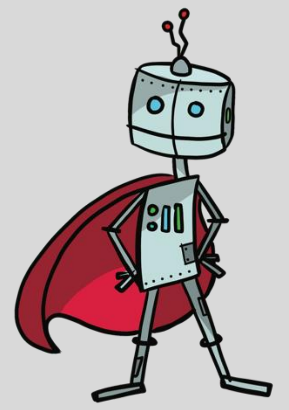
World Wide Technology
Make a new world happen

How to Automate Toil Out of Your Daily Life



WHOAMI?

Bob Longmore
World Wide Technology
Technical Solutions Architect
Global Solutions and Architecture
Minneapolis, MN

Long history in technology dating back to the US Air Force in the 1990s. Background in networking, security, and data center. In 1998, I wrote a batch file to install antivirus, kicking off decades of experimenting with automation.

wwt.com/profile/bob-longmore

linkedin.com/in/boblongmore

@boblongmore

boblongmore.com

What is Toil?

“TOIL is the **kind of work tied** to running a production service that tends to be manual, repetitive, automatable, tactical devoid of enduring value and that scales linearly as a service grows.”

What is Toil?

“TOIL is the **kind of work tied** to running a production service that tends to be manual, repetitive, automatable, tactical devoid of enduring value and that scales linearly as a service grows.”

What is boring and can't scale?

What is Toil?

“TOIL is the **kind of work tied** to running a production service that tends to be manual, repetitive, automatable, tactical devoid of enduring value and that scales linearly as a service grows.”

What is boring and can't scale?

What do you hate to do?

Why do we want to automate the toil away?

I am lazy, but I am also curious

Why do we want to automate the toil away?

I am lazy, but I am also curious

Reclaim time and mental energy, reduce context switching

Why do we want to automate the toil away?

