

“Making SimpleMDM Complicated”

“Making SimpleMDM Complicated”

*Considerations in systematically managing your devices
via a MDM vendor with a working API.*

Lucas J. Hall
@lucasjhall_
@lucas

SimpleMDM isn't just any MDM, but it can be any MDM.

- Extend these ideas generally
- 🙄, MDM Specific first, SimpleMDM features Second
- Context: AWS

A few bad assumptions...

- You're using mdm for... mdm
- Your mdm vendor, or internal solution, has a ~~robust basic any~~ API
- You have an ~~cloud~~ environment with access to to perform basic logic via scripted actions or services
- Be able to receive and process webhook (POST) payloads

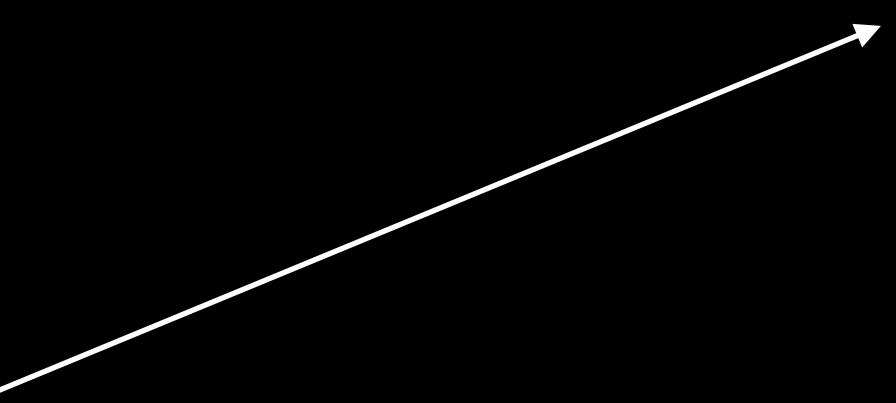
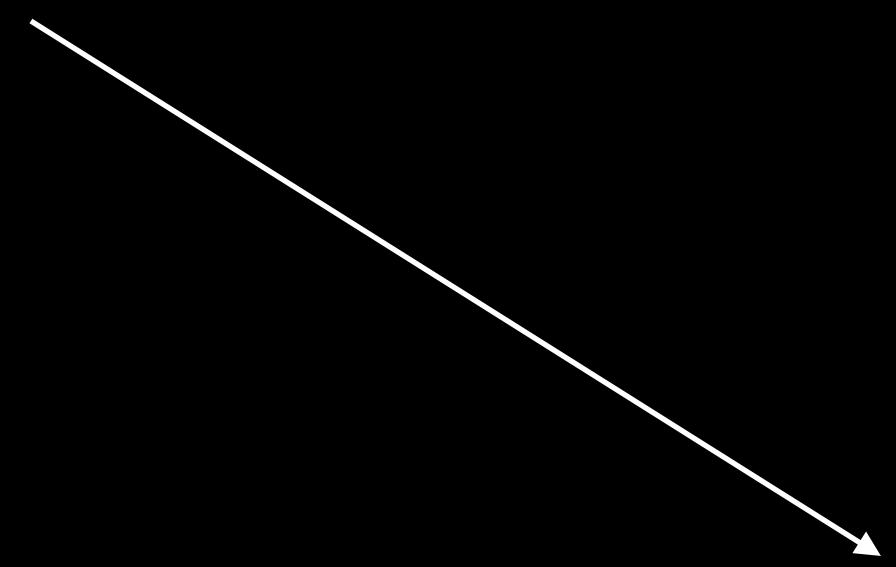
tl;dr

- All the (MDMs | Enrollments)?
- Overrides and Attribute Precedence
- API Use and Abuse

All the Enrollments?

- Multiple auto assigning enrollments?
- Inbound pipelines from ABM → MDM
- Example: Systematic MDM Enrollment based on Platform

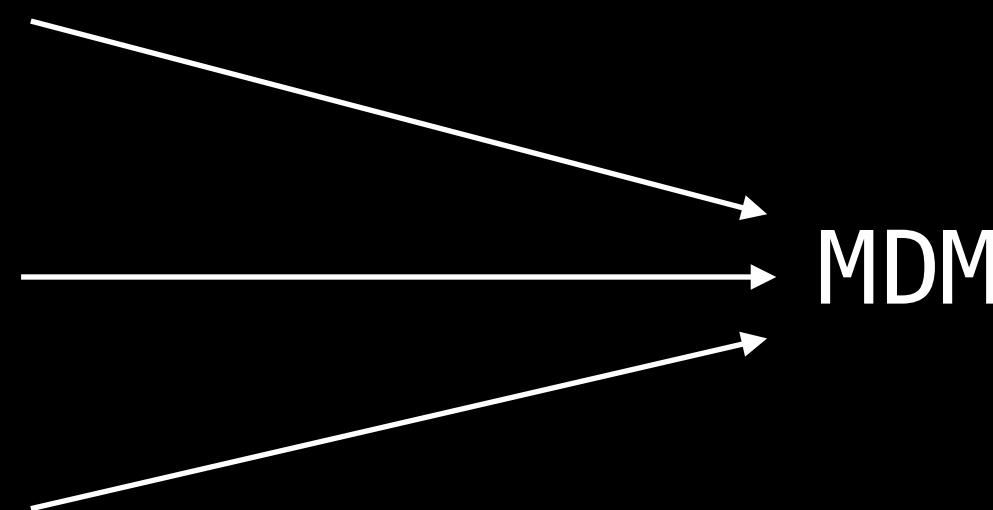
resellers



macOS

iPadOS

iOS



nd Books

gs

ent Information

c

ts

anaged Apple IDs

ource

Management Settings

evices

TV MDM Server

ces

ering MDM Server

ices

ng MDM Server

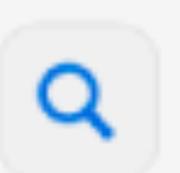
evices

IDM Server

evices

Device Management Settings

9,000 Devices



Show
Devices



Add MDM
Server

Customer Numbers ?

Edit

● 5 Active ● 4 Pending

Default Device Assignment

Done

iPad

Engineering MDM Server

iPhone

Sales MDM Server

iPod

None

Mac

Marketing MDM Server

Apple TV

Apple TV MDM Server

Devices assigned to specific MDM servers

resellers



macOS

iPadOS

iOS

macOS MDM

iPadOS MDM

iOS MDM

The screenshot shows a web browser window with the URL docs.simplemdm.com in the address bar. The page content is titled "Error Messages" and includes a section titled "How to Connect SimpleMDM to DEP".

