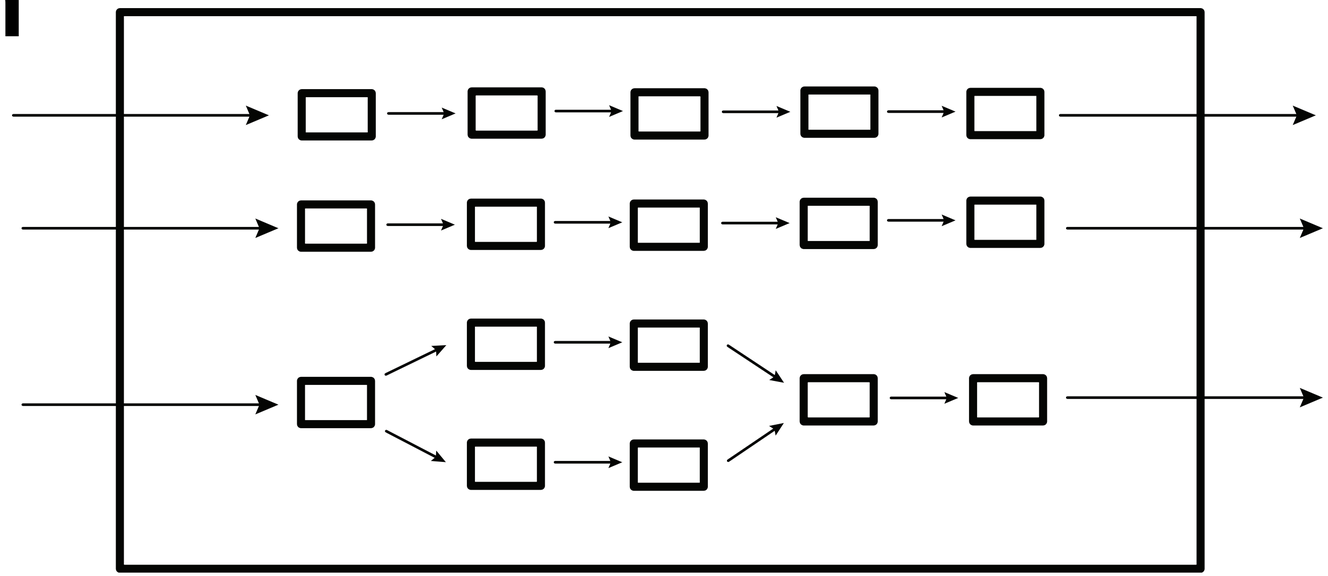
Modular Privacy Flows:

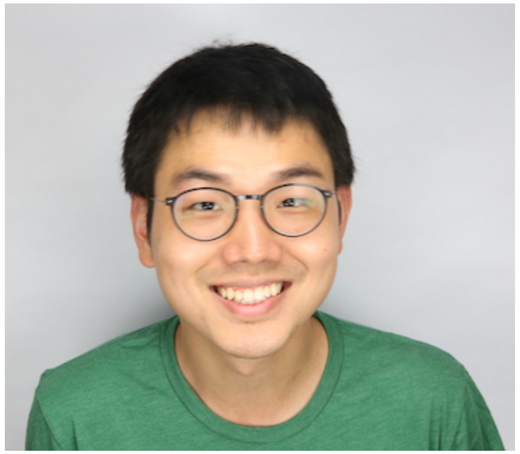
A Design Pattern for Data Minimization



Haojian Jin

Mar. 29, 2023

Bio



Haojian Jin (http://haojianj.in/)

Asst. Prof @ UCSD-HDSI

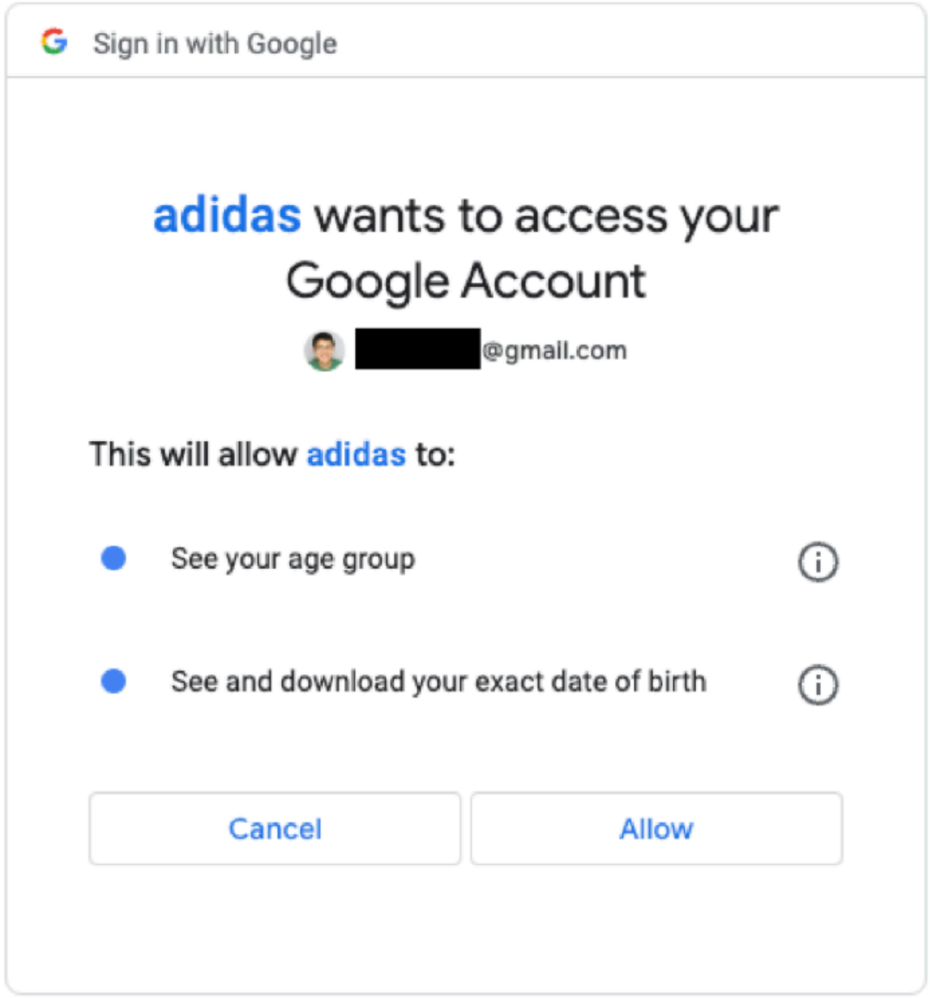
Data Smith Lab:

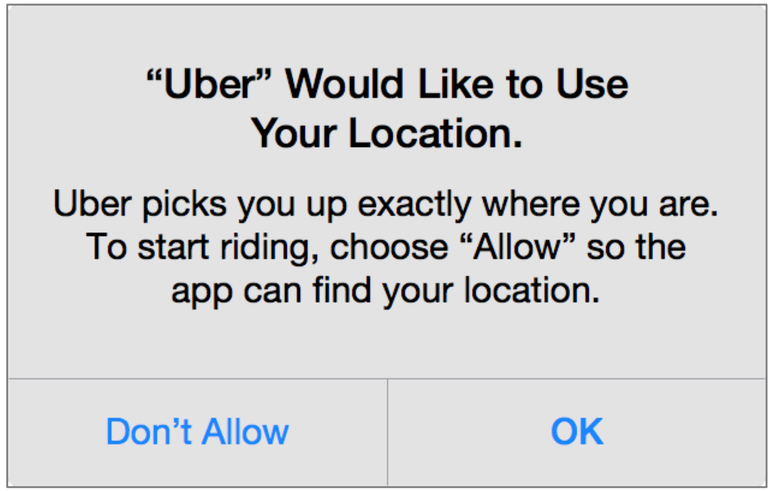
We study the security and privacy of data systems by researching the people who design, implement, and use these systems.

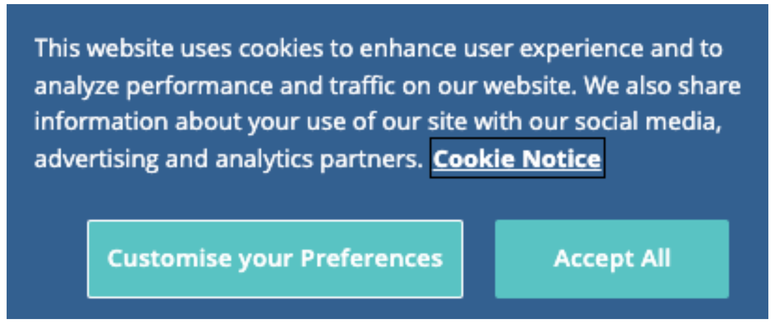
Ph.D. from CMU Human-Computer Interaction Institute

Before Ph.D.: worked at Yahoo Research, ran a startup

Permissions

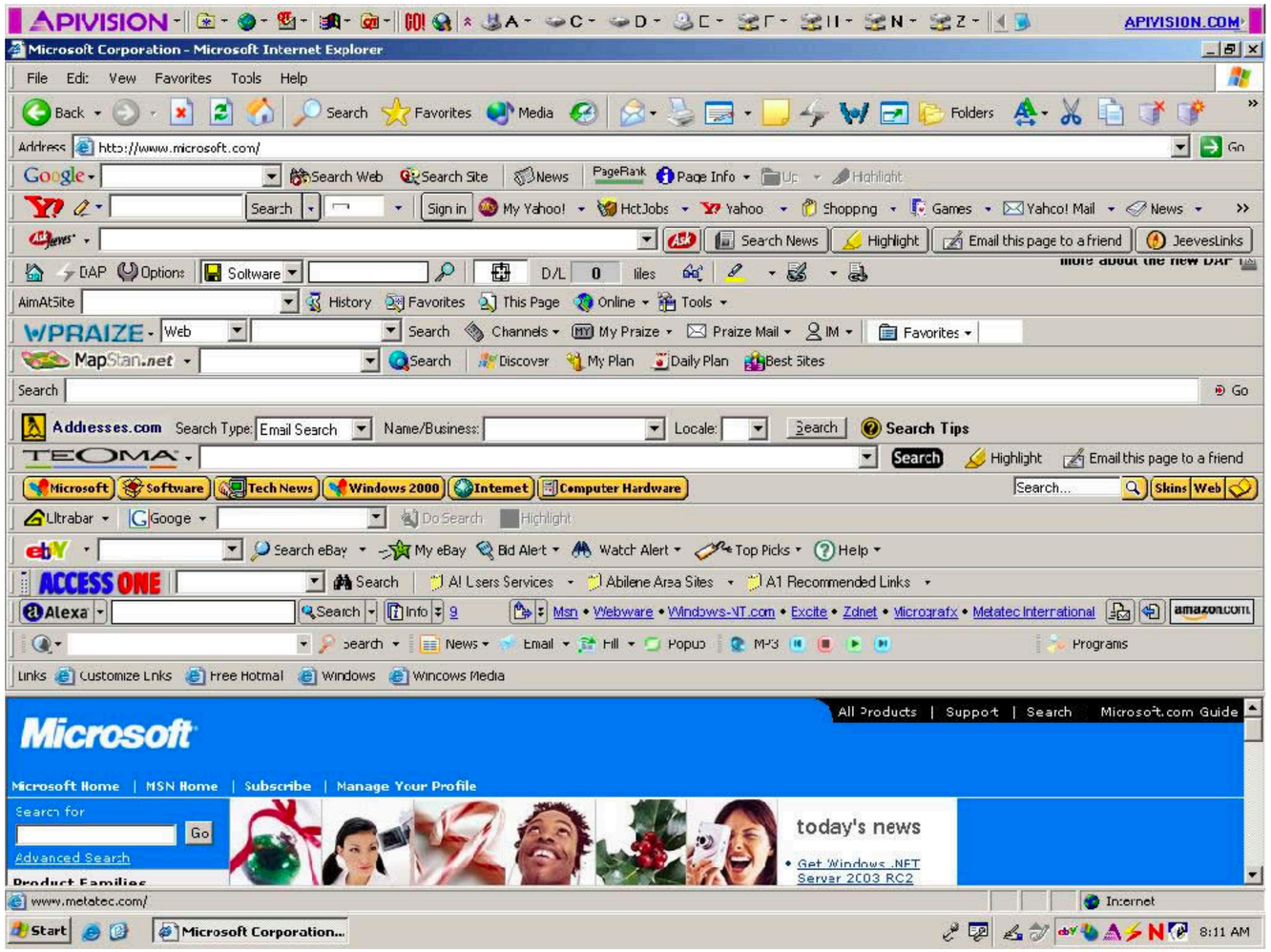






Notice and choice

Informed decisions


APIVISION
APIVISION.COM
Microsoft Corporation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media Folders
Address http://www.microsoft.com/
Go
Google
Search Web
Search Site
News
Page Info
Highlight
Search
Sign in
My Yahoo!
HotJobs
Yahoo
Shopping
Games
Yahoo! Mail
News
Search News
Highlight
Email this page to a friend
JeevesLinks
DAP
Options
Software
D/L
files
AimAtSite
History
Favorites
This Page
Online
Tools
WPRAIZE
Web
Search
Channels
My Praize
Praize Mail
IM
Favorites
MapStan.net
Search
Discover
My Plan
Daily Plan
Best Sites
Search
Go
Addresses.com
Search Type:
Email Search
Name/Business:
Locale:
Search
Search Tips
TEOMA
Search
Highlight
Email this page to a friend
Microsoft
Software
Tech News
Windows 2000
Internet
Computer Hardware
Search...
Skins
Web
Ultrabar
Google
Do Search
Highlight
eBay
Search eBay
My eBay
Bid Alert
Watch Alert
Top Picks
Help
ACCESS ONE
Search
All Users Services
Abilene Area Sites
All Recommended Links
Alexa
Search
Info
Msn
Webware
Windows-NT.com
Excite
Zdnet
Micrografx
Metatec International
amazon.com
Search
News
Email
Hill
Popup
MP3
Programs
Links
Customize Links
Free Hotmail
Windows
Windows Media
Microsoft
All Products
Support
Search
Microsoft.com Guide
Microsoft Home
MSN Home
Subscribe
Manage Your Profile
Search for
Go
Advanced Search
today's news
Get Windows .NET Server 2003 RC2
www.metatec.com/
Internet
Start
Microsoft Corporation...
8:11 AM


KASPERSKY
Kaspersky
Anti-Virus
The Smarter Way to Protect
everything on your PC
SAVING THE WORLD FOR 20 YEARS

App Stores

"changed how software development worked, and expanded the ***number of people who could comfortably, safely use a computer*** *from a few hundred million to a few billion."*

[1] Benedict Evans, App stores, trust and anti-trust

Technical idea #1

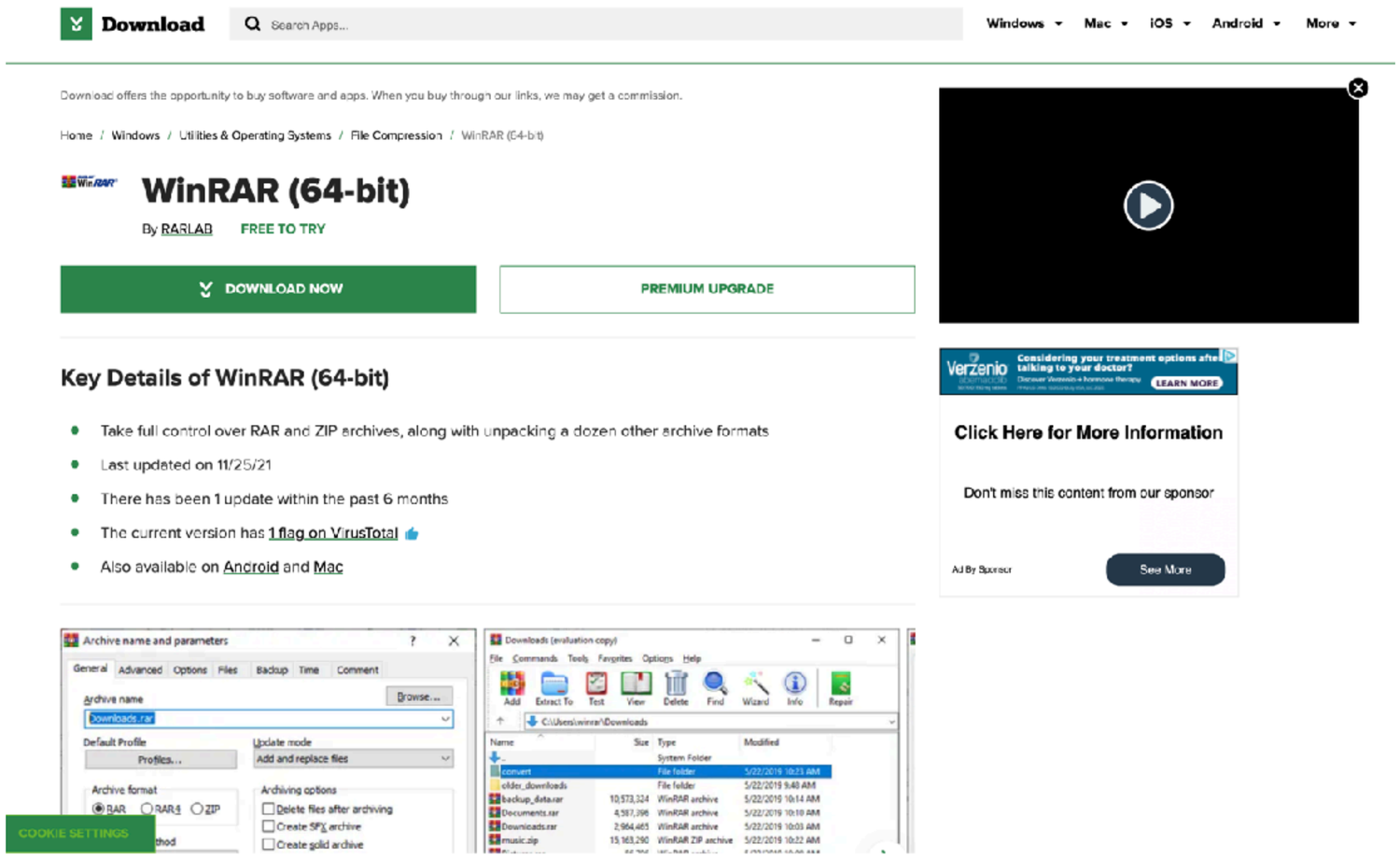
Putting apps in a sandbox

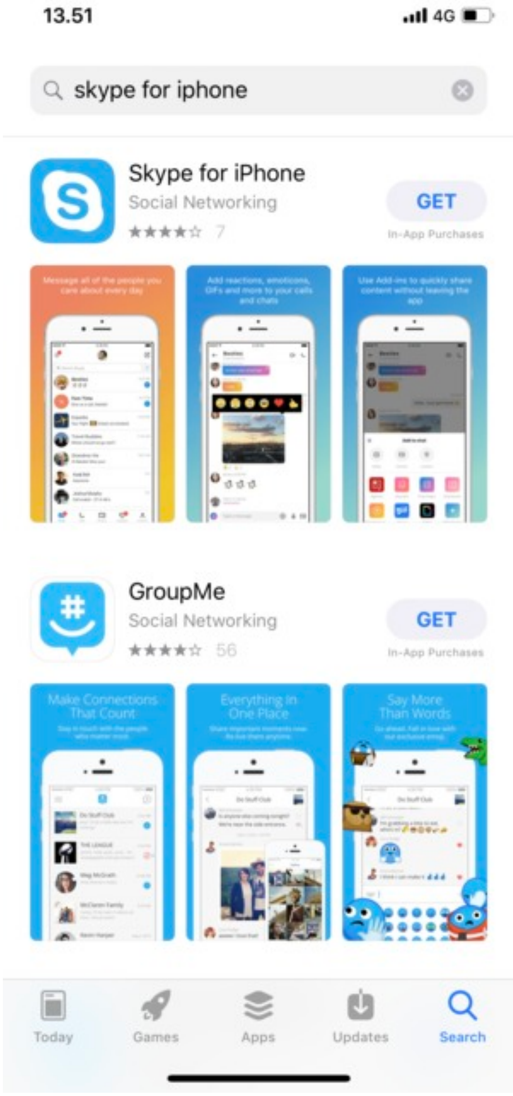
Apps can only do things that Apple allows and cannot ask (or persuade, or trick) the user for permission to do ‘dangerous’ things.

- Would this break my phone?
- Would this run my battery down?
- Steal my bank details?

