CS 528: Mobile and Ubiquitous Computing

Lecture 10b: Security and Internet of Things (IoT)

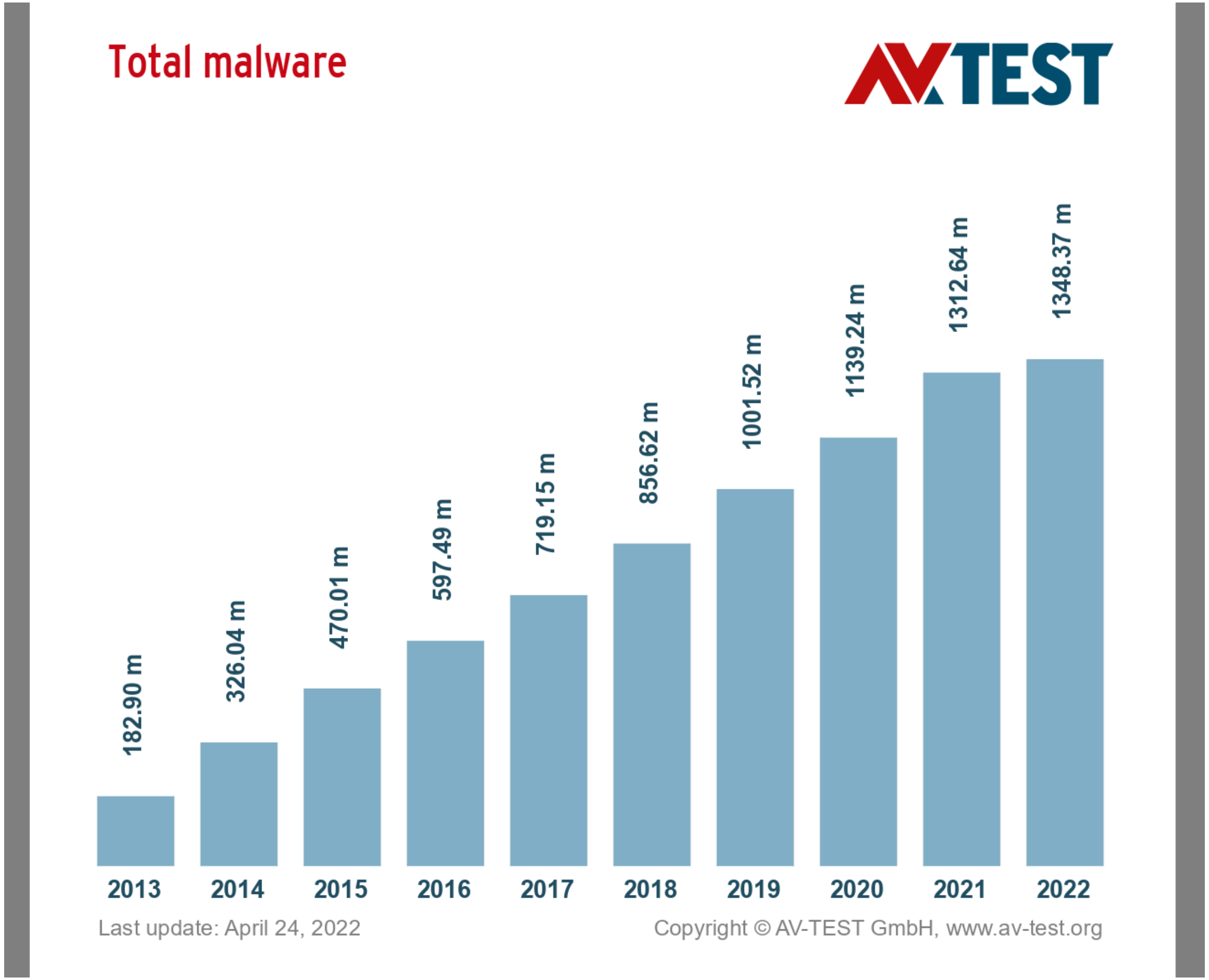
Emmanuel Agu

Mobile Security Issues

Introduction

- Millions of mobile apps
- Access to web, personal information, social media, etc
- Security problems (not previously envisaged) have resulted
- Examples:
  - Malicious apps can steal your private information (credit card information, etc)
  - Jogging map generated from paths of Fitbit users can expose locations/behavioral habits of users. E.g. US soldiers at German base
  - Malware can lock your phone till you pay some money (ransomware)
- Users/developers need better understanding of mobile security

Growth of Malware



Ref: https://www.digitalinformationworld.com/2022/04/alarming-numbers-of-malware-are-being.html

Android Security Model

Android Security: Features

1. **Application Isolation:**
   - Application sandboxing: App 1 cannot interact directly with app 2
   - Apps can only communicate using secure inter-process communication (Intent)

