Automate all the things! Using Puma to automate UI actions in Android applications

Angelina Claij-Swart, Erik Oudsen

Netherlands Forensic Institute

The Hague, The Netherlands

Hansken*

*Not open source!

Why reference data?

Verify if our code works as expected

Generated > acquired

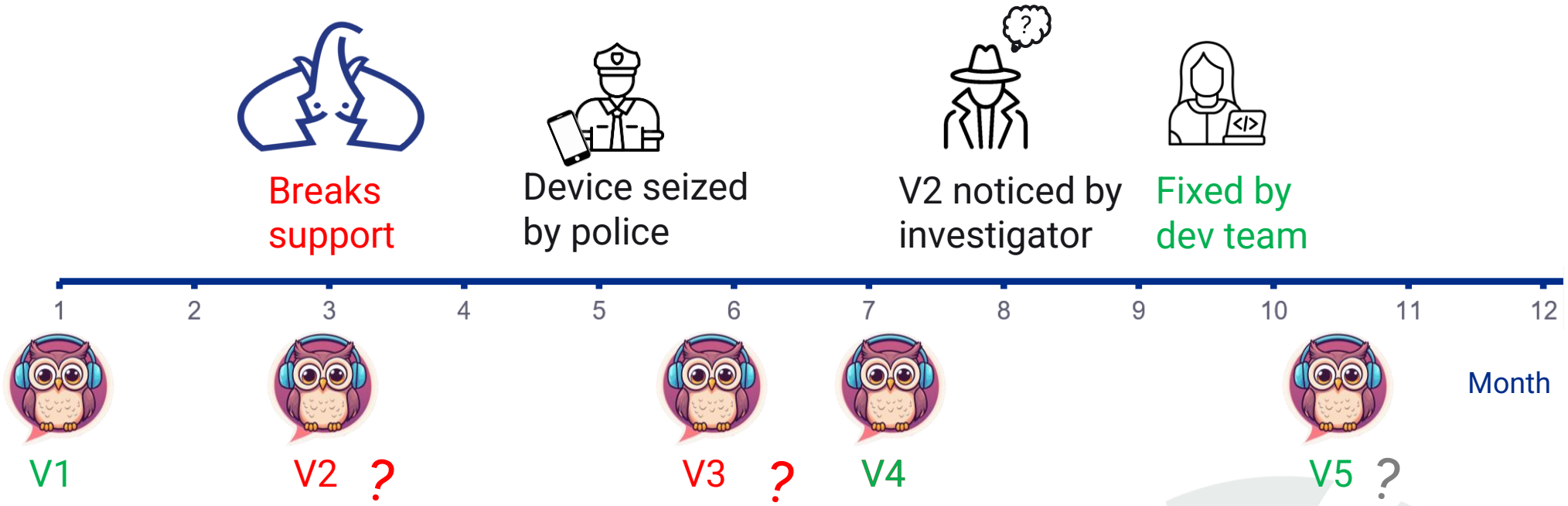
Why?

- Ground truth
- Completeness
- Privacy laws

Why not?

- Unforeseen edge cases
- Not “real-life”
- Laborious
  - Scalability
  - Updates

App updates



- 36% of top 1000 apps: weekly update
  - Average for WhatsApp: about 2 versions per week
- Old version might not be available anymore

