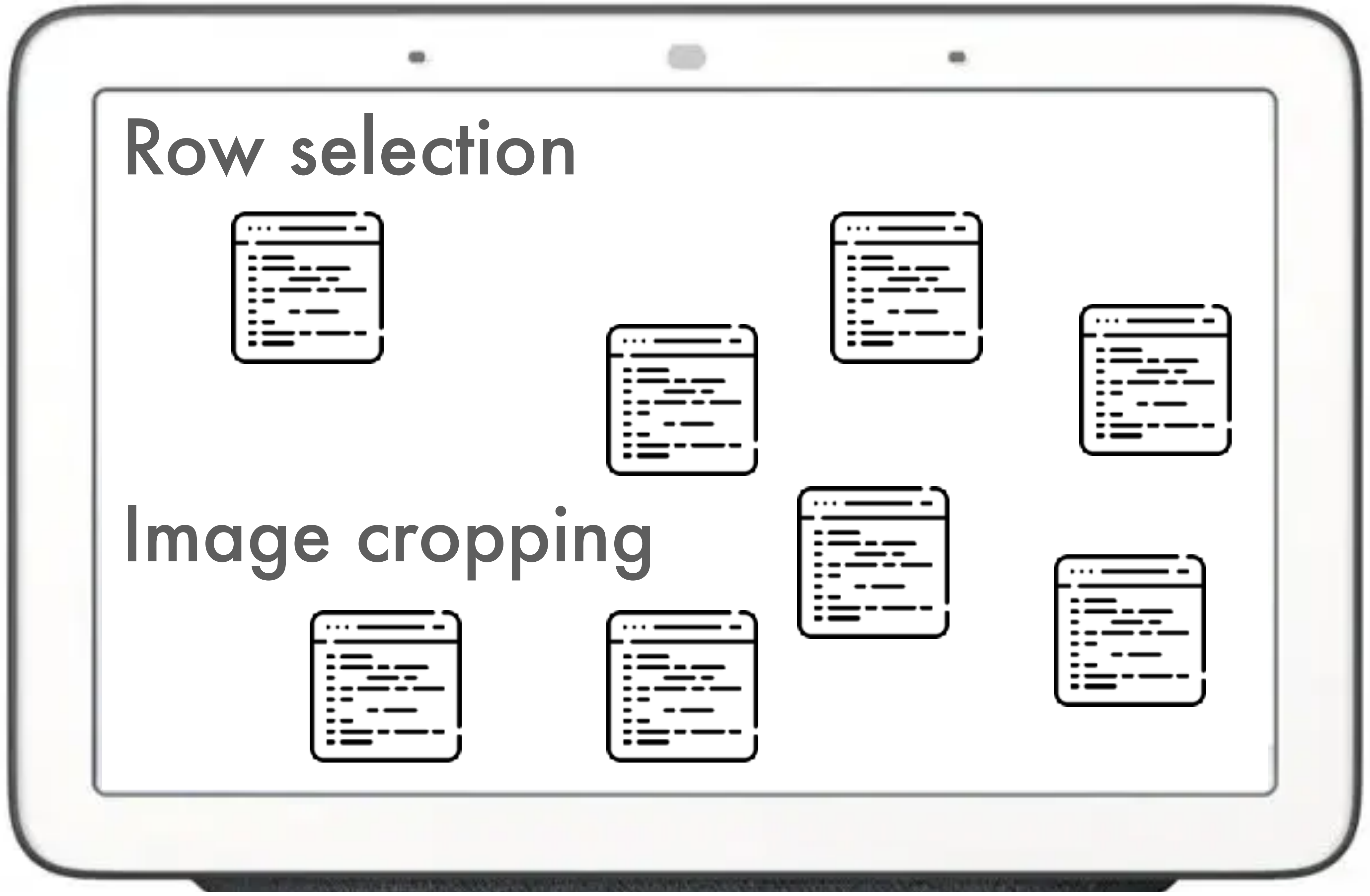
Operators are mapped to pre-loaded implementations



How Peekaboo works

A small set of pre-processing algorithms improve privacy

video #	duration	name	time	...
aaa	-	-	-	-
bbb	-	-	-	-
...	...	...	...	...



25 hours/week

Implementation (hardware)