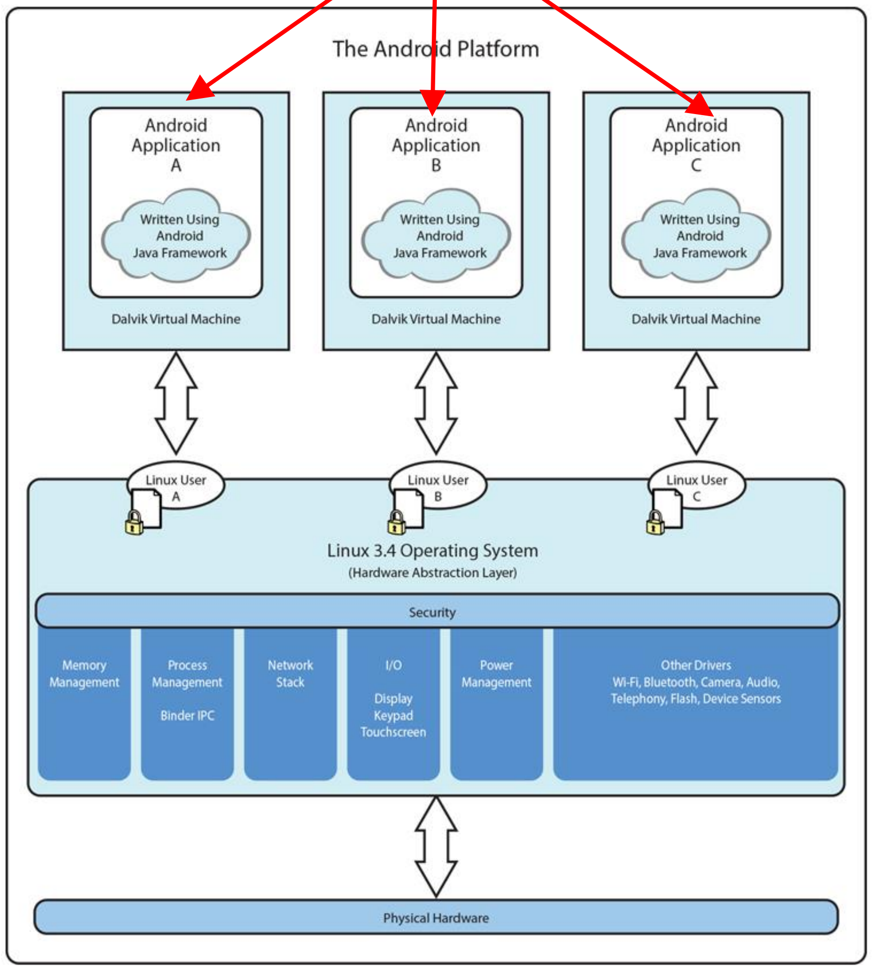
2. **Permission Requirement:**
   - Apps need permission to use certain hardware, resources, perform certain actions

3. **Encryption:** All user-created data automatically encrypted before storage on disk

4. **App signing:** Every Android app must be signed by developer, ensures future updates authentic

5. **Authentication:** either using password, fingerprint or biometrics

Apps are isolated from each other



Ref: Introduction to Android Programming, Annuzzi, Darcey & Conder

Recall: Android Software Framework

- Each Android app runs in its own security sandbox (VM, minimizes complete system crashes)
- Android OS multi-user Linux system
- Each app is a different user (assigned unique Linux ID)
- Access control: only process with the app's user ID can access its files
- Apps talk to each other only via intents, IPC or ContentProviders

Malware Evolution

Threat Types: Malware, Grayware & Personal Spyware

- **Malware:**
  - Gains access to a mobile device in order to steal data, damage device, or annoying the user, etc. **Malicious!!**
- **Personal Spyware:**
  - Collects user's personal information over of time
  - Sends information to app **installer** instead of author
  - E.g. spouse may install personal spyware to get info
- **Grayware:**
  - Collect data on user, but with no intention to harm user
  - E.g. for marketing, user profiling by a company

Mobile Malware Survey (*Felt et al*)

Mobile Malware Study?

A survey of mobile malware in the wild Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner in Proc SPSM 2011

- First major mobile malware study in 2011 by Andrienne Porter Felt *et al*
  - Prior studies focused on PC malware
  - Provided definitions, foundations for defenses today

- Analyzed 46 malwares that spread Jan. 2009 – June 2011
  - 18 – Android
  - 4 – iOS
  - 24 – Symbian (now discontinued)

- Analyzed malware:
  - in databases maintained by anti-virus companies
    - E.g., Symantec, F-Secure, Fortiguard, Lookout, and Panda Security
  - Discovered by mentions in news sources
- Just analyzed malware. Did not analyze spyware and grayware

Categorized Apps based on Behaviors

1. **Novelty and amusement**
   - Causes minor damage
   - E.g. Change user's wallpaper

2. **Selling user information**
   - Malware obtains user's personal information via API calls
     - E.g. User's location, contacts, download + browser history/preferences

   - Information can be sold to advertisers
     - E.g. Dunkin Donuts may want to know users who visit their competitors
     - 2011 price: $1.90 to $9.50 per user per month

Categorized Apps based on Behaviors

3. **Stealing user credentials**
   - People use smartphones for activities that require their passwords and payment information. E.g. shopping, banking, e-mail
   - Malwares can log keys typed by user (keylogging), scan their documents for username + password

   - User credentials (username, password) can be sold
   - In 2008, black market price of:
     - Bank account credentials: $10 to $1, 000,
     - Credit card numbers: $.10 to $25,
     - E-mail account passwords: $4 to $30

Categorized Apps based on Behaviors

4. **Make premium-rate calls and SMS**
    - Premium rate texts to specific numbers are expensive (E.g. 1-900.. Numbers)
    - Attacker can set up premium rate number, Malware sends SMS there
    - User is billed by their cell carrier (e.g. sprint), attacker makes money

5. **SMS spam**
    - Used for commercial advertising and phishing
    - Sending spam email is illegal in most countries
    - Attacker uses malware app on user's phone to send SPAM email
    - Harder to track down senders

Categorized Apps based on Behaviors

6. **Search Engine Optimization (SEO):**
    - Malware makes HTTP requests for specific pages to increase their search ranking (e.g. on Google)
    - Increases popularity of requested websites

7. **Ransomware**
    - Possess device, e.g. lock screen till money is paid
    - *Kenzero* – Japanese virus inserted into pornographic games distributed on P2P networks
        - Publishes user's browser history on public website
        - Asked **5800 Yen** (~$60) to delete information from website
        - About 12 % of users (661 out of 5510) actually paid

Frequency of Malware Categories

A survey of mobile malware in the wild Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner in Proc SPSM 2011

Exfiltrates user information	28
Premium calls or SMS	24
Sends SMS advertisement spam	8
Novelty and amusement	6
Exfiltrates user credentials	4
Search engine optimization	1
Ransom	1

Table 1: We classify 46 pieces of malware by behavior. Some samples exhibit more than one behavior, and every piece of malware exhibits at least one.