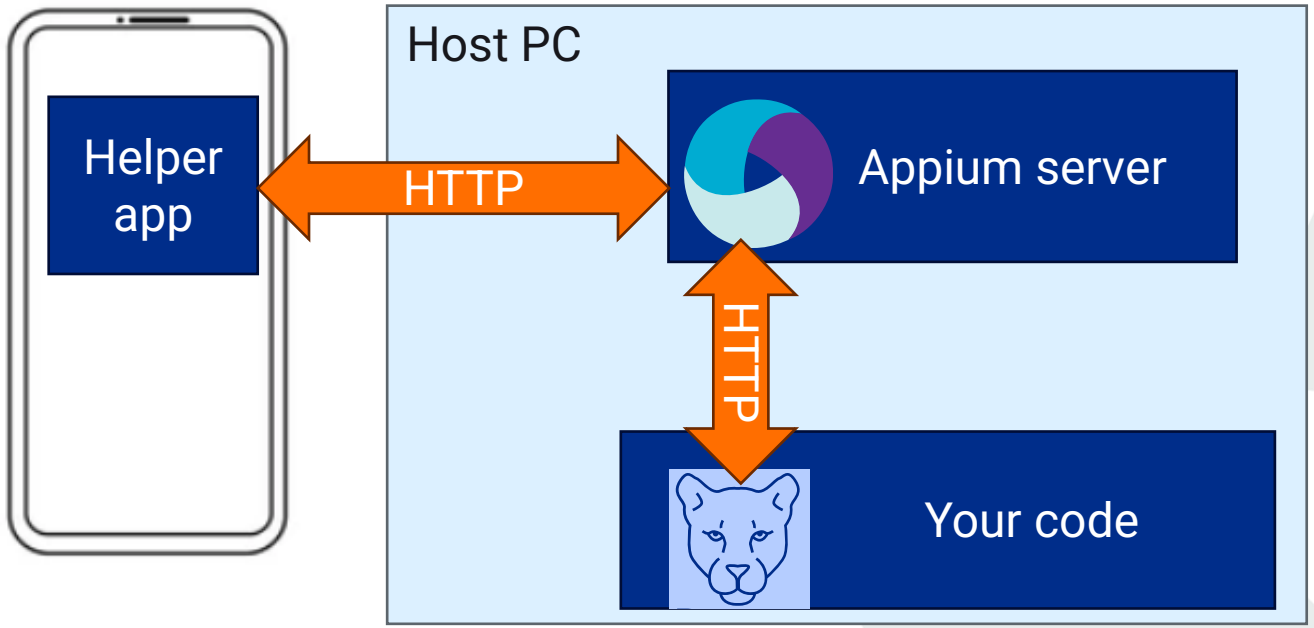
- Manual creation of reference data is too laborious, automation is needed

https://www.statista.com/statistics/1404434/google-play-store-app-updates-by-frequency/
Timeline inspired by: https://dfrws.org/wp-content/uploads/2025/03/DFRWS-EU-2025_Preservation-Spichiger-Adelstein.pdf#slide27

Programmable Utility for Mobile Automation

- Write reusable scenarios for reference data generation
- Python API: Integrate into your workflow
- Framework built on top of Appium

Puma



Appium

An open-source UI testing framework for mobile applications

- Works with production code
- No root access needed
- Works on Android and iOS

See what's on screen

Interact with screen content

Perform actions on device

Appium vs Puma

Sending a Telegram message with Appium:

```
from appium.webdriver import webdriver

driver = webdriver.Remote()
driver.activate_app('org.telegram')

chat = driver.find_element(f'//android.view.ViewGroup["Alice"]')
chat.click()

msg = driver.find_element('//android.widget.EditText[@text="Message"]')
msg.send_keys("Hello Alice!")

button = driver.find_element('//android.view.View[@content-desc="Send"]')
button.click()
```

1. Open app
2. Open chat
3. Insert message in textbox
4. Click send button

Appium vs Puma

Sending a Telegram message with Puma:

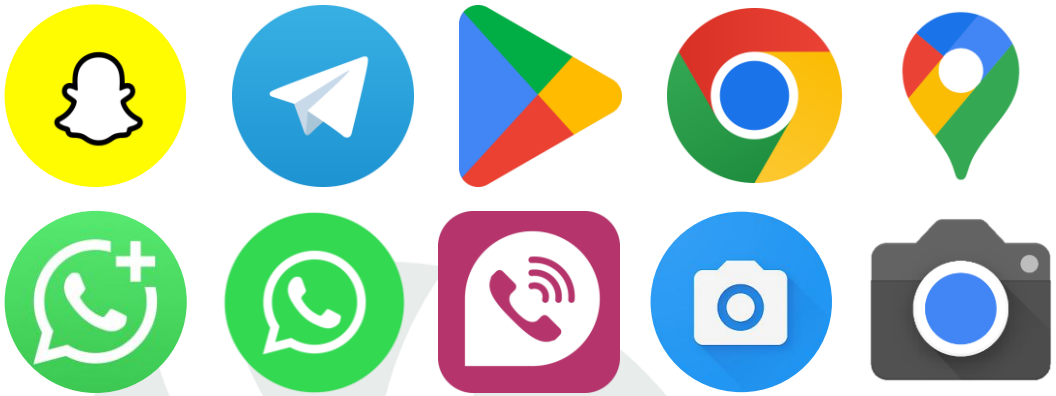
```
from puma.apps.android.telegram.telegram import Telegram

alice = Telegram("emulator-5554")
alice.send_message(message="Hi Bob, how are you?", chat="Bob")
```