Technical idea #2

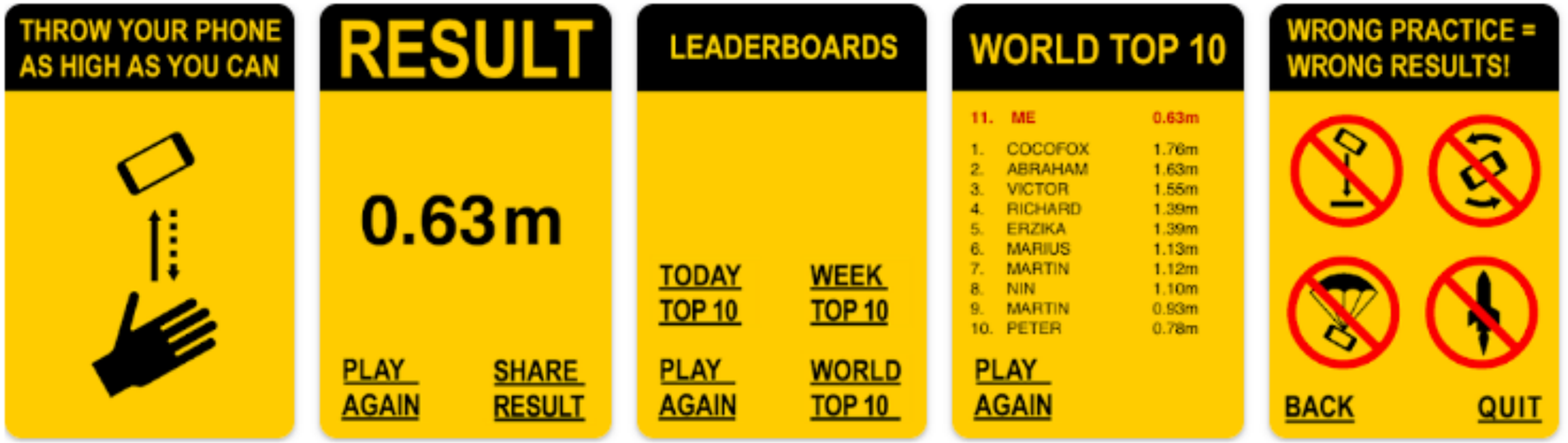
Distributing software through a centralized app store

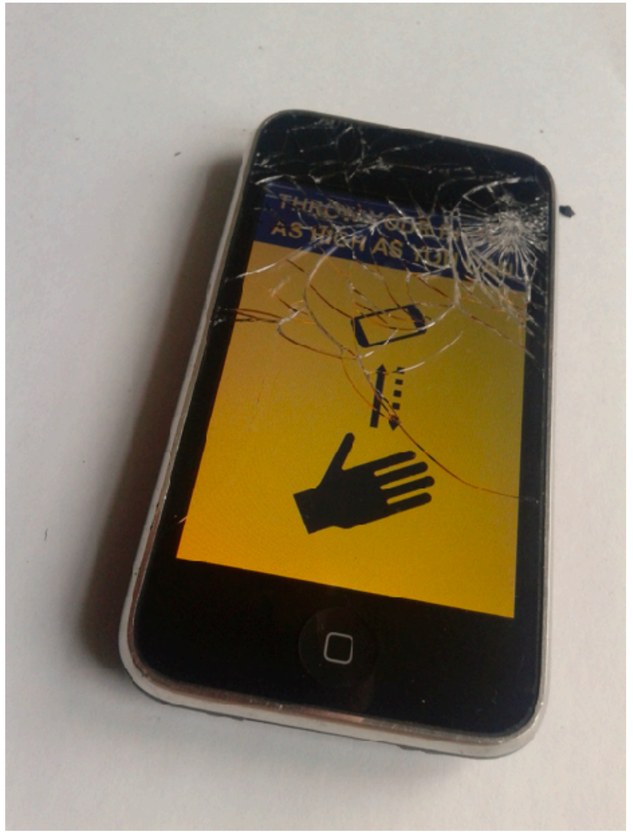




Technical idea #2

Distributing software through a centralized app store





S.M.T.H.: Send Me to Heaven

Issues around idea #1

Putting apps in a sandbox

Apps can only do things that Apple allows and cannot **ask** (or **persuade**, or **trick**) the user for **permission** to do '**dangerous**' things.

- What are 'dangerous' things?
- How can we trust Apple?
- How can we detect if apps trick the user?
- ……

Issues around idea #1

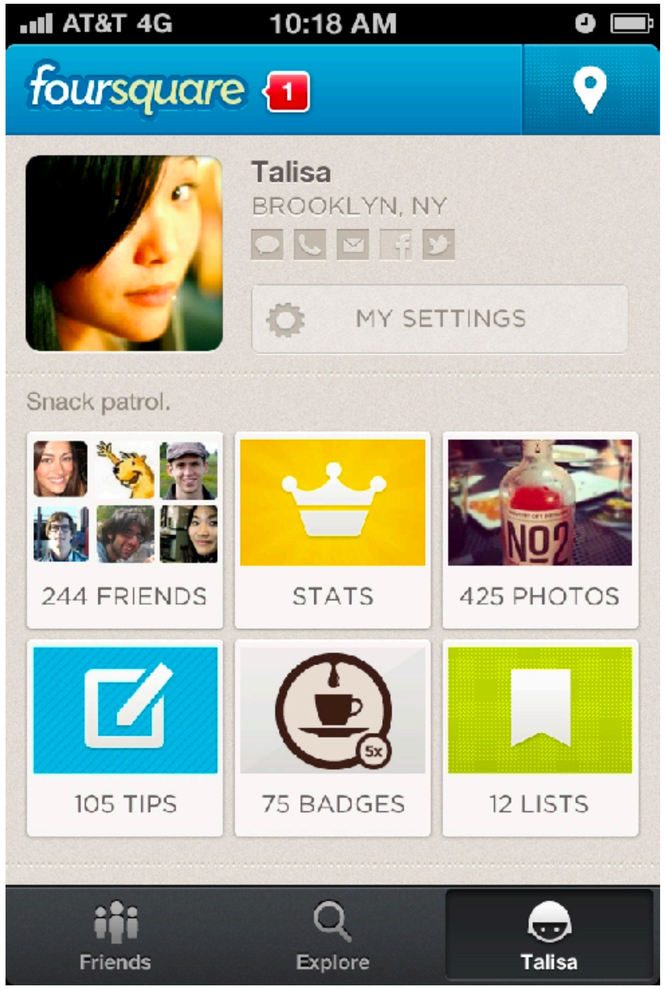
After 15 years - still iterating

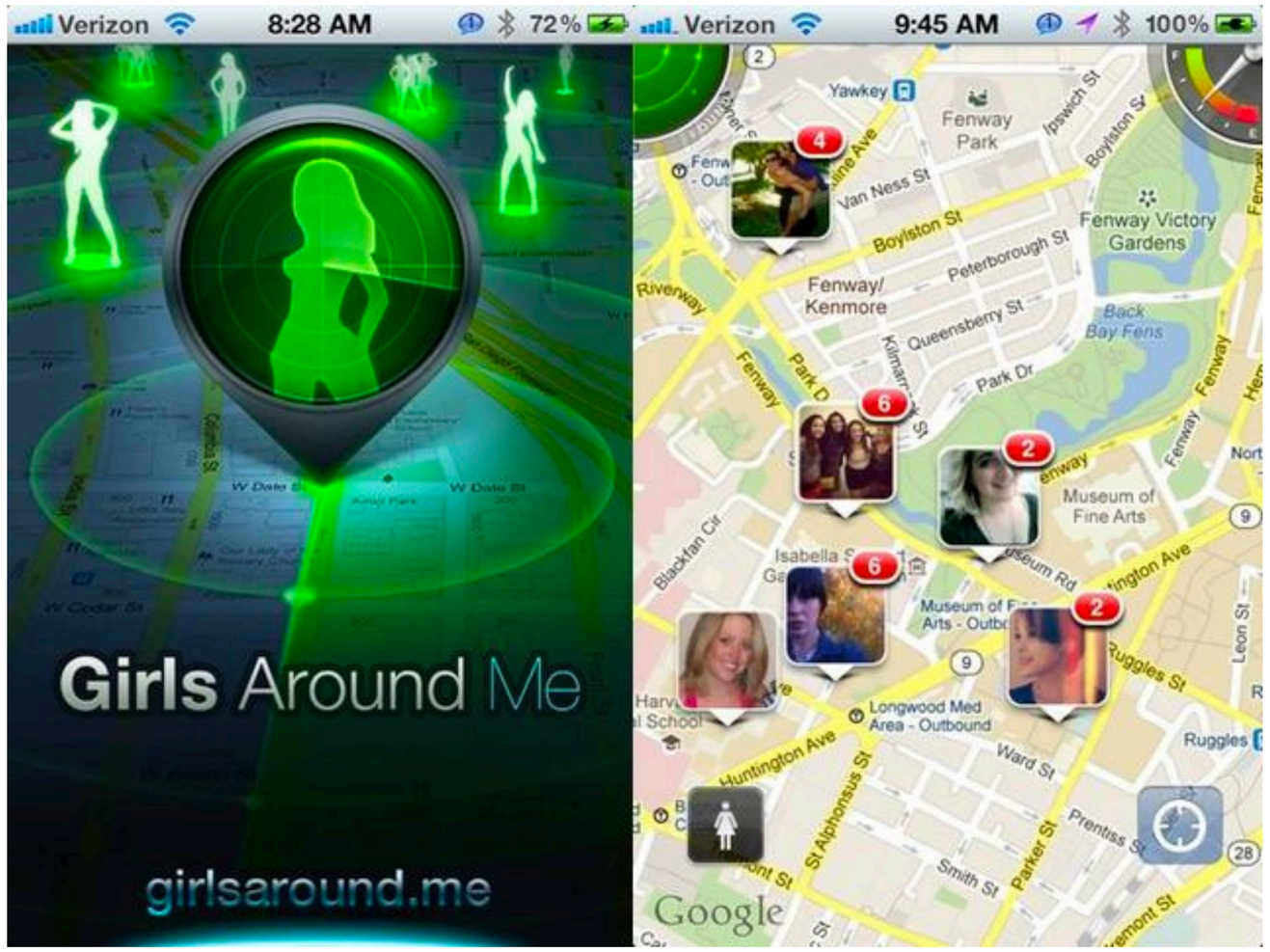
Issues around idea #2

Distributing software through a centralized app store

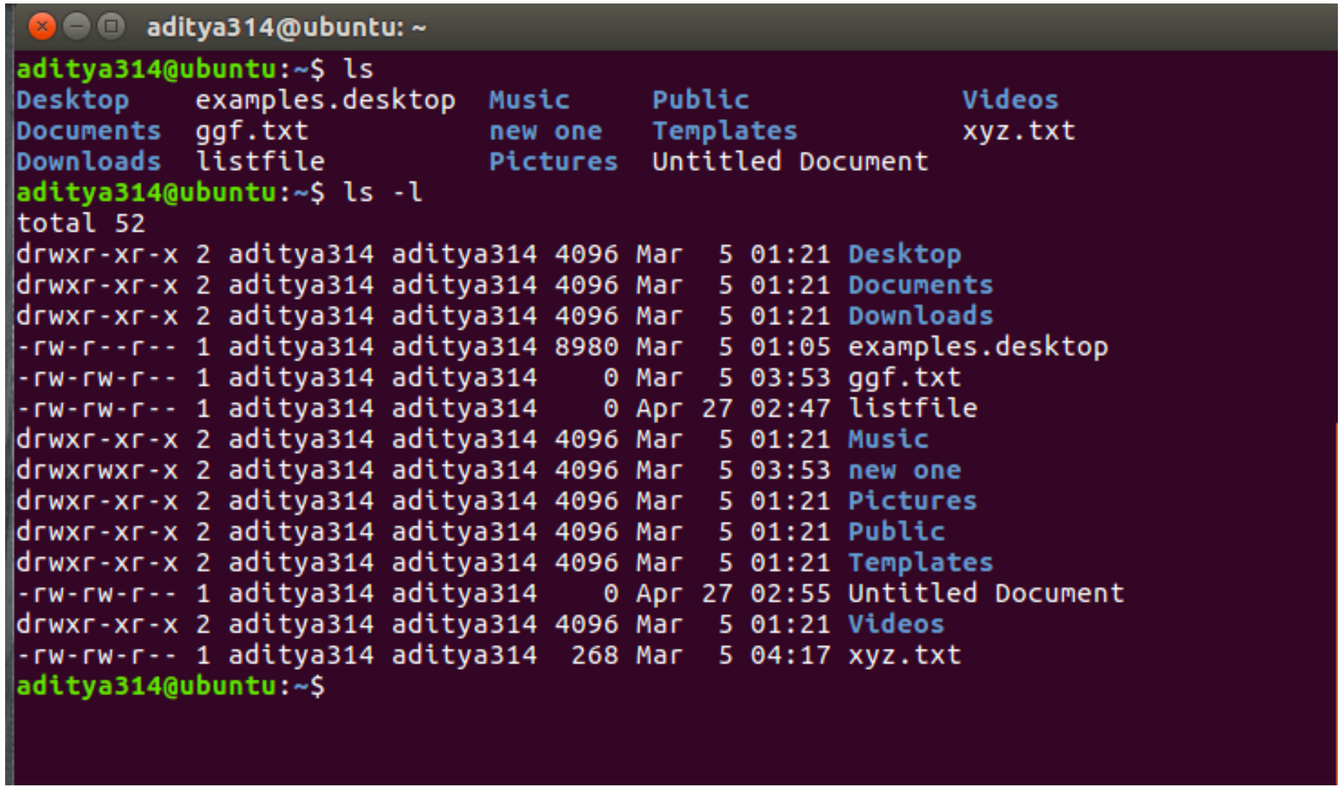
Issues around idea #2

Software often has a cloud component.