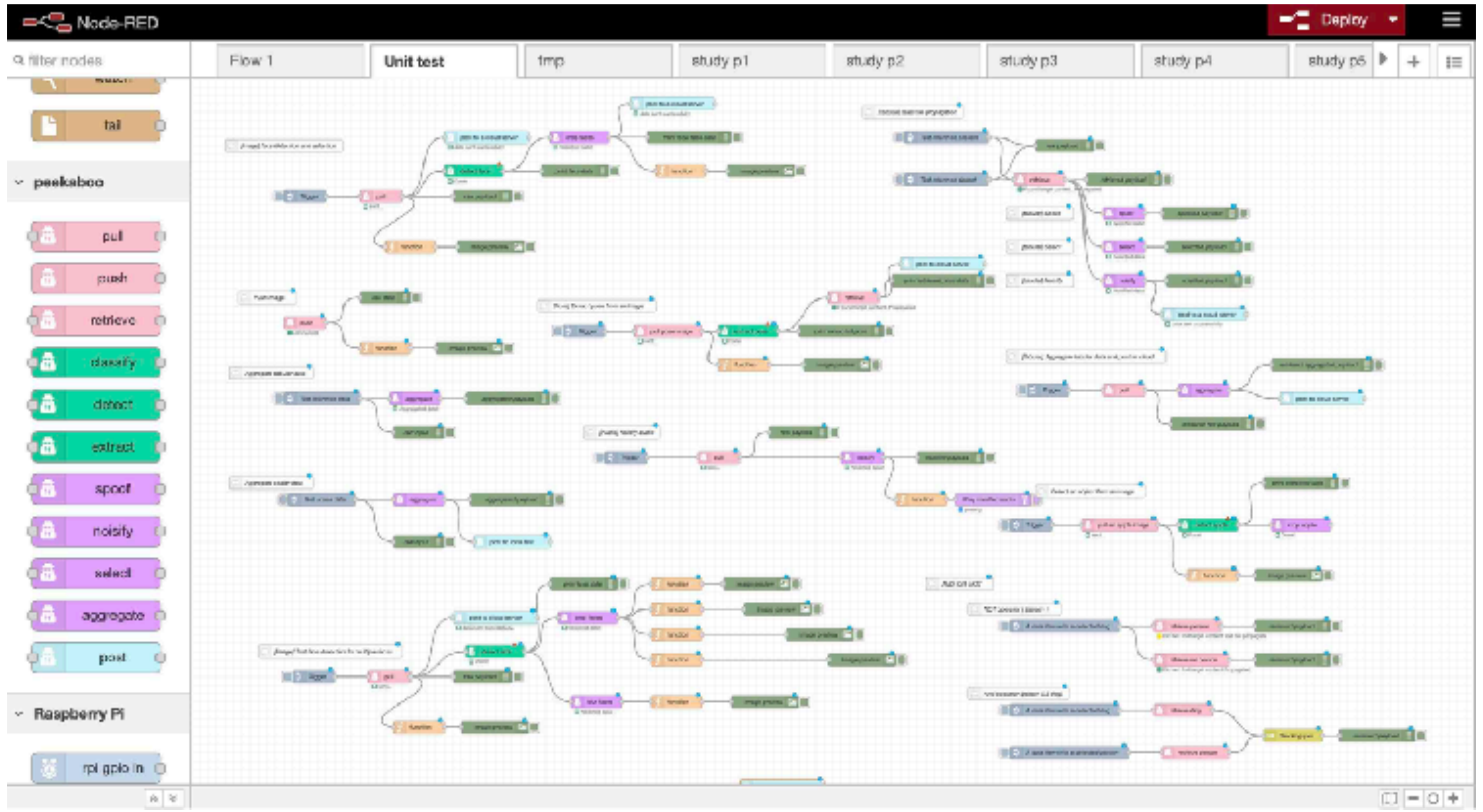
Edge devices

Raspberry PI + TPU

Cloud

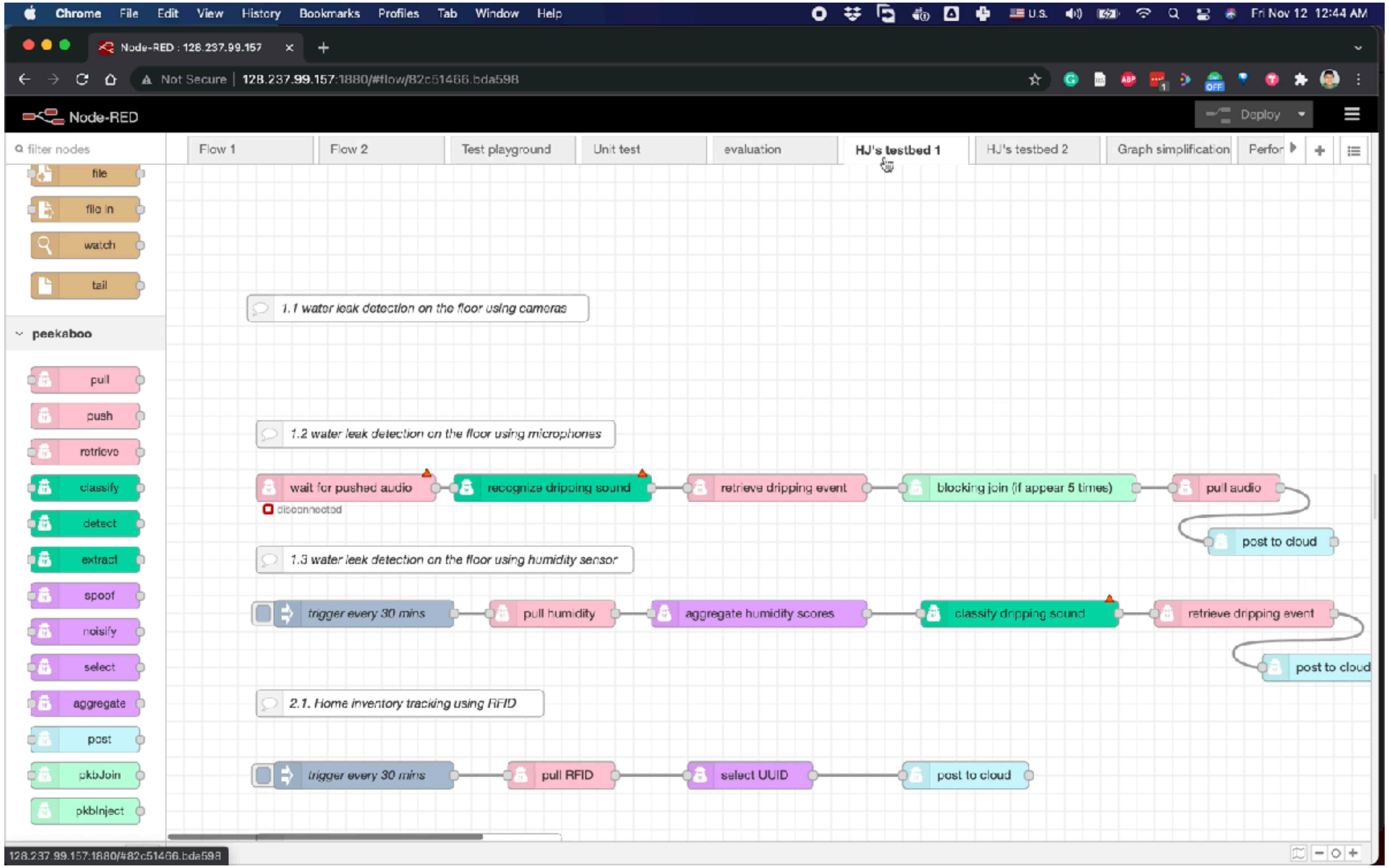
Implementation (software)

1. Operators: Node.JS package
2. Programming IDE: NodeRed
3. Drivers: 5 data types
4. 23 Preloaded implementations



Evaluation

Expressiveness (200+ smart home cases)



1.1 water leak detection on the floor using cameras

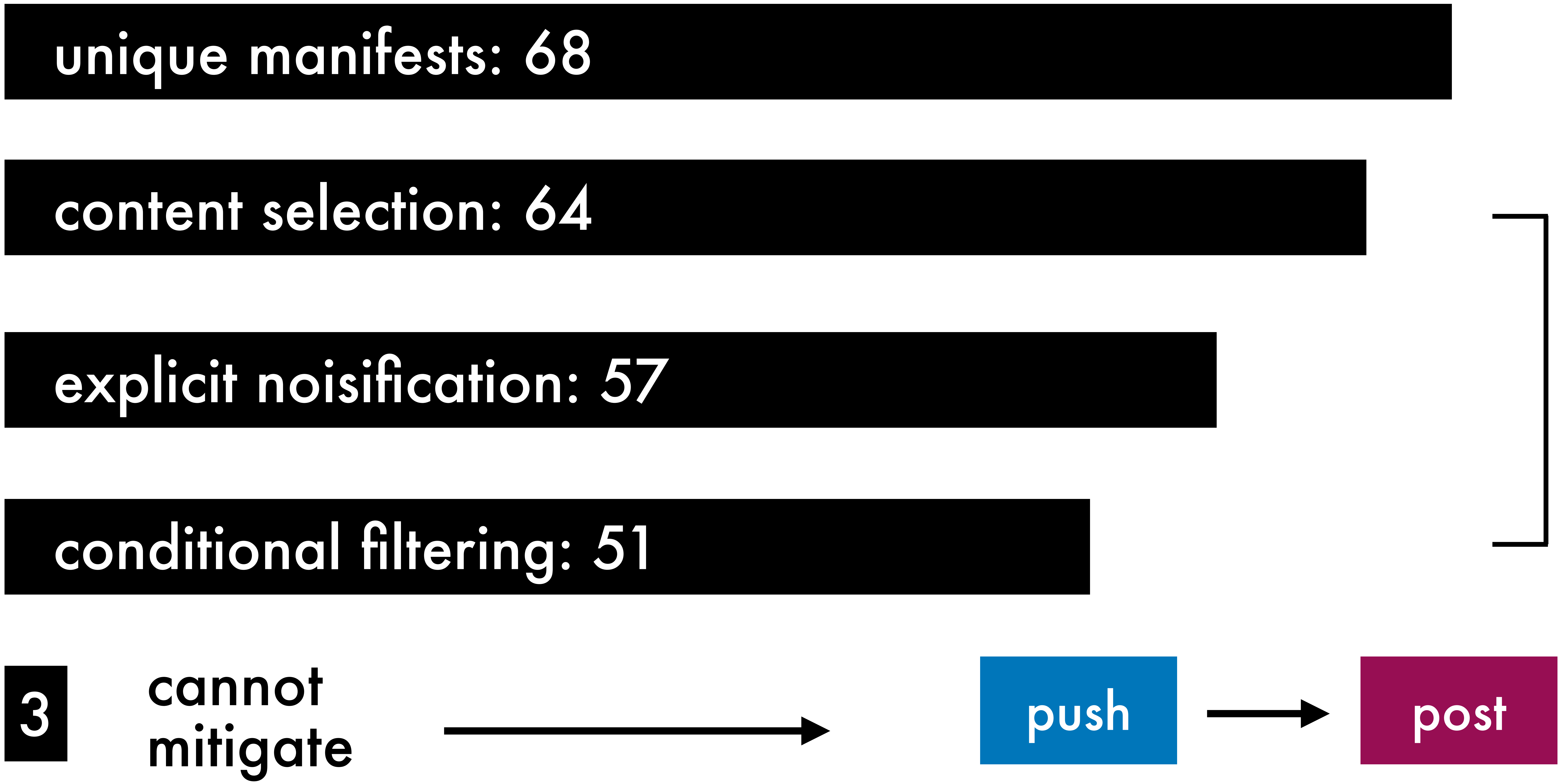
1.2 water leak detection on the floor using microphones

1.3 water leak detection on the floor using humidity sensor

2.1. Home inventory tracking using RFID

Evaluation (expressiveness)