Error Messages

- Skip basic setup steps

When paired, SimpleMDM configures the devices in your DEP account according to your settings in SimpleMDM.

How to Connect SimpleMDM to DEP

SimpleMDM needs to link to your DEP account so that it can configure devices to use the SimpleMDM service. To link Apple DEP to SimpleMDM, a certificate must be exchanged between the accounts to establish a secure, trusted relationship.

Complete the following steps to link your DEP account:

1. Sign into SimpleMDM.
2. Click the "Devices" link on the left hand side of the screen.
3. Click the "Enrollments" sub-menu option.
4. On the Enrollments page, click the "Create Enrollment" button.
5. Select "Automated Enrollment (DEP)".
6. Follow the on-screen steps based on whether you are using Apple Business Manager or a legacy Apple DEP account. These steps will guide you through the certificate exchange process.

Once you have uploaded your Apple server token, SimpleMDM will link to your Apple Business Manager account and be granted access for [automated enrollments](#).

Last updated on March 5, 2020

resellers



macOS

iPadOS

iOS

macOS MDM

iPadOS MDM

iOS MDM

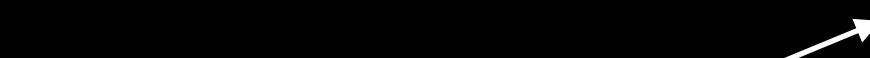
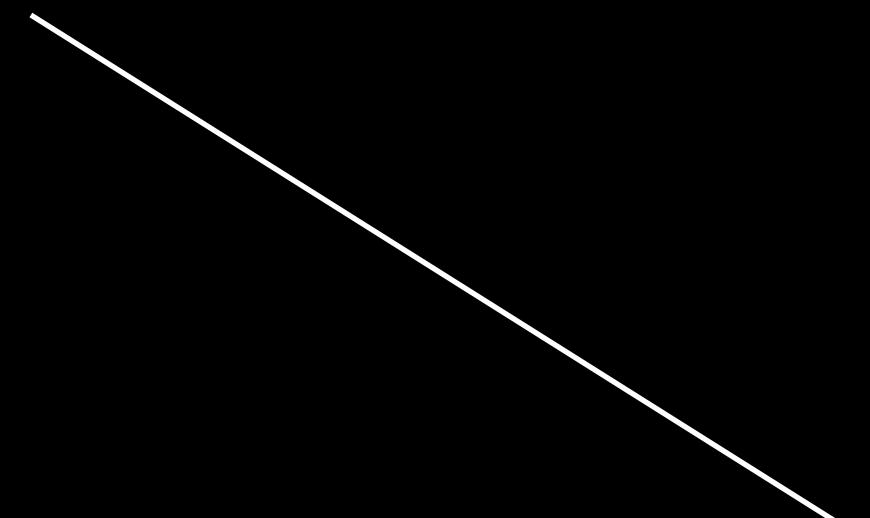
Testing

CICD

ZOOM macOS

Zoom iOS

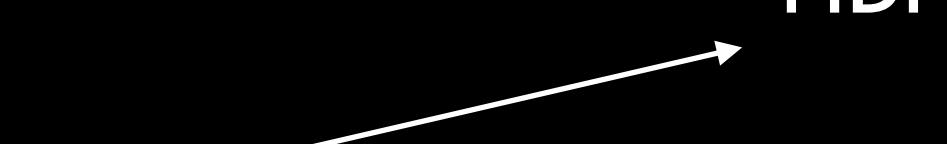
resellers

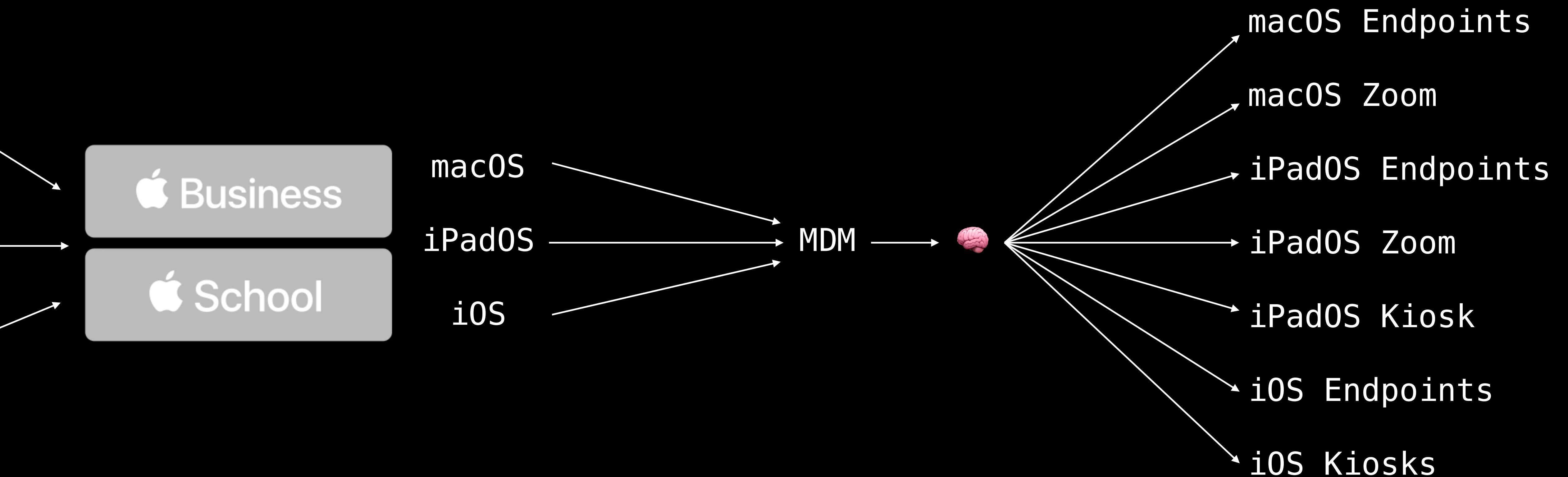


macOS

iPadOS

iOS





Overrides and Attribute Precedence

- SimpleMDM Specific Examples
 - What are custom attributes?
 - How is precedence determined?
 - Logically extend to any system that supports attributes, or attribute precedence

Attributes

- SimpleMDM's oob include:
 - name, model, phone number, serial number, UDID
- Lock Screen Example:
 - Asset Name: {{device_name}}. Serial Number: {{serial_number}}.