I am lazy, but I am also curious

Reclaim time and mental energy, reduce context switching

★★★ Improve quality and reduce human errors

Toil in our day-to-day lives

manual hose by ProSymbols

Dishwasher by Maan Icons

cleaning robot by WR Graphic Garage

Toil in our day-to-day lives

Vacuuming

Toil in our day-to-day lives

Vacuuming

Dish Washing

manual hose by ProSymbols

Dishwasher by Maan Icons

cleaning robot by WR Graphic Garage

Toil in our day-to-day lives

Vacuuming

Dish Washing

Lawn Watering

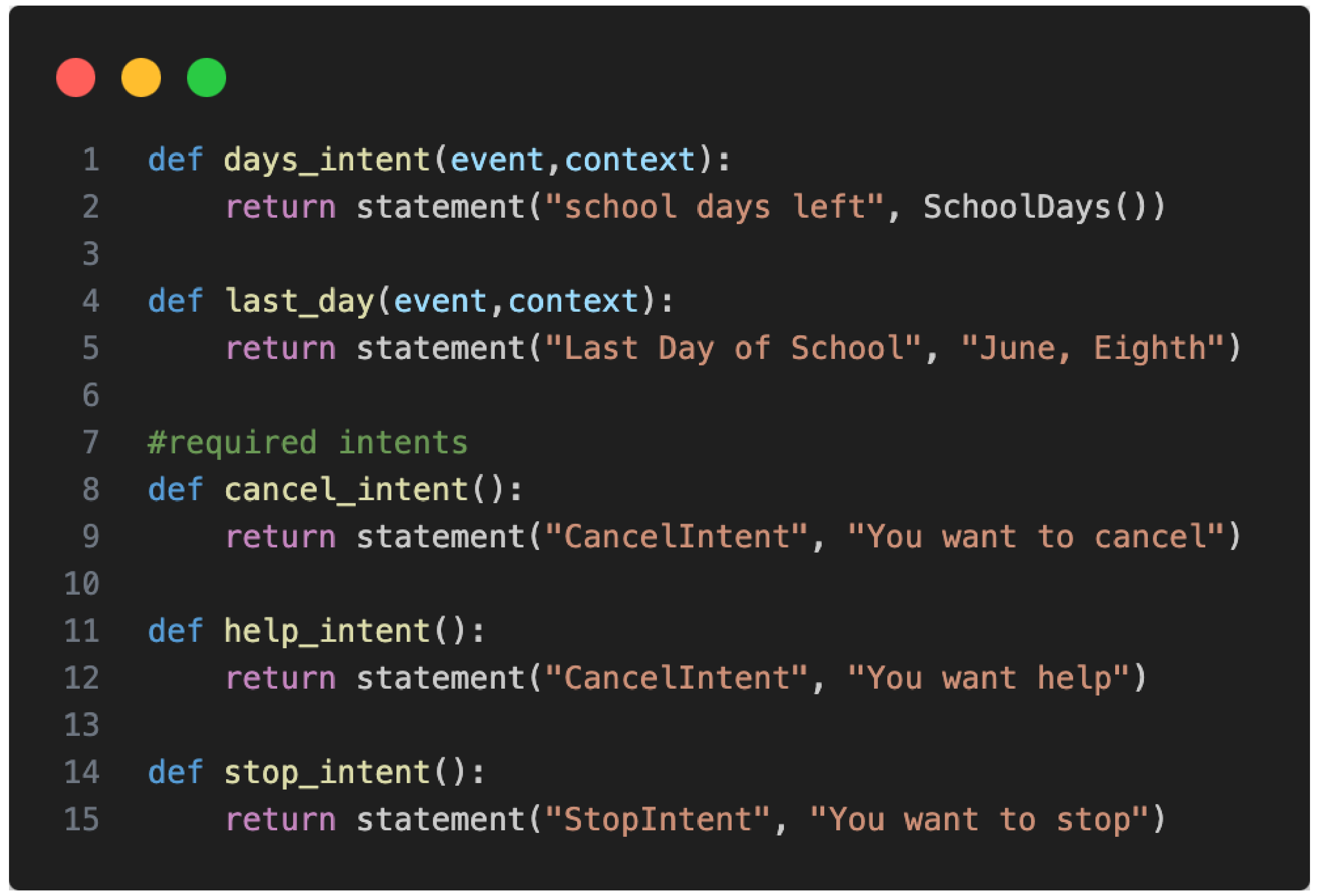
manual hose by ProSymbols

Dishwasher by Maan Icons