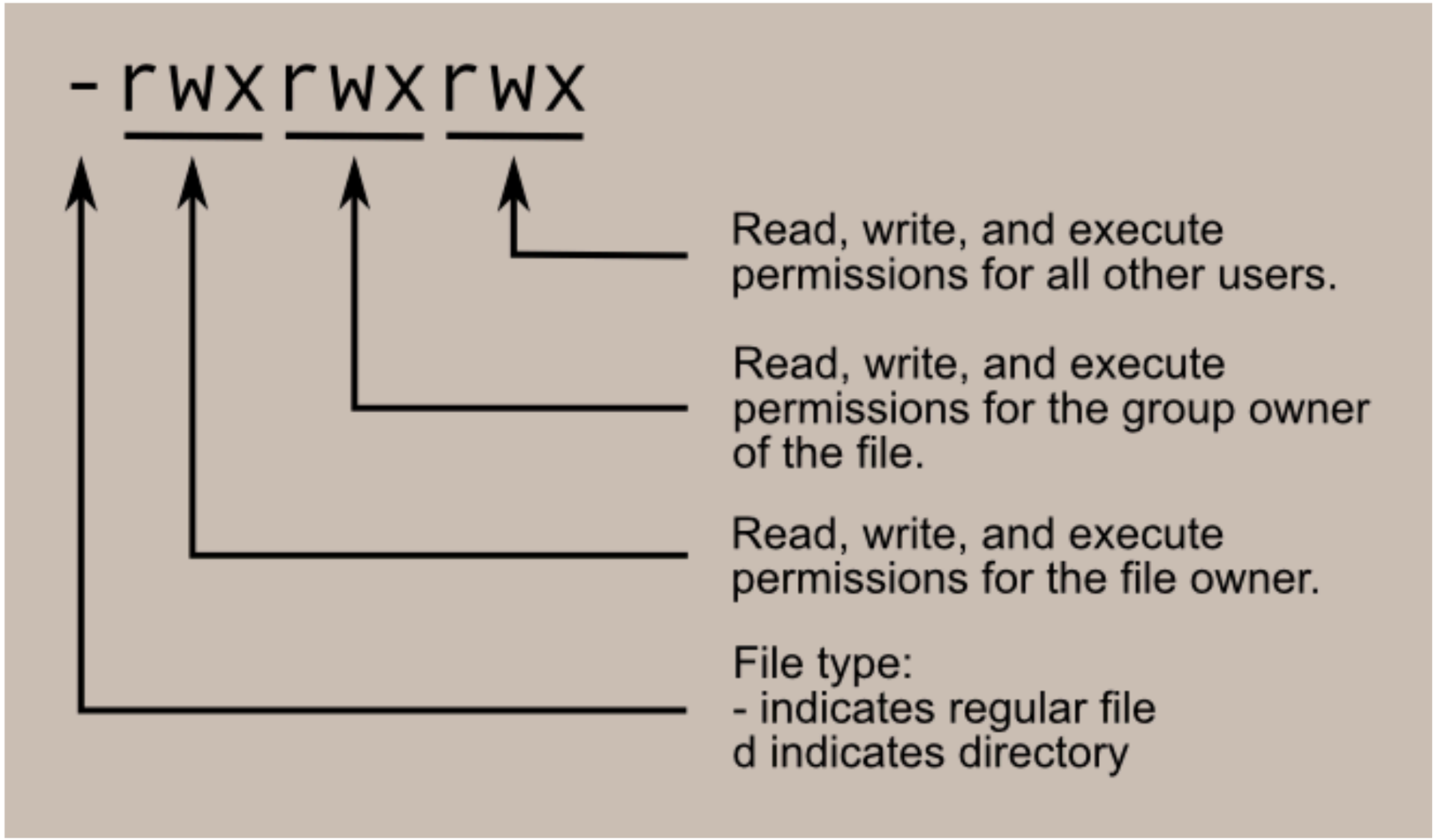


Linux systems



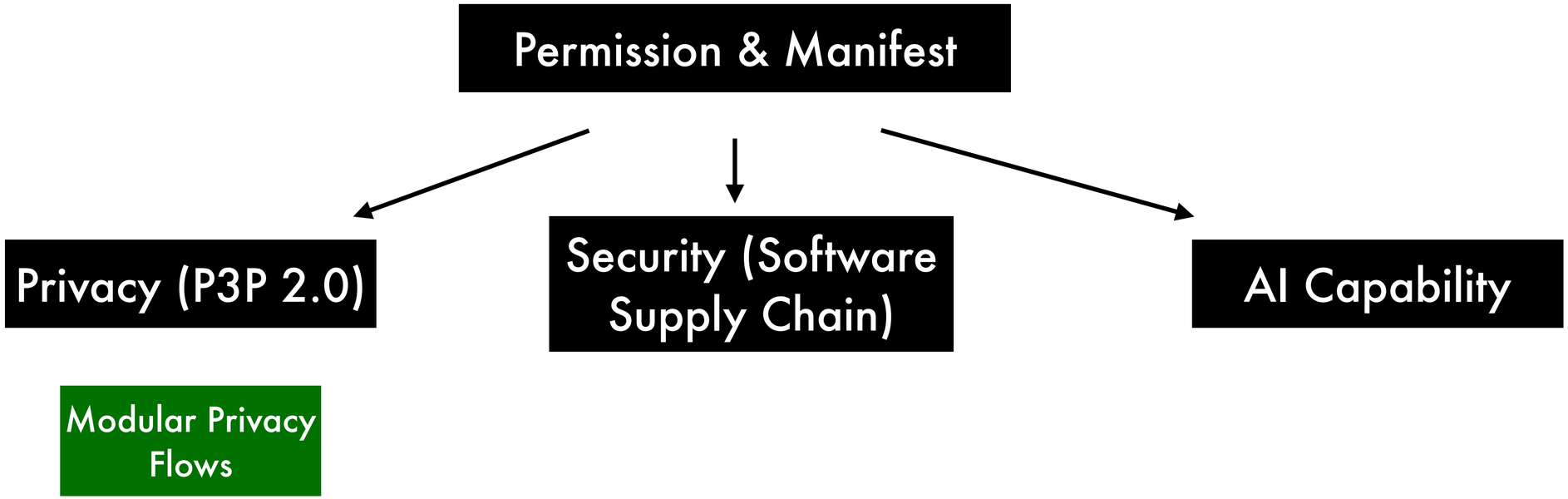
- Users
- File permissions

File permissions

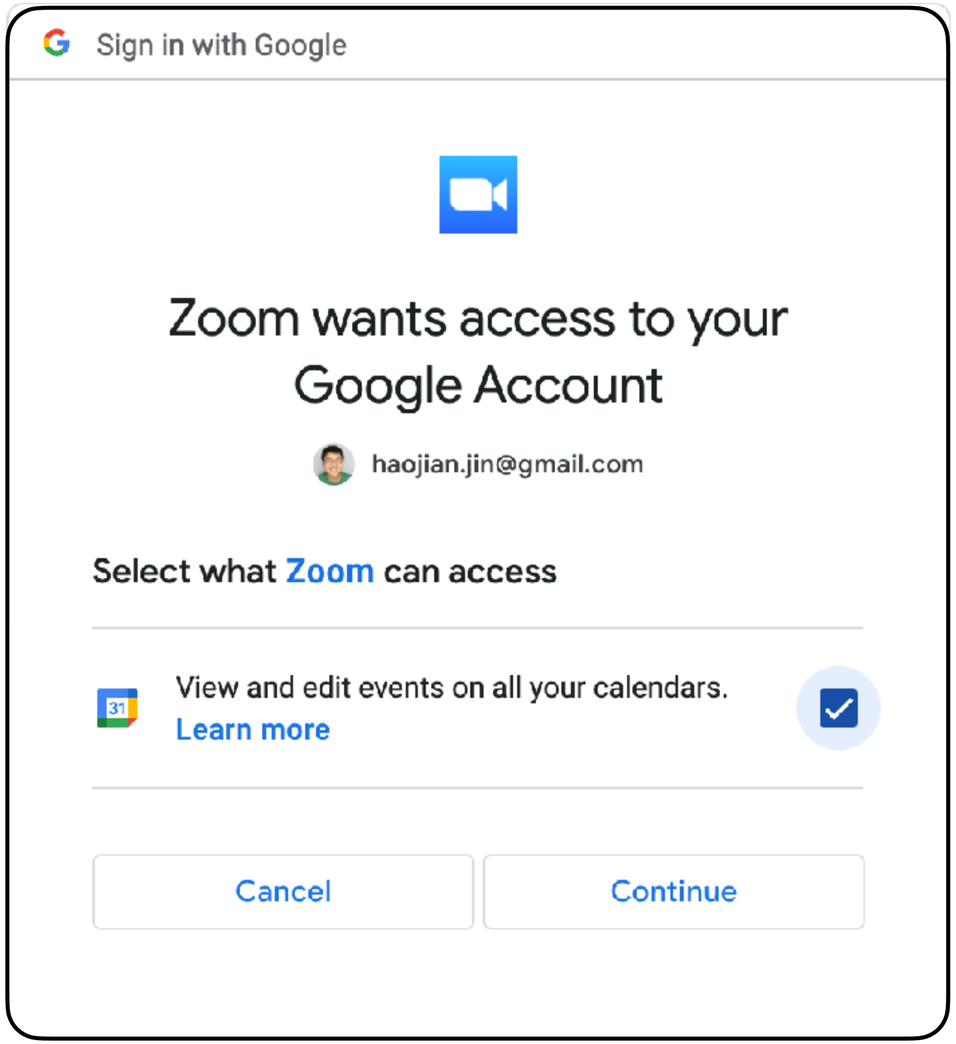


Current research focus

Next generation permission system

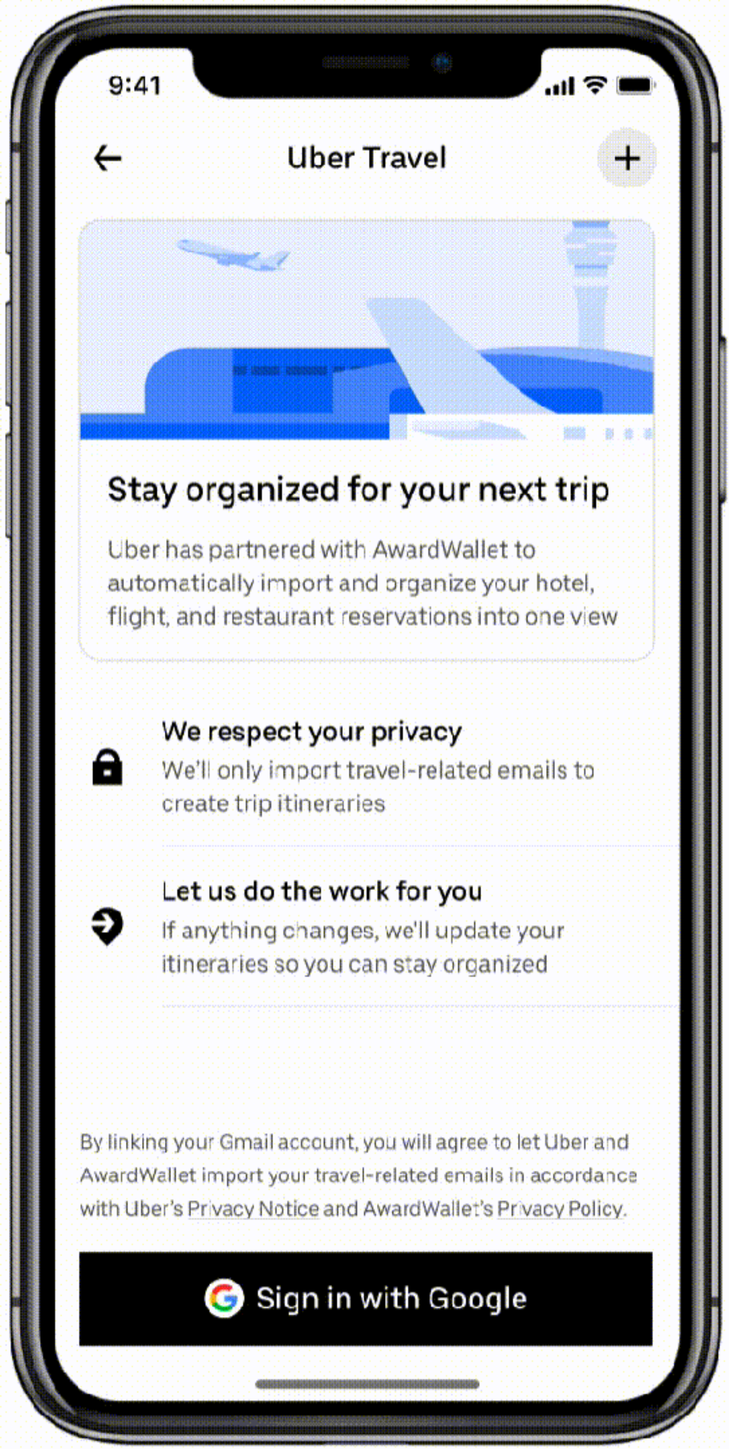


Zoom accesses **all** your calendar events **continuously**!



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

Uber wants to see all your emails.



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)

Principle of least privilege

"A security architecture should be designed so that each entity is granted the ***minimum system resources*** *and authorizations that the entity needs to perform its function.."*

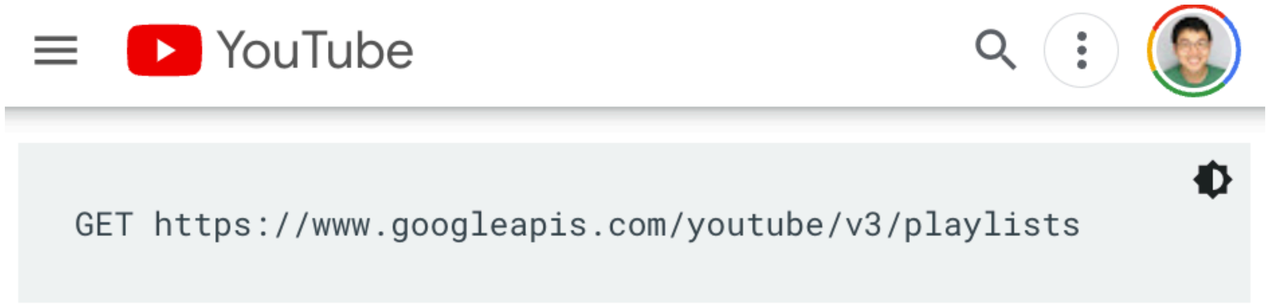
Google APIs - All-or-nothing binary permissions

Scope	Meaning
`https://www.googleapis.com/auth/calendar`	read/write access to Calendars
`https://www.googleapis.com/auth/calendar.readonly`	read-only access to Calendars
`https://www.googleapis.com/auth/calendar.events`	read/write access to Events
`https://www.googleapis.com/auth/calendar.events.readonly`	read-only access to Events
`https://www.googleapis.com/auth/calendar.settings.readonly`	read-only access to Settings
`https://www.googleapis.com/auth/calendar.addons.execute`	run as a Calendar add-on

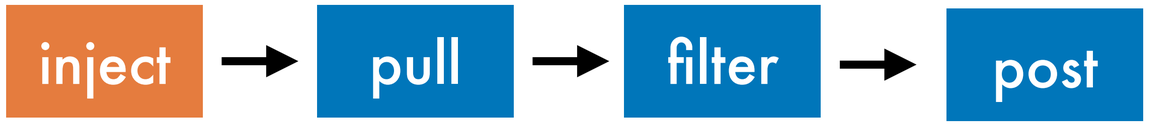
https://developers.google.com/calendar/api/guides/auth

Program data transformation functions using chainable *operators*

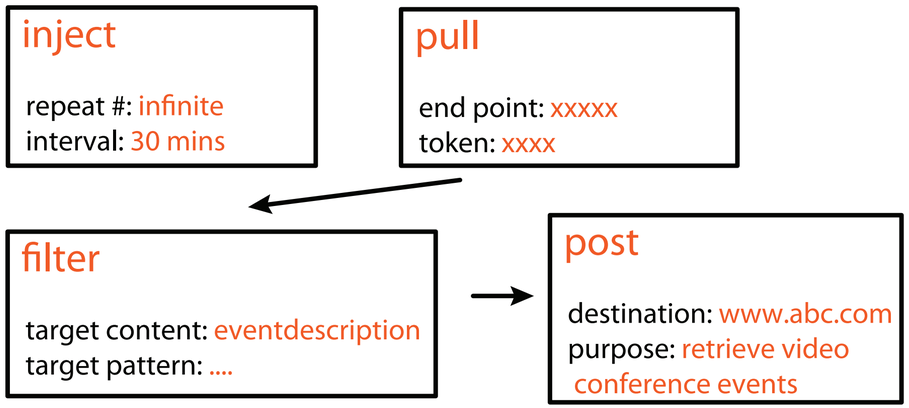
URL-based APIs



Operator-based APIs

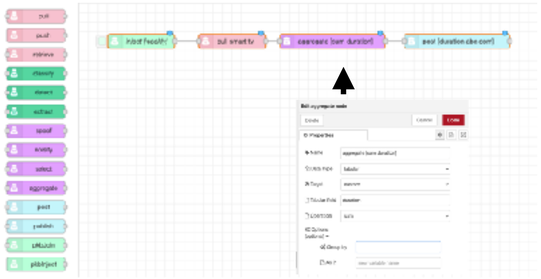


A text-based whitelist *manifest* (i.e., program representation)




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

System builders



Offer a set of operators as the API

Execute the manifest using preloaded implementations

Developers

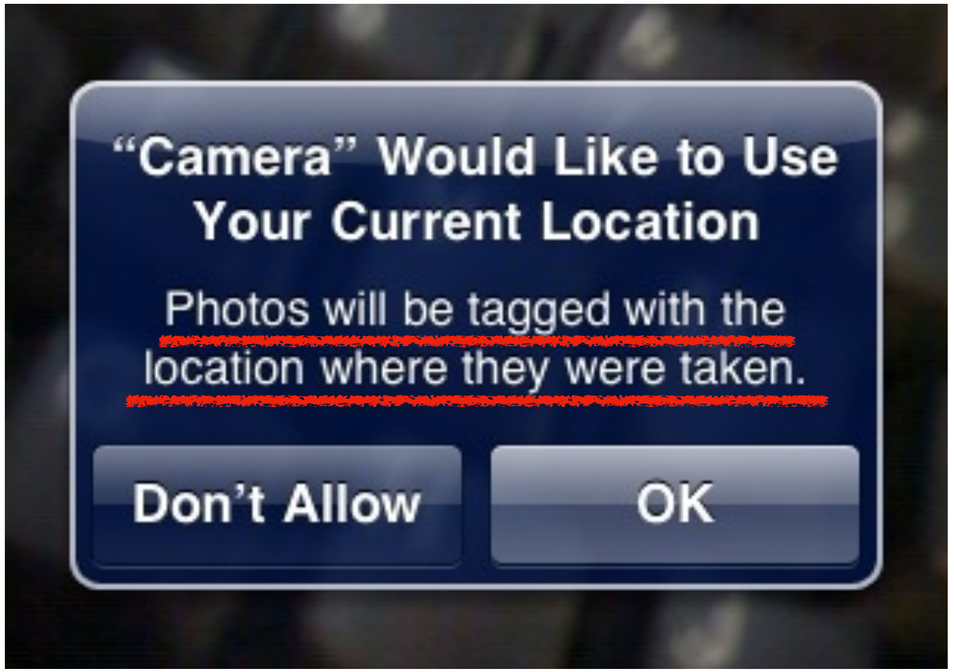
```
@purpose: The app can access calendar events
which contains a zoom link.
ZoomCalendarIntegration{
  // operator [properties]
  inject[...] -> pull Calendar[...] ->
  filter [Zoom join link pattern] ->
  post [Zoom events]
}
```

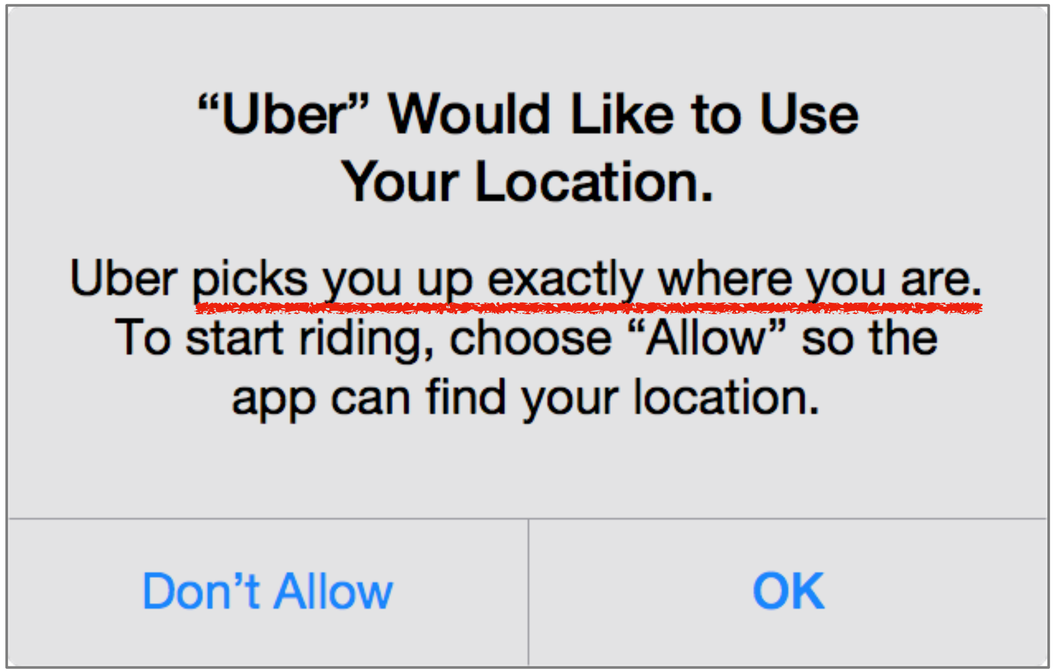
Author a manifest by connecting operators

Talk outline

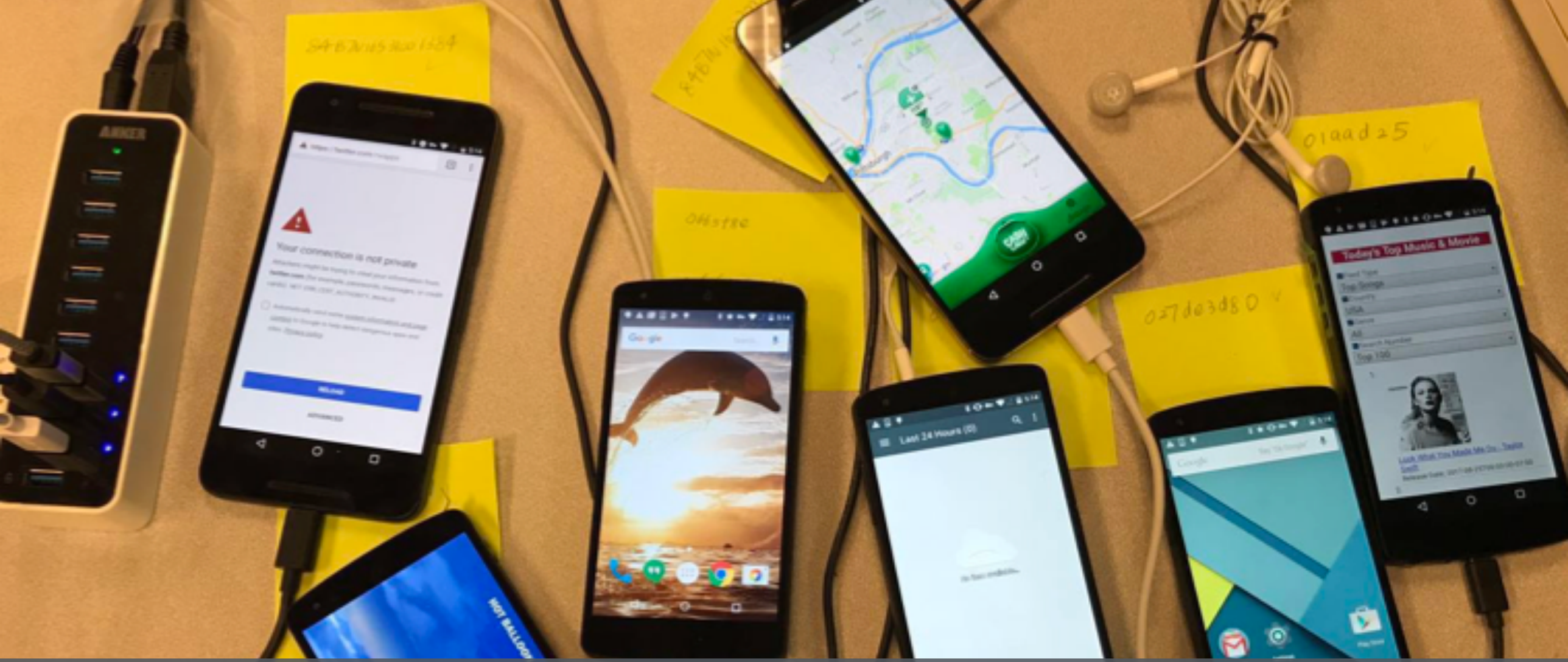
1. Modular Privacy Flows (MPF) in a Nutshell
2. Why MPF
3. How MPF
4. When and when not MPF
5. Future Work

Purpose strings.





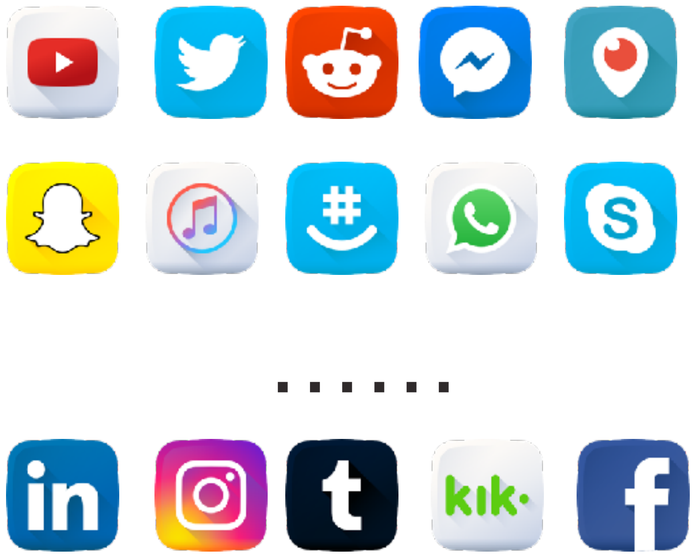
arbitrary text,
manually annotated,
hard to validate/assess.



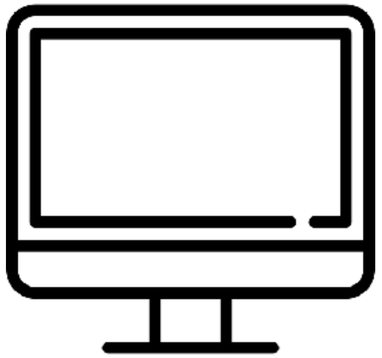
MobiPurpose is a scalable in-lab solution that can index fine-grained privacy attributes (who, where, what, why) of outgoing network requests.

Jin et al. "Why Are They Collecting My Data?": Inferring the Purposes of Network Traffic in Mobile Apps, IMWUT' 18

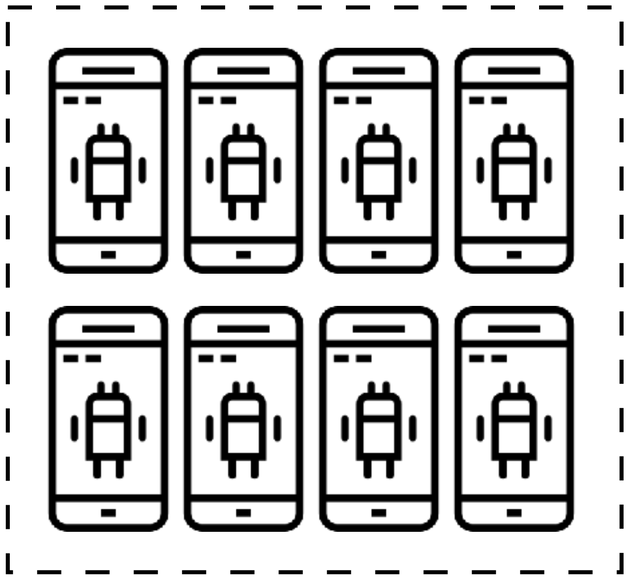
MobiPurpose - network tracing



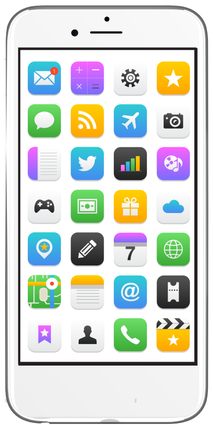
185k apps



Intercepted 2 million
unique traffic requests



MobiPurpose - network tracing



Traffic request snapshot

source app:
com.inkcreature.predatorfree → Who?

connect to host:
inkcreature.com → Where?

server path:
/_predatorServer/

key-value pairs in request body:
myLat: 40.4435877
myLon: -79.9452883
....
→ Key-value pairs

Traffic request snapshot

source app:
com.inkcreature.predatorfree
connect to host:
inkcreature.com
server path:
/_predatorServer/

key-value pairs in request body:
myLat: 40.4435877
myLon: -79.9452883
....