Malware Detection based on Permissions

- Malware request more permissions!!
- Analyzed permissions of 11 Android malware

- **Findings: Yes!**
  - 8 of 11 malware request SMS permission (73%)
    - Only 4% of non-malicious apps ask for this
  - Dangerous permissions: requests for personal info (e.g. contacts), etc
  - Malware requests 6.18 dangerous permissions
    - 3.46 for Non-malicious apps

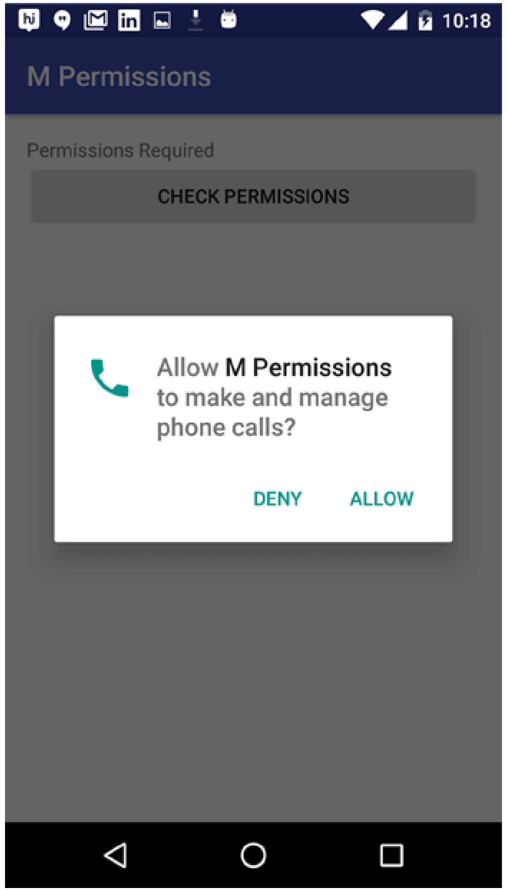
Number of Dangerous permissions	Number of non-malicious applications		Number of malicious applications
0	75	(8%)	-
1	154	(16%)	1
2	182	(19%)	1
3	152	(16%)	-
4	140	(15%)	2
5	82	(9%)	1
6	65	(7%)	-
7	28	(3%)	2
8	19	(2%)	1
9	21	(2%)	1
10	10	(1%)	1
11	6	(0.6%)	1
12	7	(0.7%)	-
13	4	(0.4%)	-
14	4	(0.4%)	-
15	2	(0.2%)	-
16	1	(0.1%)	-
17	1	(0.1%)	-
18	-		-
19	-		-
20	1	(0.1%)	-
21	-		-
22	-		-
23	1	(0.1%)	-
24	-		-
25	-		-
26	1	(0.1%)	-

Table 2: The number of "Dangerous" Android permissions requested by 11 pieces of malware and 956 non-malicious applications [28].

Android Run-Time Permissions Changed in Marshmallow (Android 6.0)

- Pre Android 6.0: Permissions during install
- Android 6.0: Changes!!
- “Normal” permissions don’t require user consent
  - E.g. change timezone
  - Normal permissions can do very little to harm user
  - Automatically granted
- Dangerous permissions (e.g. access to contacts can harm user

- Android 6.0: Run-time permissions now required for “dangerous” permissions

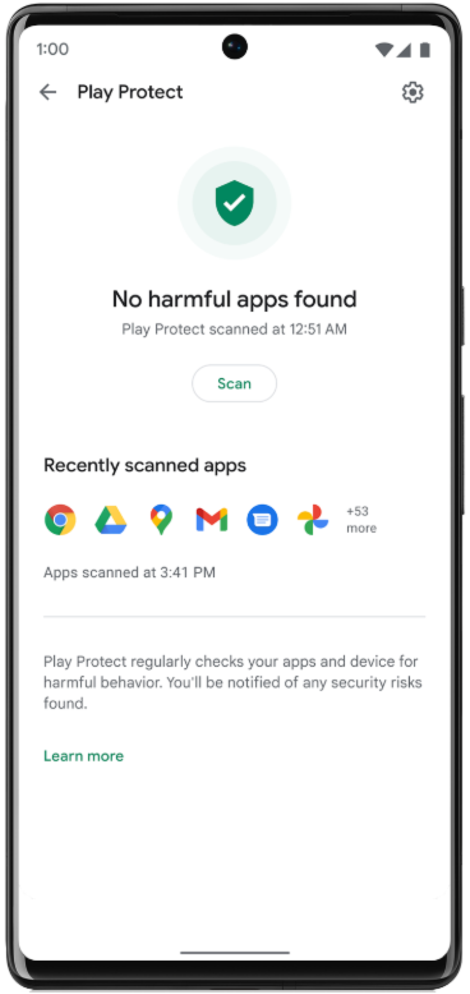


Google Play Protect

Google Play Protect

https://security.googleblog.com/2023/10/enhanced-google-play-protect-real-time.html
https://www.airdroid.com/mdm/google-play-protect/

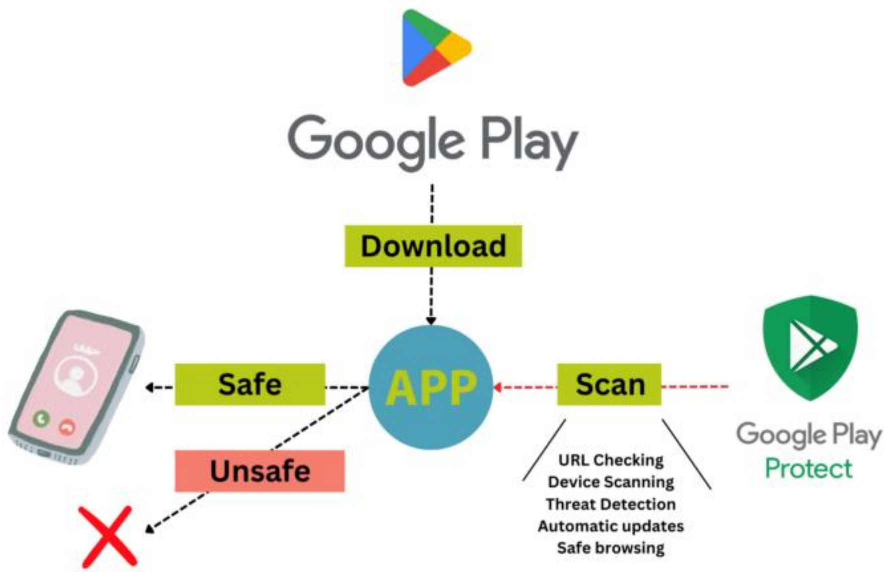
- Automatically scans device, detects malicious apps
  - **Daily PHA scan:** Scans 125 billion apps daily regardless download source repository
  - **On-demand PHA scan:** User initiated can initiate full-device scan
  - **Offline PHA scan:** Scan for well-known PHAs when trying to install apps offline
- Blocks 300 million Potentially Harmful Applications (PHAs) annually
  - PHAs classified as harmful are automatically remove
  - PHAs classified as less harmful are disabled, user notified to decide to enable