Puma features

- Multiple app & device support
- Ground truth logging
- Location spoofing
- Popup handling: e.g. permissions
- Robust UI navigation

android

Supported apps:



Demo of a full scenario

Multiple phones

Multiple apps

Coherent narrative

```
bob_maps = GoogleMapsActions('34281JEHN03866')
bob_telegram = Telegram('34281JEHN03866')
charlie_telegram = Telegram('32131JEHN38079')

# send messages
bob_telegram.send_message( message: 'Hey Charlie!',conversation='Charlie')
bob_telegram.send_message("I'm heading to the office now")
charlie_telegram.send_message( message: 'Ok, see you soon!', conversation='Bob')

# start navigation
bob_maps.start_route(from_query="Schiphol Airport",
                     to_query="Laan van Ypenburg 6, Den Haag",
                     speed=50)
# send a picture from device
charlie_telegram.send_message('Bob, we might have a problem...')
charlie_telegram.send_media_from_gallery(media_index=1,
         caption="The servers don't look great, we need you here ASAP!")

# change speed
bob_telegram.driver.activate_app()
bob_maps.activate_app()
bob_maps.route_simulator.update_speed(180, variance_absolute=10)
```

Demo

Demo video can be found here:

https://archive.org/details/puma-demo-2025

Use case 1: Initial data population

Populate an application from scratch

Happens when

- Researching a new app
- Creating a new (large) dataset

Manual: multiple people, lots of time

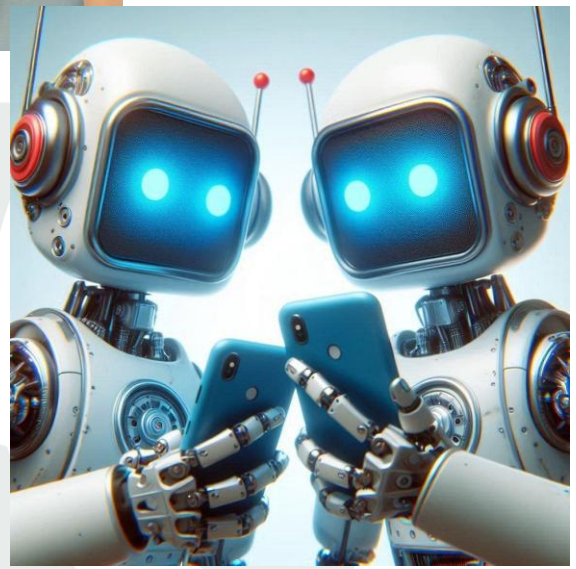
With Puma: run a script (and supervise)

Caveats:

New apps: adding support to Puma also takes time

UI changes can break Puma support

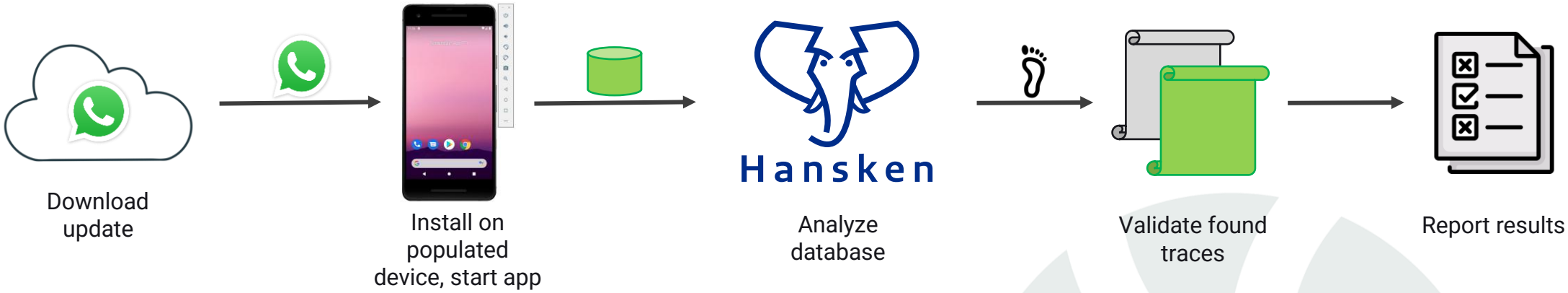
New accounts: only needed once, we haven't built this into Puma



Use case 2: monitor app updates

Use an app that was already populated

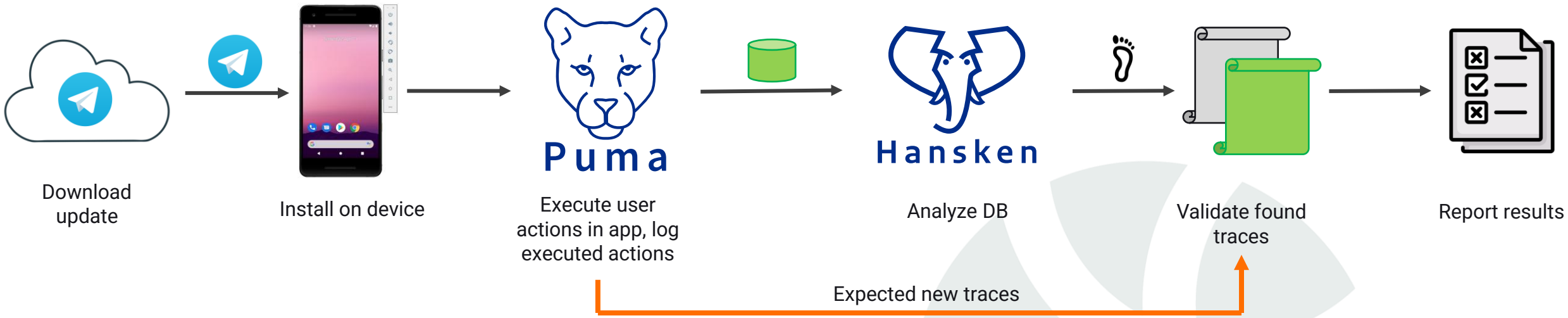
Pipeline to install updates and validate tools



Useful for apps that change datastructure on update, e.g. WhatsApp

Use case 3: monitor app updates with post-update UI interaction

Expand previous pipeline:



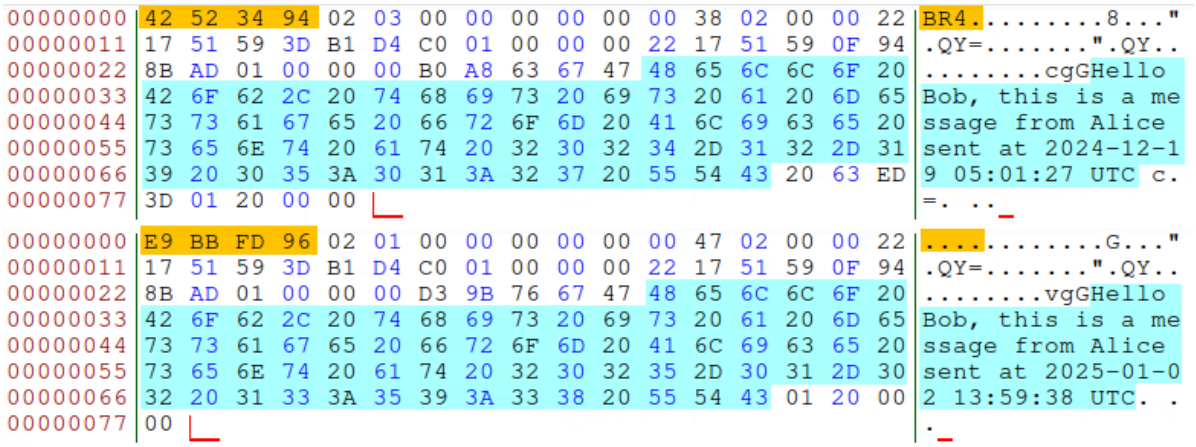
Useful for apps that only store new actions in new datastructure, e.g. Telegram