2,008,912 unique traffic requests

from 14,910 apps

contacting

12,046 unique domains

302,893 unique URLs

We publish the dataset at:
http://bit.ly/purposedata

Traffic Data stats

DATA TYPES

DATA PURPOSES

EXAMPLES

location

nearby search

DATA TYPES

DATA PURPOSES

EXAMPLES

location

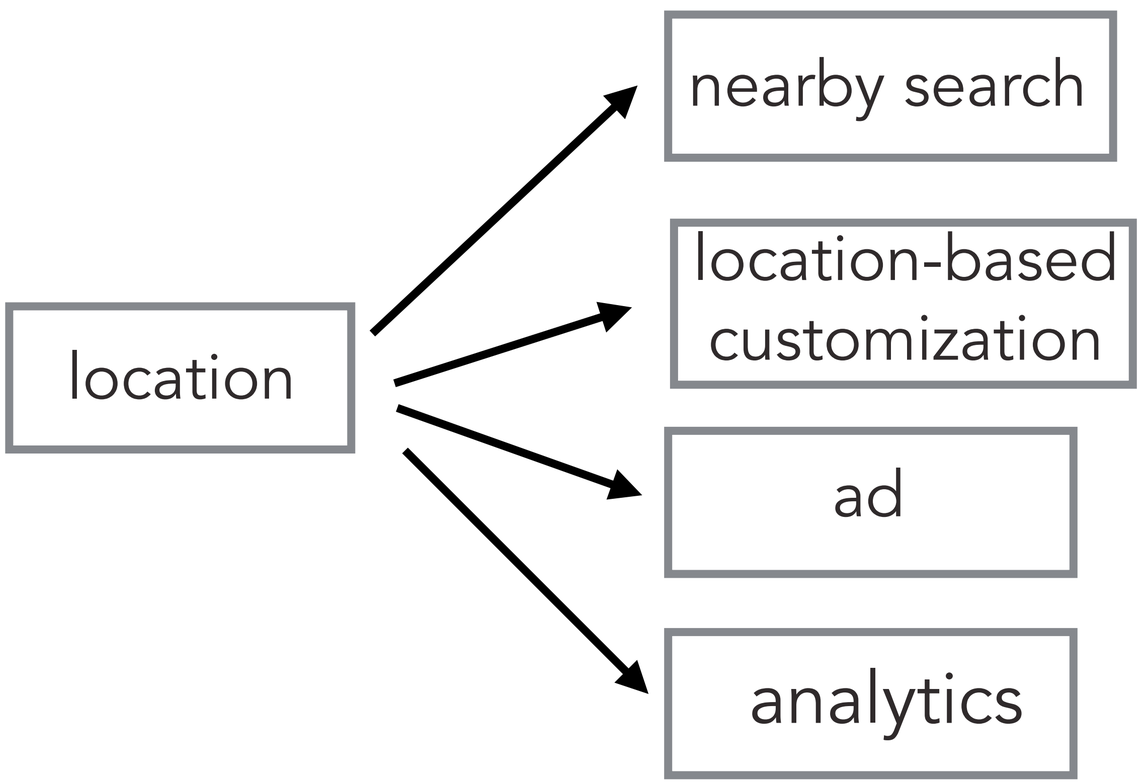
nearby search

location-based customization

DATA TYPES

DATA PURPOSES

EXAMPLES



.....

yelp

YAHOO! WEATHER

FLURRY

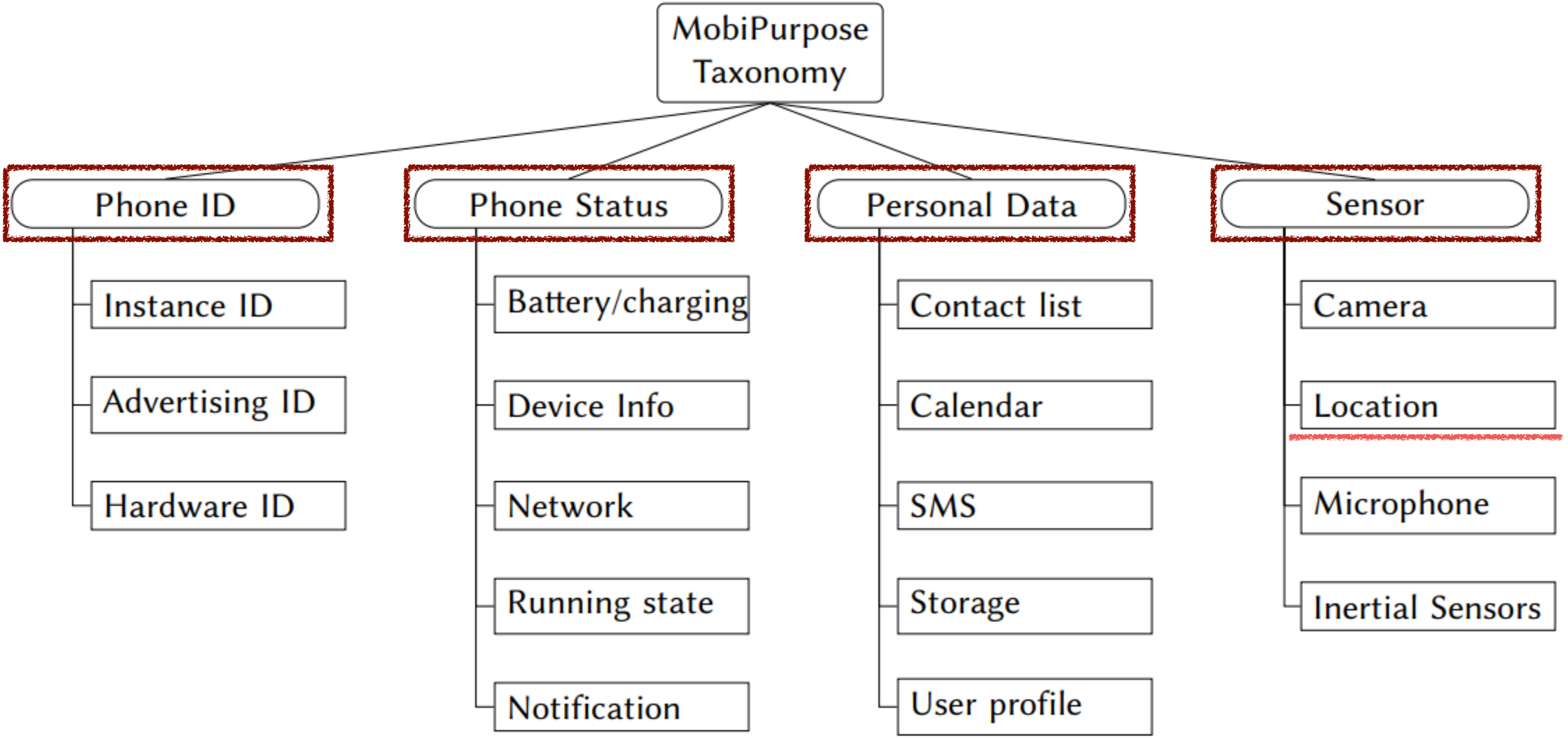
crashlytics

.....

13 common data collection purposes for location data

Location[7]	Nearby Search	Search nearby POIs/real estates
	Location-based Customization	Fetch local weather/radio information
	Query Transportation Information	Estimate the trip time through Uber API
	Recording	Track the running velocity
	Map and Navigation	Find the user location in Map apps
	Geosocial Networking	Find nearby users in the social network
	Geotagging	Tag photos with locations
	Location Spoofing	Set up fake GPS locations
	Alert and Remind	Remind location-based tasks
	Location-based game	Play games require users' physical location
	Reverse geocoding	Use the GPS coords to find the real world address.
	Data collection for analytics	Collect data for marketing analysis
	Data collection for ad	Collect data for ad personalization

See the complete taxonomy at:
http://bit.ly/mobitaxonomy


MobiPurpose Taxonomy
Phone ID
Instance ID
Advertising ID
Hardware ID
Phone Status
Battery/charging
Device Info
Network
Running state
Notification
Personal Data
Contact list
Calendar
SMS
Storage
User profile
Sensor
Camera
Location
Microphone
Inertial Sensors

data types

Three empirical studies

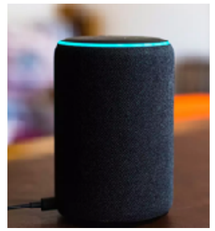
Large-scale mobile network tracing [1]

Smart home [2] applications

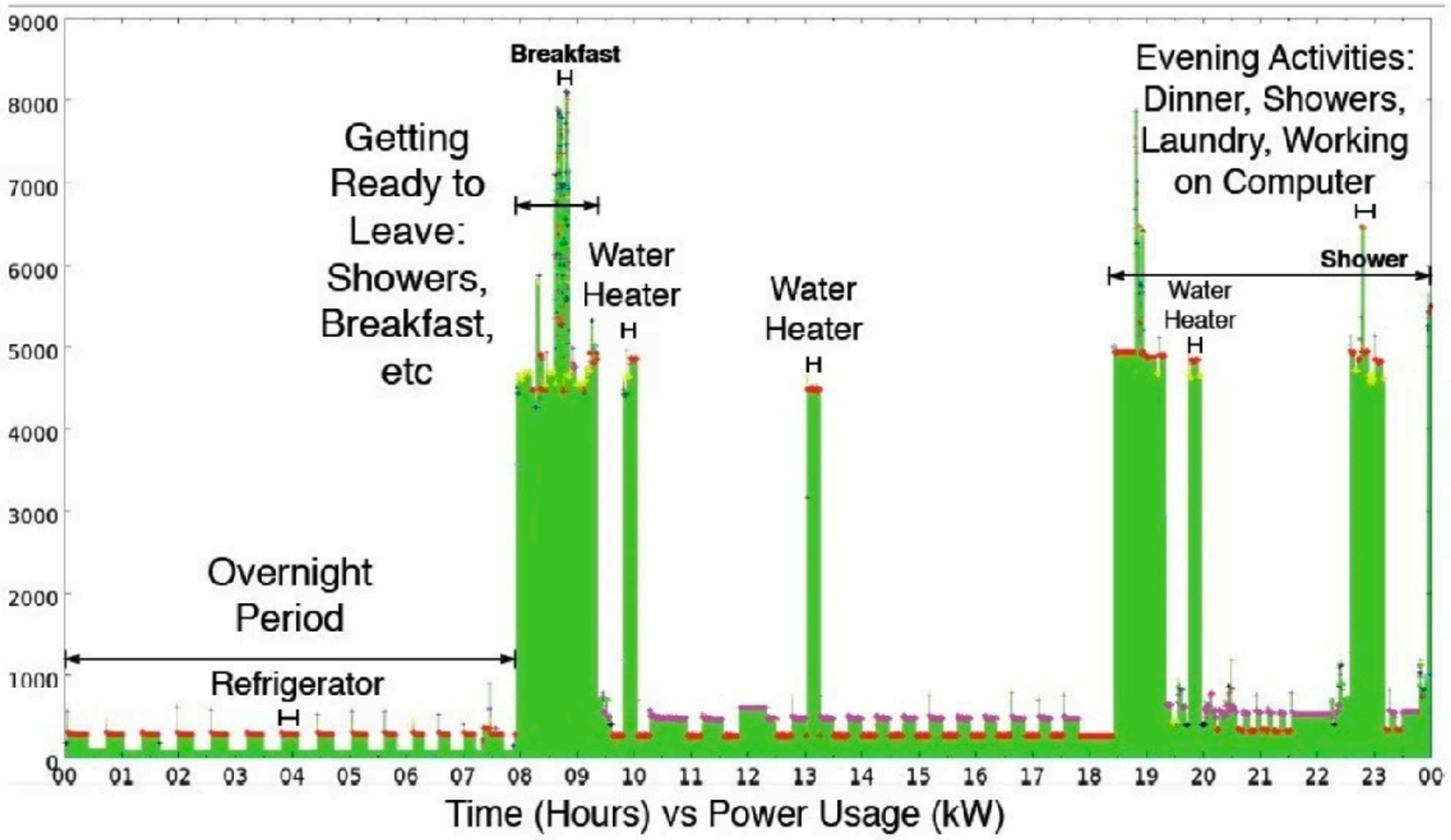
Smart city [3] applications

[2] Peekaboo, Jin et al., S&P'22

77% Smart home apps do not need raw data.

	Sensor	*Raw*	*Needed data*
Hello visitor			
Noise level			55 db

Oversensing



Private Memoirs of a Smart Meter, BuildSys 20

Three empirical studies

Large-scale mobile network tracing [1]

Smart home applications [2]

Smart city applications [3]

[2] Peekaboo, Jin et al., S&P'22

76/80 Smart city apps only need aggregated data.

Aggregation types	Applications
Count/Sum/Average	Electricity usage (sum/average), water consumption (sum/average), heat consumption (sum/average), water leaks (sum/average), fire alerts (average), CO2 emission tracking (sum), power-efficient appliances (sum/percentage), indoor vs. outdoor water usage (group sum), building vs. home energy efficiency (group sum), electric vehicle usage (sum), waste management (sum), renewable energy (sum), rainwater collection (sum), public transport tracking (count), intersection traffic (count), residents physical activity (average), pedestrian entrance tracking (count), pedestrian foot traffic (count), user mobility (count), time spent asleep (average), number of users that pass by a billboard (count), DMV visitor counting (GPS), fire department delay (average), traffic light wait time (average), irrigation tracking (sum/average), electromagnetic field level monitoring (sum/average), food consumption (sum/average), indoor home temperature (average), parking occupancy (count/average/sum)
Rank/Median/Top X	Snow plowing fairness, public space noise level monitoring
Histogram/Heatmap	Network speed monitoring (group average), water quality monitoring (group average), food consumption (group average), street light brightness (group average), noisiest neighborhoods (group average)
Route	Delivery route with best parking availability, Smart garbage pickup route, least-polluted biking path
TF-IDF	Determine restaurant type for region, trending TV search queries
Trilateration	Gunshot localization, air pollution source localization
Time-series forecasting	Parking occupancy prediction
Non-aggregatable apps	Searching for full trash cart, On-demand rides with autonomous vehicles, DMV visitors Counting through a camera, Garbage collection prediction

New understanding about privacy

State of the art

> 25% users were surprised this app sent their **approximate location** to dictionary.com for searching nearby words.

arbitrary text
manually annotated
hard to validate/assess

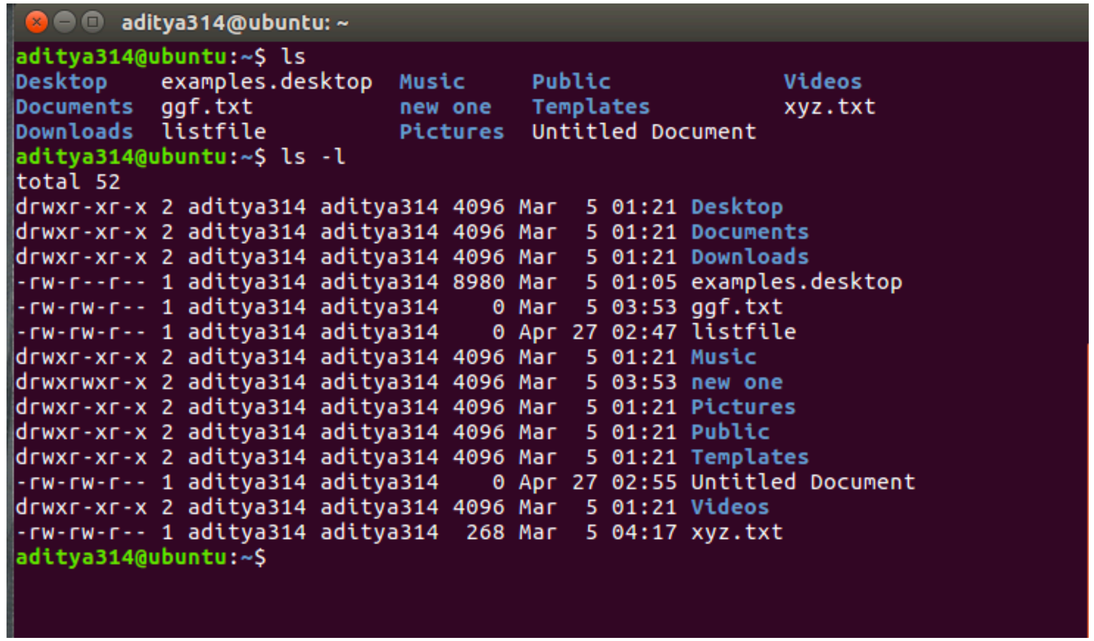
MPF

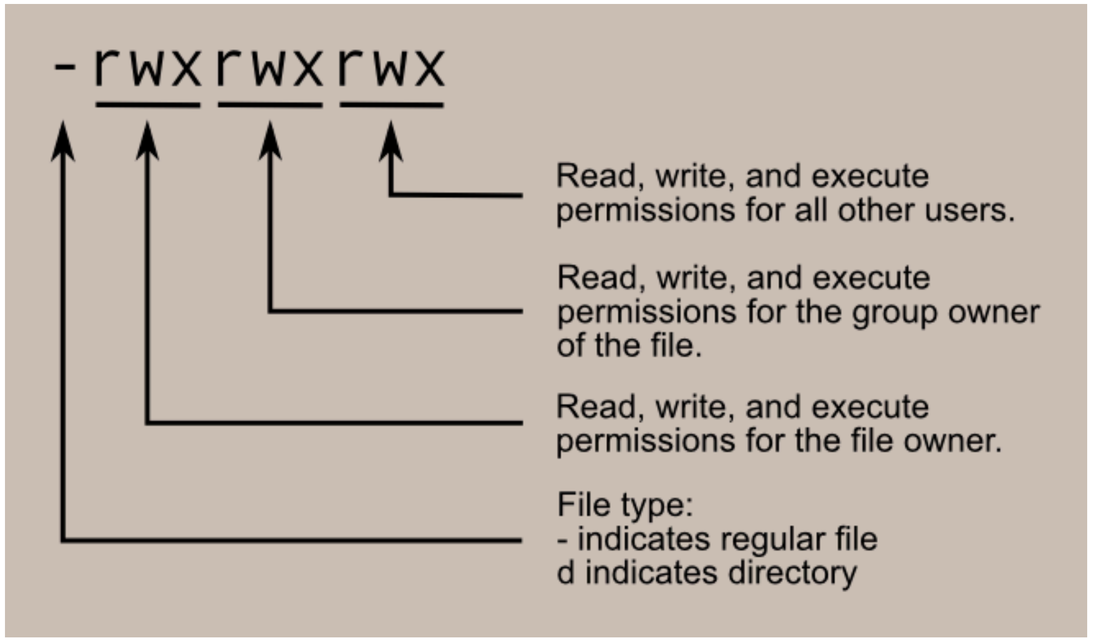
	Sensor	*Raw*	*Needed data*
Hello visitor			

Enumerable data collection purposes.

Given a purpose, developers do not need raw data.

File permissions are insufficient.

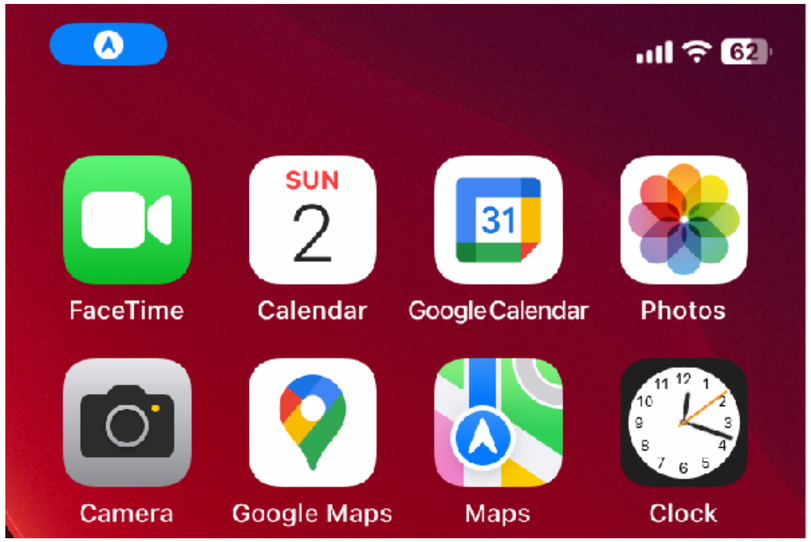


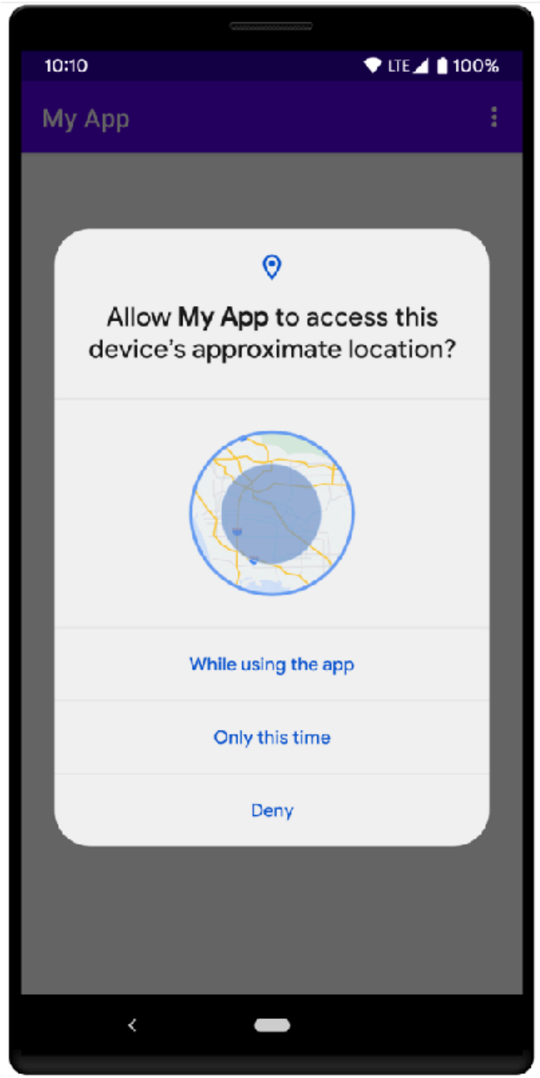


State of the Art: fine-grained permission manifest

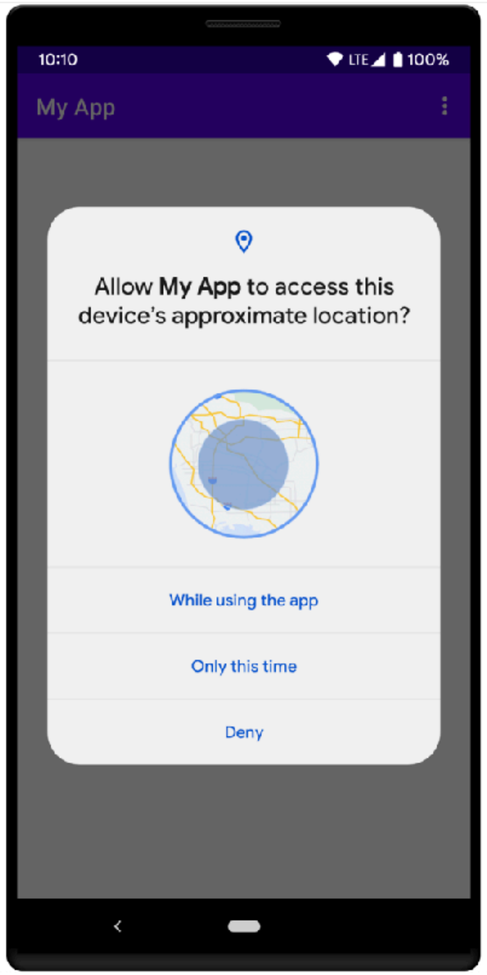
```
<manifest ... >
  <!-- Always include this permission -->
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

  <!-- Include only if your app benefits from precise location access. -->
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
</manifest>
```