Google Play Protect: How it works

https://security.googleblog.com/2023/10/enhanced-google-play-protect-real-time.html
https://www.airdroid.com/mdm/google-play-protect/

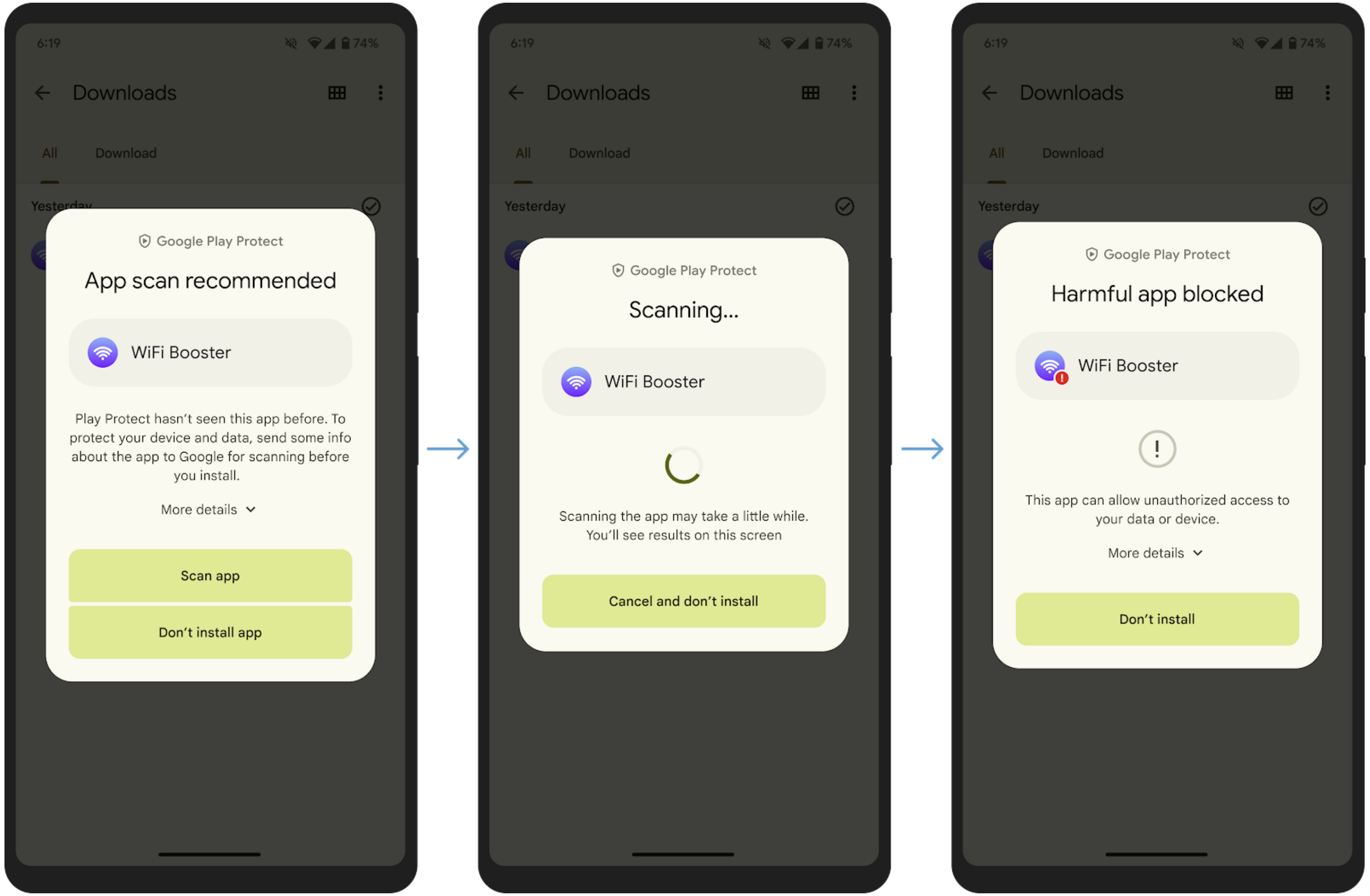
- Scans device to detect Potentially Harmful Applications (PHAs) based on
  - App's behavior
  - Permissions (types and number), etc.
  - Comparisons with similar apps, its catalog of malicious apps
  - Static as well as code-level scanning (As of October 2023)
- During install, scans for PHAs downloaded from Google Play or any source



Google Play Protect: How it works

https://security.googleblog.com/2023/10/enhanced-google-play-protect-real-time.html
https://www.airdroid.com/mdm/google-play-protect/