cleaning robot by WR Graphic Garage

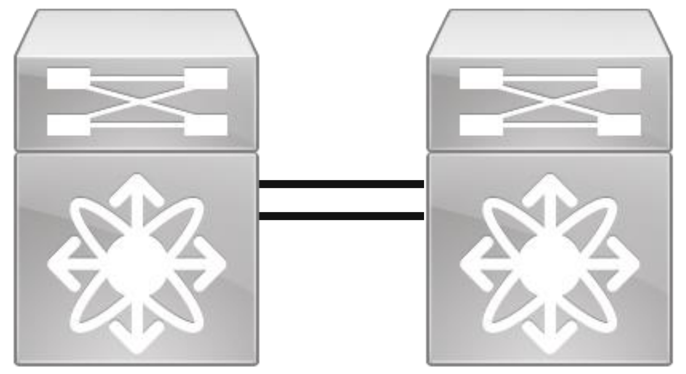
Personal examples of solving for toil

How Many School Days Left

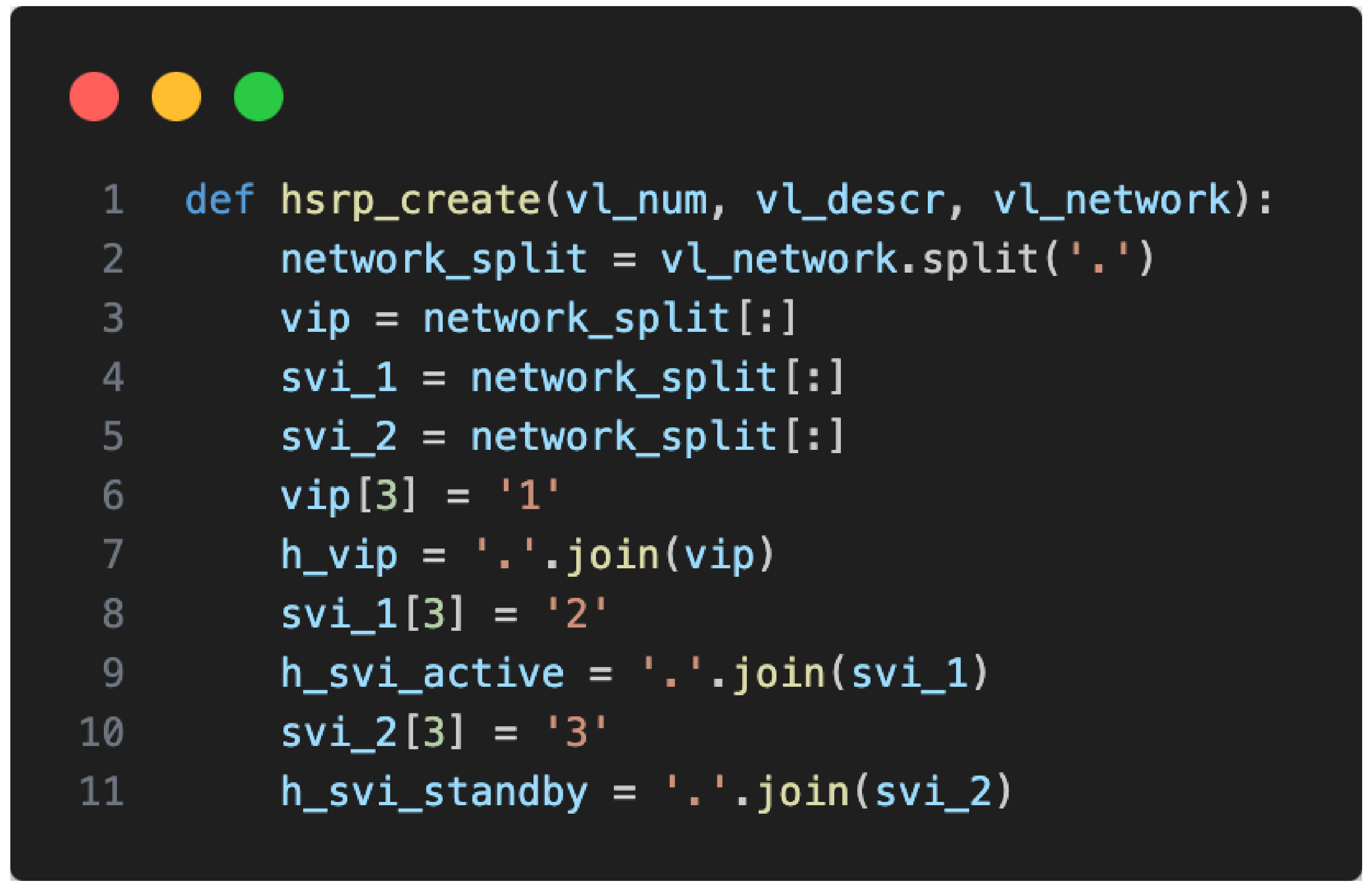


```
def days_intent(event,context):
    return statement("school days left", SchoolDays())

def last_day(event,context):
    return statement("Last Day of School", "June, Eighth")

#required intents
def cancel_intent():
    return statement("CancelIntent", "You want to cancel")

def help_intent():
    return statement("CancelIntent", "You want help")

def stop_intent():
    return statement("StopIntent", "You want to stop")
```

