UNIVERSITÀ DEGLI STUDI DELL'INSUBRIA

PrivacyAssist: A User-Centric Agent Framework for Detecting Privacy Inconsistencies in Android Apps

19th ACM Conference on Security and Privacy in Wireless and Mobile Networks

(WiSec 2026)

Lam Nguyen, Edoardo Di Tullio, Barbara Carminati, and Elena Ferrari

Email: lam.nguyen@uninsubria.it

OUTLINE

1. Introduction
   - current problem & threat model
2. Methodology
3. Result
4. Future work

1. Introduction: Current Problem

- General Data Protection Regulation (**GDPR**) & California Consumer Privacy Act (**CCPA**) introduced to protect user privacy, have two main principles:
  - **Lawfulness** → requires user's consent for data collection
  - **Transparency** → requires informing to the user how and why their data is collected and shared.
- Android provides 2 mechanisms:
  - **Permission model (PM):** the user can grant permission at runtime to consent app access to sensitive data
  - **Data Safety (DS):** the developer discloses what sensitive data the app will collect and share

➡ **PM & DS** still have many weaknesses!

1. Introduction: Current Problem

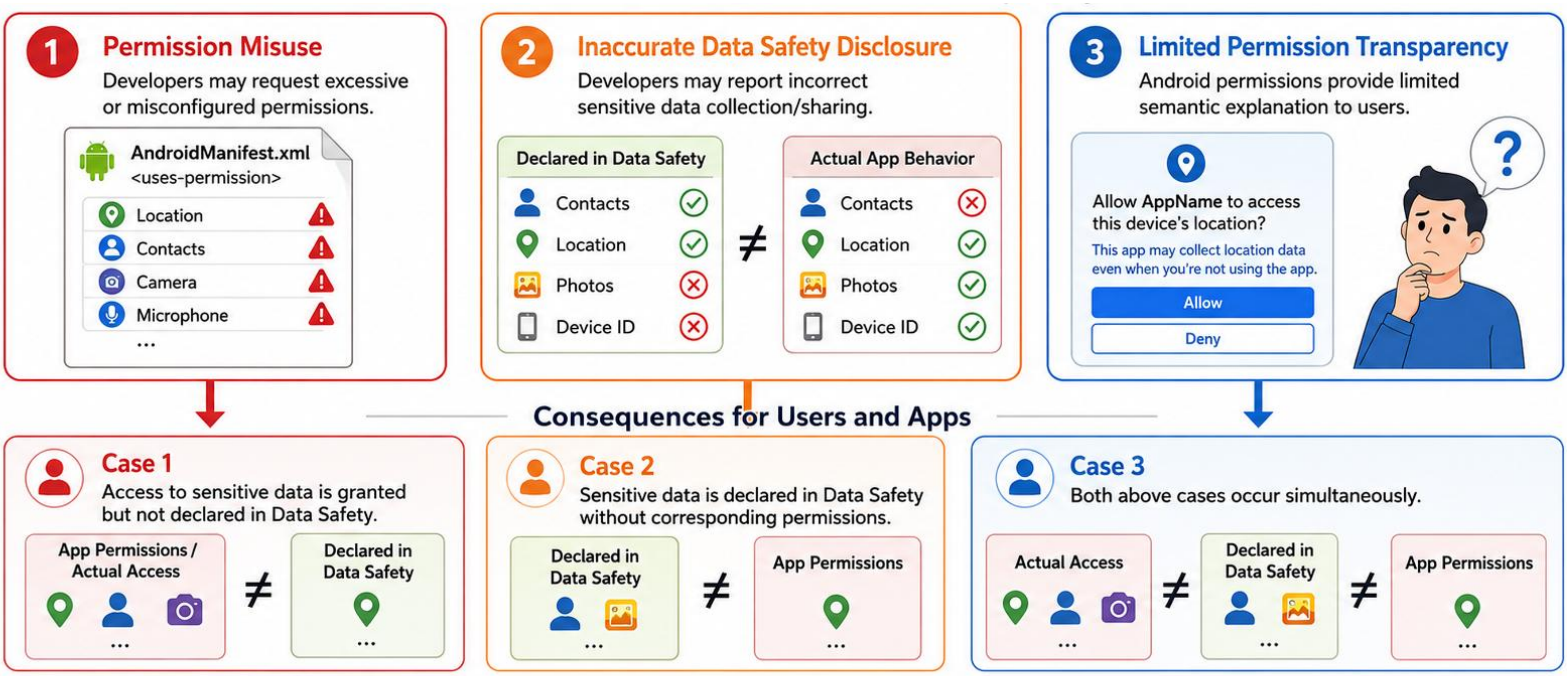
Permission model (PM)