Unique insights thanks to automation

- Telegram updates some blobs every X.Y version
  - Example of a message, 11.5.5 vs 11.6.1

- WhatsApp schema is very stable:
  - 2 changes in 3 years
  - Minor change to absent timestamps
  - Major schema update

- Major schema update: staged rollout!
  - New tables appeared, remained empty
  - Data migration not tied to version, but account



Conclusion

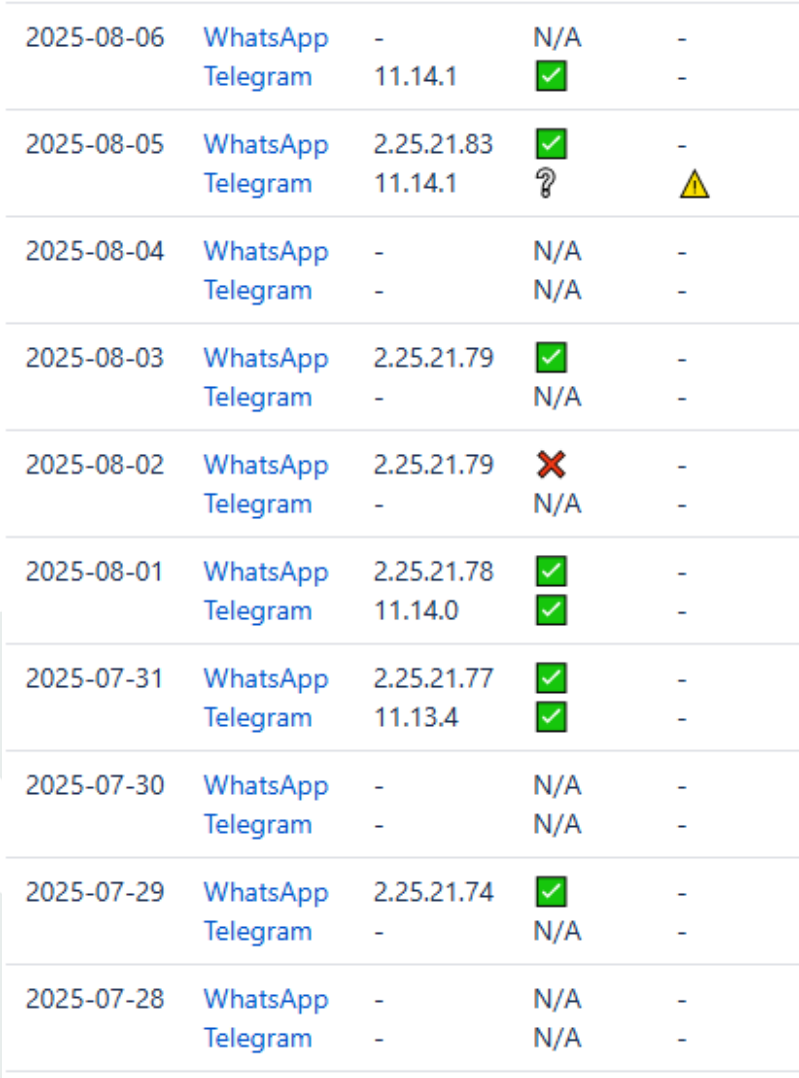
Puma has greatly improved our processes for research and continuous validation.

- Initial app population saves time
- Update-triggered validation gives us exact and complete coverage
  - Time between update and bug report < 24h
  - Complete list of validated app versions
- **Win:** Hansken supports Telegram updates faster than Cellebrite UFED

High potential for all tool developers, researchers, educators, and hobbyists!