Personal examples of solving for toil

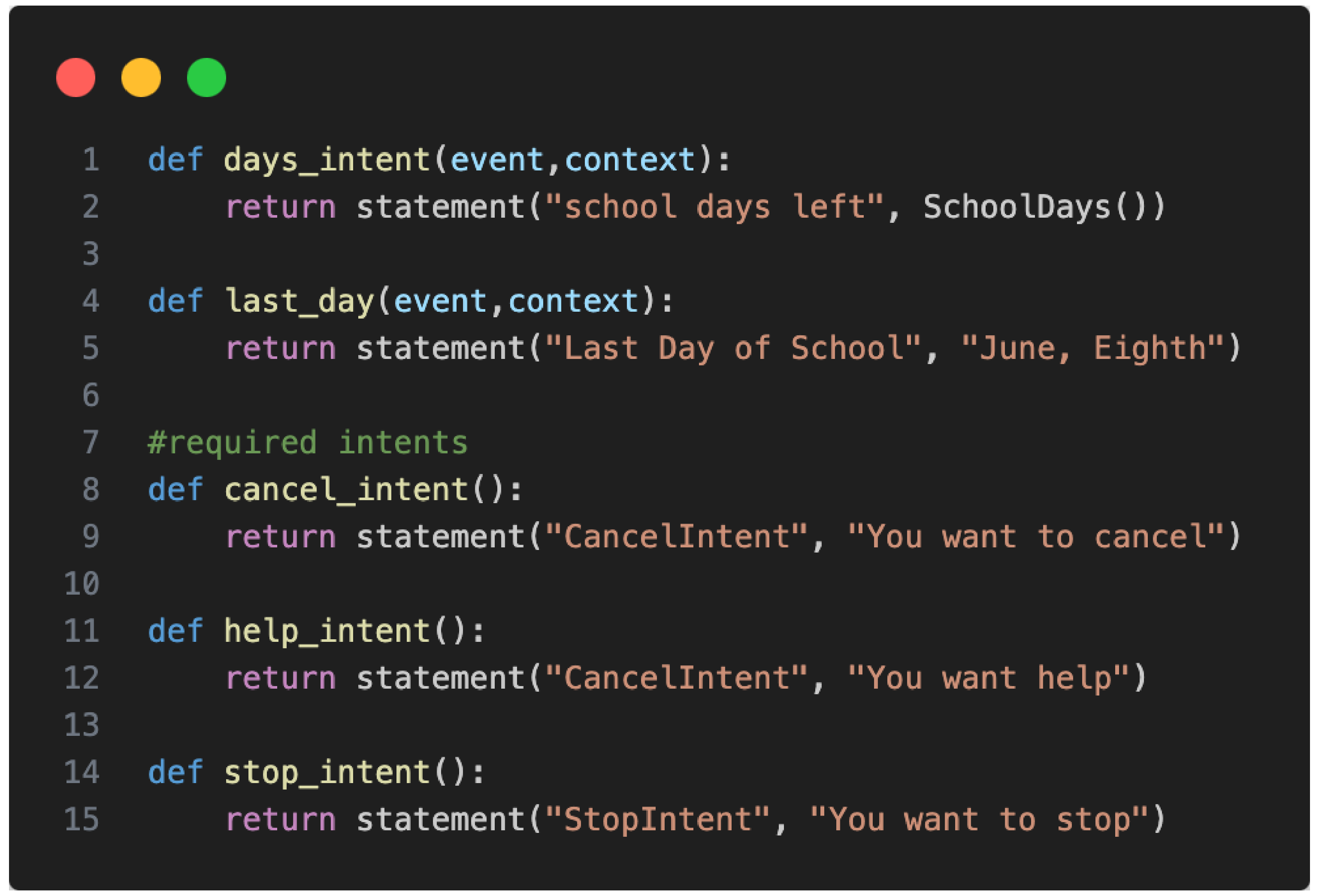


Nexus HSRP Configuration



```python
def hsrp_create(vl_num, vl_descr, vl_network):
    network_split = vl_network.split('.')
    vip = network_split[:]
    svi_1 = network_split[:]
    svi_2 = network_split[:]
    vip[3] = '1'
    h_vip = '.'.join(vip)
    svi_1[3] = '2'
    h_svi_active = '.'.join(svi_1)
    svi_2[3] = '3'
    h_svi_standby = '.'.join(svi_2)
```