Custom Attributes

- Key: value relationship
 - “screensaver_default”: “10”
 - Set within Configs > Attributes

Custom Attribute Precedence

highest



device

5

group

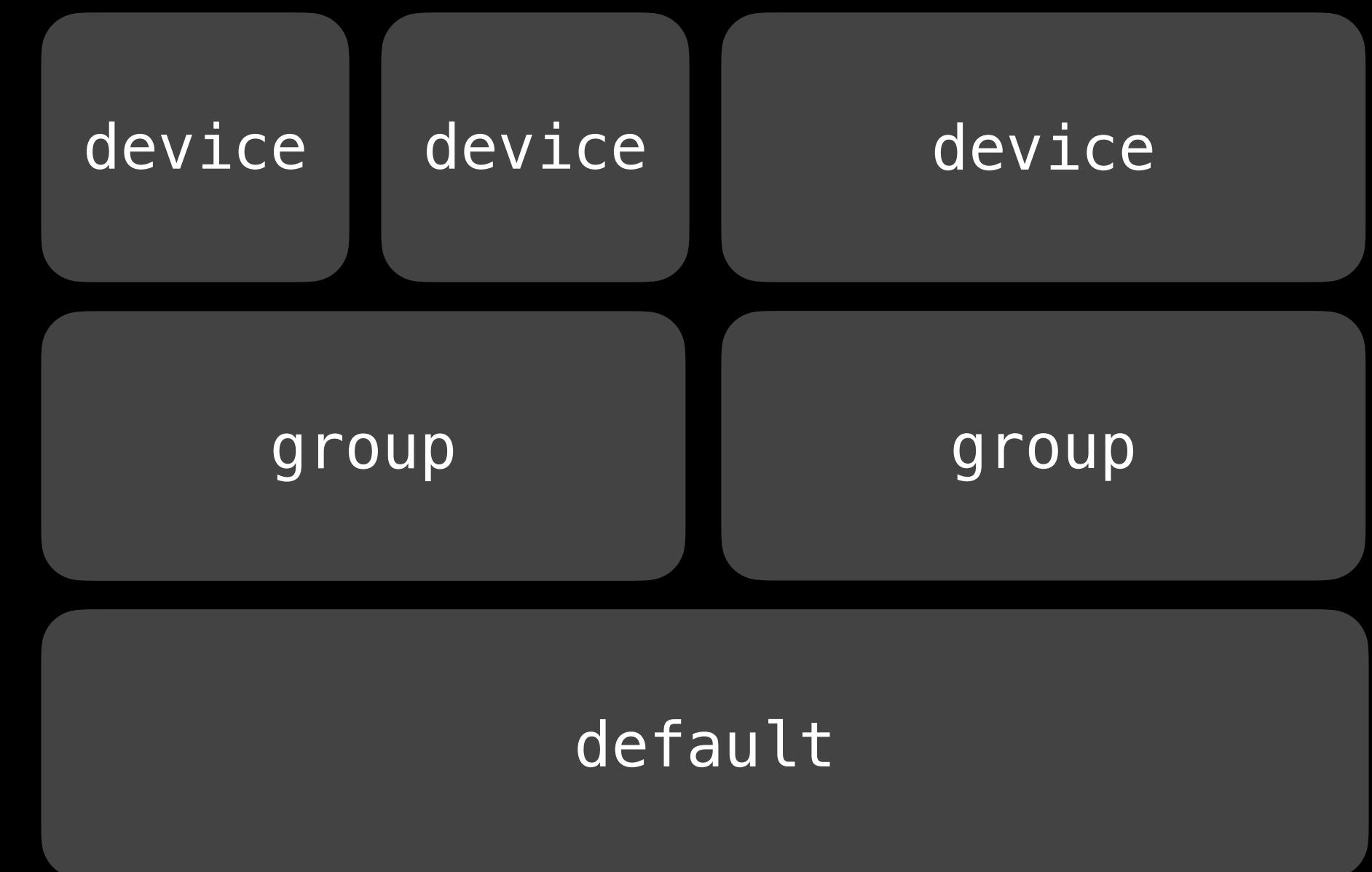
10

lowest

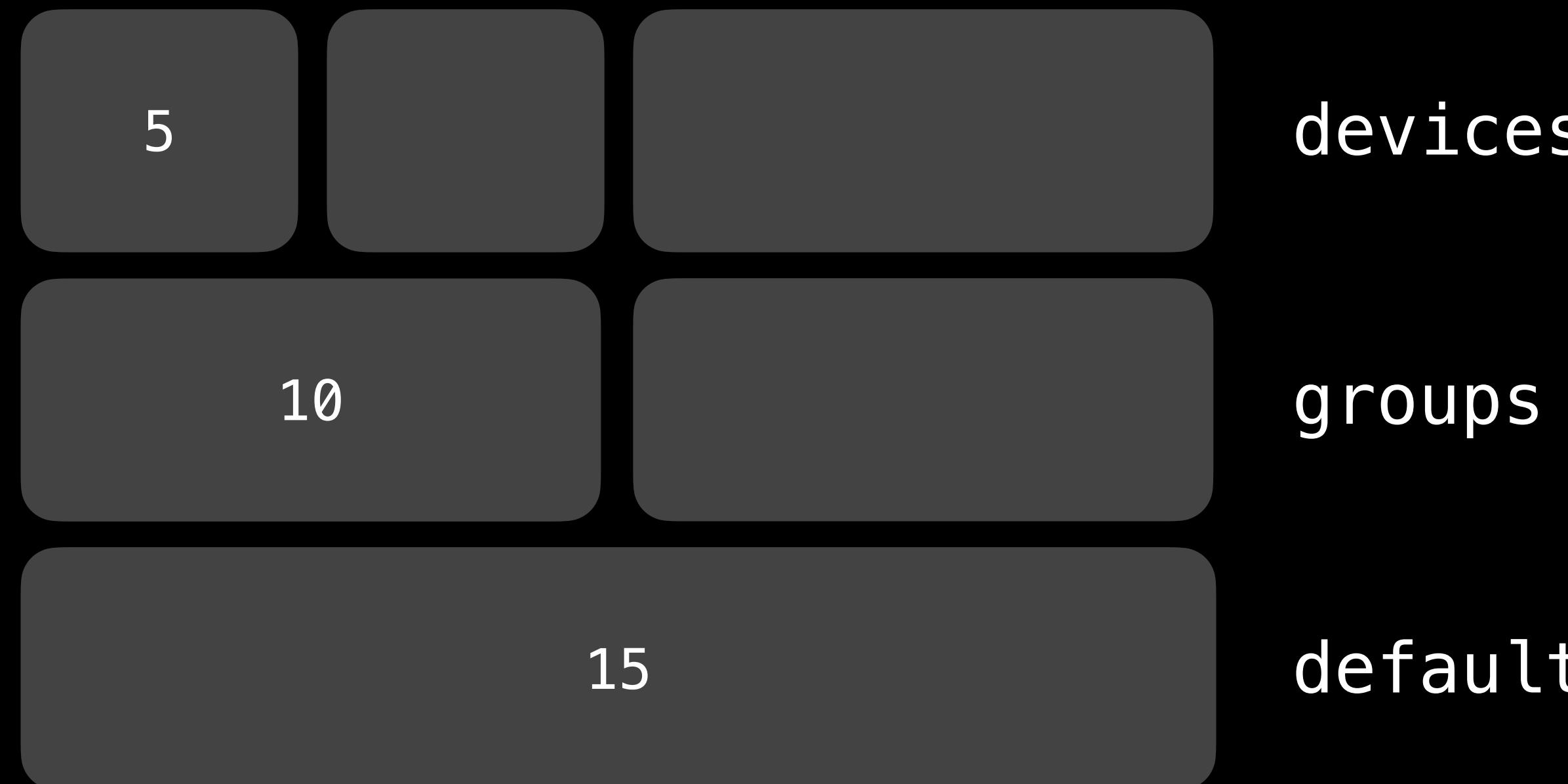
default

15

Custom Attribute Precedence



Custom Attribute Precedence



Custom Attribute Overrides and Precedence

- Beneficial to Systematically Updating the Attribute's for Enrollment, Group or Device
- Not maintaining profiles per control, per device

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>PayloadContent</key>
  <array>
    <dict>
      <key>PayloadDescription</key>
      <string>>A configuration profile which enables the Screensaver.</string>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>PayloadContent</key>
  <array>
    <dict>
      <key>PayloadDescription</key>
      <string>>A configuration profile which enables the Screensaver.</string>
      <key>askForPassword</key>
      {{screensaver_ask_for_password_bool}}
      <key>askForPasswordDelay</key>
      <integer>{{screensaver_ask_for_password_delay}}</integer>
      <key>idleTime</key>
      <integer>{{screensaver_idle_time}}</integer>
      <key>loginWindowIdleTime</key>
      <integer>{{screensaver_login_window_idle_time}}</integer>
    ...
  
```

API Use and Abuse

- What *is* Supported?
- API Libraries?
- Webhooks
- Examples:
 - Log Ingestion
 - Systematic Group Assignment
 - Systematic Attribute Assignment



/usr/bin/profiles

What is supported?



What is an MDM?

What is the MDM specification?

I APNS therefore I am?

Not now.

What is supported?

- Probably a question that should be asked when considering an MDM.
- What can / can't you do from API?
- How do those things align with your need?

The screenshot shows a web browser window with the URL simplemdm.com in the address bar. The page displays the SimpleMDM API documentation for version 1.27. On the left, a sidebar lists various API endpoints under categories like Introduction, Authentication, Errors, Pagination, Account, Apps, Assignment Groups, Custom Attributes, Custom Configuration, DEP Servers, and Devices. Under the Devices category, the 'Retrieve one' option is selected, which leads to the current view. The main content area has a title 'Retrieve one' and a sub-section 'HTTP Request' containing the command `curl https://a.simplemdm.com/api/v1/devices/{DEVICE_ID}`. To the right of this, a