Data overaccess mitigation breakdown



unique manifests: 68

content selection: 64

explicit noisification: 57

conditional filtering: 51

3 cannot mitigate → push → post

See details in the paper

Evaluation

System performance

≈$100

25 inference/s

100 filtering/s

1-80 ms per request

Evaluation

Utility privacy tradeoff example

incognito voice assistant

6 speakers
112 audio files [1]

noisify <5% random pitch shift

Microsoft Cognitive Services

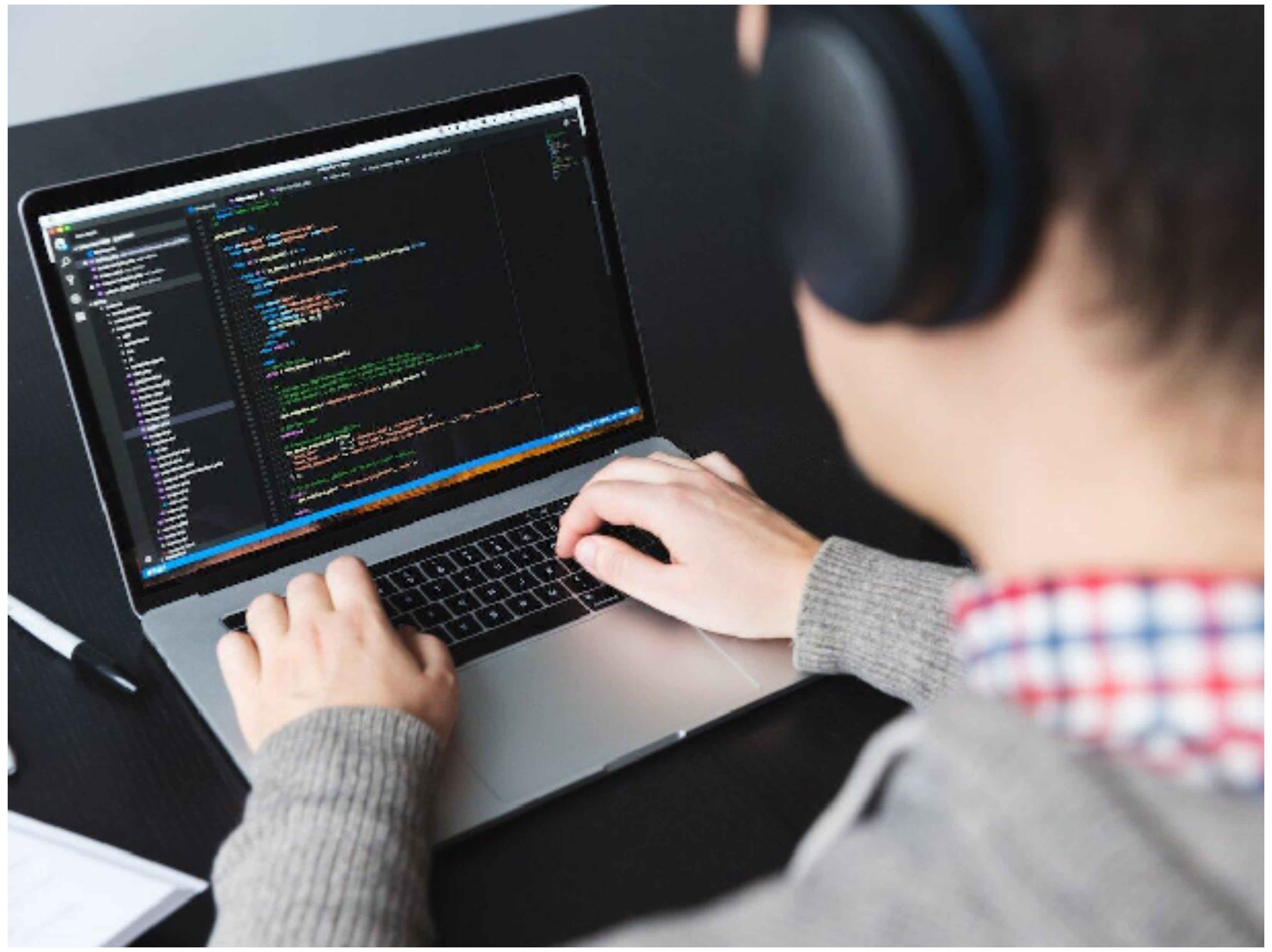
Speech word error rate:
9.27% → 11.88%

Speaker recognization:
100% → 27.7%

[1] CMU PDA Speech Database

Evaluation

Developer studies



Task descriptions

IDE & Unit tests

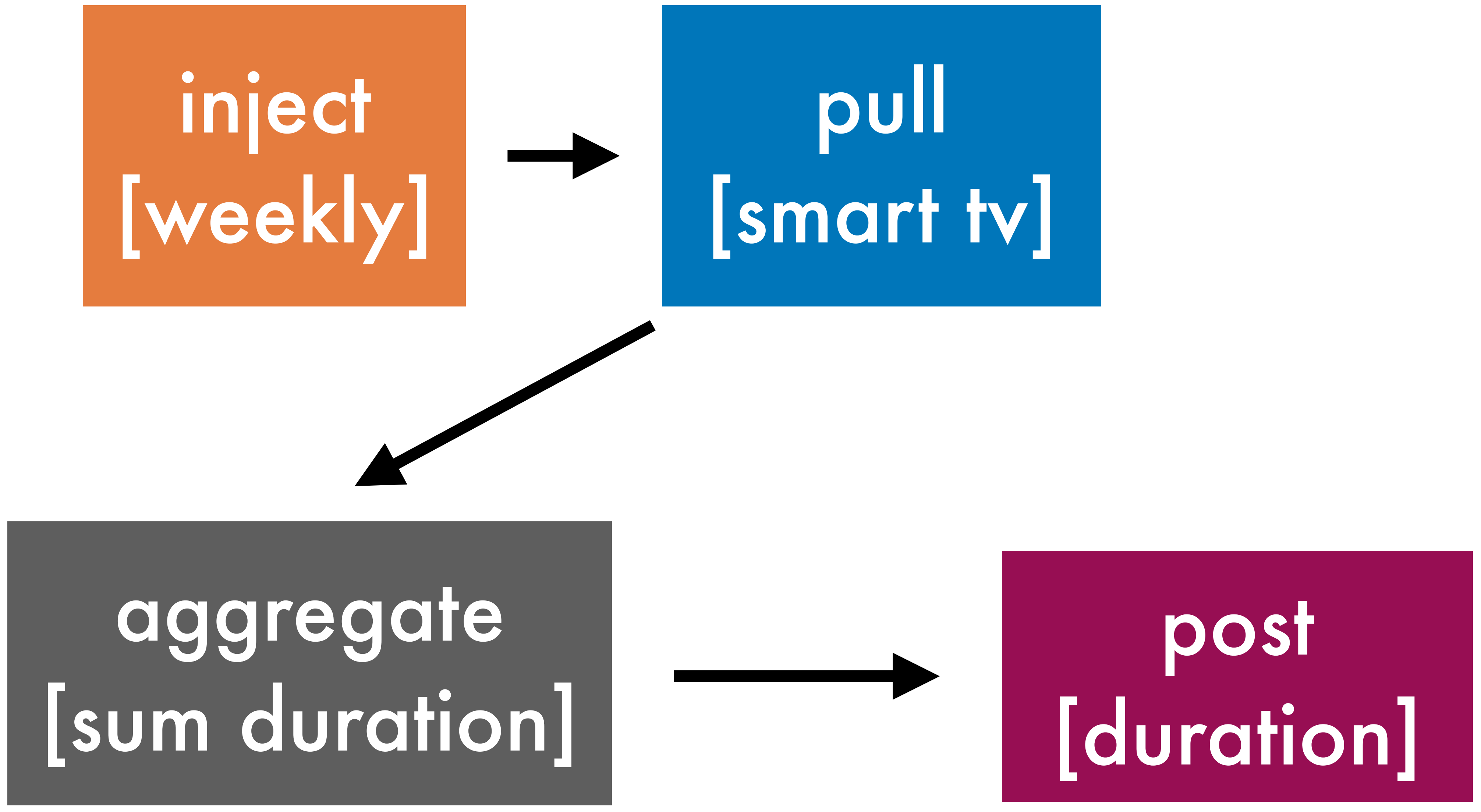
6 - 15 mins to author a manifest

Advantages

Manifests enforce fine-grained data collection

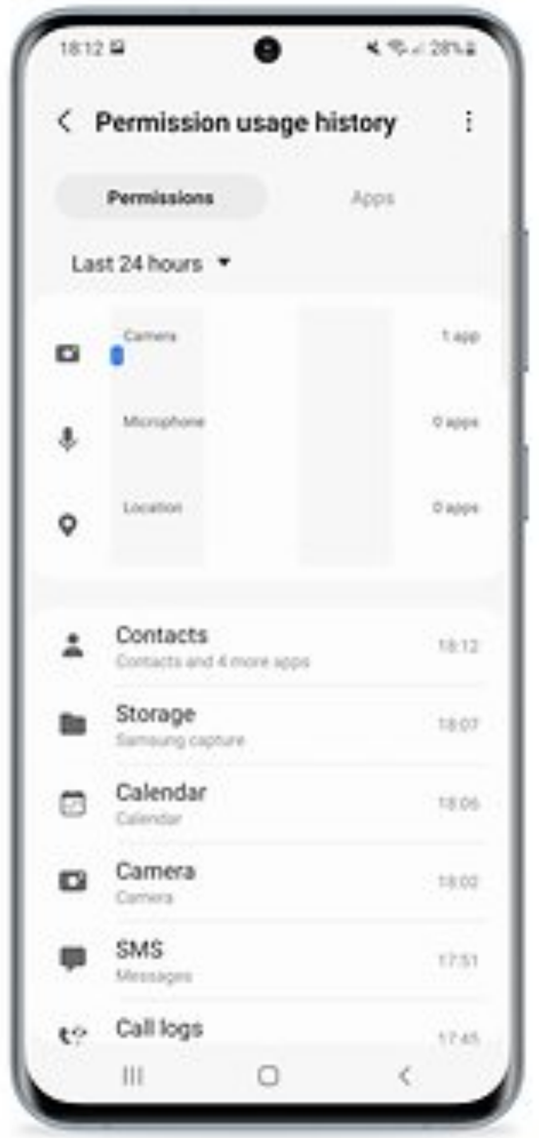
```
@purpose: To measure device engagement.
WeeklyUsageHours{
  // operator [properties]
  inject [weekly] ->
  pull [smart TV driver] ->