- Developer selected permissions; users can only accept or deny.
- Same sensitive data, different permissions, different meanings.
- *COARSE_LOCATION*: *700-1000m*
- *FINE_LOCATION*: *3-5m*
- No explanation is provided when users grant permissions, so they may not fully understand what they have allowed.

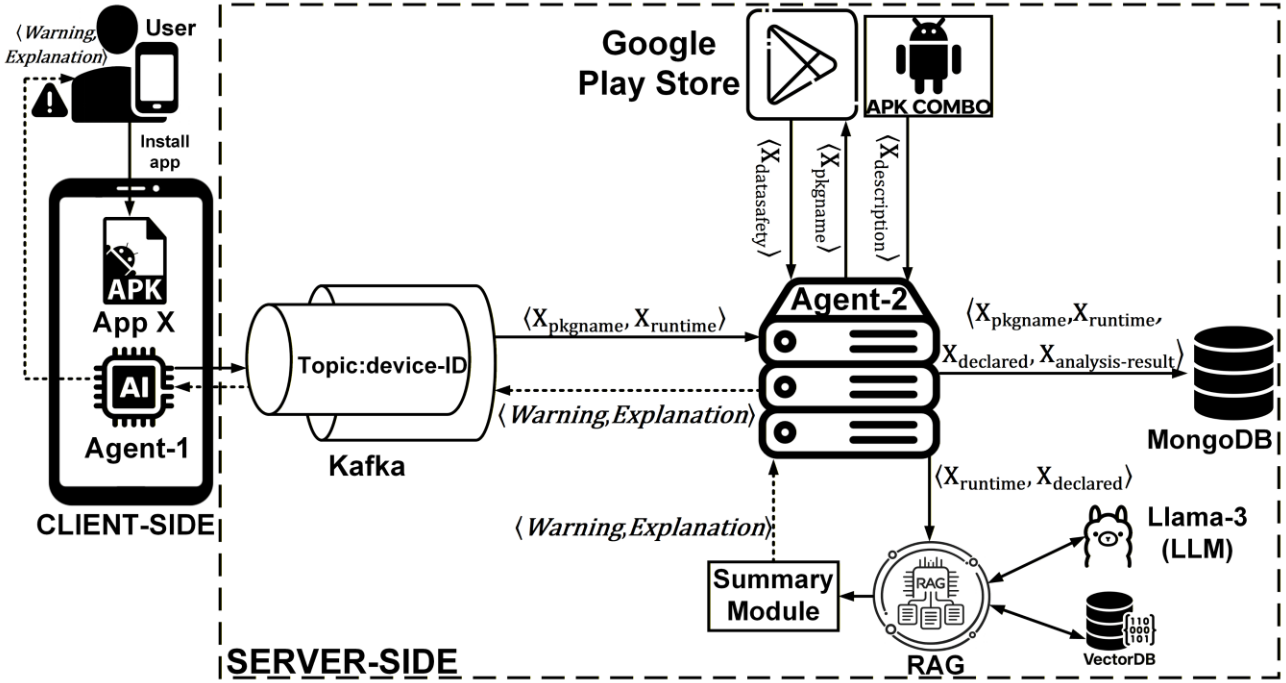
Data Safety (DS)

- DS is self-declared by developers without verification mechanism, so it may not reflect the app's actual behavior.
- No Data Safety notice is shown during app installation; users must search for it on Google Play.