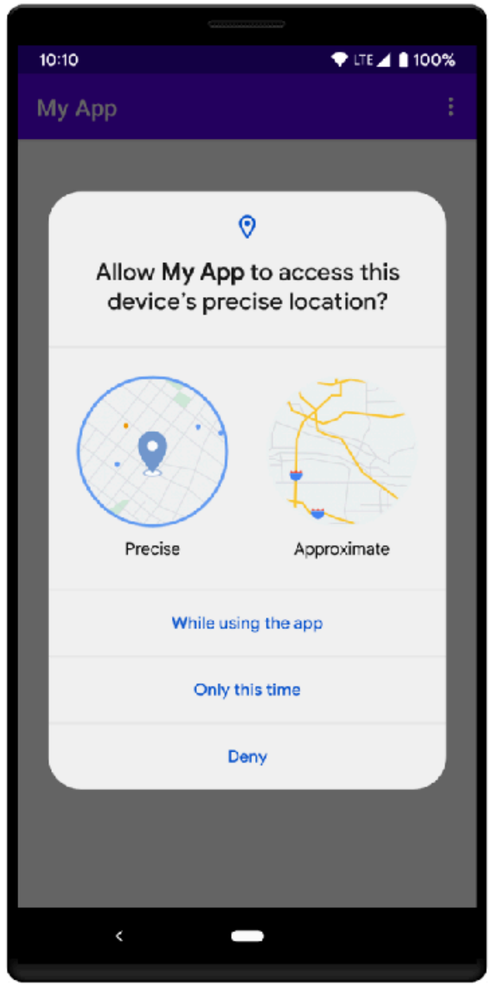
```
<manifest ... >
  <!-- Required only when requesting background location access on
       Android 10 (API level 29) and higher. -->
  <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
</manifest>
```





State of the Art: User choices





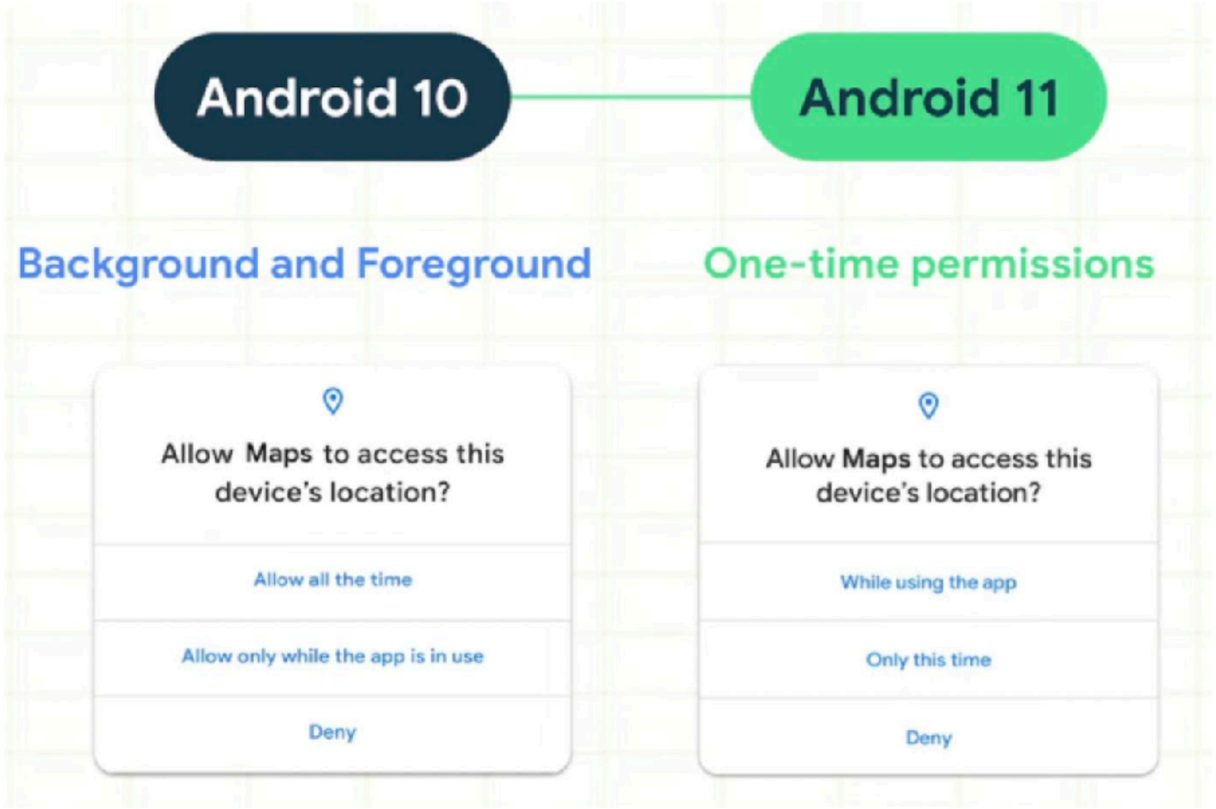
	Precise	Approximate
While using the app	`ACCESS_FINE_LOCATION` and `ACCESS_COARSE_LOCATION`	`ACCESS_COARSE_LOCATION`
Only this time	`ACCESS_FINE_LOCATION` and `ACCESS_COARSE_LOCATION`	`ACCESS_COARSE_LOCATION`
Deny	No location permissions	No location permissions

The permission granularity dilemma

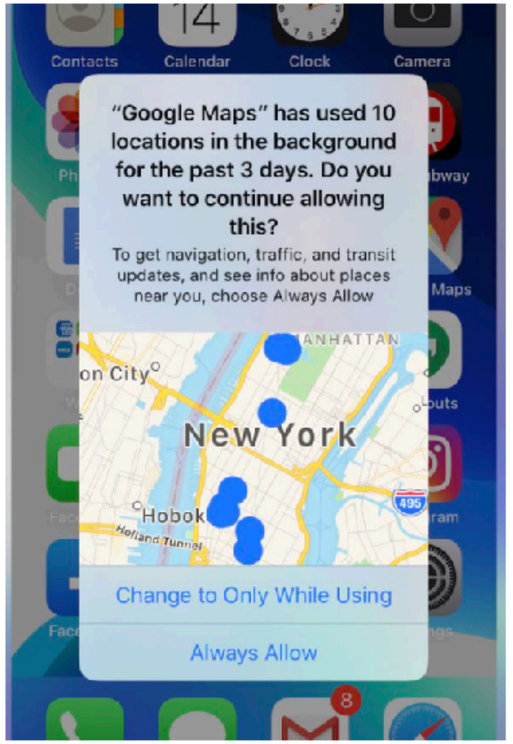
More fine-grained permissions
→ Better privacy
→ More management burden for users
  Harder learning curve for app developers
  More implementation efforts for system builders

More coarse-grained permissions
→ Worse privacy
→ Oversensing risks
  More users deny data requests
  More complaints for system builders
  Hard to gain trust from users for app developers

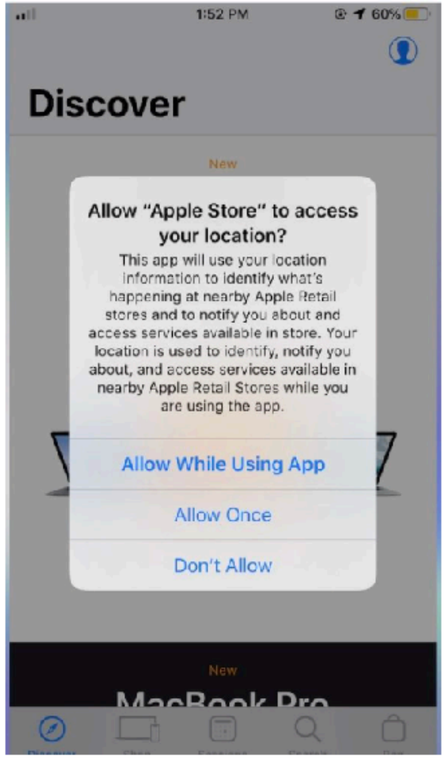
On-going permission dilemma



iOS 12



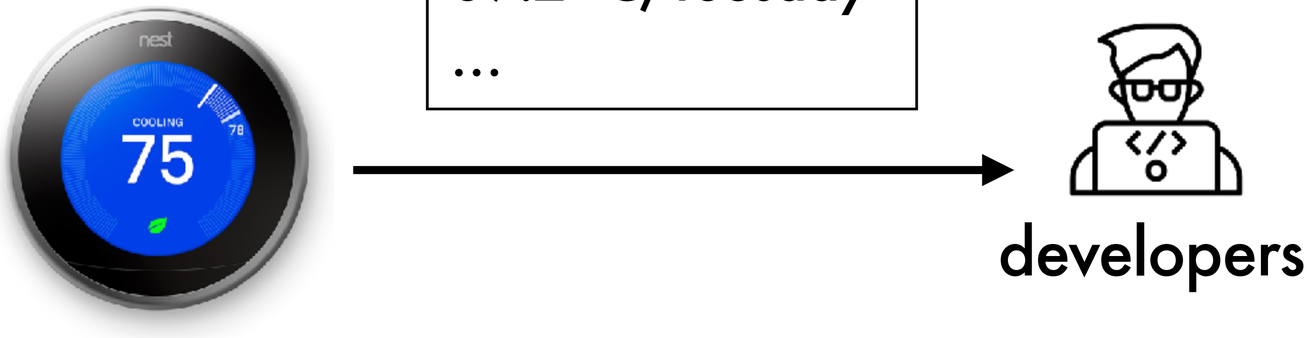
iOS 13



Talk outline

1. Modular Privacy Flows (MPF) in a Nutshell
2. Why MPF
3. How MPF
4. When and when not MPF
5. Future Work

How can Nest prove that they only collect aggregated data?

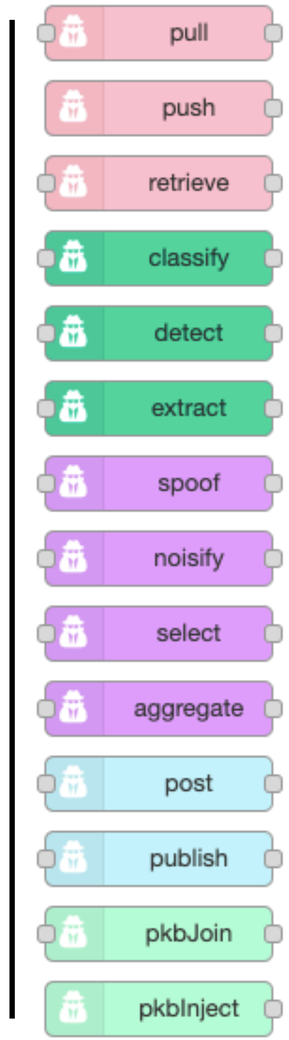


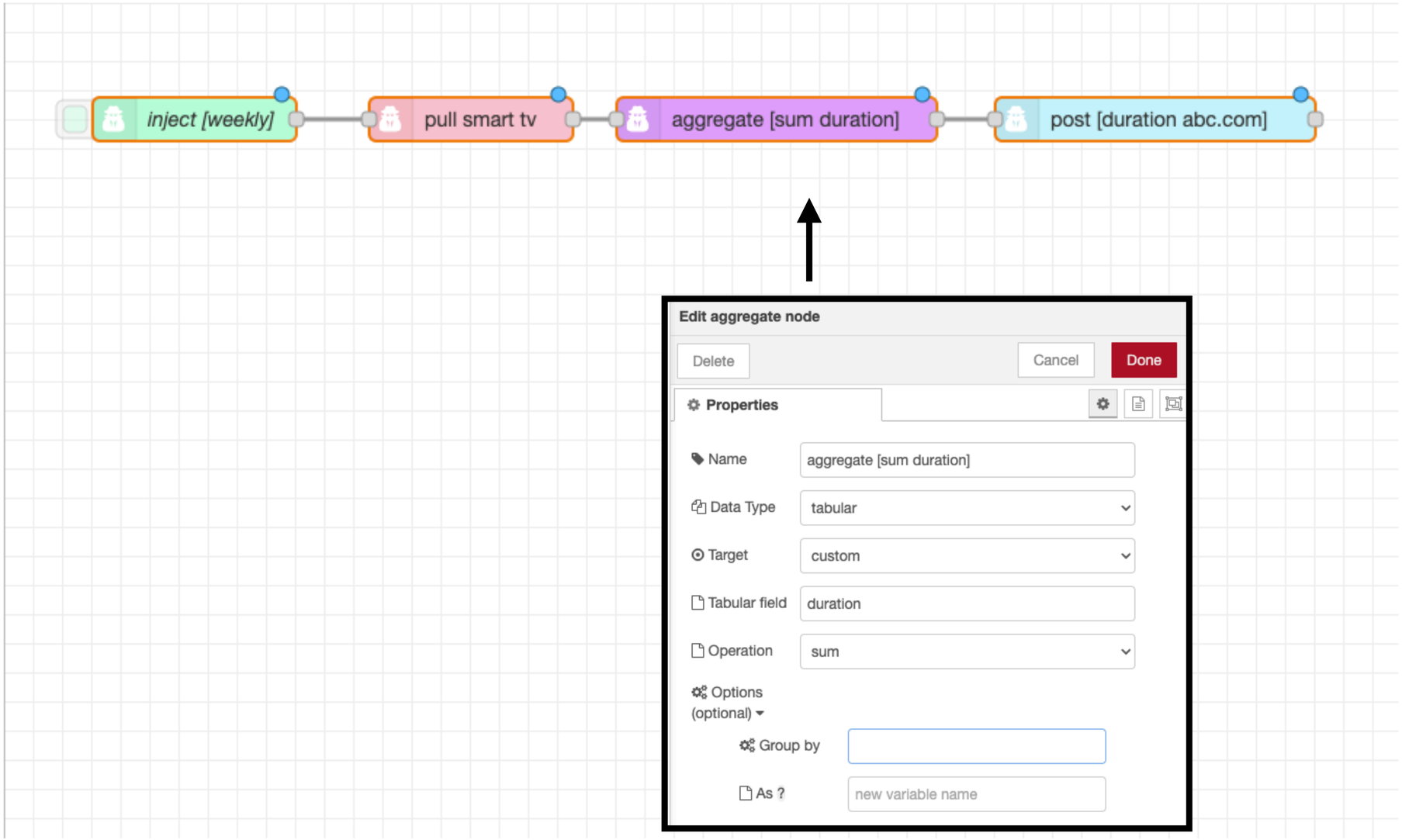
Open source?

[1] Jin et al., Peekaboo: A Hub-Based Approach to Enable Transparency in Data Processing within Smart Homes. S&P'22

Program pre-processing functions using chainable *operators*

A fixed set of operators





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

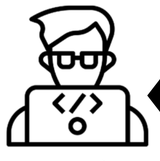
How much time does the user spend on the TV?

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

A trusted runtime with pre-loaded implementations

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

25 hours



How much time does the user spend on the TV?

A local hub

inject [weekly]

pull [smart tv]

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

aggregate [sum duration]

25 hours

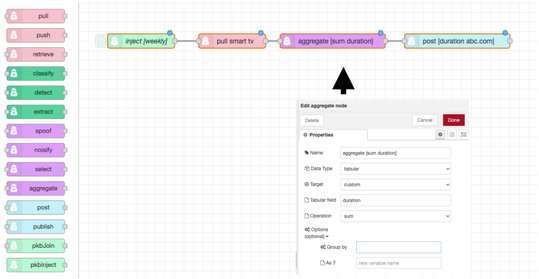
post [duration]

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

A text-based manifest

A trusted runtime with pre-loaded, open-source implementations

Smart home app store



Programming environment with operators

Runtime with preloaded implementations

App developers

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

Manifest

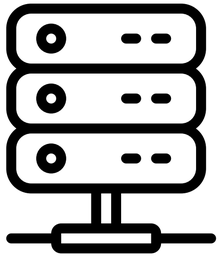
Smart home hub → privacy firewall

Smart home app →

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

Edge devices

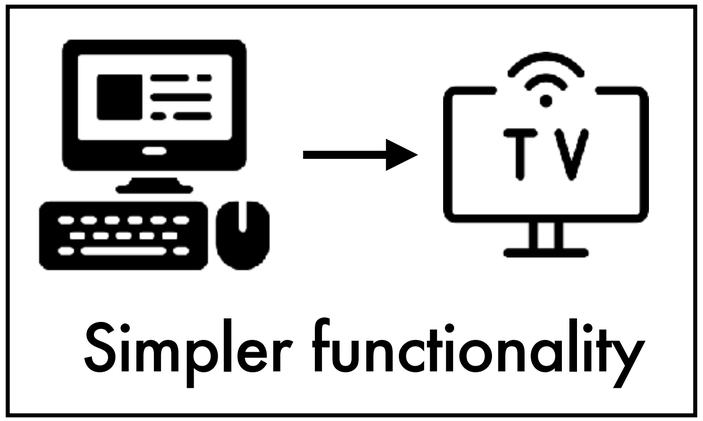
A local hub



Cloud

"Privacy firewall"

Peekaboo v.s. Firewall



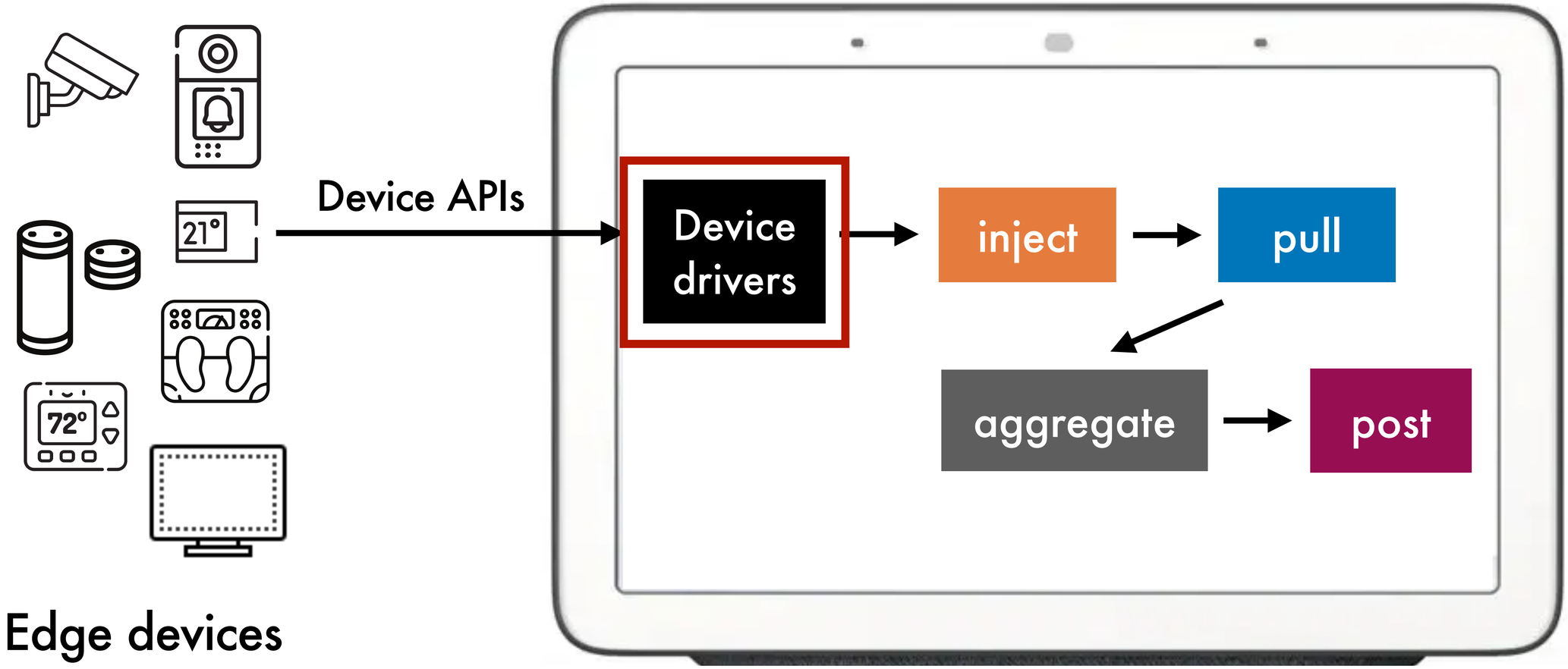
Whitelist-only

Developer-in-the-loop

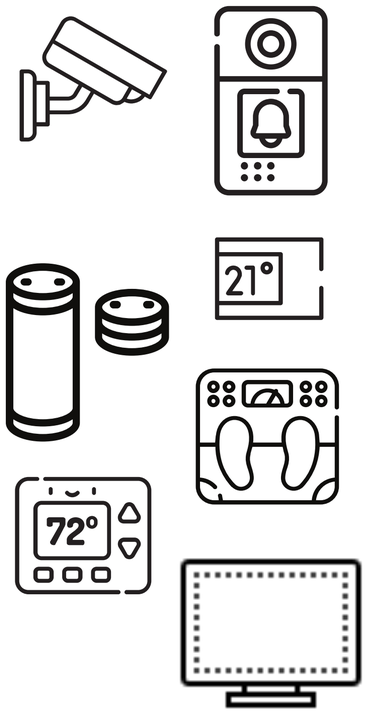
77% Apps do not need raw data.