  aggregate [sum duration] ->
  post [duration]
}
```

public, non-proprietary



Repetitive implementation and *distributed* interfaces

Samsung

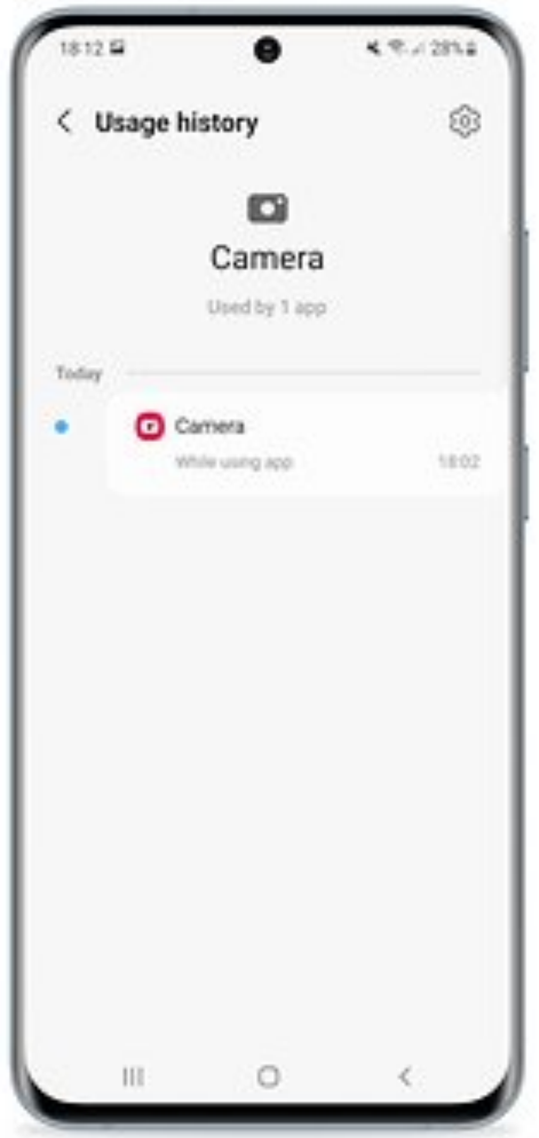

18:12 28%

Permission usage history

Permissions Apps

Last 24 hours ▾

Contacts Contacts and 4 more apps	18:12
Storage Samsung capture	18:07
Calendar Calendar	18:06
Camera Camera	18:02
SMS Messages	17:51
Call logs	17:45


18:12 28%

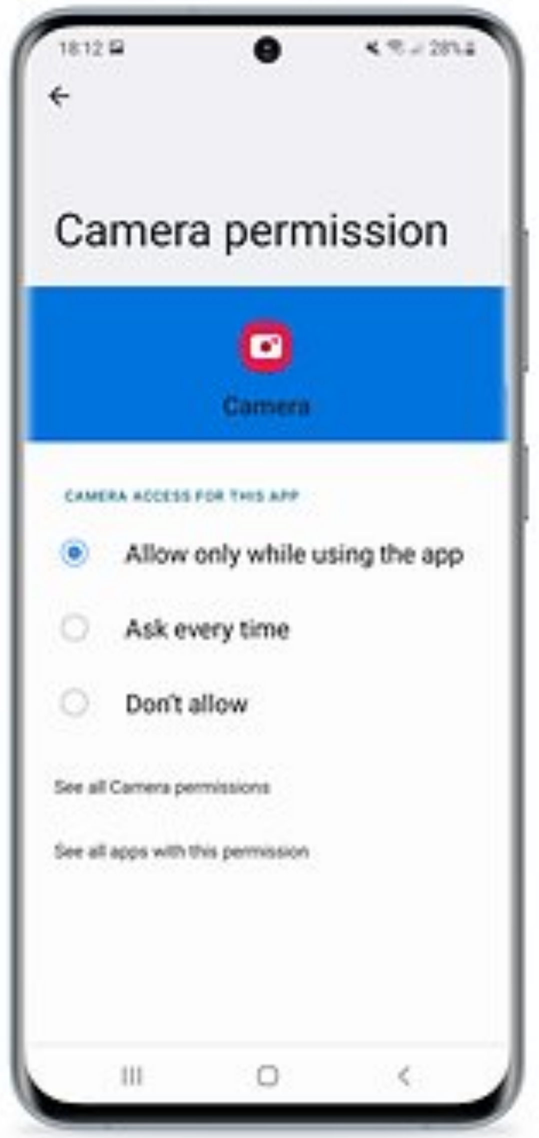
Usage history

Camera

Used by 1 app

Today

Camera
While using app 18:02


18:12 28%

Camera permission

Camera

CAMERA ACCESS FOR THIS APP

● Allow only while using the app

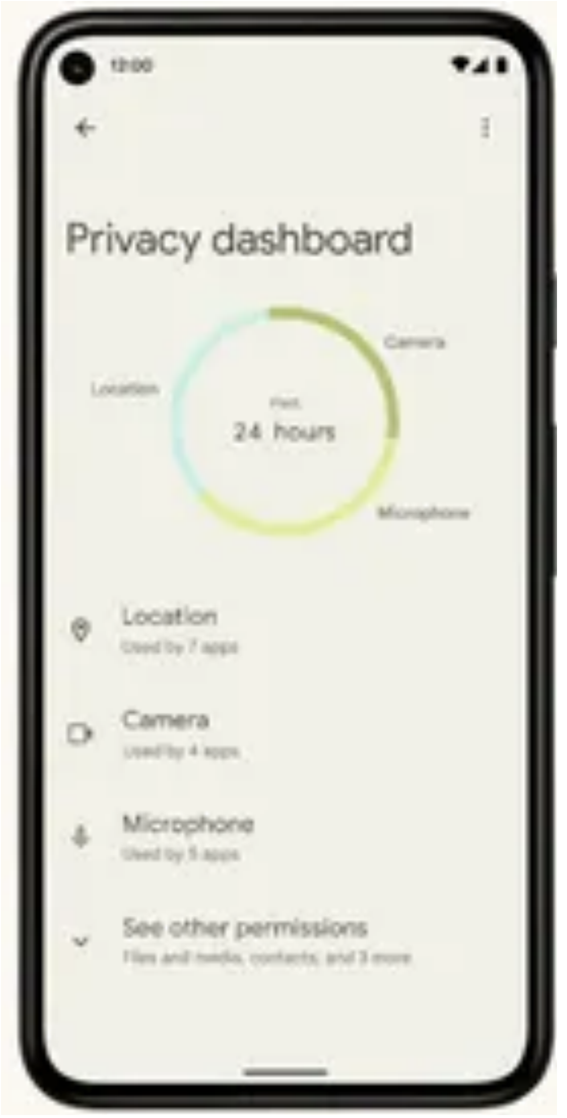
○ Ask every time

○ Don't allow

See all Camera permissions

See all apps with this permission

Nest


12:00

Privacy dashboard

Location
Used by 7 apps

Camera
Used by 4 apps

Microphone
Used by 5 apps

See other permissions
Files and media, contacts, and 3 more

Small developers?