1. Introduction: Threat Model



2. Methodology: Workflow



PrivacyAssist has two agents

Agent-1: Monitoring stage

→ Agent-1 runs on the user's phone to detect newly installed apps X ($X_{pkgname}$) and collect the list of runtime permissions granted by the user ($X_{runtime}$).

Then Agent-1 sends < $X_{pkgname}, X_{runtime}$ > to Agent 2.

- **Agent-2: Analysis stage**

→ Agent-2 uses $X_{pkgname}$ to retrieve app's Data Safety ($X_{datasafety}$) and app's description ($X_{description}$).

$X_{datasafety}, X_{description}$ **&** $X_{runtime}$ **is sent to LLMs to** detects inconsistencies in runtime permission & data safety → analysis result (i.e., X_{result})

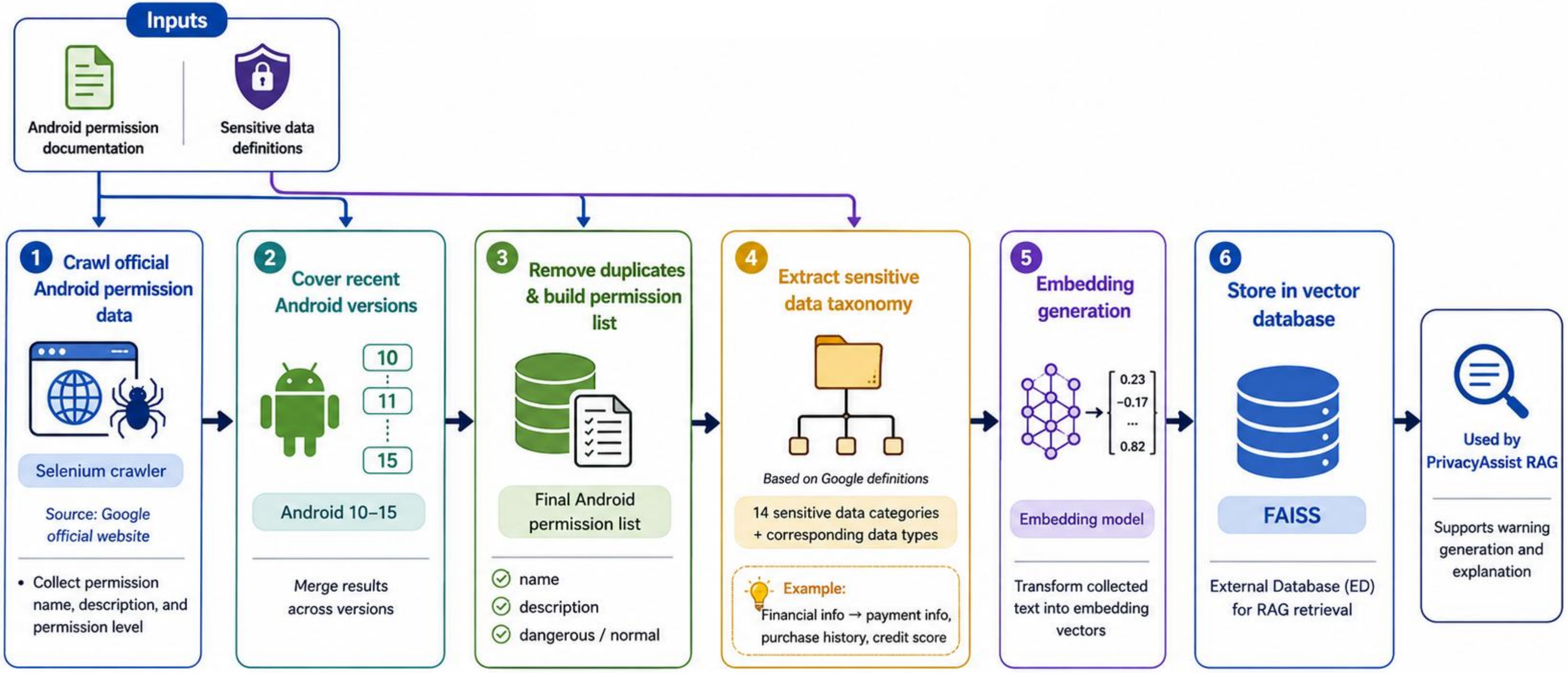
- **Warning and explanation stage:**

→ X_{result} is summarized by the summarization module because LLM responses are often lengthy → warnings and explanations → displayed on the user's phone.