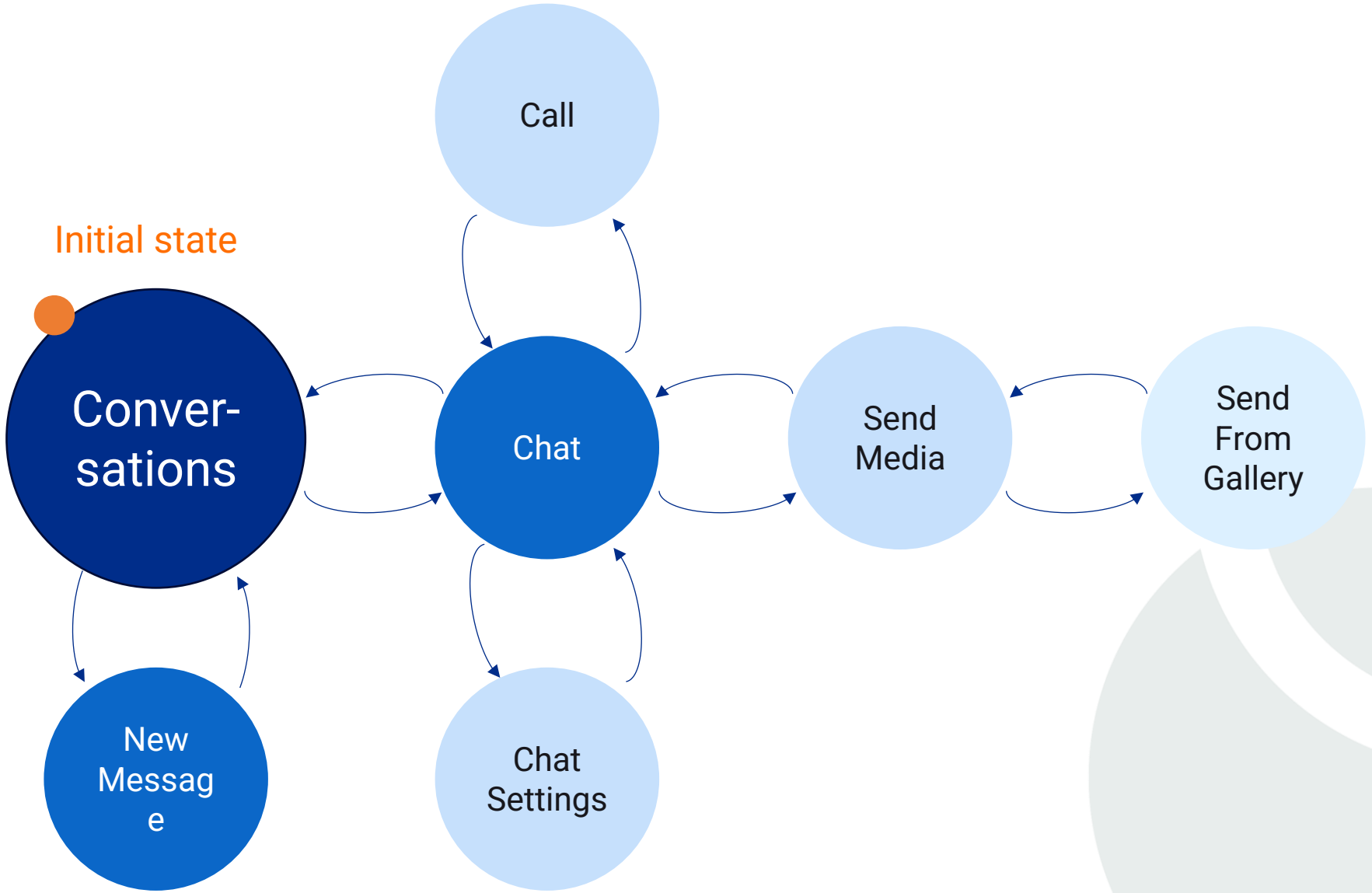
Caveats:

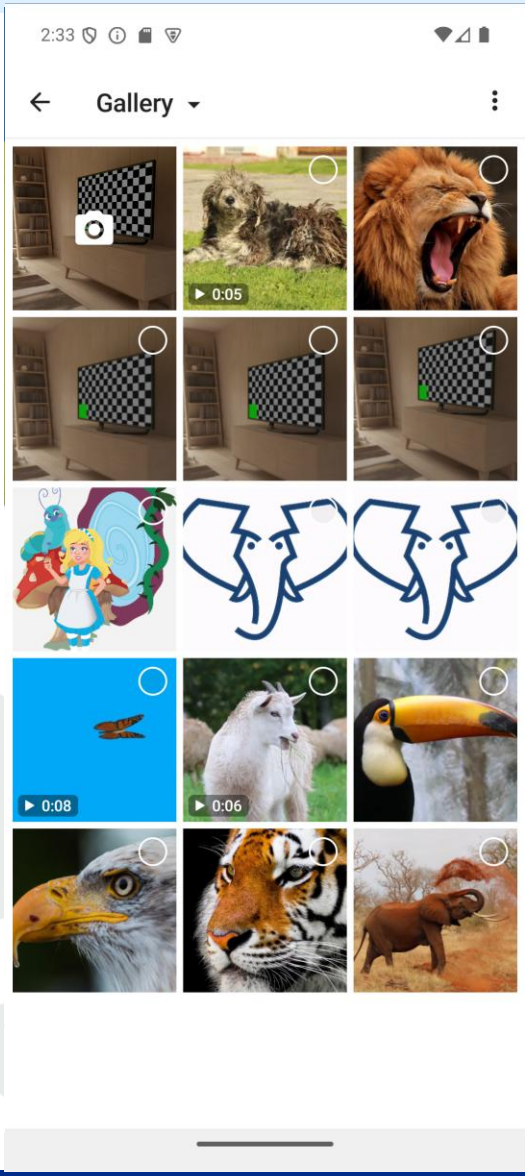
- Puma has development costs
- Puma is intrusive
- Be mindful of staged rollouts!



2025-08-06	WhatsApp	-	N/A	-
	Telegram	11.14.1	✅	-
2025-08-05	WhatsApp	2.25.21.83	✅	-
	Telegram	11.14.1	❓	⚠
2025-08-04	WhatsApp	-	N/A	-
	Telegram	-	N/A	-
2025-08-03	WhatsApp	2.25.21.79	✅	-
	Telegram	-	N/A	-
2025-08-02	WhatsApp	2.25.21.79	❌	-
	Telegram	-	N/A	-
2025-08-01	WhatsApp	2.25.21.78	✅	-
	Telegram	11.14.0	✅	-
2025-07-31	WhatsApp	2.25.21.77	✅	-
	Telegram	11.13.4	✅	-
2025-07-30	WhatsApp	-	N/A	-
	Telegram	-	N/A	-
2025-07-29	WhatsApp	2.25.21.74	✅	-
	Telegram	-	N/A	-
2025-07-28	WhatsApp	-	N/A	-
	Telegram	-	N/A	-