Pre-process users' data

Handle heterogeneous hardware with device drivers



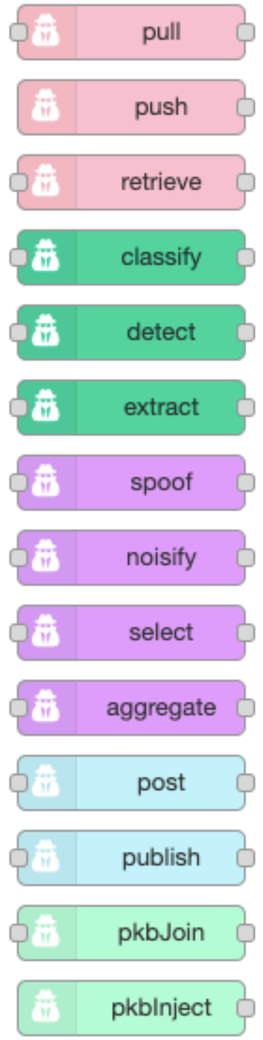
A fixed set of operators



Edge devices

video, image, audio, tabular, scalar

A *fixed* set of operators



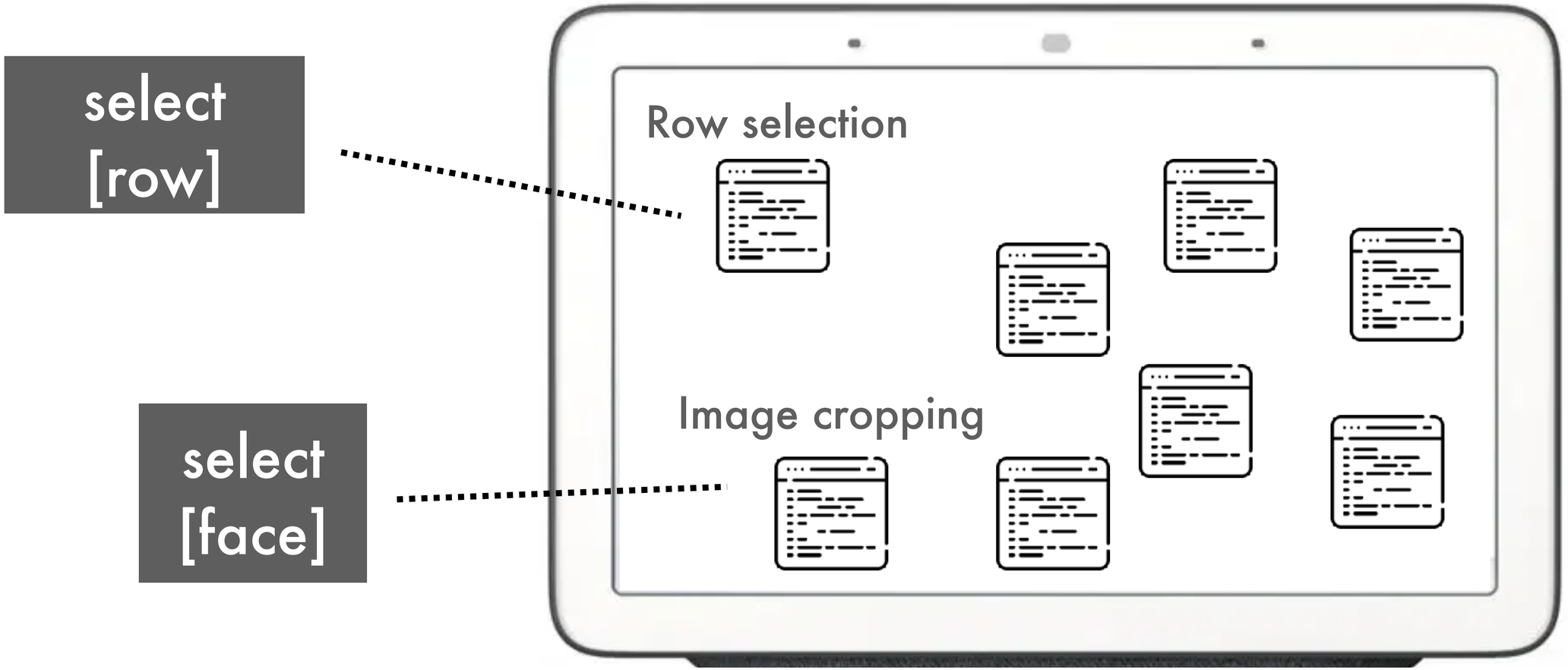
An operator = A verb keyword

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

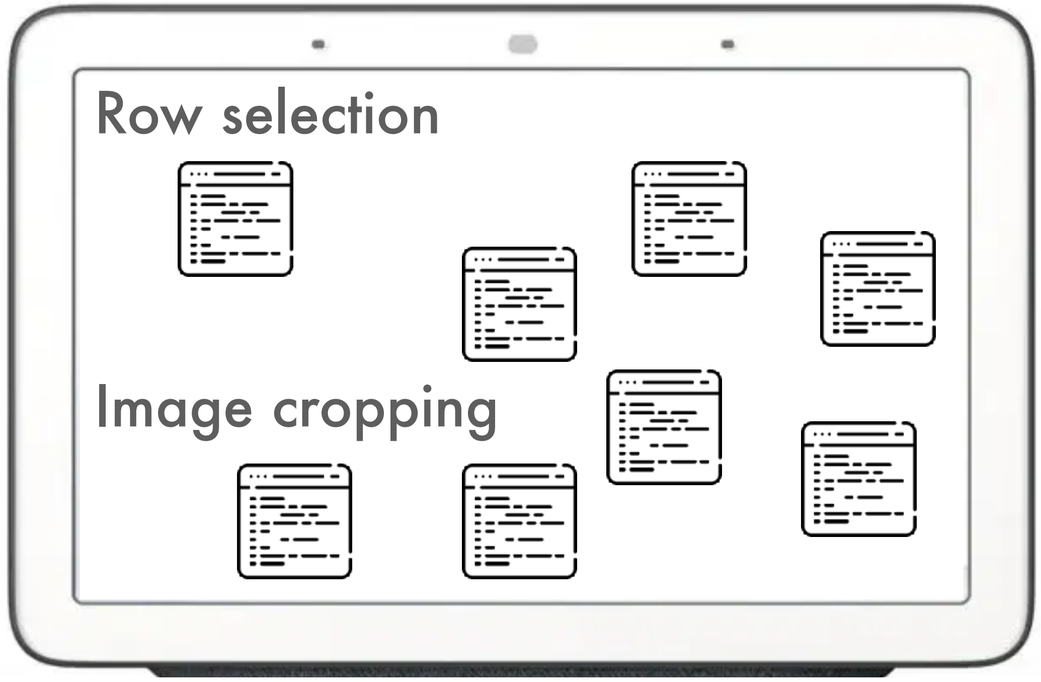
→ detect [face] → select [face] →

Operators are mapped to pre-loaded implementations



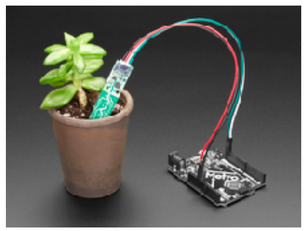
A small set of pre-processing algorithms improve privacy

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



25 hours/week

Implementation (hardware)



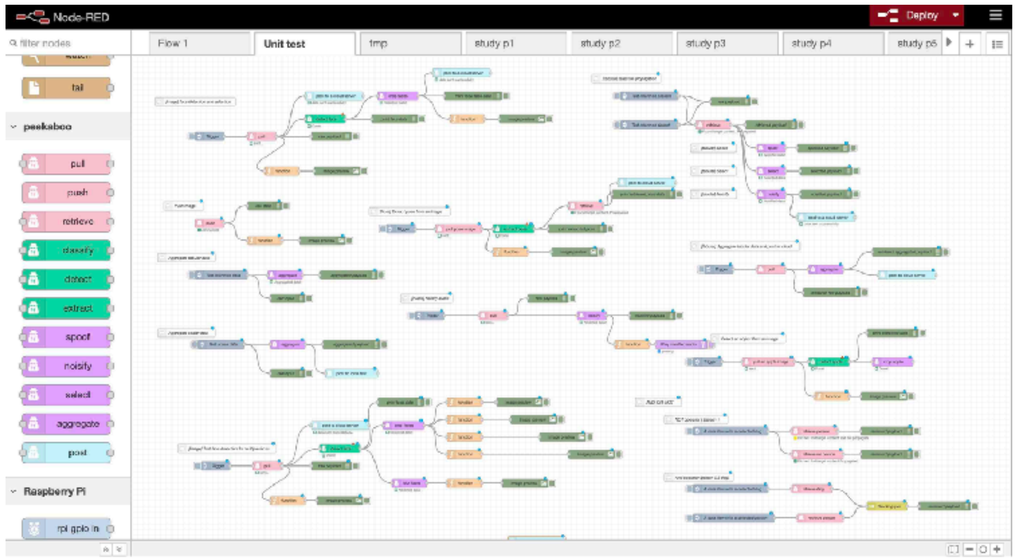
Edge devices

Raspberry PI + TPU

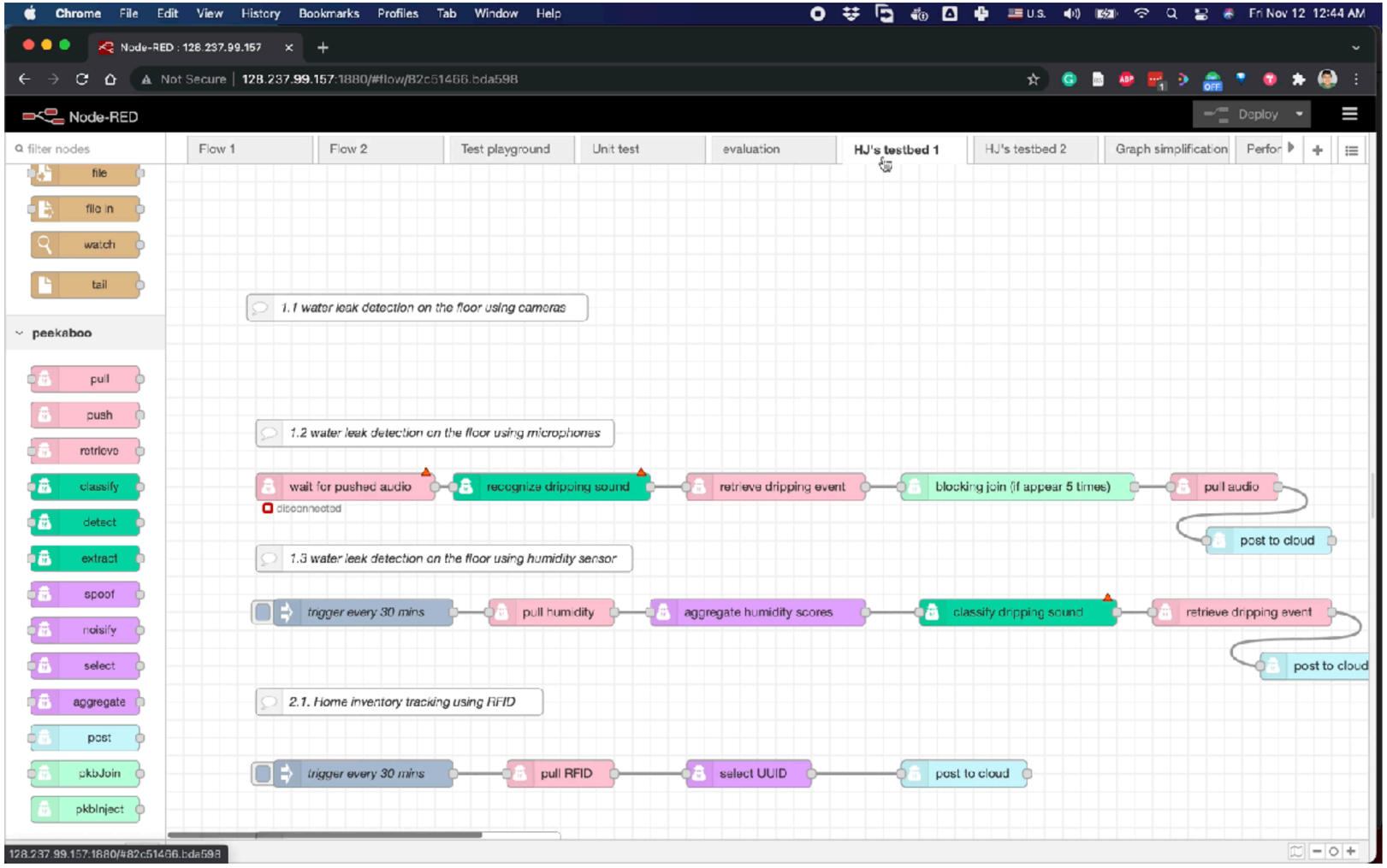
Cloud

Implementation (software)

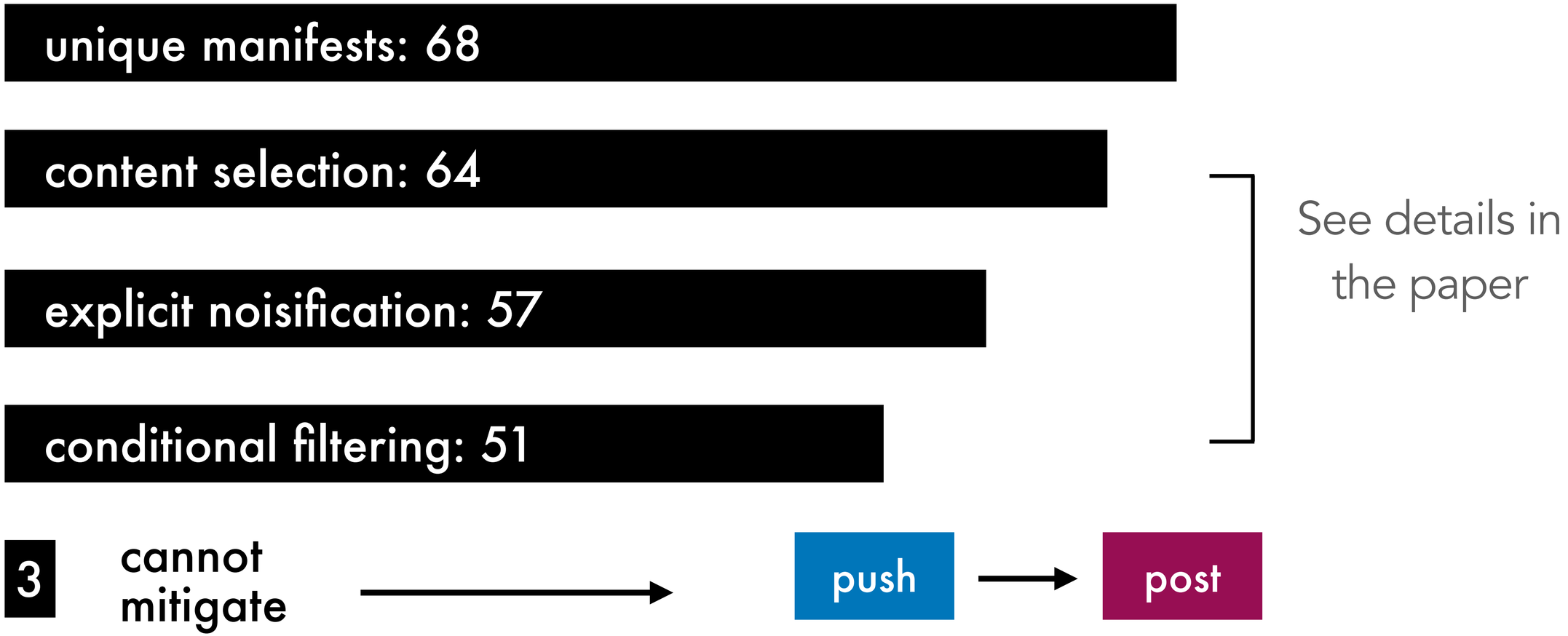
1. Operators: Node.JS package
2. Programming IDE: NodeRed
3. Drivers: 5 data types
4. 23 Preloaded implementations



Expressiveness (200+ smart home cases)



Data overaccess mitigation breakdown



System performance

≈$100

25 inference/s

100 filtering/s

1-80 ms per request

Utility privacy tradeoff example

incognito voice assistant

6 speakers
112 audio files [1]

noisify <5% random pitch shift

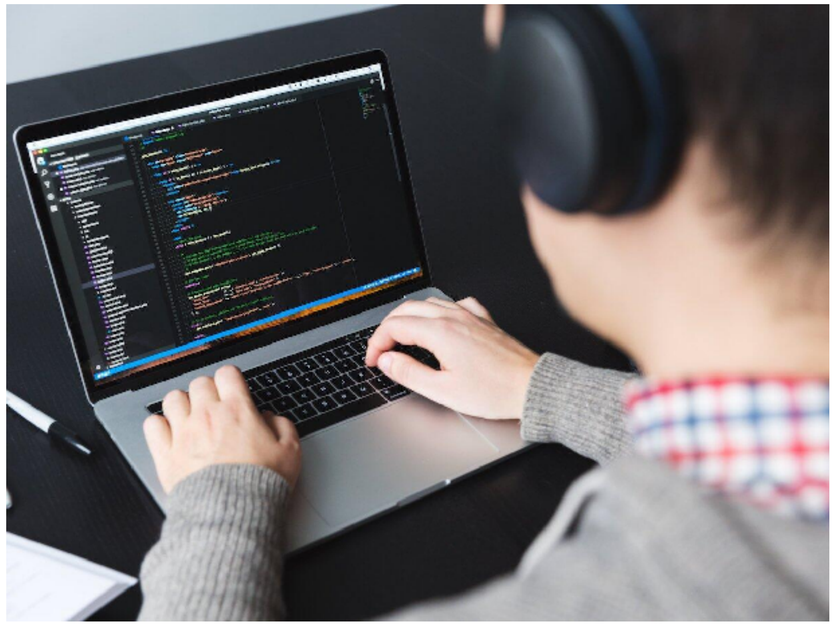
Microsoft Cognitive Services

Speech word error rate:
9.27% → 11.88%

Speaker recognization:
100% → 27.7%

[1] CMU PDA Speech Database

Developer studies



Task descriptions

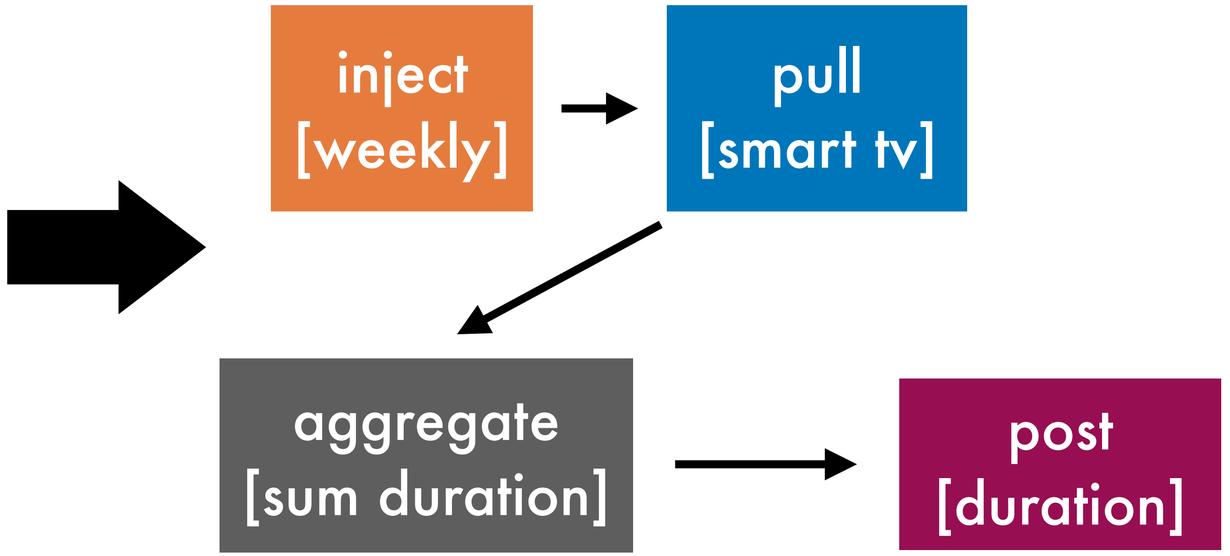
IDE & Unit tests

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]
}
```

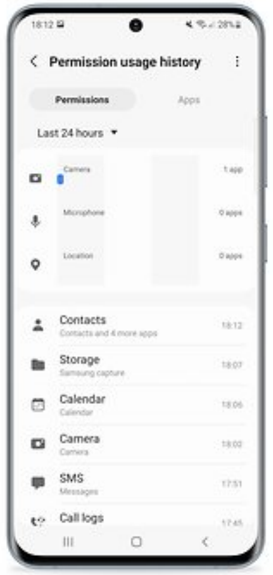


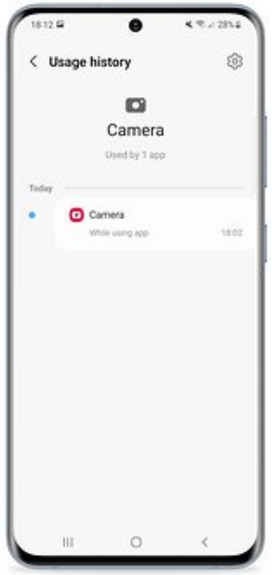
public, non-proprietary

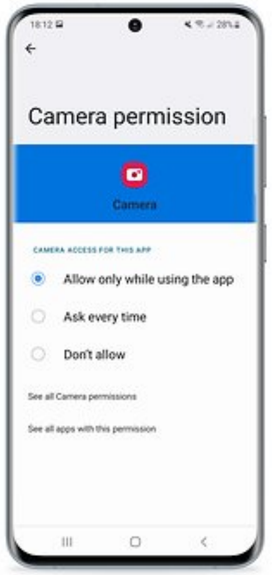
Advantages

Repetitive implementation and *distributed* interfaces

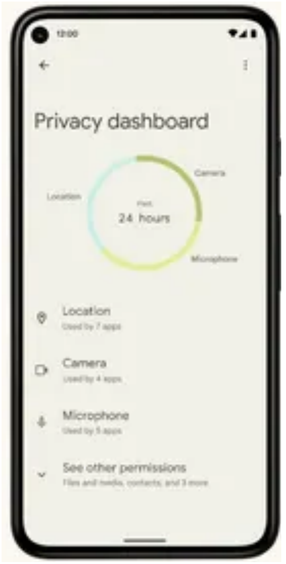
Samsung







Nest



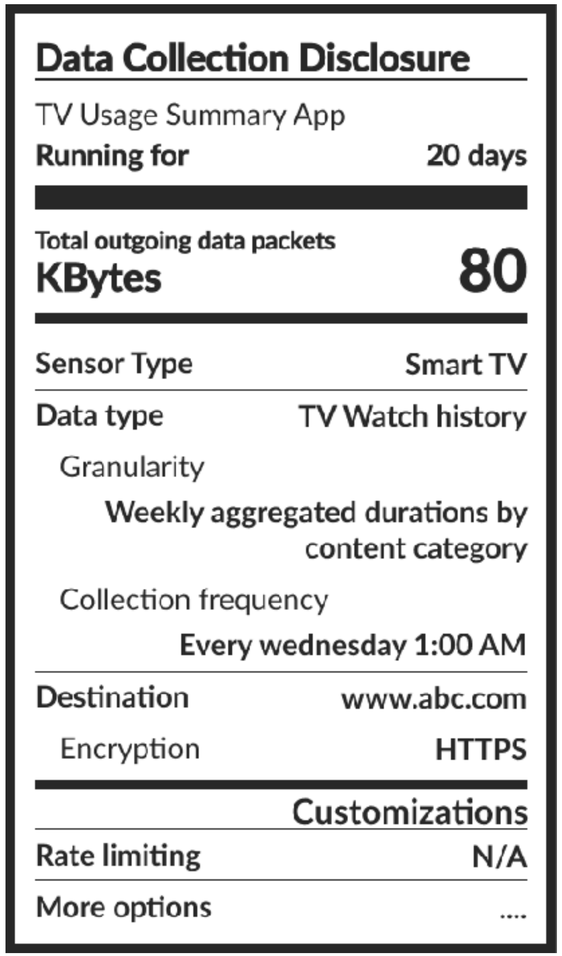
Small developers?

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]
}
```