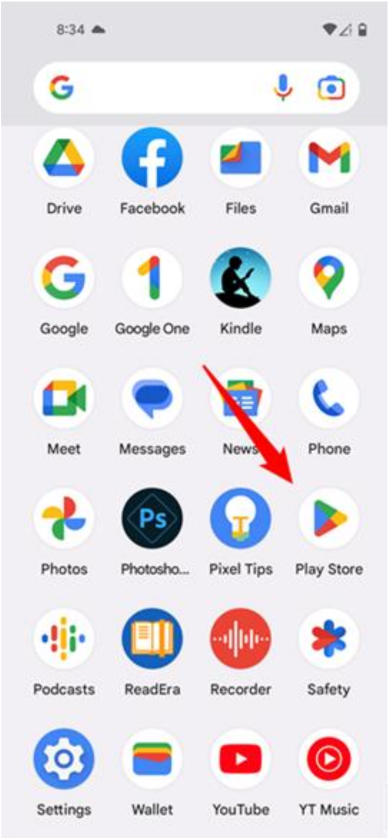
- If PHA detected, either
  - Block PHA installation
  - Notify user to remove PHA if already installed



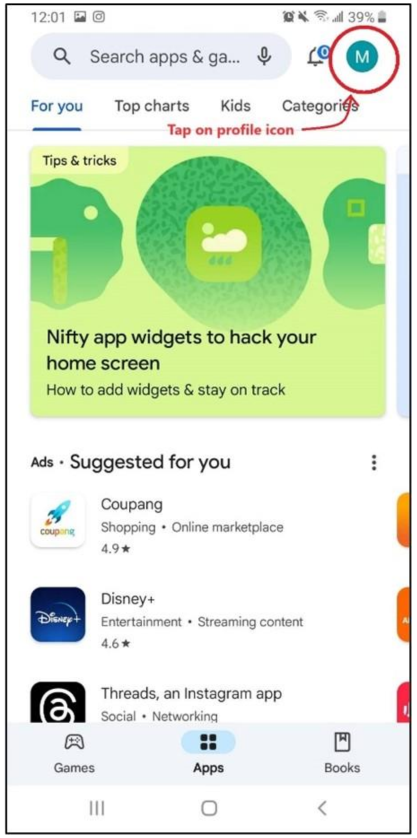
Google Play Protect: How to Turn on/off

https://security.googleblog.com/2023/10/enhanced-google-play-protect-real-time.html
https://www.airdroid.com/mdm/google-play-protect/

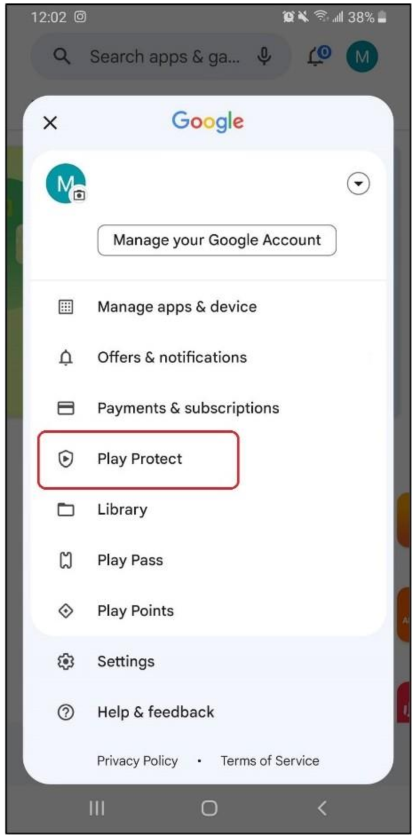
- Turned on by default
- Can access it, turn it off/on using Google PlayStore app



Open Google Play Protect app



Tap profile icon



Select Play Protect on menu options

Google Play Protect: Success Rate

https://security.googleblog.com/2023/10/enhanced-google-play-protect-real-time.html
https://www.airdroid.com/mdm/google-play-protect/

- Google Play Protect is not perfect
  - 93% of PHAs detected
  - In independent tests by AV-TEST, 80.7% of real-time threats were detected
  - Detected only 90% of live viruses compared to 100% for Norton 360 and Bitdefender
  - Higher false positive rate (tag harmless apps as malicious) than competitors (MalwareBytes, Bitdefender),
    - 12 vs 1 in one study

Android Analysis Tools

Analyzing Android Apps

- Attacker can use analysis tools to get more information about an Android app
- **Source code recovery:** generate app source code from executable
- **Static analysis (binaries or source code):** Understand app design without running it.
  - Examine application logic, flow, APIs used
- **Dynamic analysis:** Observe how app executes
  - App memory usage, network usage, response time, performance, etc
- Many available (open source?) tools for all of the above!

Android Analysis Tools

- APKinspector
- Androguard
- ApkTool
- Appknox.
- CharlesProxy
- ClassyShark
- DeGuard
- DevKnox
- Dex2Jar.

- **Scary!!**

Internet of Things (IoT)

Recall: IoT: Definitions

- New technology paradigm
- Internet extended to connect Physical Devices
- Physical devices contain sensors
- Internetworked smart machines and devices can
  - Interacting with each other
  - Exchanging information
  - Can be controlled over the Internet

Lee, I. and Lee, K., 2015. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons, 58(4), pp.431-440.