2. Methodology: Challenge

- PrivacyAssist built on LLM-based agents → **hallucination**.
- Training data for LLM may be outdated while Android continues to update.

→ **Retrieval-Augmented Generation (RAG)**



2. Methodology: Illustrative example

Example: PrivacyAssist Analysis of Facebook (FB)

Facebook

FB_runtime

READ_CALENDAR, WRITE_CALENDAR, READ_CALL_LOG, CAMERA, READ_CONTACTS, WRITE_CONTACTS, READ_EXTERNAL_STORAGE, ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION, RECORD_AUDIO, BLUETOOTH_CONNECT, BLUETOOTH, BLUETOOTH_ADMIN, READ_PHONE_STATE, READ_BASIC_PHONE_STATE, READ_PHONE_NUMBERS, CALL_PHONE, ANSWER_PHONE_CALLS, MANAGE_OWN_CALLS

FB_datasafety

Files and docs; App activity; Audio; Photos and videos; Health and fitness; Personal info; Web browsing; App info and performance; Calendar; Financial info; Contacts; Messages; Device or other IDs; Location

FB_description

Facebook is a social networking app for discovering content (personalized Feed, Reels, search, Marketplace, Meta AI), joining communities, messaging friends, making voice/video calls, and creating/sharing posts, stories, and stickers/images.

Matched mappings

- READ_CALENDAR, WRITE_CALENDAR (Calendar ✓)
- READ_CONTACTS, WRITE_CONTACTS (Contacts ✓)
- ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION (Location ✓)
- RECORD_AUDIO (Audio ✓)
- CAMERA (Photos and videos ✓)
- READ_EXTERNAL_STORAGE (Photos and videos / Files and docs ✓)
- READ_PHONE_STATE, READ_BASIC_PHONE_STATE, READ_PHONE_NUMBERS, BLUETOOTH_CONNECT, BLUETOOTH, BLUETOOTH_ADMIN (Device or other IDs ✓)

Detected inconsistencies

1 Case 1

Runtime-granted permissions		Declared in Data Safety
READ_CALL_LOG, CALL_PHONE, ANSWER_PHONE_CALLS, MANAGE_OWN_CALLS	≠ Not declared	No declared Data Safety category: *Call logs*

2 Case 2

Declared in Data Safety		Corresponding granted runtime permissions
Web browsing; App activity; App info and performance; Messages; Personal info; Financial info; Health and fitness	≠ Not supported by runtime permissions	No corresponding granted runtime permissions

Potential over-collection

The app belongs to the *Social* category, and its core features do not relate to collecting *Web browsing* and *Health and fitness*, which may increase the risk of over-collection.

Warning & explanation for user

- The app can collect your call history, but the app developer does not mention it in the privacy information shown on Google Play.
- The app states that it collects data such as your browsing activity, app usage details, messages, personal information, financial information, and health data, even though it has not received your consent to access this information on your device.
- The app may over-collect because web browsing, financial, and health data may not relate to its *Social* features.