code block shows the JSON response for a device with ID 121. The JSON object contains detailed information about the device, including its name, last seen at, status, device name, OS version, build version, model name, model, product name, serial number, IMEI, MEID, device capacity, available device capacity, battery level, modem firmware version, ICCID, Bluetooth MAC, WiFi MAC, current carrier network, SIM carrier network, subscriber carrier network, carrier settings version, phone number, and voice roaming enabled status.

```
curl https://a.simplemdm.com/api/v1/devices/121 \
-u {API_KEY}:

{
  "data": {
    "type": "device",
    "id": 121,
    "attributes": {
      "name": "Mike's iPhone",
      "last_seen_at": "2015-10-01T18:38:47.277-07:00",
      "status": "enrolled",
      "device_name": "Mike's iPhone",
      "os_version": "9.0.2",
      "build_version": "13A452",
      "model_name": "iPhone 6",
      "model": "NG4W2LL",
      "product_name": "iPhone7,2",
      "serial_number": "DNFJE9DNG5MG",
      "imei": "35 445506 652132 5",
      "meid": "35404596608032",
      "device_capacity": 55.62955093383789,
      "available_device_capacity": 15.19466781616211,
      "battery_level": "93%",
      "modem_firmware_version": "4.02.00",
      "iccid": "8914 8110 0002 8094 4264",
      "bluetooth_mac": "f0:db:e2:df:e9:11",
      "wifi_mac": "f0:db:e2:df:e9:2f",
      "current_carrier_network": "Verizon",
      "sim_carrier_network": "Verizon",
      "subscriber_carrier_network": "Verizon",
      "carrier_settings_version": "21.1",
      "phone_number": "5035551234",
      "voice_roaming_enabled": true
    }
  }
}
```

API Libraries

- Language specific libraries or SDKs to interact with your MDM
- SimpleMDMpy

API Libraries

- Simplify the lift
- Handle API calls, pagination, etc

macadmins / simpleMDMpy

forked from SteveKueng/simpleMDMpy

Code Issues (4) Pull requests (1) Actions Wiki Security Insights Settings

main ▾ 6 branches 10 tags Go to file Add file ▾ Code ▾

This branch is 18 commits ahead, 1 commit behind SteveKueng:master.

Contribute ▾ Fetch upstream ▾

Readme MIT License

lucasjhall Merge pull request #22 from macadmins/issue-21 ... 5221aa3 on Mar 2 89 commits

Example: Log Ingestion

- GET the logs
- Optionally pick up where you left off in a past sync
- Process / ship the logs as needed for you

```
import SimpleMDMpy

simplemdm_api_key = 'Your_API_Key'
filterID = 'last_log_uuid'