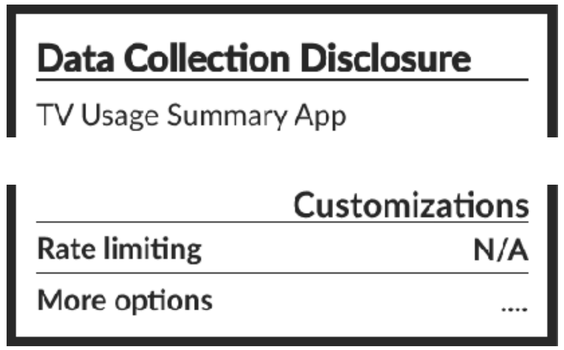


[1]

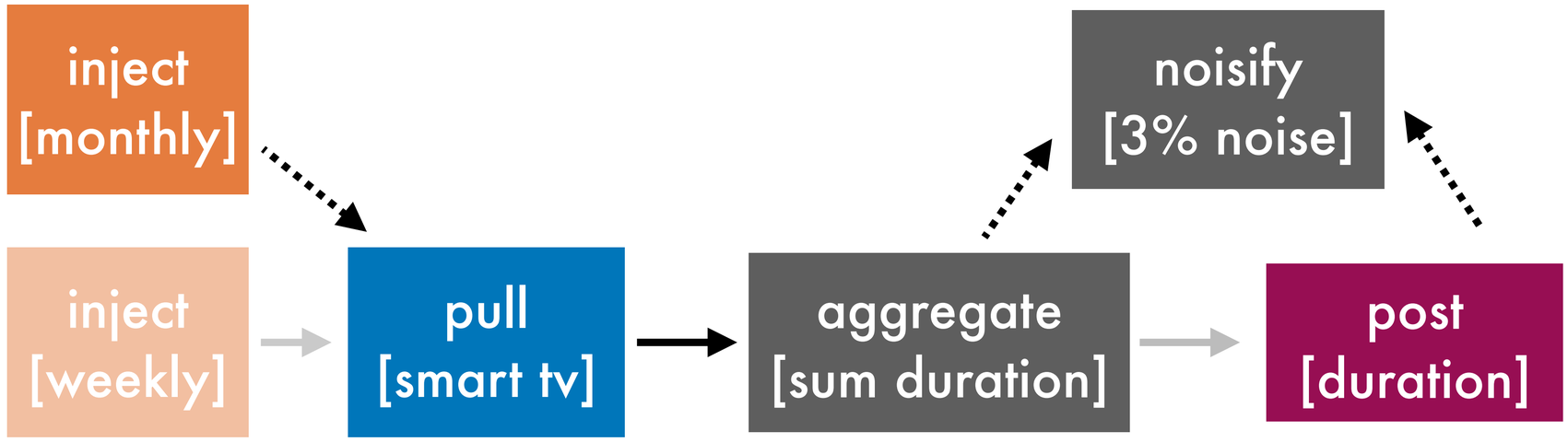
[1] Security and Privacy "Nutrition" Label, P. Emami-Naeini et al, IEEE S&P'20

Advantages

Built-in fine-grained control through manifest rewriting



Change the rate to monthly



Revisit: The permission granularity dilemma

More fine-grained permissions
→ Better privacy
→ More management burden for users
  Harder learning curve for app developers
  More implementation efforts for system builders

More coarse-grained permissions
→ Worse privacy
→ Overaccess risks
  More users deny data requests
  More complaints for system builders
  Hard to gain trust from users for app developers

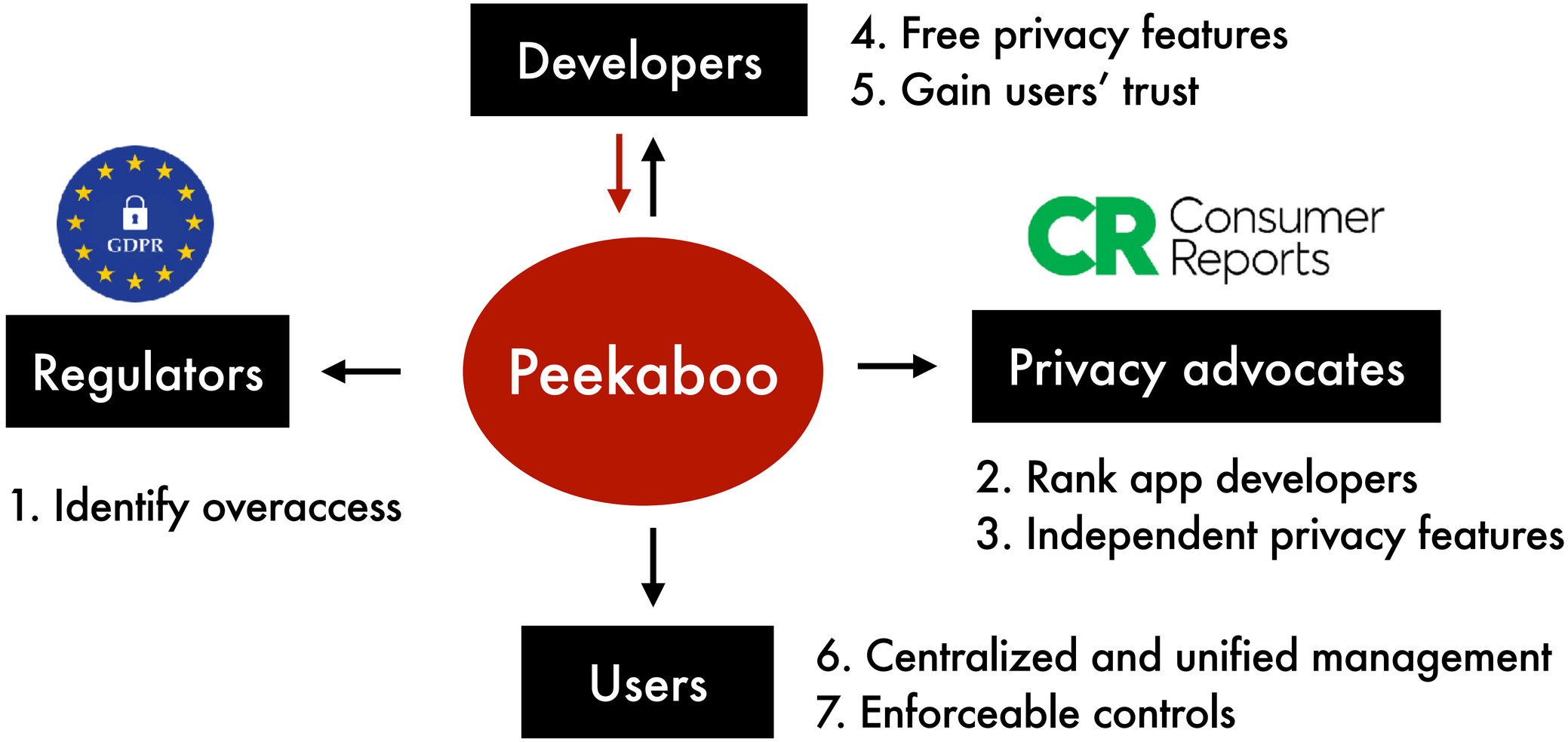
Revisit: The permission granularity dilemma

More fine-grained permissions.
→ Better default options.

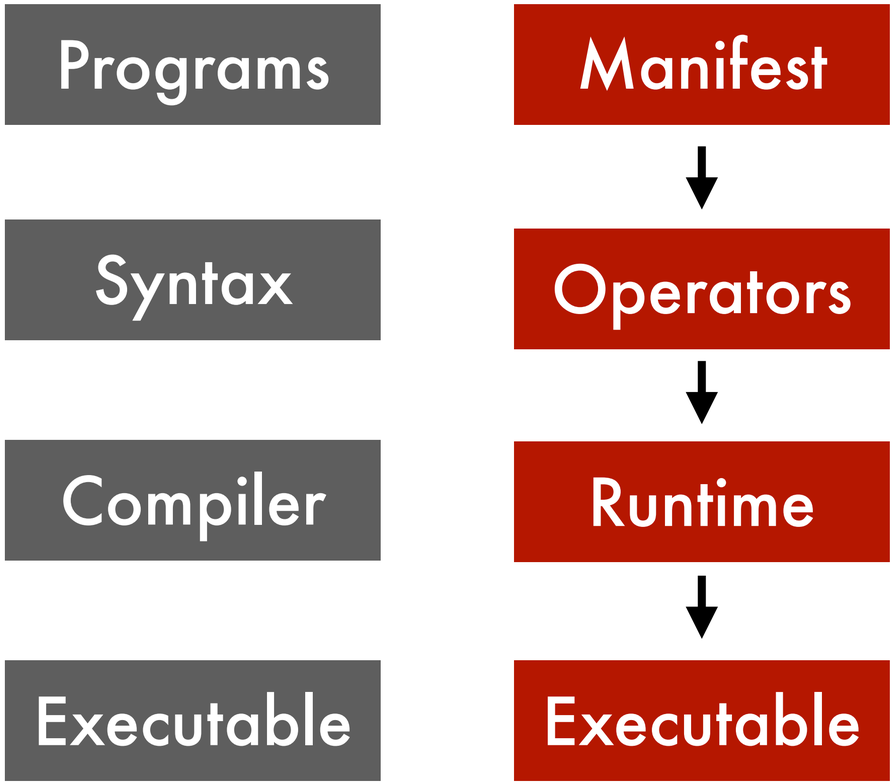
Machine-readable permissions
→ Easier to audit.
  → Better ecosystem. Good privacy drive-out bad privacy.
→ Aggregated management.

Decomposable (operator-based) permissions.
→ Fast development.

Let the good privacy drive out the bad privacy



MPF is a simpler compiler architecture.



A fixed set of operators

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

Talk outline

1. Modular Privacy Flows (MPF) in a Nutshell
2. Why MPF
3. How MPF
4. When and when not MPF
5. Future Work

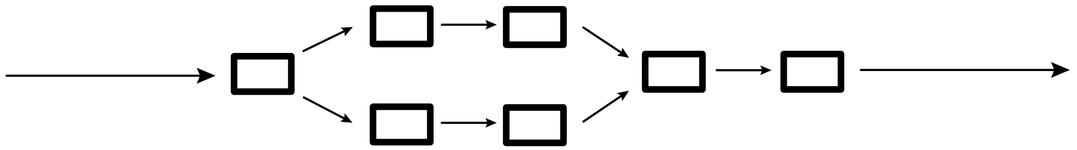
Recap:

Privacy as *modular* information flow

Who (which app) sends the data?
Where the data is being sent to?
What data is being collected?
Why the data is being collected?
How the data is being stored?

Users

Developers



Flow-based programming

Future work:

Broader application domains

Peekaboo

Smart City

Vox recode

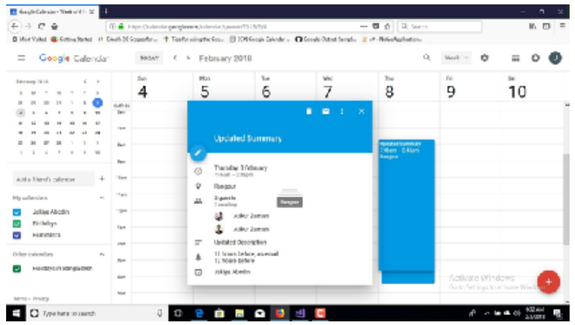
Here's how Facebook allowed Cambridge Analytica to get data for 50 million users

Facebook says it isn't at fault.

By Kurt Wagner | Mar 17, 2018, 3:47pm EDT

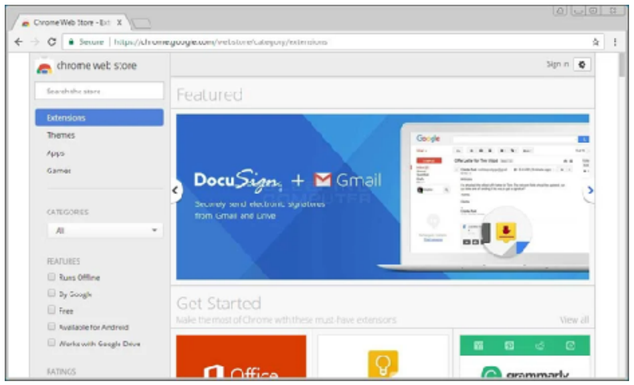
SHARE

Social network?



Personal data API?

Mobile apps?



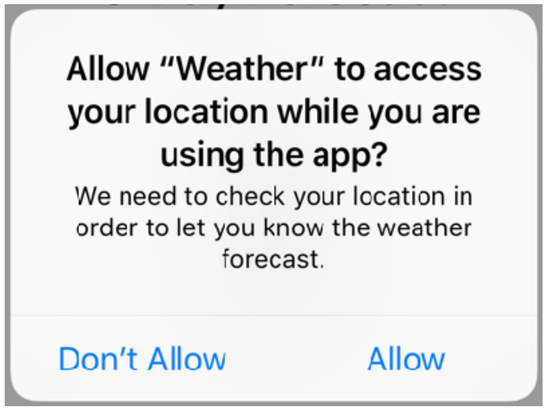
Chrome extensions?

ChatGPT Plugin?

MPF v.s. Binary permissions

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

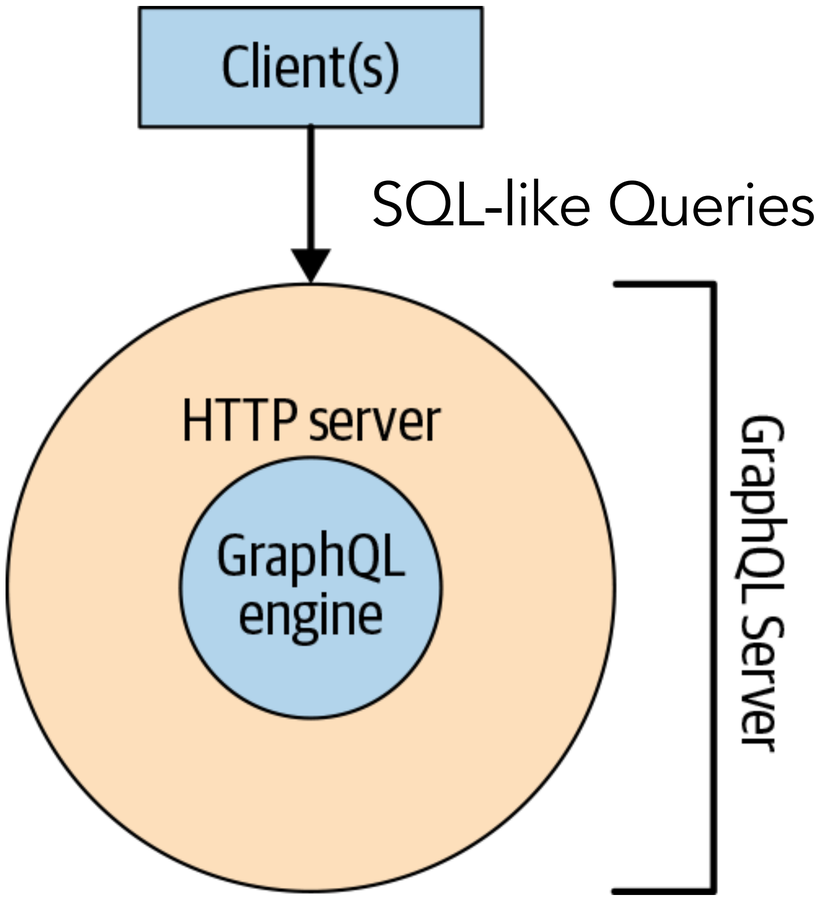
Android Permission Manifest



Popup window

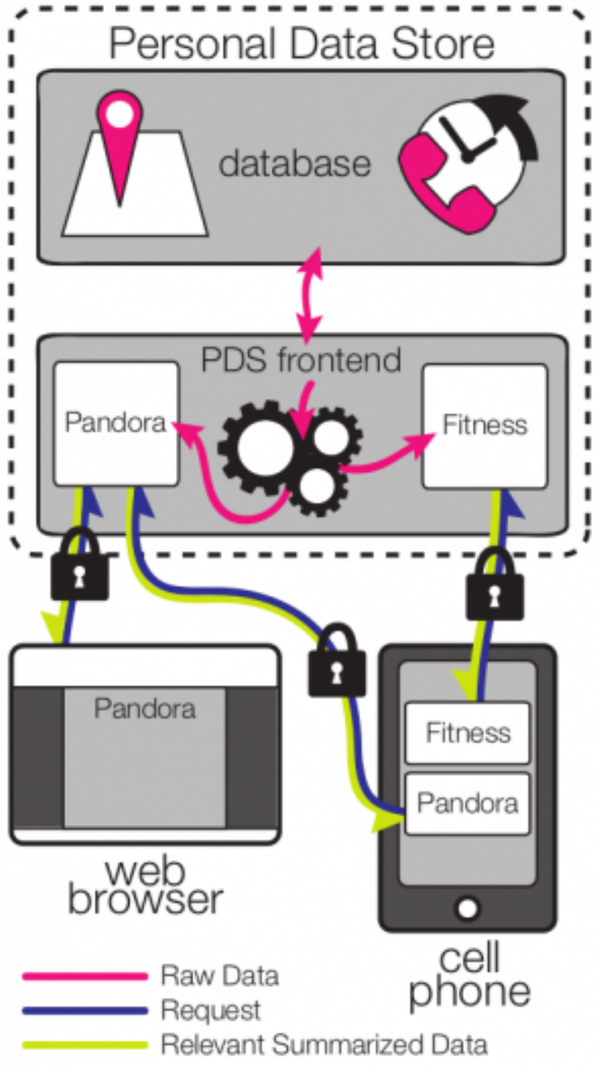
1. System implementation
2. API complexity
3. End-user management

MPF v.s. Database approaches (e.g., GraphQL)



1. Flexibility/Extendability
2. Auditability

MPF v.s. Remote Code Execution

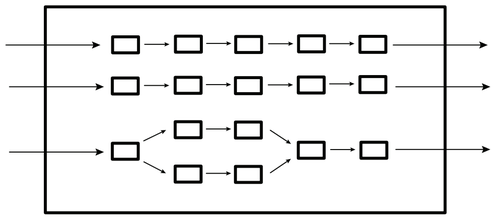


Developer-uploaded arbitrary code

1. Auditability
2. App development
3. Security

https://openpds.media.mit.edu/

Modular Privacy Flows today is REST in 2000.