3. Data Set & Results

Dataset

Dataset Construction

200 participants → 15 selected apps each → 3,000 app selections → duplicate removal → 2,347 unique apps

Participant Profile

30.8%
18–24 years

23.1%
45–54 years

19.2%
25–34 years

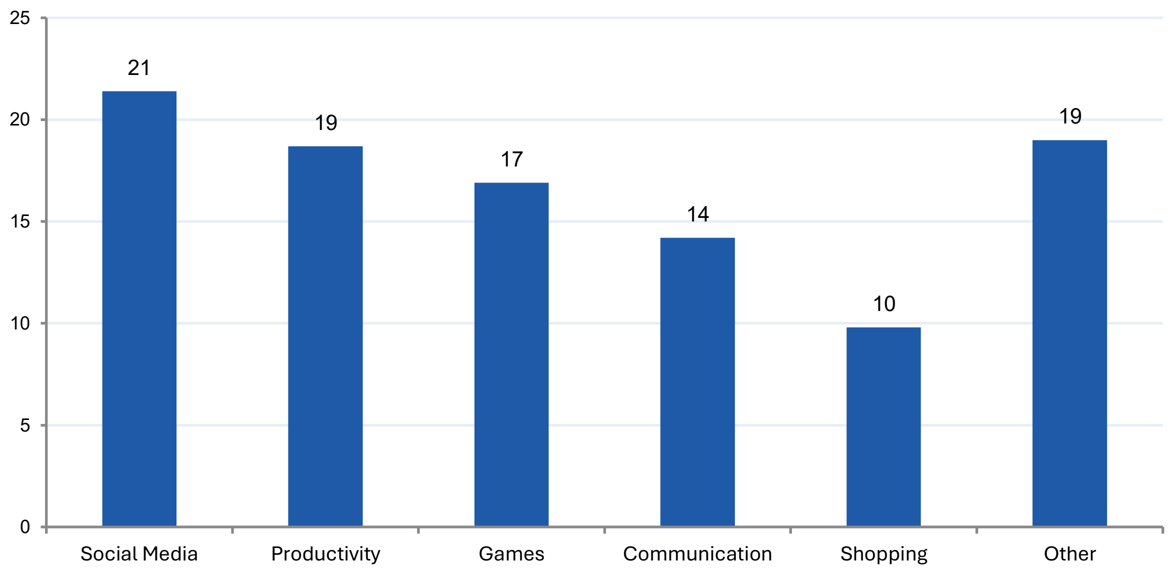
15.4%
≥55 years

46.2%
basic security knowledge

38.5%
intermediate knowledge

15.4%
advanced knowledge

App Dataset Composition



Popularity

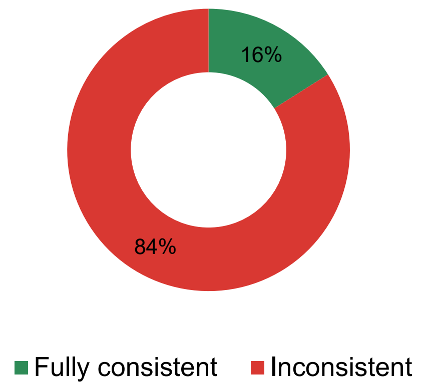
62.5%
apps have >100,000 downloads

28.3%
apps exceed 1 million downloads

3. Data Set & Results

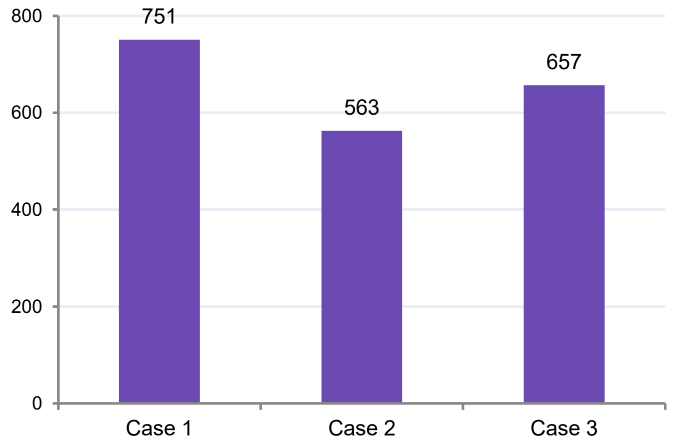
Experimental Results

Consistency Analysis



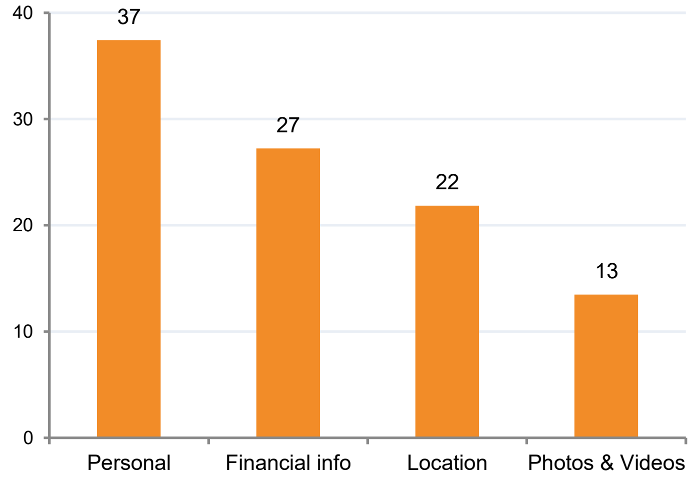
Only 376/2347 apps (16.02%) fully match granted permissions and Data Safety declarations; 1,971/2347 apps (83.98%) show inconsistencies.

Inconsistency Breakdown



Case 1: (751) granted PM but not declared DS