Logs = SimpleMDMpy.Logs(simplemdm_api_key)
logObj = Logs.get_logs(id_override=filterID)

for log in logObj['data']:
    print(log)
```

SimpleMDM Webhooks

- Events that trigger web hooks:
 - device:
 - enrolled
 - unenrolled
 - changed_group
- These JSON payloads are sent to your API as a POST to be parsed within your system

SimpleMDM Webhooks

- Device enrolled:

```
{  
  "type": "device.enrolled",  
  "at": "2020-08-19T14:46:55.094+00:00",  
  "data": {  
    "device": {  
      "id": 4444444,  
      "udid": "00008020-000834AA2E09002E",  
      "serial_number": "C023456789",  
      "device_group_id": 55555  
    }  
  }  
}
```

SimpleMDM Webhooks

- Device enrolled:

```
{  
  "type": "device.enrolled",  
  "at": "2020-08-19T14:46:55.094+00:00",  
  "data": {  
    "device": {  
      "id": 4444444,  
      "udid": "00008020-000834AA2E09002E",  
      "serial_number": "C023456789",  
      "device_group_id": 55555  
    }  
  }  
}
```



SimpleMDM Webhooks

- Device enrolled:

```
{  
  "type": "device.enrolled",  
  "at": "2020-08-19T14:46:55.094+00:00",  
  "data": {  
    "device": {  
      "id": 4444444,  
      "udid": "00008020-000834AA2E09002E",  
      "serial_number": "C023456789",  
      "device_group_id": 55555  
    }  
  }  
}
```

SimpleMDM Webhooks

- Device enrolled:

```
{  
  "type": "device.enrolled",  
  "at": "2020-08-19T14:46:55.094+00:00",  
  "data": {  
    "device": {  
      "id": 4444444,  
      "udid": "00008020-000834AA2E09002E",   
      "serial_number": "C023456789",  
      "device_group_id": 5555  
    }  
  }  
}
```

SimpleMDM Webhooks

- Device enrolled:

```
{  
  "type": "device.enrolled",  
  "at": "2020-08-19T14:46:55.094+00:00",  
  "data": {  
    "device": {  
      "id": 4444444,  
      "udid": "00008020-000834AA2E09002E",  
      "serial_number": "C023456789",   
      "device_group_id": 5555  
    }  
  }  
}
```

SimpleMDM Webhooks

- Device enrolled:

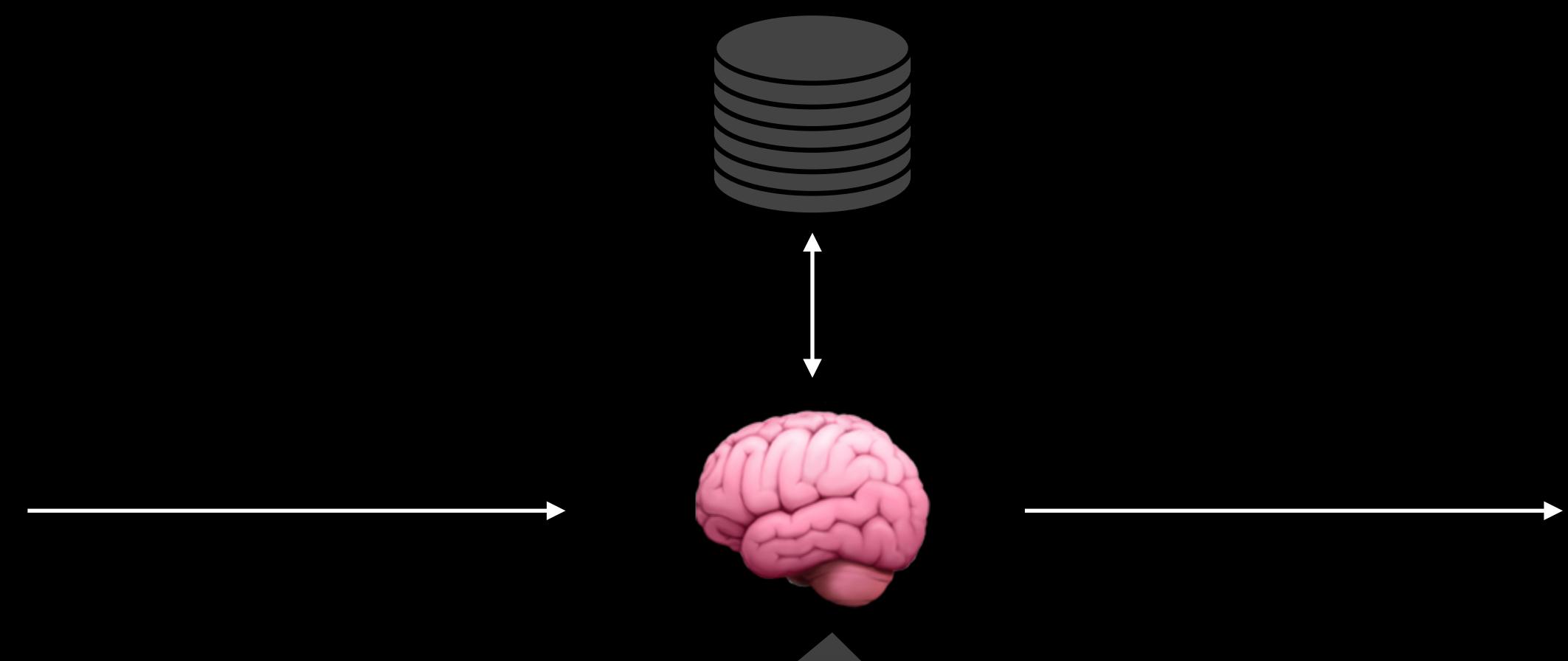
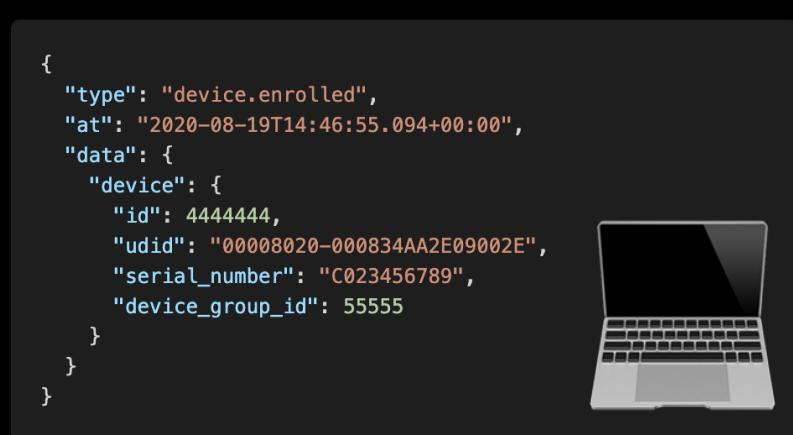
```
{  
  "type": "device.enrolled",  
  "at": "2020-08-19T14:46:55.094+00:00",  
  "data": {  
    "device": {  
      "id": 4444444,  
      "udid": "00008020-000834AA2E09002E",  
      "serial_number": "C023456789",  
      "device_group_id": 55555  
    }  
  }  
}
```



Example: Systematic Group Assignment

- We know dis:
 - Device serial number *UUID / PK device*
 - Device's ID within the MDM *How manipulate device*
 - Group ID of Device's assignment *Current configuration*

Example: Systematic Group Assignment



Group: Advertising