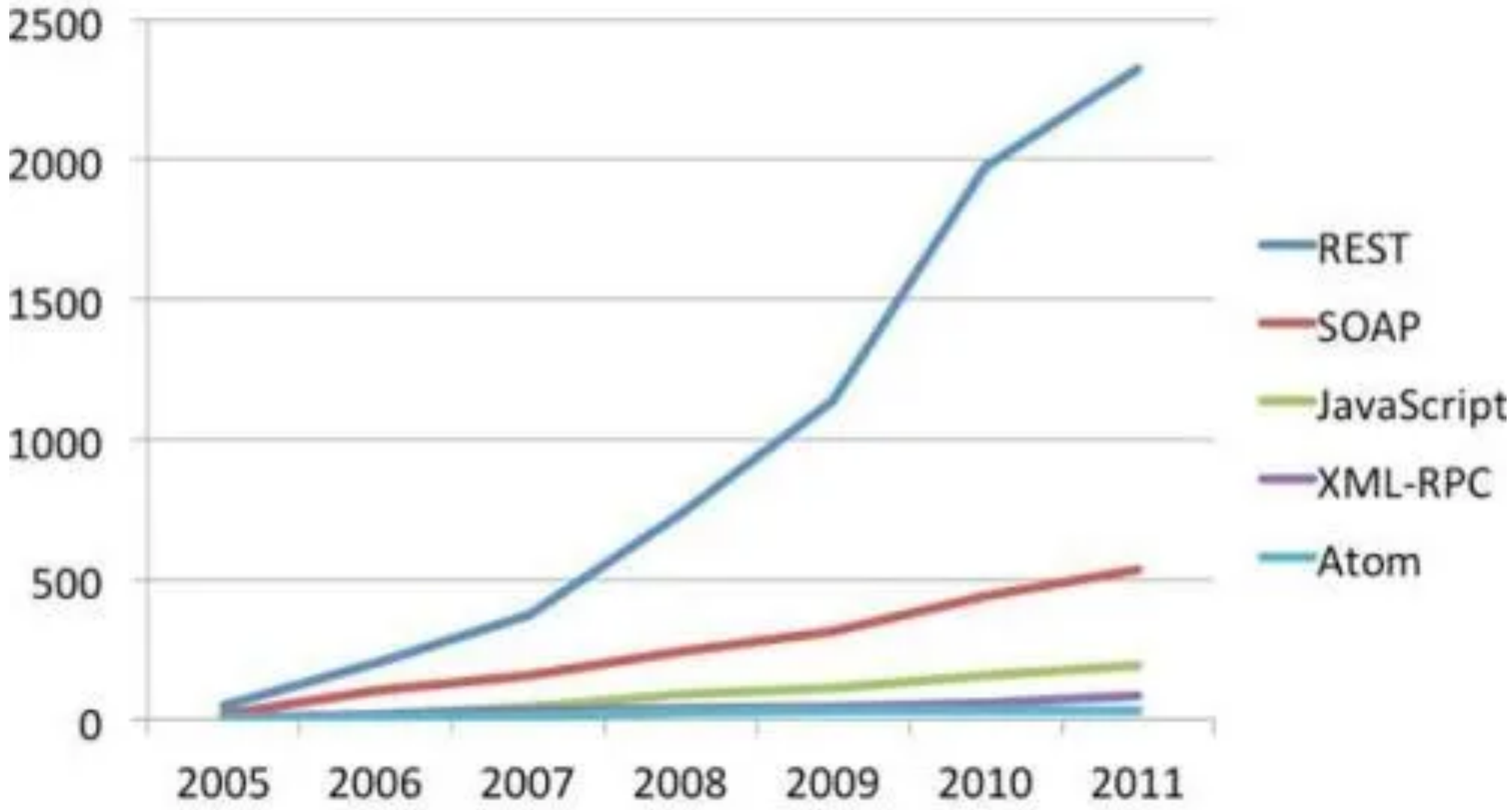


GET /pet/{petId} Find pet by ID

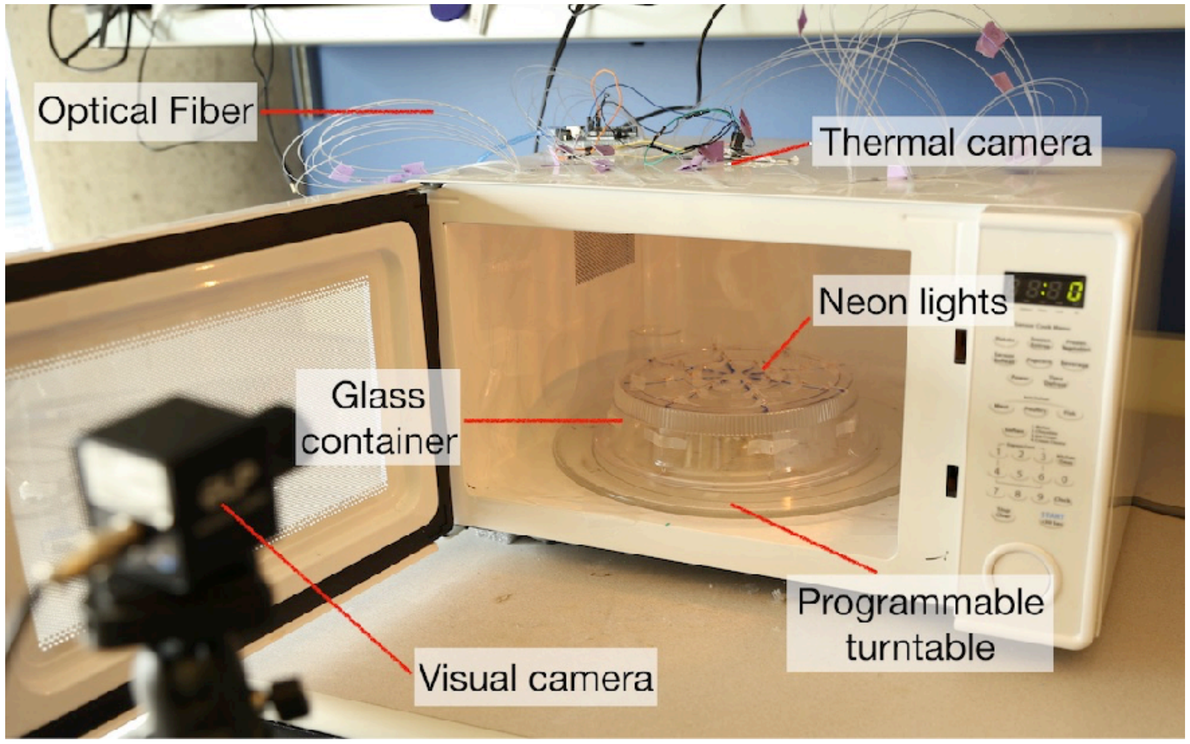
PUT /pet Update an existing pet

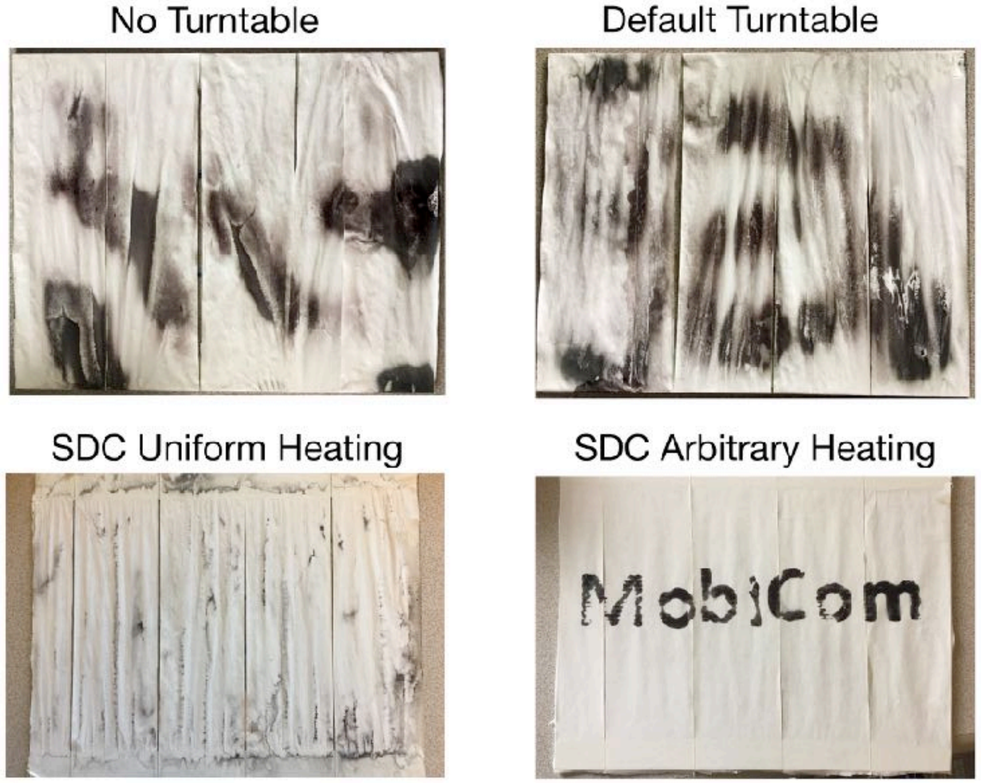
DELETE /pet/{petId} Deletes a pet

POST /pet/{petId}/uploadImage uploads an image



Software-defined Cooking

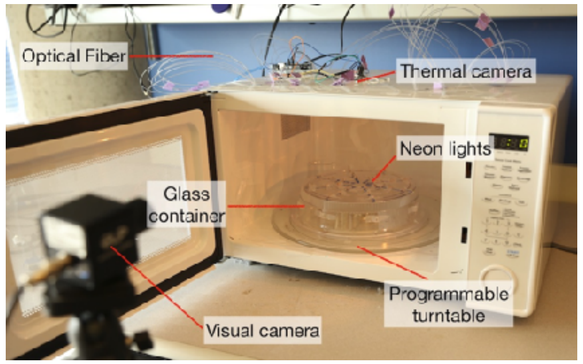





:58
Sensor Cook Menu
Potato
Frozen Entree
Frozen Vegetables
Sensor Reheat
Popcorn
Beverage
Power
Time Defrost
Meat
Poultry
Fish
Soften
Express Cook
1
2
3
Kitchen Timer
4
5
6
0
7
8
9
Clock
Stop Clear
START
+30 Sec
SHARP
Carousel

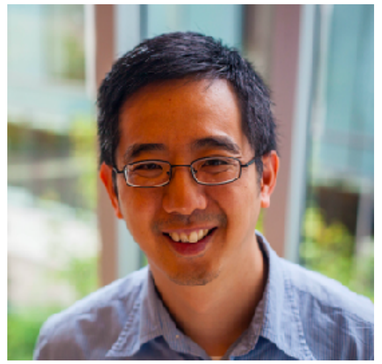
A story behind Modular Privacy Flows

Software Defined Hardware



+Operators

+Runtime



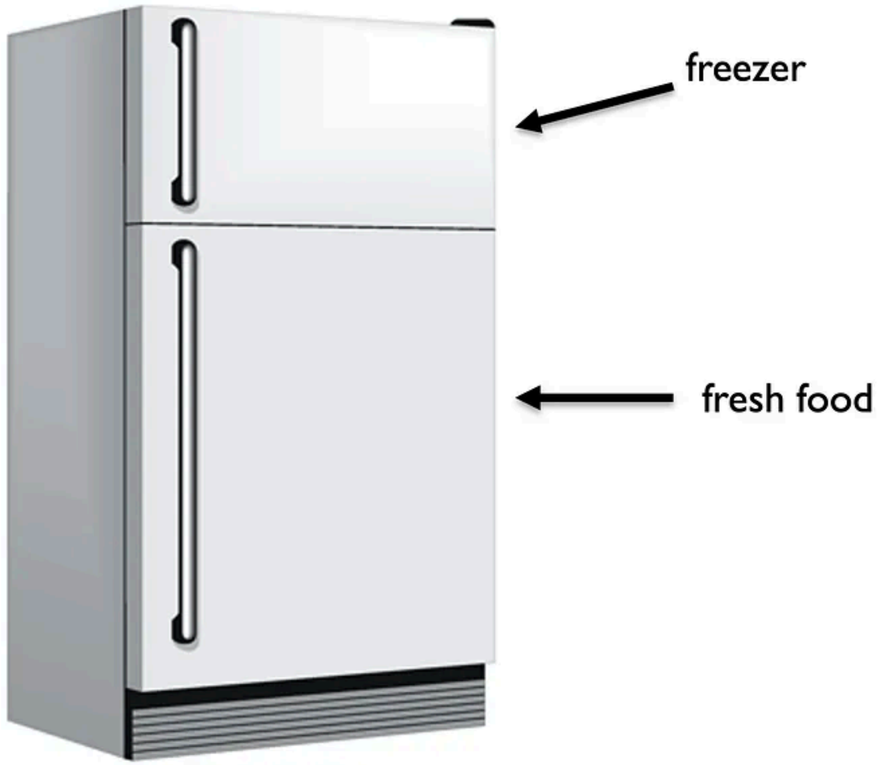
+Manifest

Talk outline

1. Modular Privacy Flows (MPF) in a Nutshell
2. Why MPF
3. How MPF
4. When and when not MPF
5. Future Work

Computing systems are increasingly complex, we need something on the order of a single page demonstrating that the system will work as intended.

Mental models

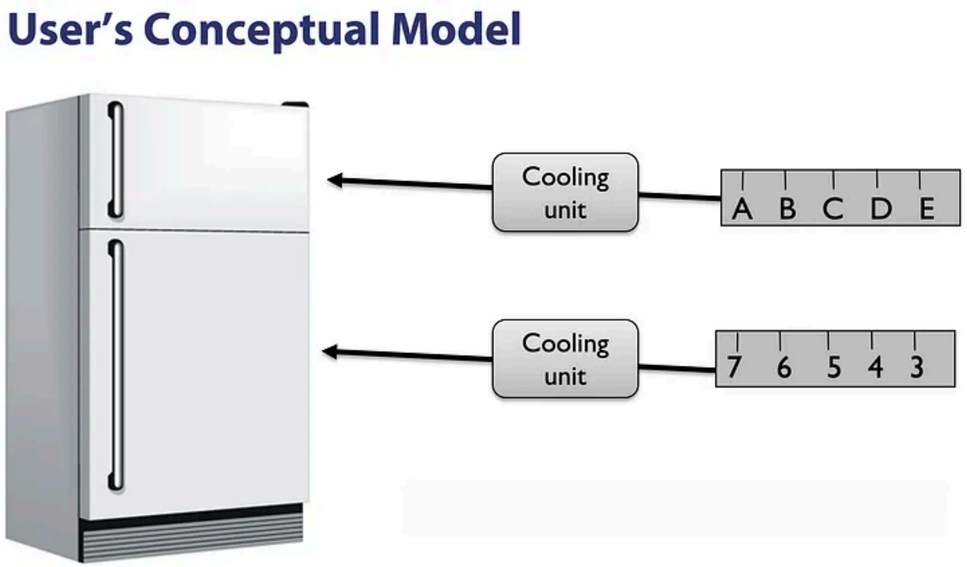


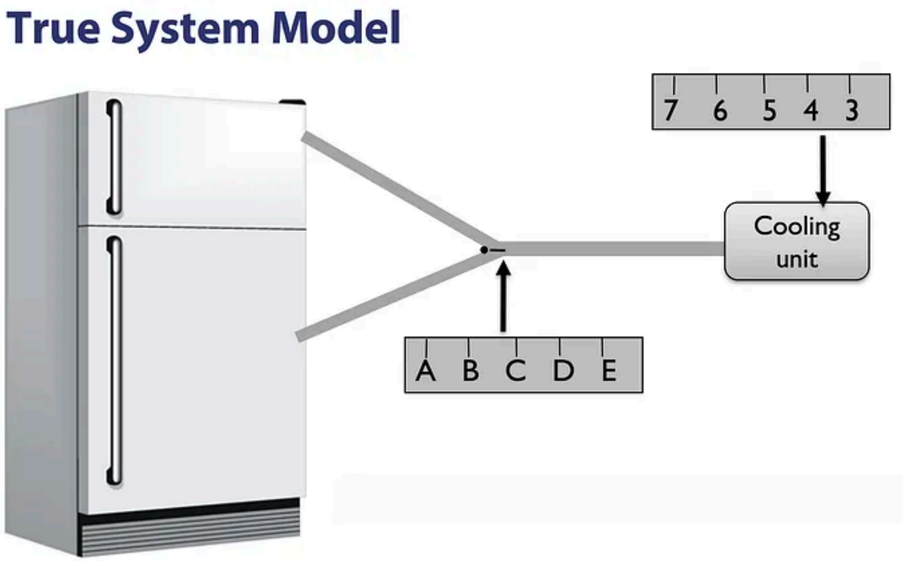
Normal Settings	C and 4
Colder Fresh Food	C and 5-6
Coldest Fresh Food	B and 7
Colder Freezer	D and 6-7
Warmer Fresh Food	C and 3-1
OFF (both)	

Freezer

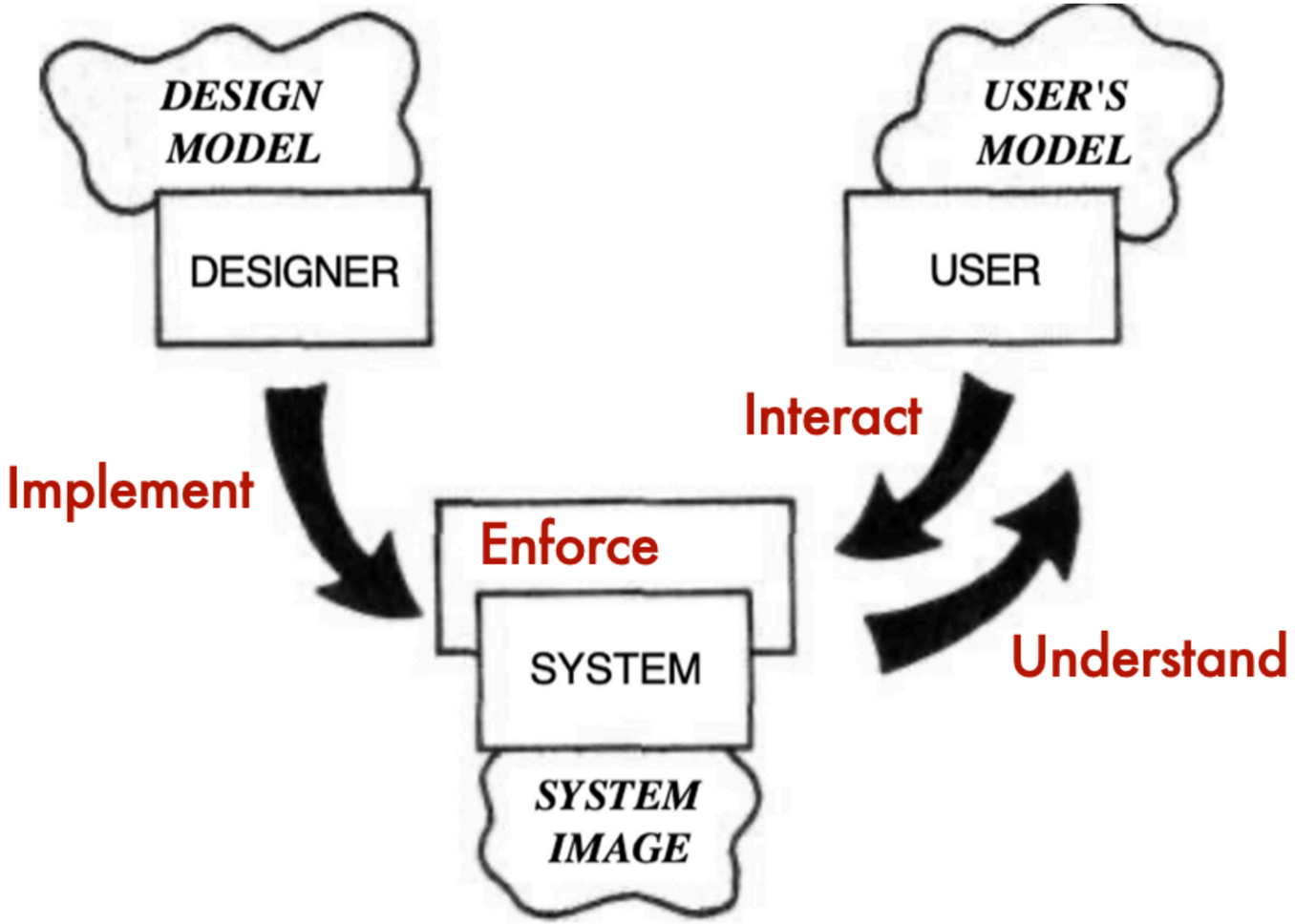
Fresh Food

The Gap


User's Conceptual Model
Cooling unit
A B C D E
Cooling unit
7 6 5 4 3

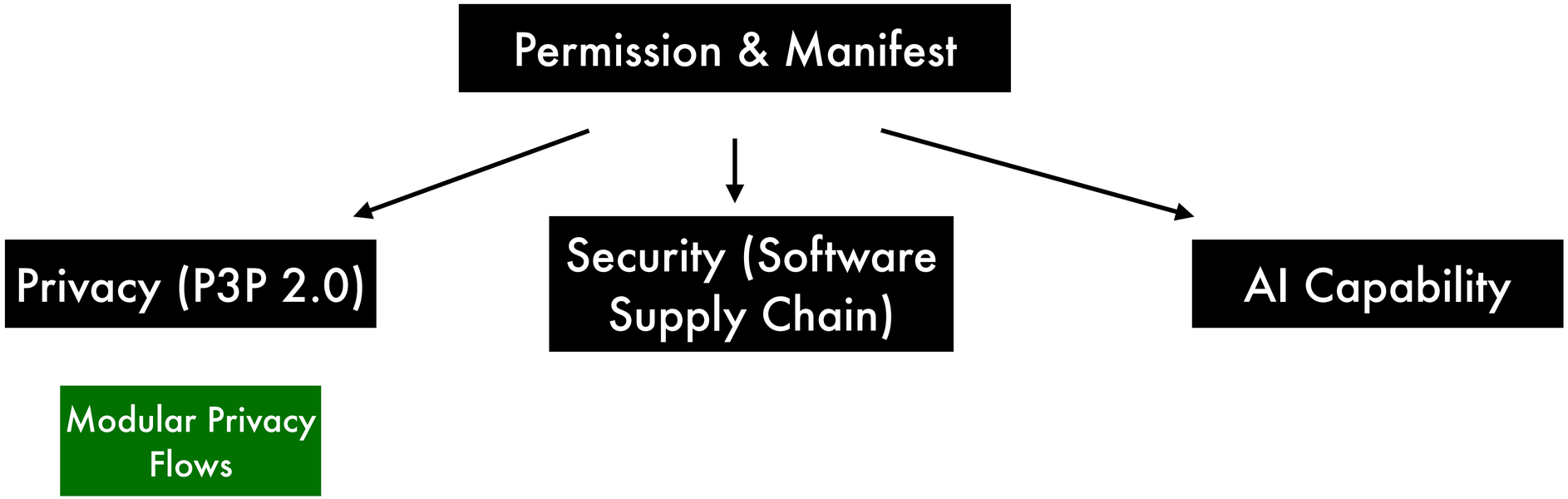

True System Model
7 6 5 4 3
Cooling unit
A B C D E

Our approach



Current research focus

Next generation permission system



Data Smith Lab is recruiting!

We study the security and privacy of data systems by researching the people who design, implement, and use these systems.

Contact: haojian@ucsd.edu
http://haojianj.in/

Acknowledgment