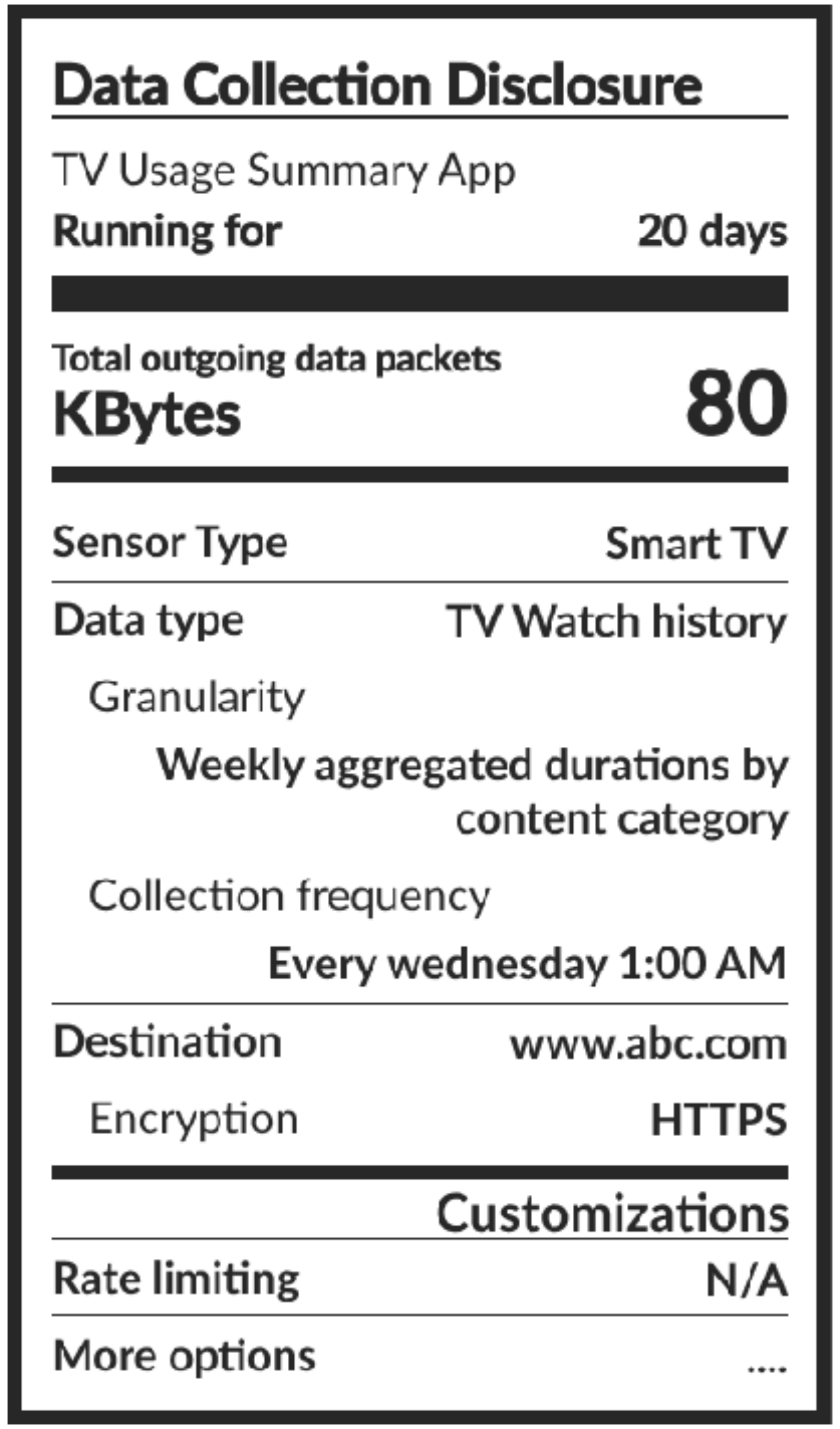
Users?

Advantages

Manifests → *enforceable/dynamic* privacy nutrition labels

```
@purpose: To measure device engagement.
WeeklyUsageHours{
  // operator [properties]
  inject [weekly] ->
  pull [smart TV driver] ->
  aggregate [sum duration] ->
  post [duration]
}
```

→



[1]

Data Collection Disclosure

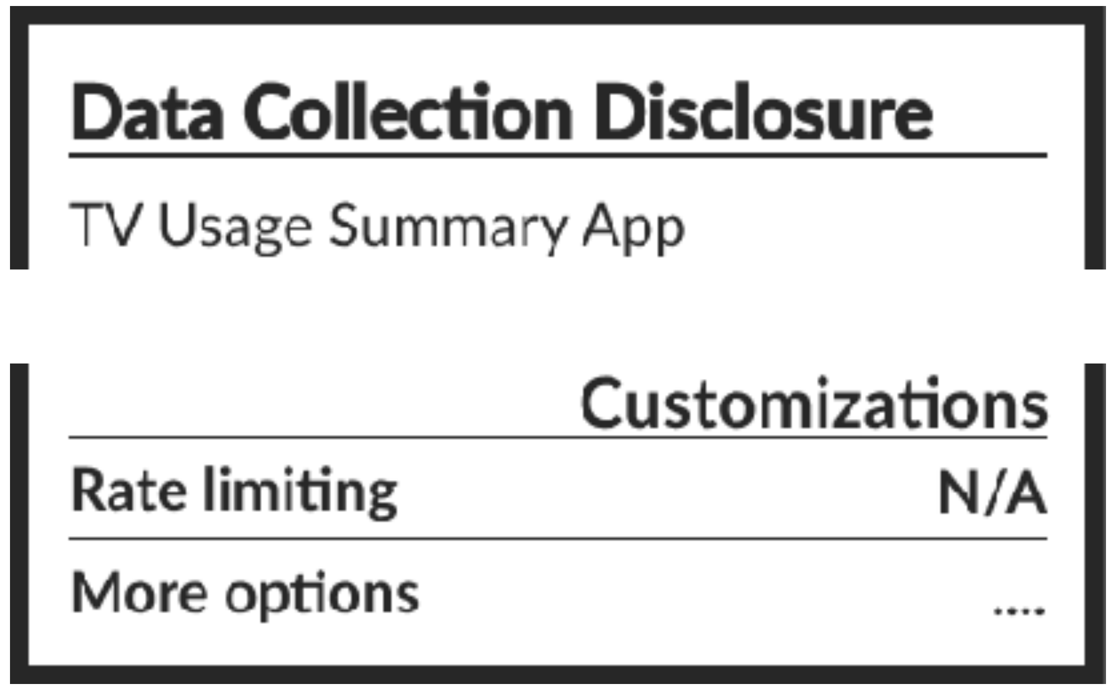
TV Usage Summary App

Running for	**20 days**
Total outgoing data packets **KBytes**	**80**
Sensor Type	Smart TV
Data type	TV Watch history
Granularity	
	Weekly aggregated durations by content category
Collection frequency	
	Every wednesday 1:00 AM
Destination	www.abc.com
Encryption	HTTPS
	Customizations
Rate limiting	N/A
More options	....