```
import SimpleMDMpy

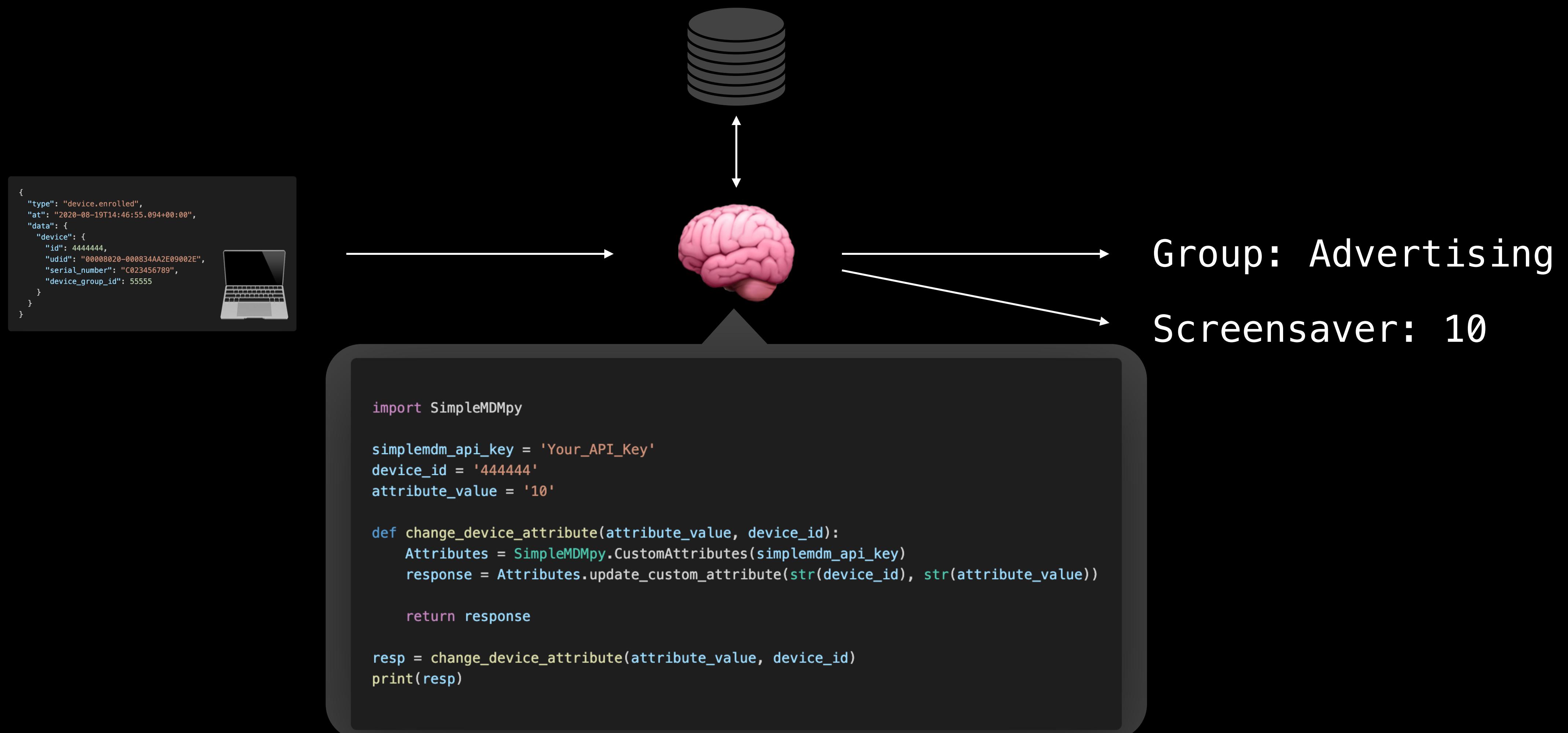
simplemdm_api_key = 'Your_API_Key'
device_id = '444444'
destination_group_id = '123456'

def change_simple_group(device_group_id, device_id):
    DeviceGroups = SimpleMDMpy.DeviceGroups(simplemdm_api_key)
    response = DeviceGroups.assign_device(str(device_id), str(device_group_id))