Implementation: State machine



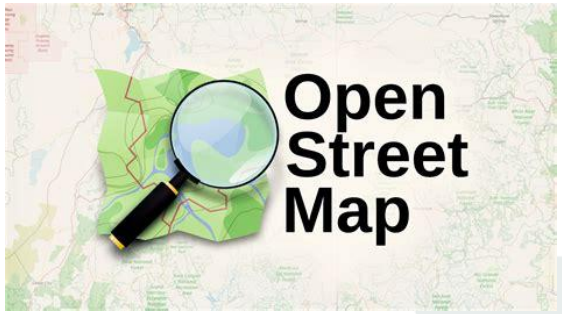


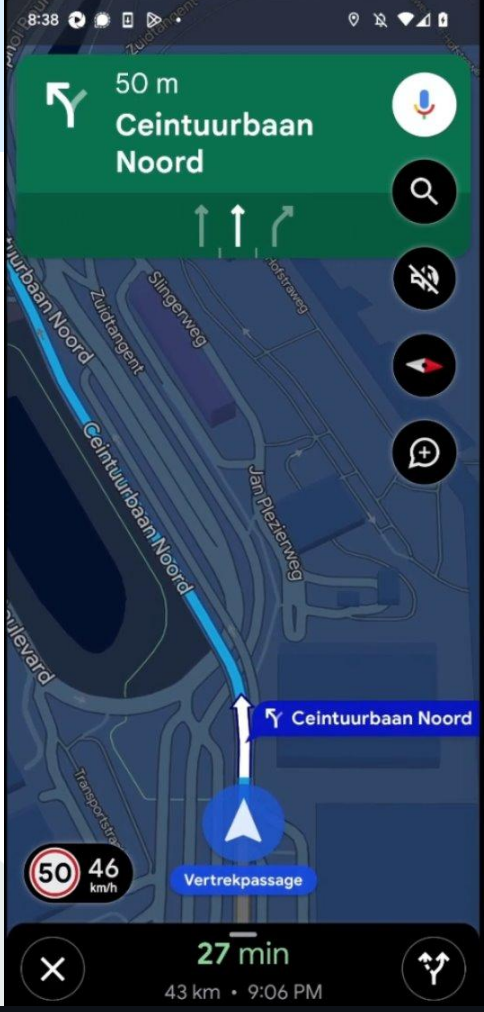
*Do not use on your own phone!

Implementation: location spoofing*

- Spoof location

- Navigation
  - Request route; update location

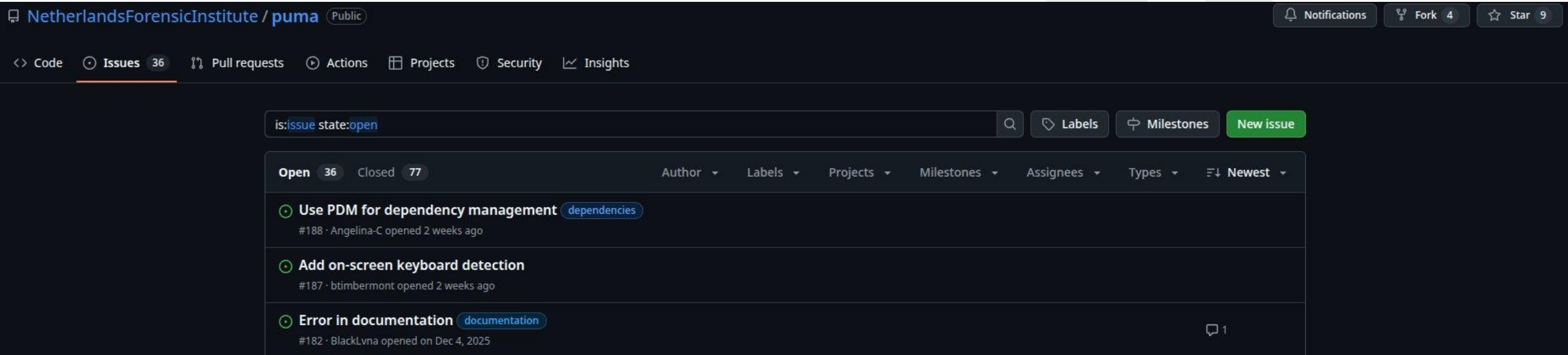





```
class RouteSimulator:
    """
    Class that allows you to spoof the location along a given route on a device available through appium.
    Starting a route is done with the execute_route_(...) methods. See their documentation on how to call these methods.
    The speed can be changed while the route is being traveled. See update_speed().
    Location spoofing can be stopped with stop_route().
    """
```

Please contribute!

- Report bugs
- Feature requests
- Add new features of an application
- Add new applications



Questions?

Contact us about Puma or Hansken: a.claij@nfi.nl

e.oudsen@nfi.nl

More about Hansken at

https://hansken.org

You can check out Puma at

https://github.com/NetherlandsForensicInstitute/puma

Or scan:

Hansken

The open digital forensic platform

Puma

Claij-Swart, A. A., Oudsen, E., Timbermont, B., Hargreaves, C., & Voigt, L. L. (2025). Automatically generating digital forensic reference data triggered by mobile application updates. Forensic Science International: Digital Investigation, 54(Supplement), 301985. https://doi.org/10.1016/j.fsidi.2025.301985