How Many School Days Left



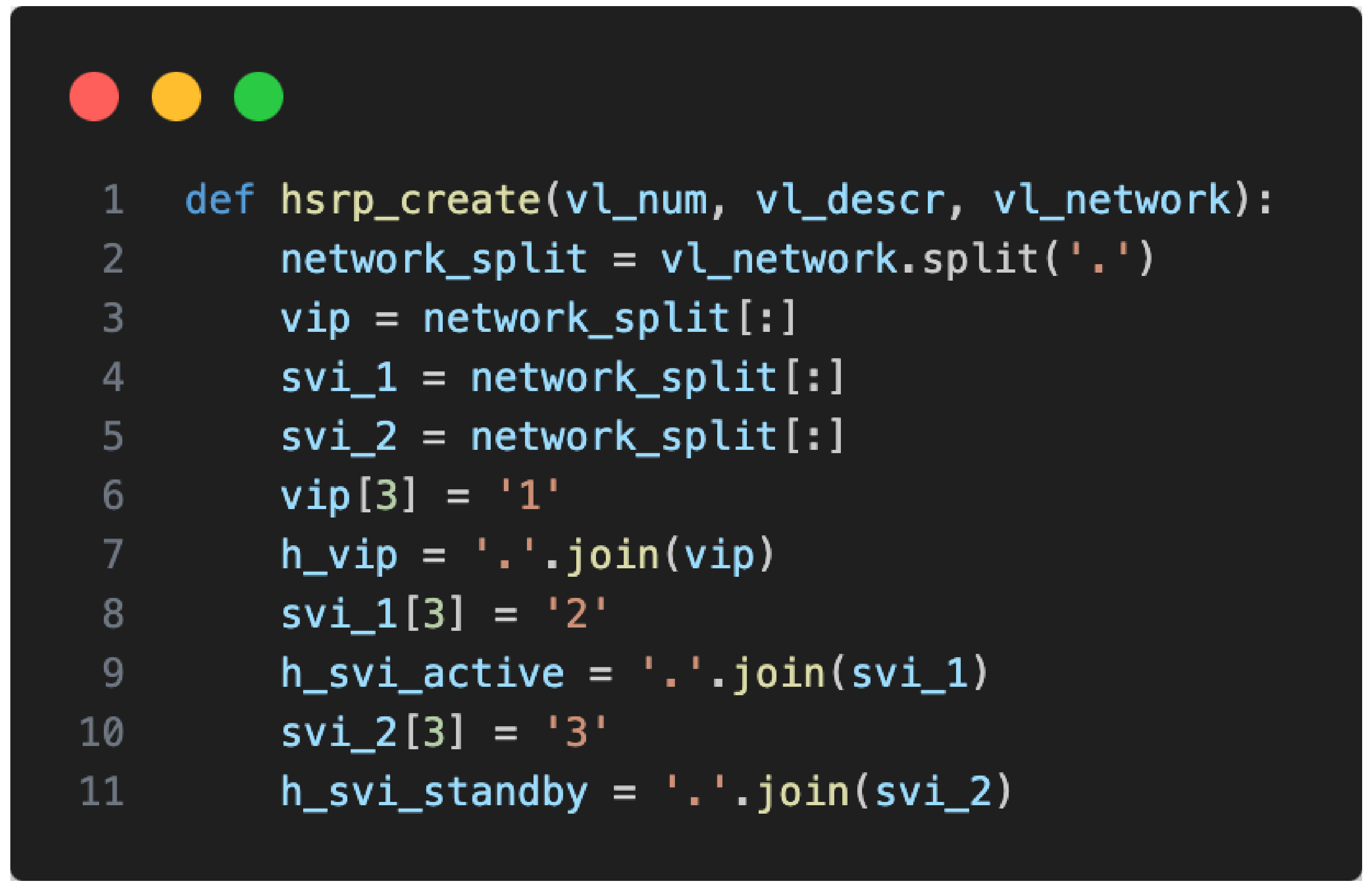
```python
def days_intent(event,context):
    return statement("school days left", SchoolDays())

def last_day(event,context):
    return statement("Last Day of School", "June, Eighth")

#required intents
def cancel_intent():
    return statement("CancelIntent", "You want to cancel")

def help_intent():
    return statement("CancelIntent", "You want help")

def stop_intent():
    return statement("StopIntent", "You want to stop")
```