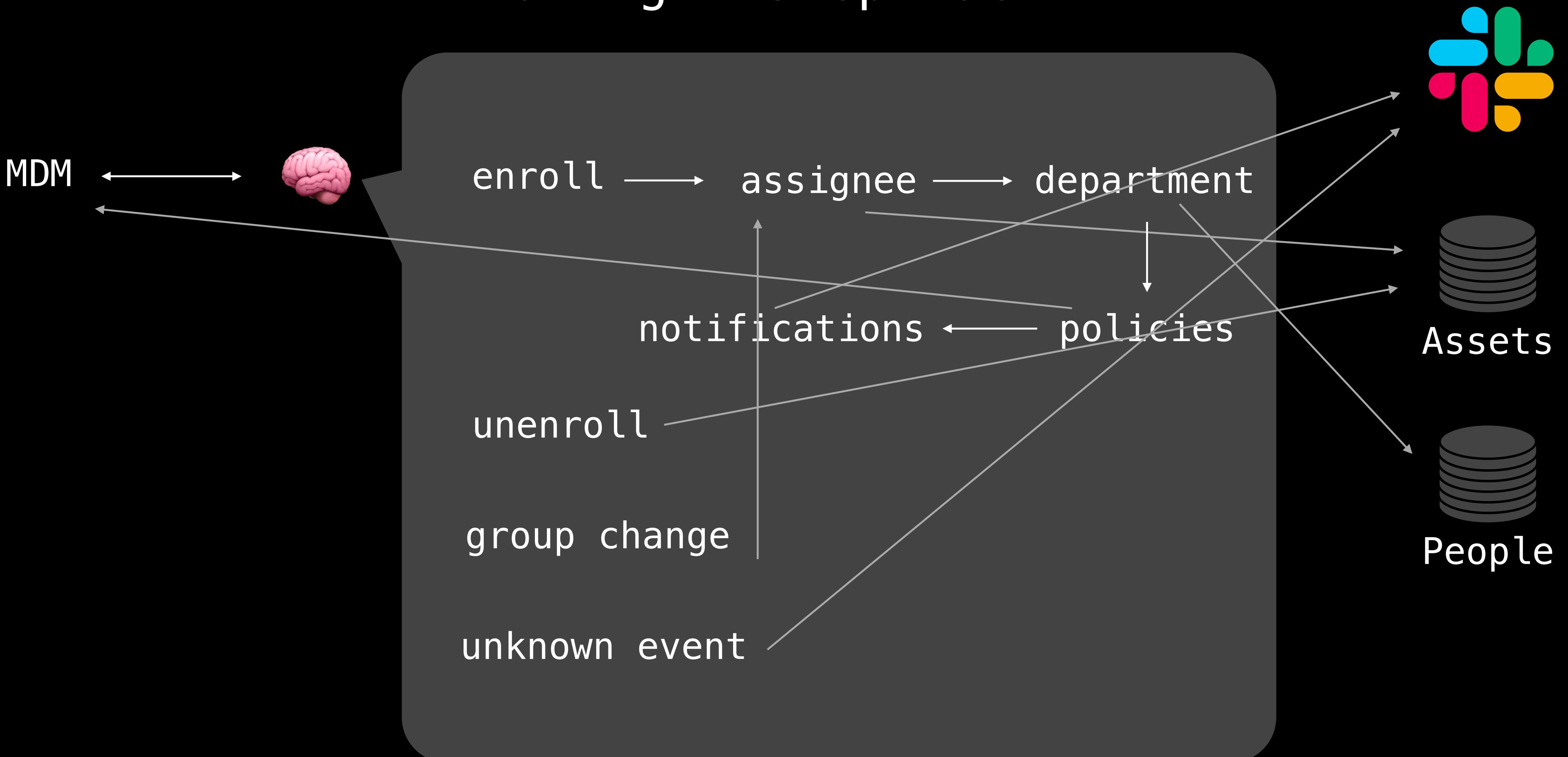
    return response

resp = change_simple_group(destination_group_id, device_id)
print(resp)
```

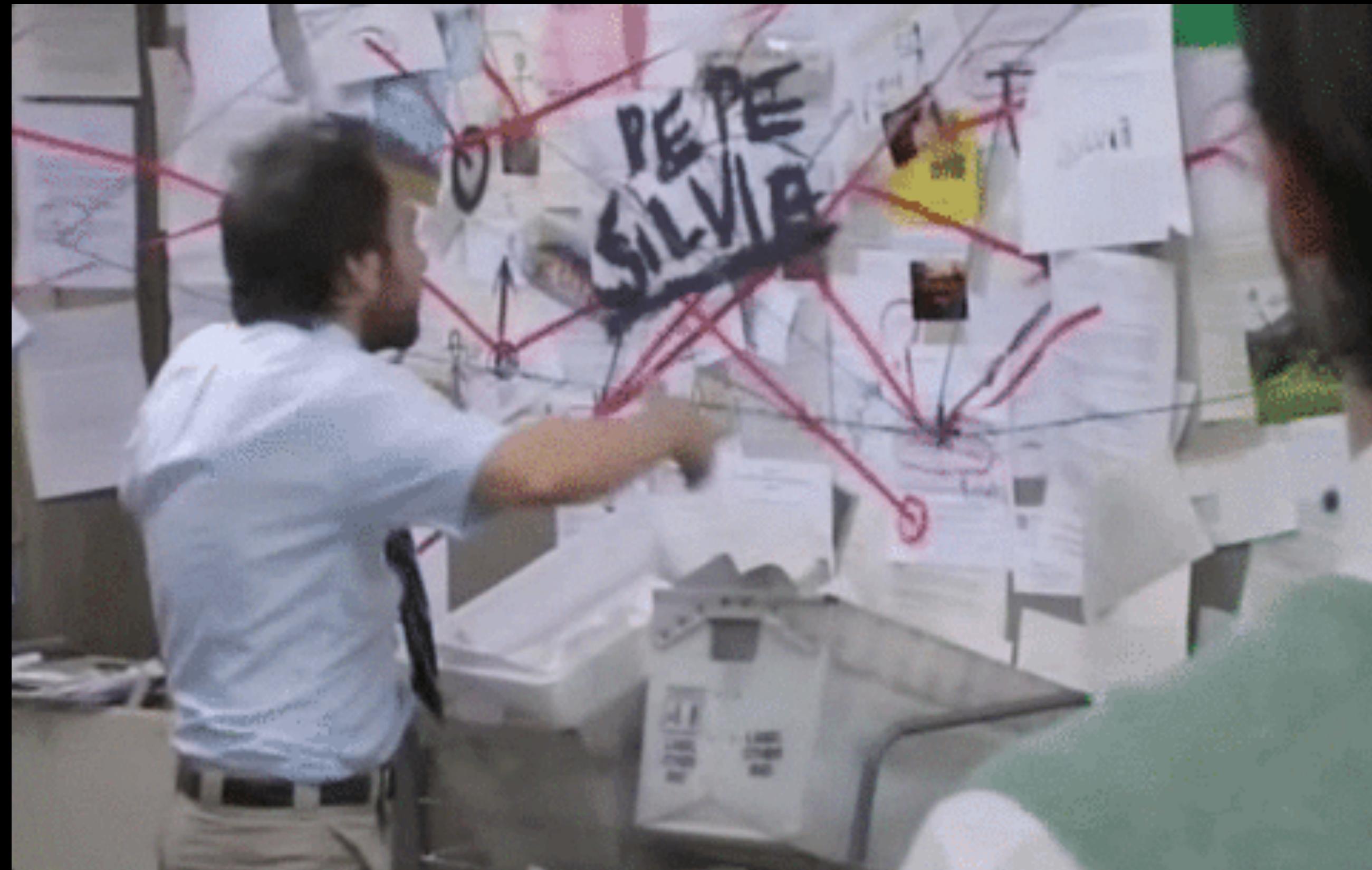
Example: Systematic Attribute Assignment



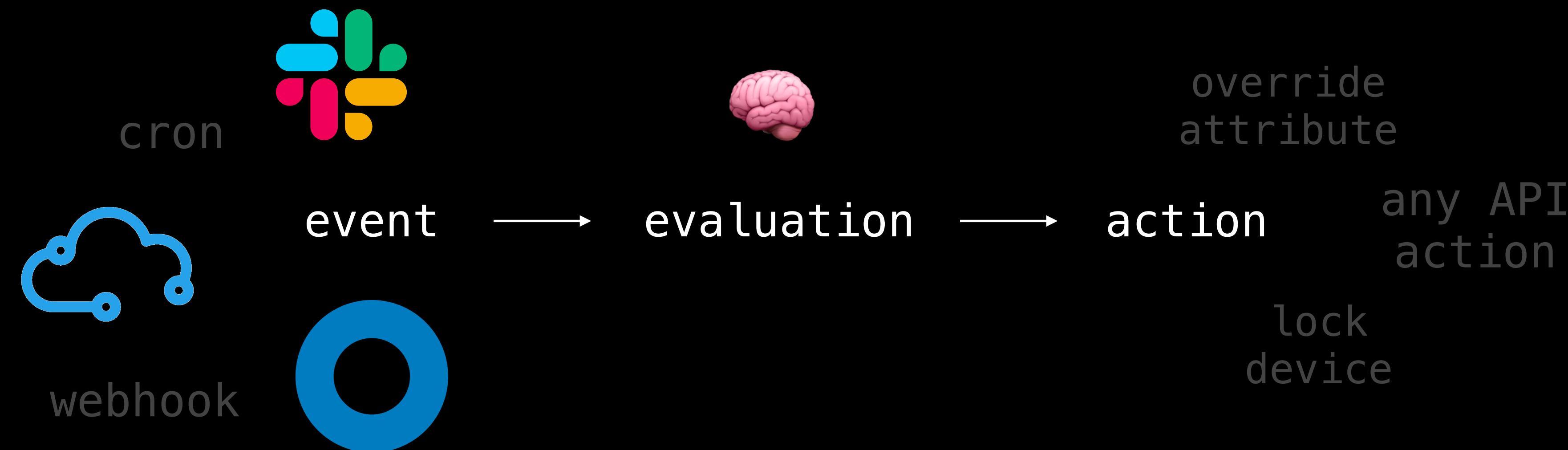
Taking A Step Back