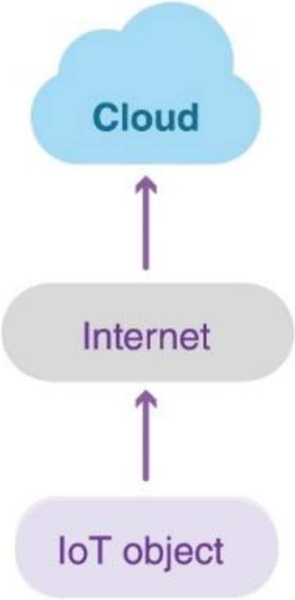
Recall: IoT: Networked Smart Devices

- Smart devices: can be accessed, controlled over the network

Smart Fridge
- See groceries in fridge from anywhere on companion app

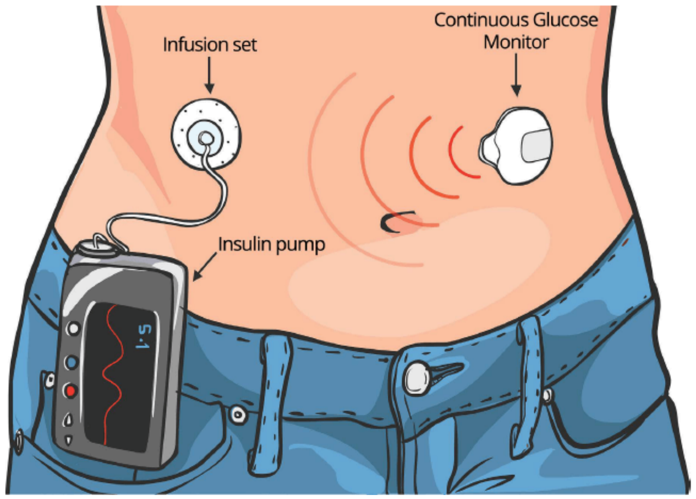
IoT Ecosystem of Smart Devices: Scenarios

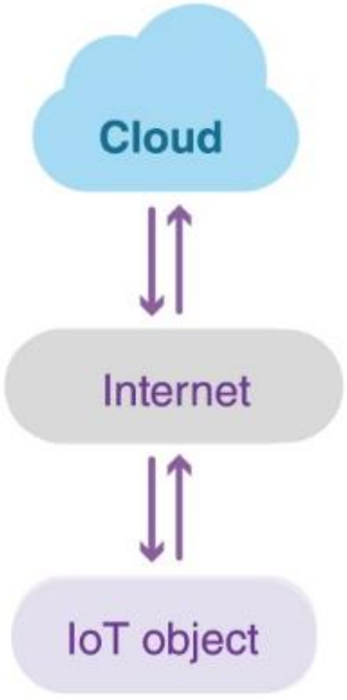
- 3 usage scenarios
- Scenario 1: Data from smart device gathered, **analyzed non-real time**, to gain intelligence
- Example:
  - Visitors in museum wear IoT wristbands with location tracking
  - Data from tracker pushed to cloud, analyzed later, non-real time
  - Determine: which paintings are most, least popular



IoT Ecosystem of Smart Devices: Scenarios

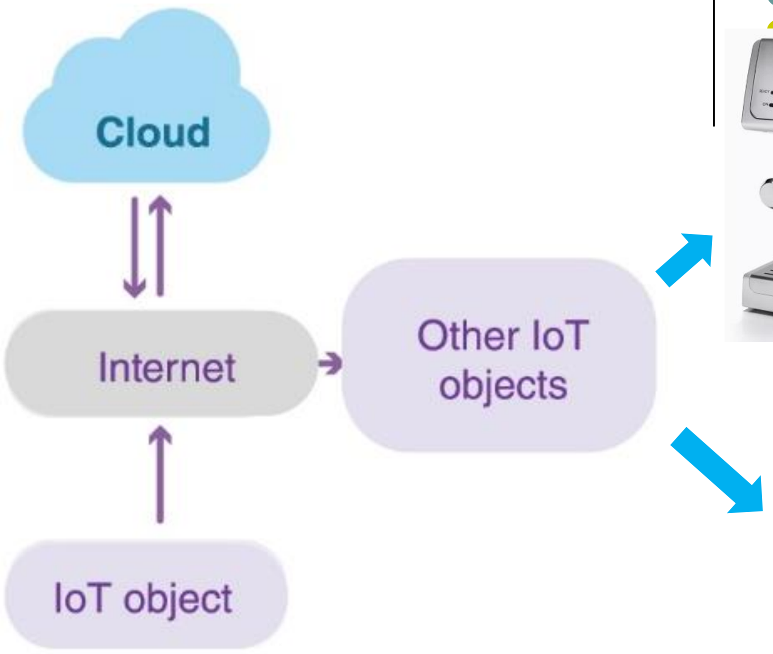
- Scenario 2: Data from smart device gathered, **analyzed in real time**, to inform intelligent action on **same device**
- Example:
  - Patient wearing IoT medical device, transmits blood glucose to cloud
  - Patient blood glucose response to foods, activities monitored real time
  - Real-time Analyses can inform immediate action: command to insulin pump (in IoT medical device) to regulate insulin

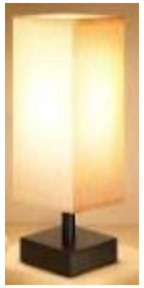




IoT Ecosystem of Smart Devices: Scenarios

- Scenario 3: Data from smart device gathered, **analyzed in real time**, to inform intelligent action on **other device**
- Example:
  - Patient wearing wearable device that transmits information about body positions, movements to cloud
  - Real time analyses of patients sleep patterns
  - If person usually wakes up, turns on light and coffee maker
  - IoT system can automatically:
    - Detect user waking up
    - Send commands to light, coffee maker to turn on