Case 2: (563) declared DS without corresponding permission
Case 3: (657) both cases occur

Most Omitted Sensitive Data Categories

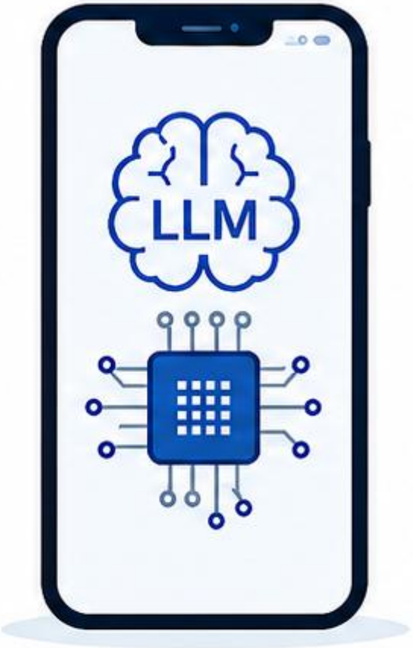


Validation, RAG Impact, and User Acceptance

90%	0	~35%	59.6%	73.1%	1.8%	10%	15 sec
LLM-agent accuracy on 100 manually checked apps	missed manually identified critical issues	false positives without RAG	Rated PrivacyAssist very useful	would cancel installation after warning	average Agent-1 CPU usage	battery over 24 hours	cached warning generation

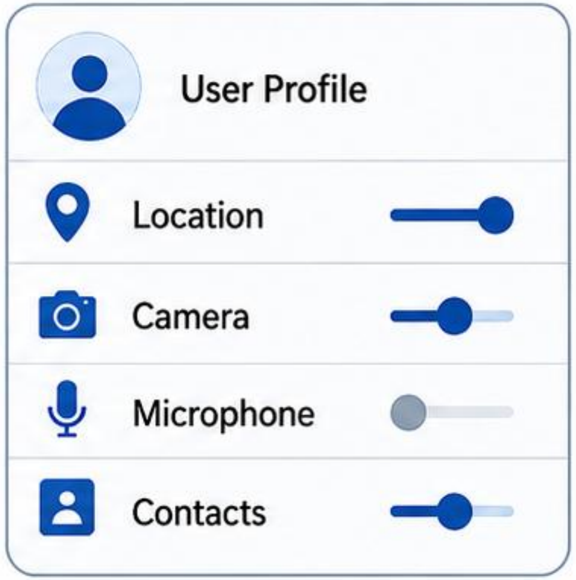
4. Future Works

1 On-device LLM



- Embed the LLM directly on Android devices
- Reduce off-device transmission of sensitive data
- Strengthen privacy guarantees

2 Adaptive Permissions



- Enhance the runtime permission mechanism
- Adapt permissions to individual user preferences


THANK YOU!