[1] Security and Privacy "Nutrition" Label, P. Emami-Naeini et al, IEEE S&P'20

Advantages

Built-in fine-grained control through manifest rewriting



Data Collection Disclosure

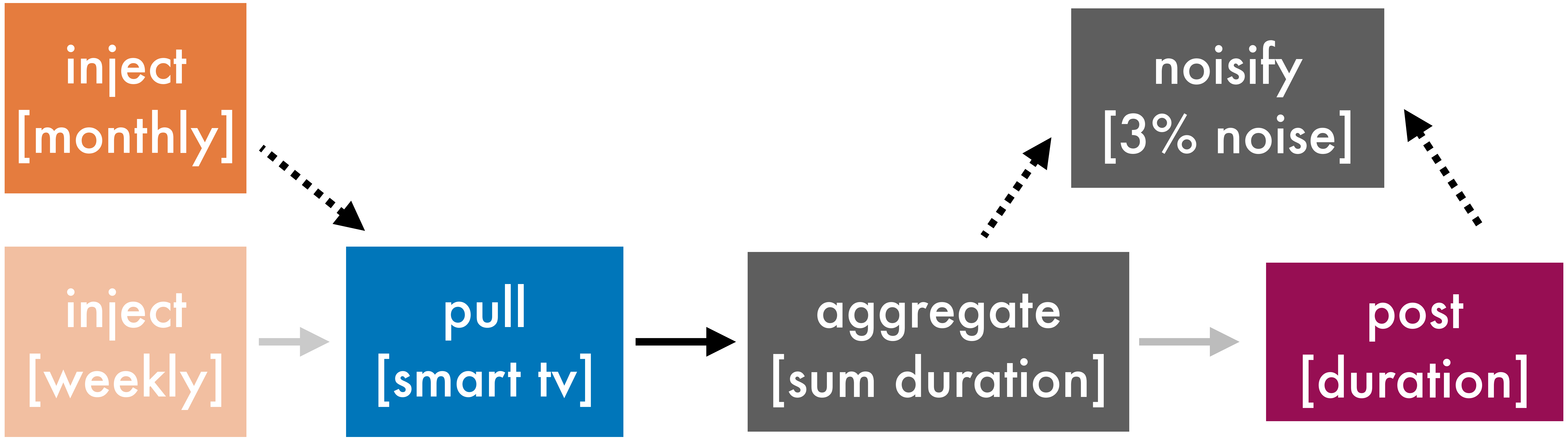
TV Usage Summary App

Customizations

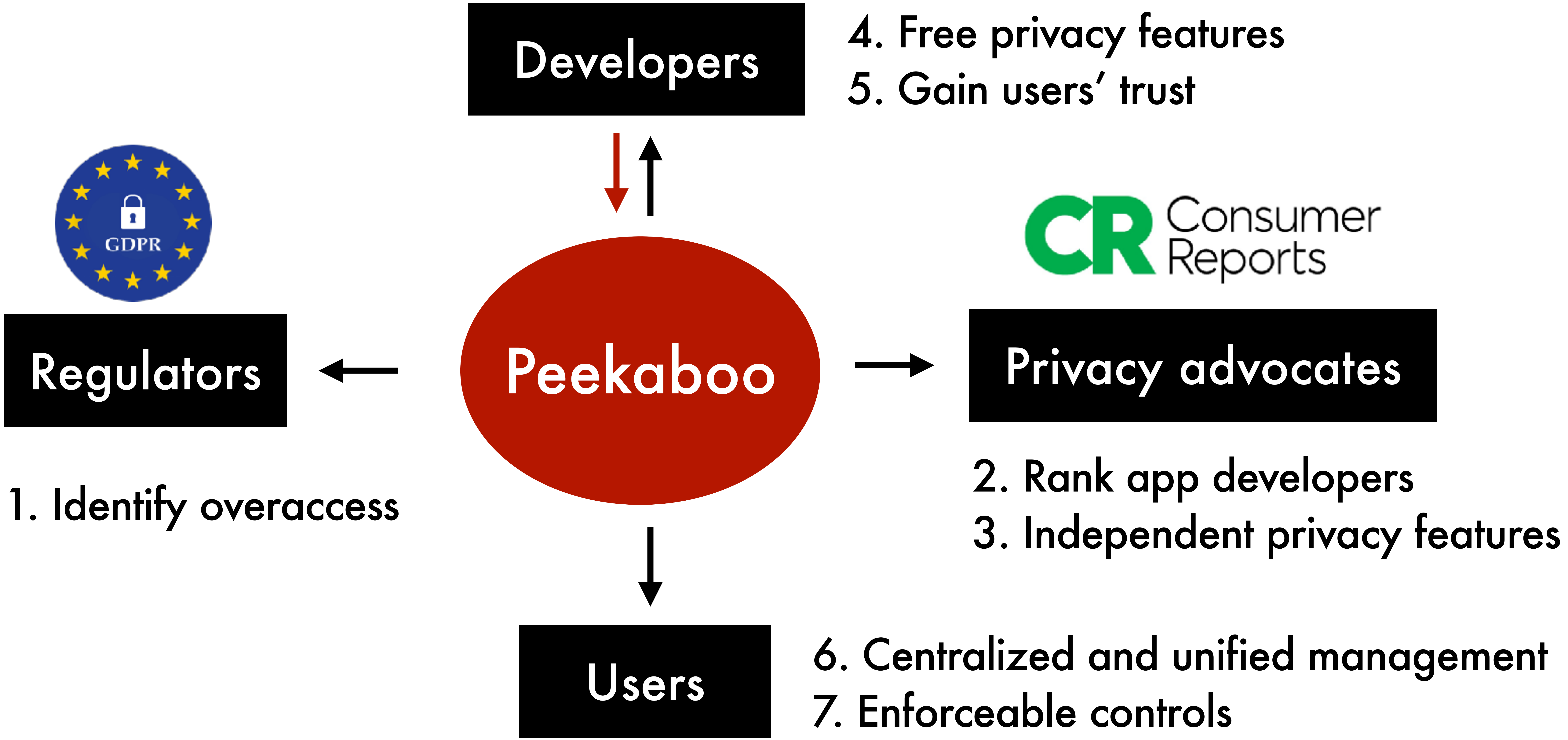
Rate limiting N/A

More options ....

Change the rate to monthly



Let the good privacy drive out the bad privacy



Developers

4. Free privacy features
5. Gain users' trust

Regulators

Peekaboo

Privacy advocates

1. Identify overaccess

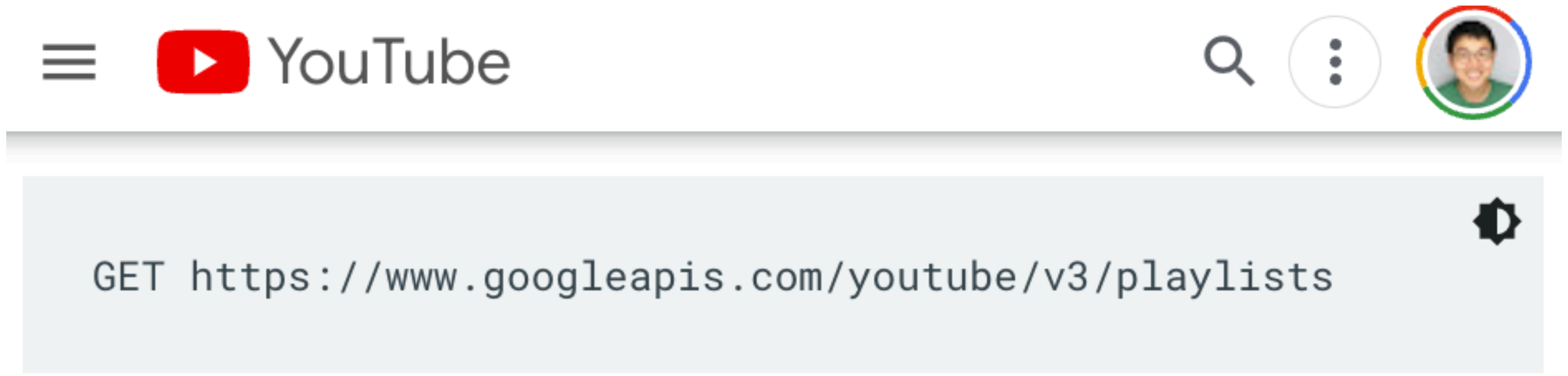
2. Rank app developers
3. Independent privacy features

Users

6. Centralized and unified management
7. Enforceable controls

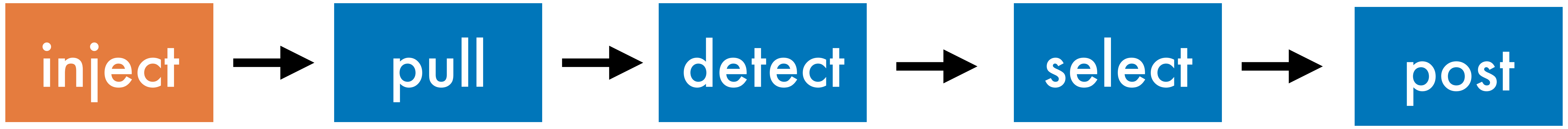
Design data access for third-party developers