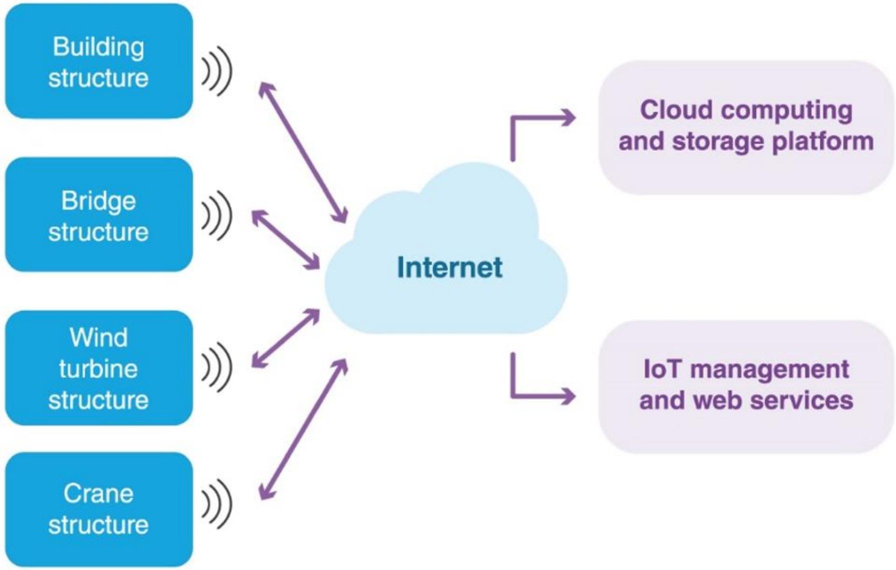


IoT: Verticals (Target Domains) and Use Cases

Verticals	Use cases
Energy	Oil/gas Renewable Smart grid
Agriculture	Greenhouse Precision farming Agricultural drones
Environment	Air pollution Water quality Chemical leakage
Automotive	Autonomous driving In-vehicle infotainment Electric cars
Industrial	Smart factory Predictive maintenance Smart robotics
Transportation	Vehicles Rail system Fleet management

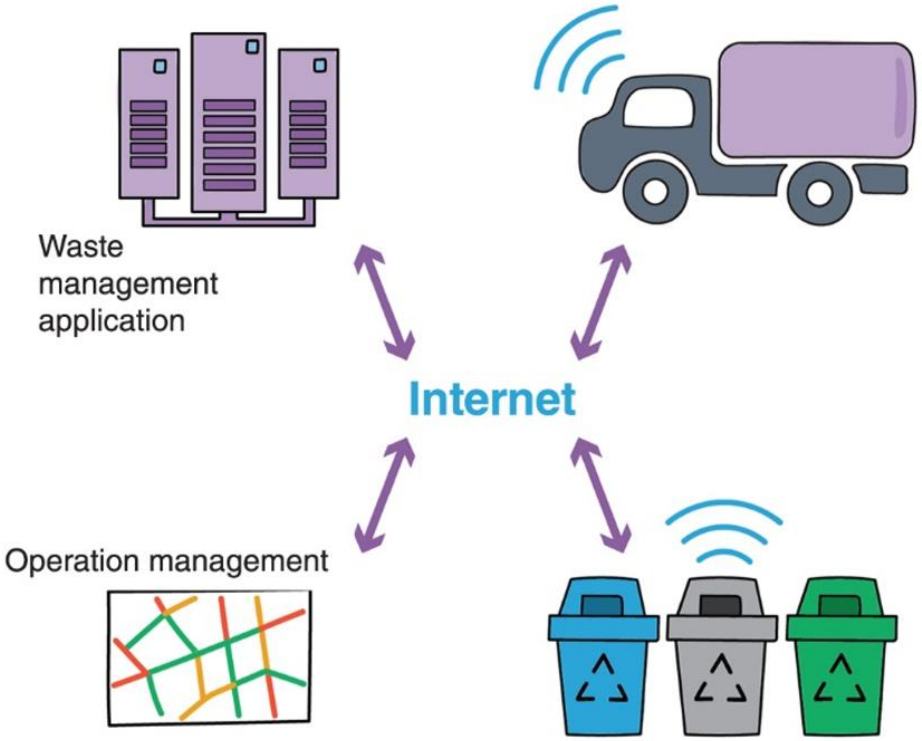
IoT Use Case: Structural Health Monitoring

- Monitor current health of structures (bridges, wind turbines, buildings)
- Predict future failures, collapse
- Requires installing sensors in structure, periodically sends data to cloud for analyses
    - **Sensors:** Strain gauges, accelerometers, crack detectors, tilt sensors, corrosion sensors



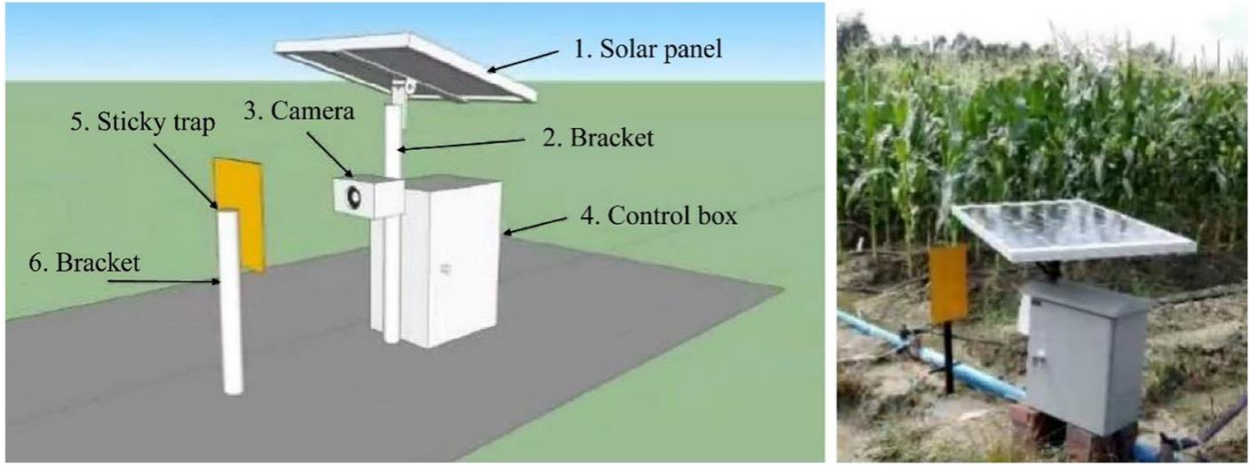
IoT Use Case: Waste Management

- Currently use scheduled waste/garbage pickup (e.g. weekly), archaic, outdated
- Weekly pickup: too often for some homes, not frequent enough for others
- IoT approach:
  - Install sensors in garbage bins
  - Notifies garbage collectors when full
- Can use data to decide areas that need more bins



IoT Use Case: Smart Agriculture

- Many current agricultural approaches can be improved using IoT
- Pest management:
  - **Current:** if farmer sees some pests, spray entire farm (crops + pests)
  - **IoT solution:** Use sensors, cameras to detect pests and exact location, then precision pesticide spraying
- Watering of plans:
  - **Current:** Water plants on a schedule (may be too much, or too little)
  - **IoT solution:** Use moisture sensors to detect dryness/moisture levels, water when needed