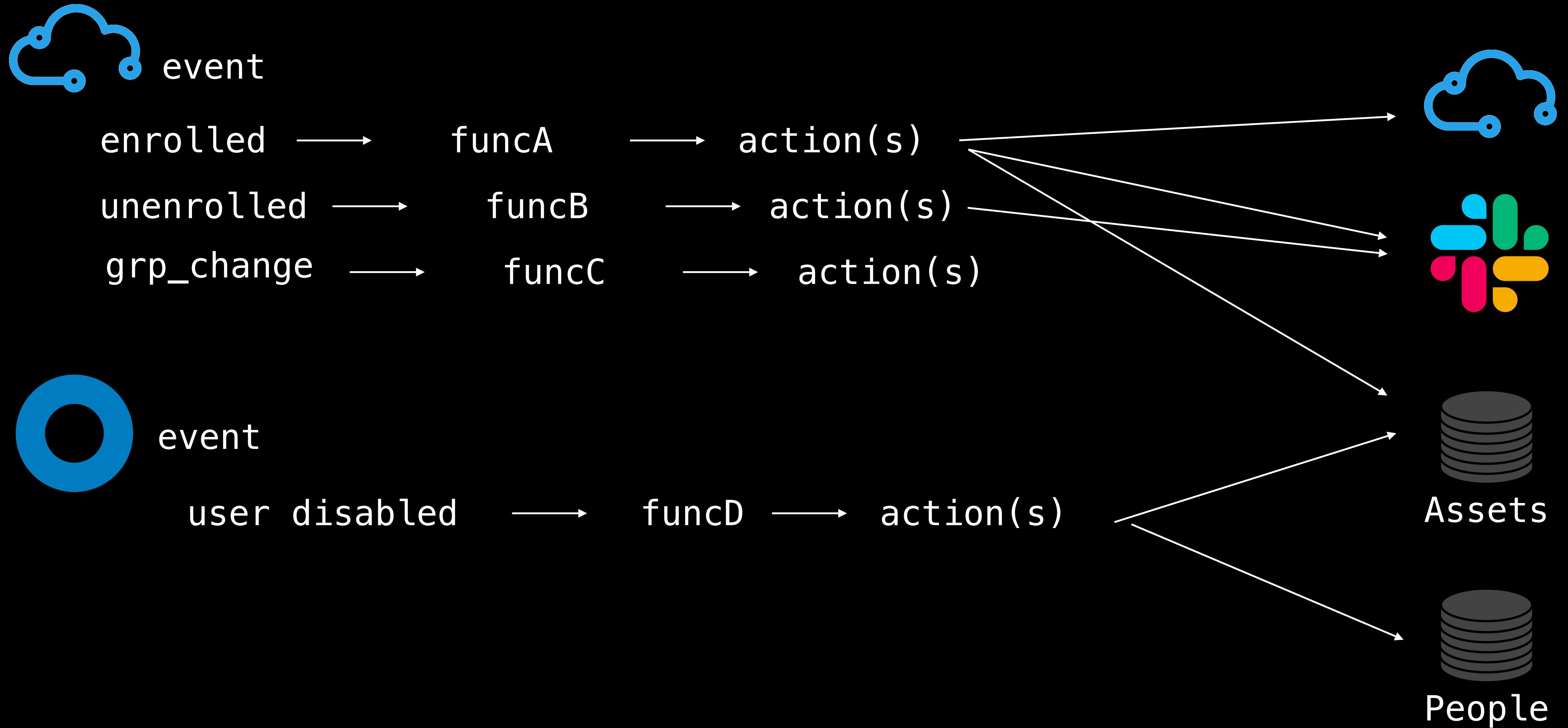


Taking A Step Back



Taking A Step Back





tl;dr

- Use the API
- Ponder, What *is* MDM?
- Complexity in increments as needed is 🤝

Resources

- SimpleMDMpy
- SimpleMDM API Documentation
- SimpleMDM on Macadmins
- Apple Device Management Documentation

Resources