URL-based APIs



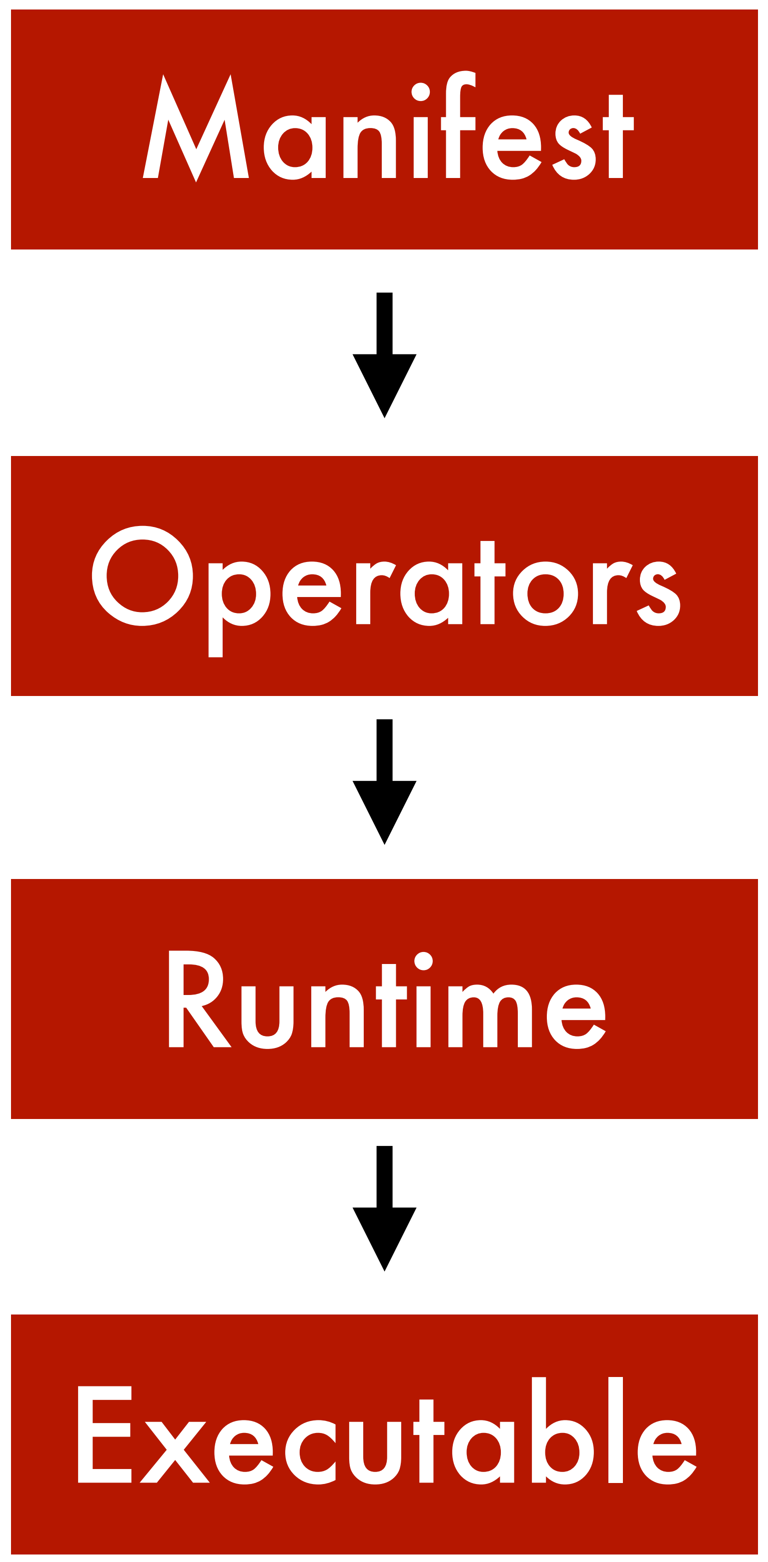
```
GET https://www.googleapis.com/youtube/v3/playlists
```

Operator-based APIs



inject → pull → detect → select → post

Peekaboo recap & implications

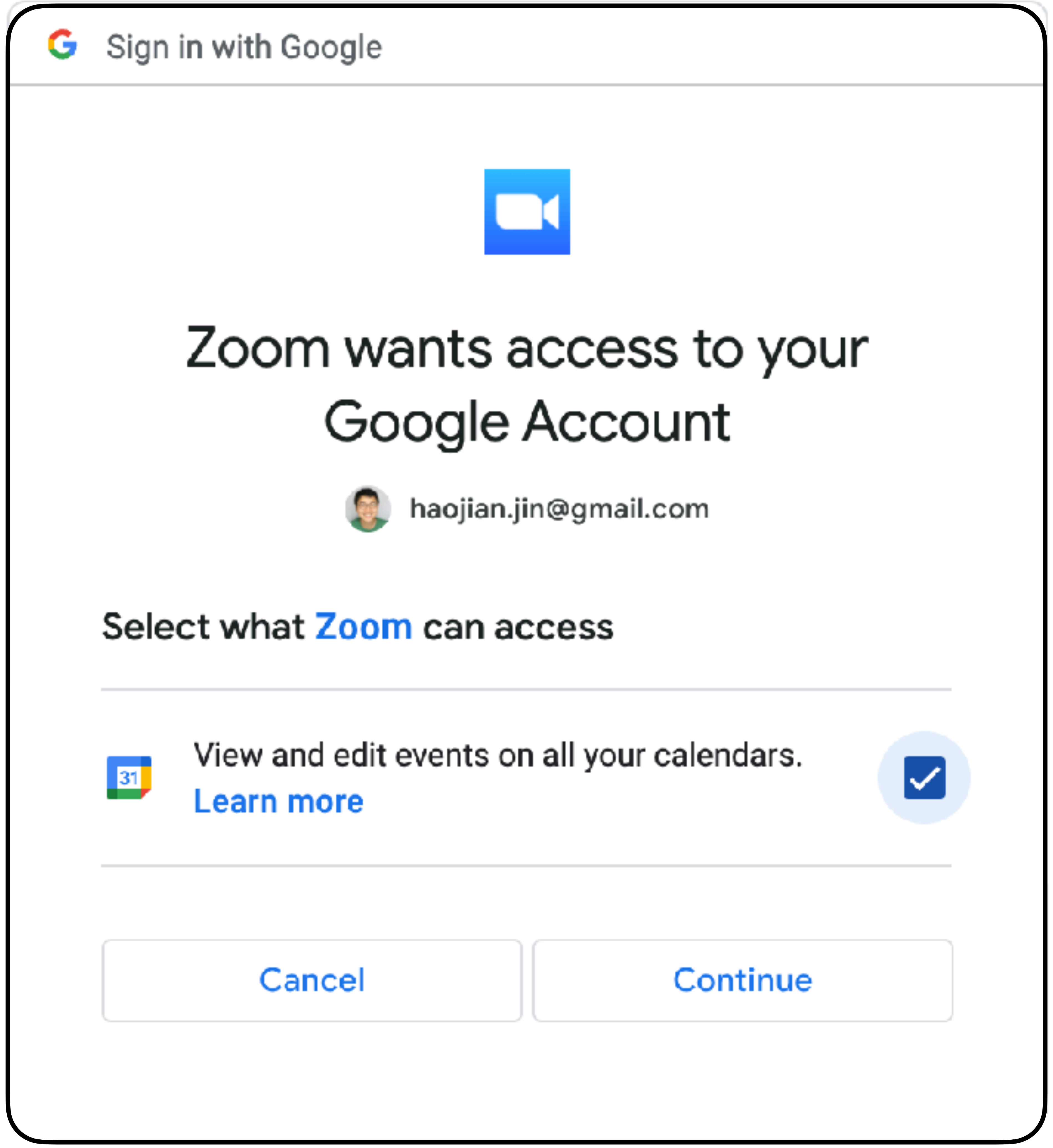


A fixed set of operators

A trusted runtime with a small set of pre-loaded implementations

Another example

Zoom accesses **all** your calendar events **continuously**!