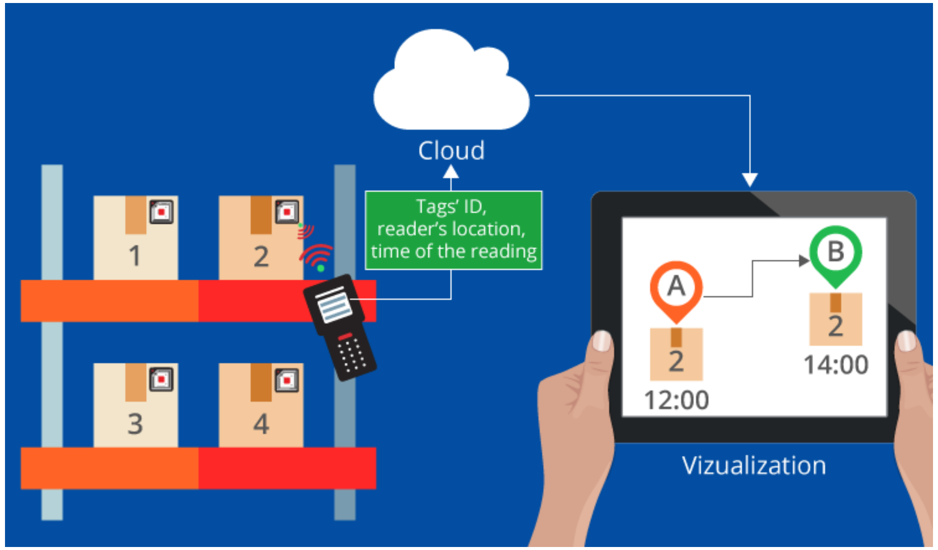


IoT pest detection system

Soil moisture sensor

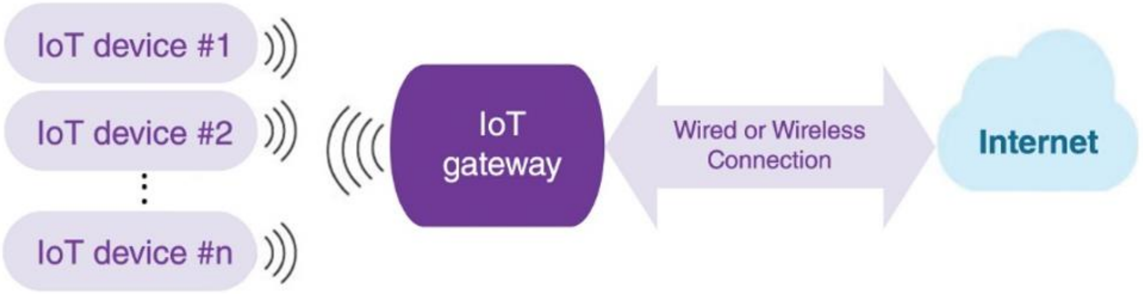
IoT Use Case: Inventory Management

- Current inventory management: Manually check warehouse, shelves overnight
  - Only 65% accurate
  - Products either overstocked or out-of-stock => Loss of revenue
  - Some reasons: customers buying things online, then returning to store
- IoT Solution: Install sensors (RFID tags) on boxes of goods, or using shelves with sensors (smart shelves)
- Can improve analytics: more accurate estimate of time on shelf, predict demand

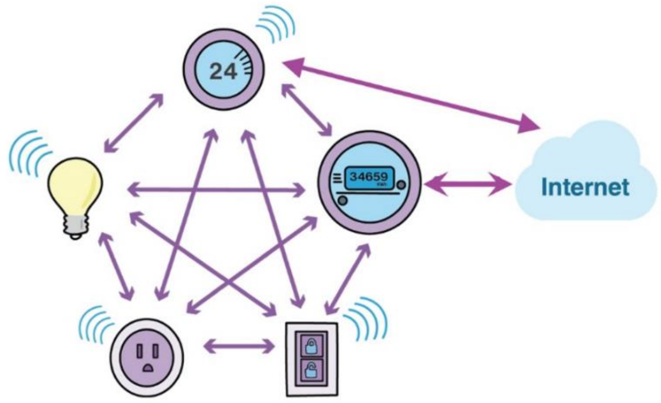


IoT Connectivity

- IoT needs:
  - Low power devices, transmission (reduce frequency of changing batteries)
  - Long range transmission. E.g. Sensor on farm can transmit data over long distances
- But these 2 needs oppose each other:
  - Longer range transmission consumes more power
- **IoT solutions:**
  - **IoT Mesh Network:** Multiple cooperating devices, multiple hops, some devices Internet gateways
  - Cellular network
  - Satellite



IoT Conceptual Connectivity Diagram



IoT Mesh Network of Multiple Cooperating Devices

IoT Traffic

- IoT traffic patterns differ from human-generated traffic patterns in terms of
  - Traffic volume over course of a day
  - Traffic volume on up vs. down link
  - Transmission rate

Traffic generated by humans	Traffic generated by IoT devices
Higher traffic during the day compared to the night	Almost uniform traffic all the time
Higher downlink traffic compared to the uplink	Substantially higher or lower traffic as compared to the traffic generated by humans
High data transmission rate	Low or high data transmission rate

IoT Standards

- IoT needs multiple devices to interoperate, communicate, trust each other
- IoT Standards are needed
- Open Connectivity Foundation (OCF): industry organization, specifies standards, ensures interoperability
- OCF member companies: Intel, Cisco, Qualcomm, Samsung, Microsoft, Electrolux
- In 2018, OCF created IoTivity (http://iotivity.org/), open source software framework for seamless IoT device-to-device connectivity

References

- Dian, F.J., 2022. *Fundamentals of Internet of Things: For Students and Professionals*. John Wiley & Sons.