Sign in with Google

Zoom wants access to your Google Account

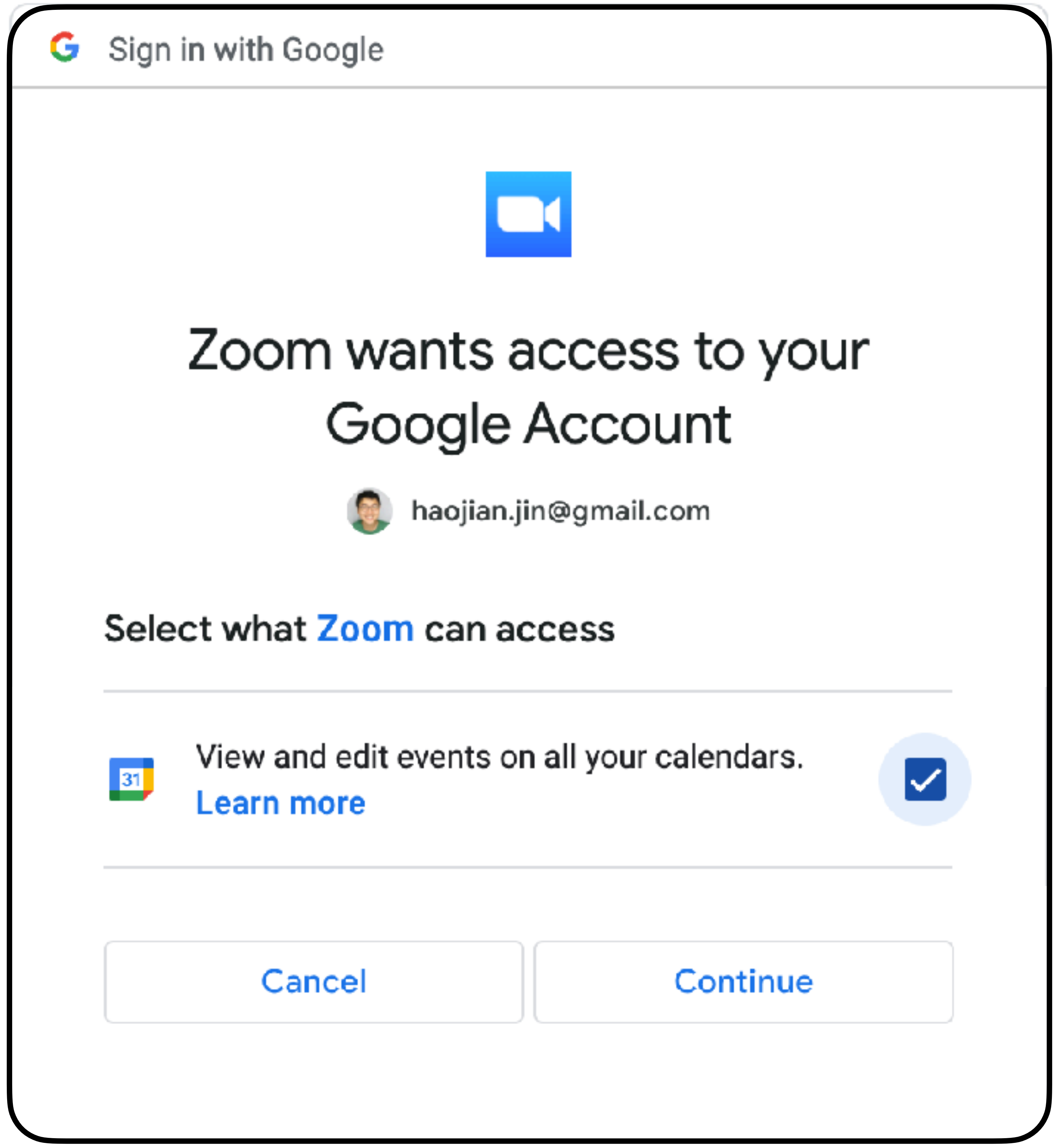
haojian.jin@gmail.com

Select what Zoom can access

View and edit events on all your calendars. Learn more ☑

Calendar events that contain https://zoom.us/xxxxx

Future third-party calendar API



Sign in with Google

Zoom wants access to your Google Account

haojian.jin@gmail.com

Select what Zoom can access

View and edit events on all your calendars. ☑
Learn more

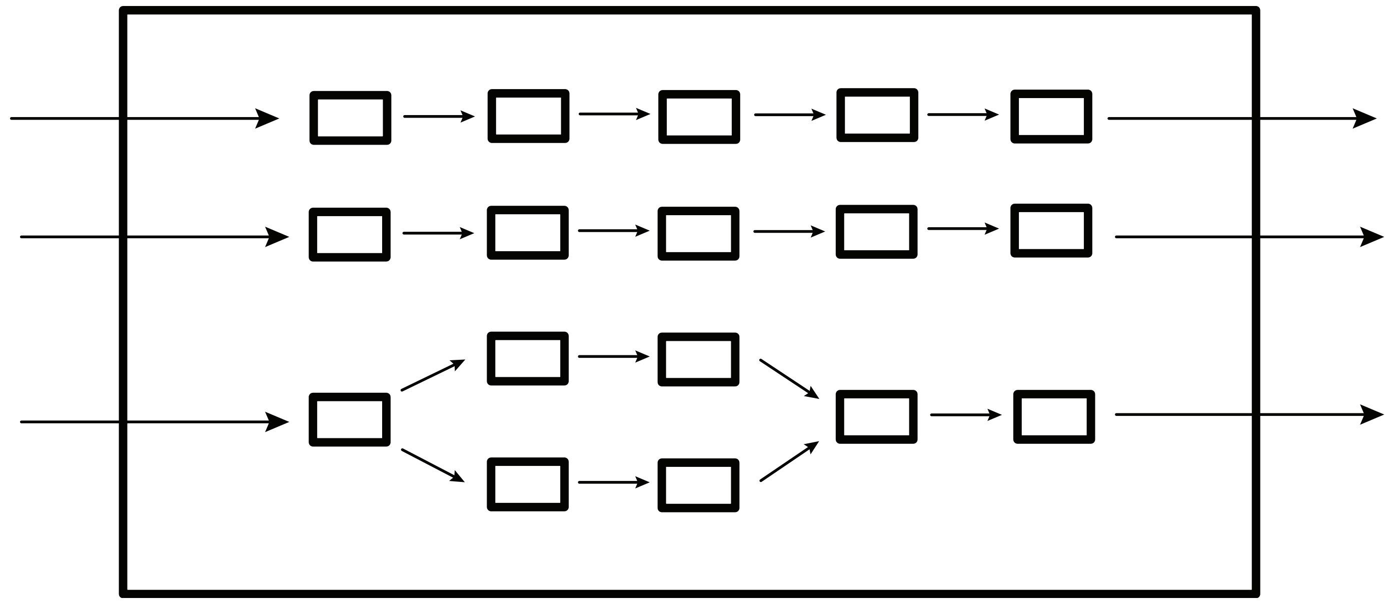
inject pull select check

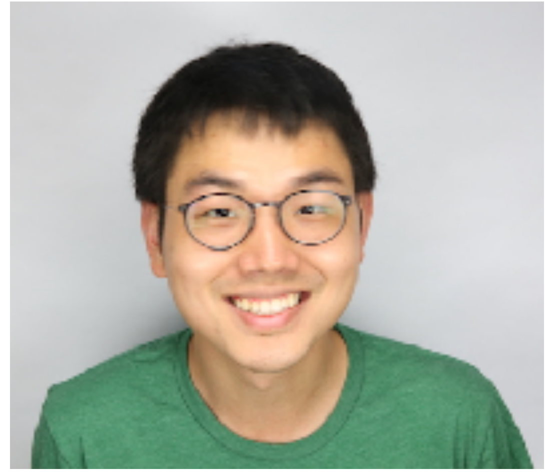
...

```
@purpose: The app can access calendar events
which contains a zoom link.
ZoomCalendarIntegration{
  // operator [properties]
  inject[...] -> pull Calendar[...] ->
  check Zoomlink[...] ->
  post [Zoom events]
}
```

Peekaboo

A Hub-Based Approach to Enable Transparency in Data Processing within Smart Homes





Haojian Jin

Gram Liu

David Hwang

Swarun Kumar

Yuvraj Agarwal

Jason Hong

Carnegie Mellon University

Principle of data minimization

> *"Personal data shall be limited to* ***what is necessary*** *in relation to the* ***purposes*** *for which they are processed."*

- GDPR, Article 5 (1) (c)