Personal examples of solving for toil

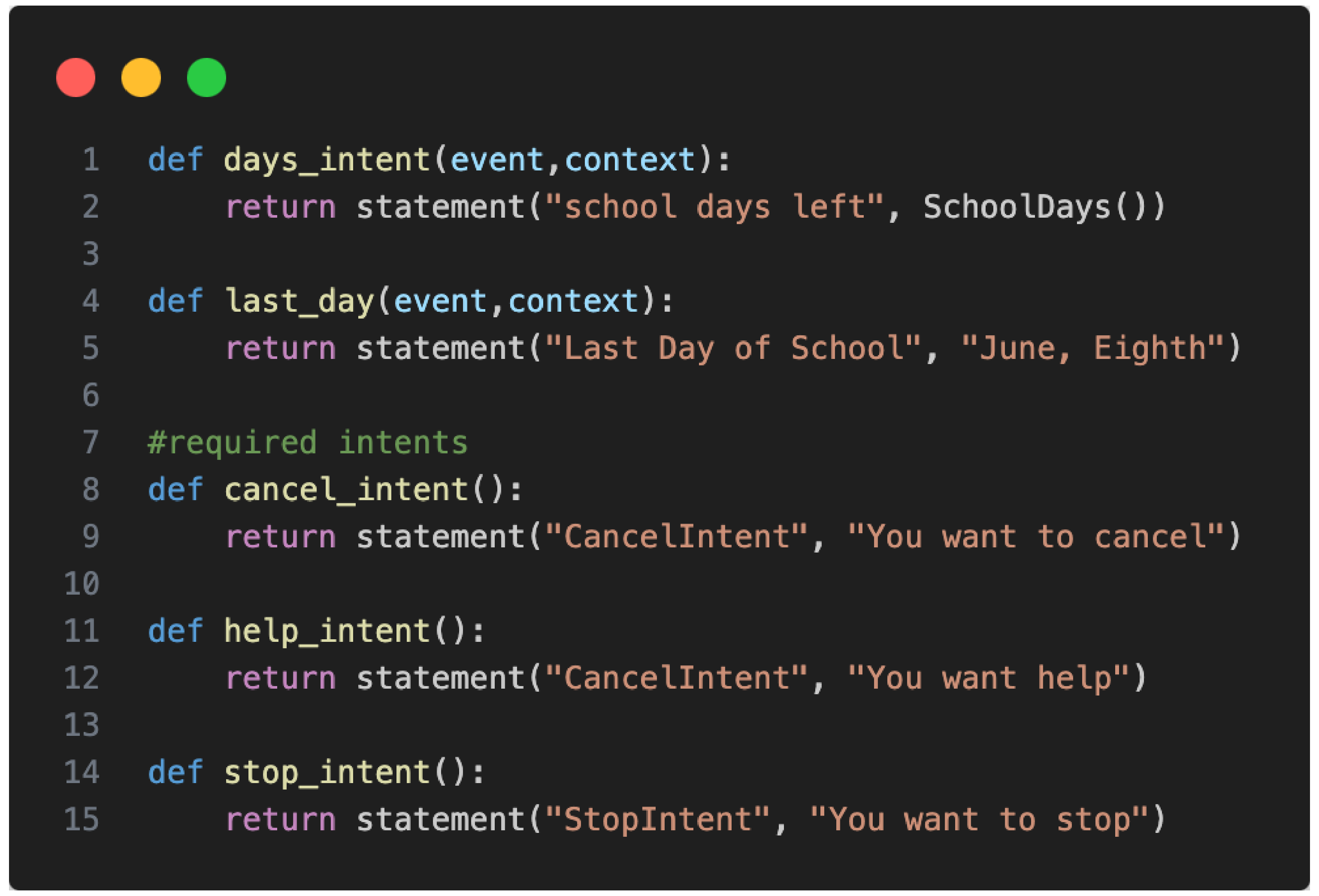
Nexus HSRP Configuration

How Many School Days Left

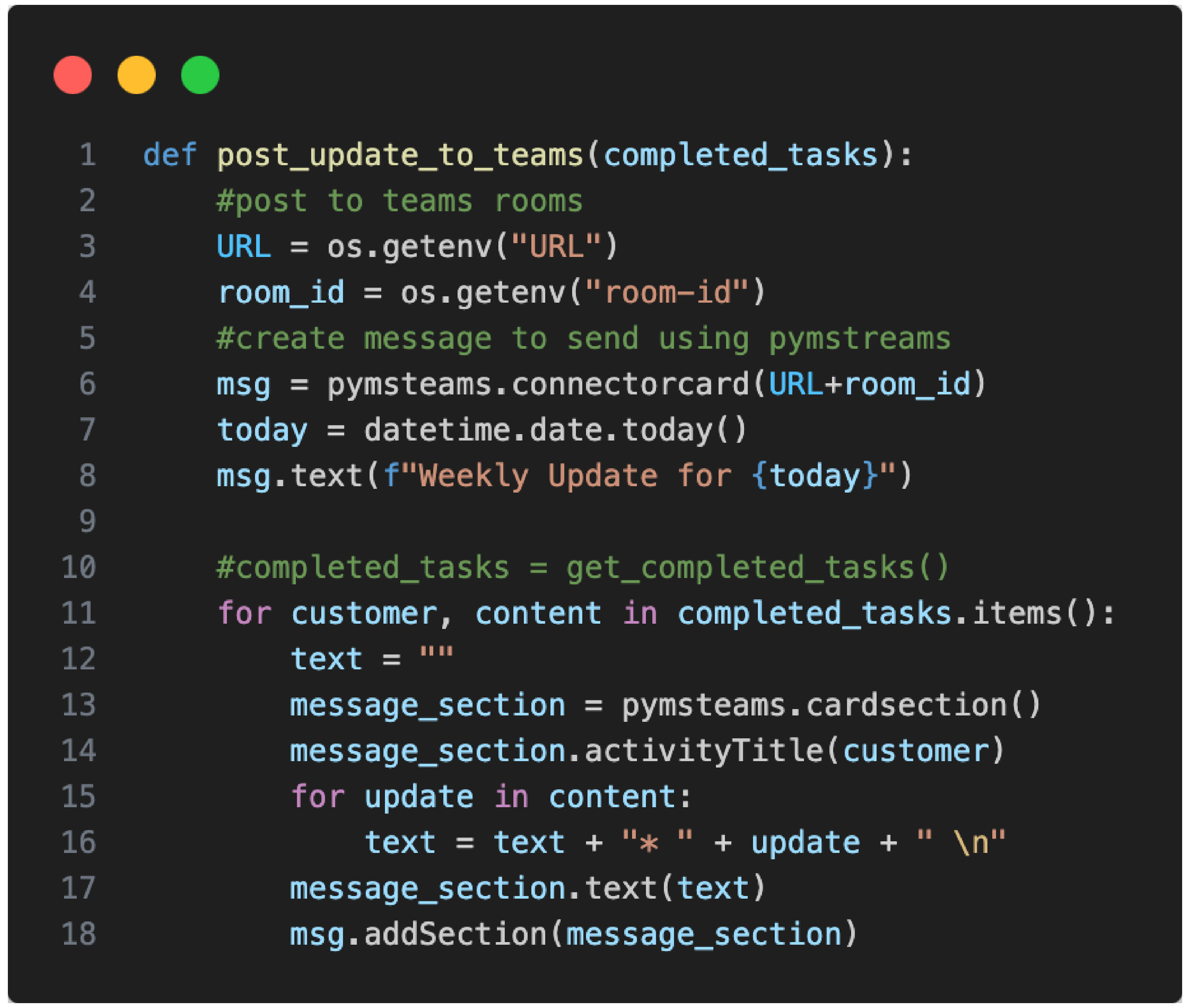
Teams Updates



```python
def hsrp_create(vl_num, vl_descr, vl_network):
    network_split = vl_network.split('.')
    vip = network_split[:]
    svi_1 = network_split[:]
    svi_2 = network_split[:]
    vip[3] = '1'
    h_vip = '.'.join(vip)
    svi_1[3] = '2'
    h_svi_active = '.'.join(svi_1)
    svi_2[3] = '3'
    h_svi_standby = '.'.join(svi_2)
```



```python
def days_intent(event,context):
    return statement("school days left", SchoolDays())

def last_day(event,context):
    return statement("Last Day of School", "June, Eighth")

#required intents
def cancel_intent():
    return statement("CancelIntent", "You want to cancel")

def help_intent():
    return statement("CancelIntent", "You want help")

def stop_intent():
    return statement("StopIntent", "You want to stop")
```



```python
def post_update_to_teams(completed_tasks):
    #post to teams rooms
    URL = os.getenv("URL")
    room_id = os.getenv("room-id")
    #create message to send using pymstreams
    msg = pymsteams.connectorcard(URL+room_id)
    today = datetime.date.today()
    msg.text(f"Weekly Update for {today}")

    #completed_tasks = get_completed_tasks()
    for customer, content in completed_tasks.items():
        text = ""
        message_section = pymsteams.cardsection()
        message_section.activityTitle(customer)
        for update in content:
            text = text + "* " + update + " \n"
        message_section.text(text)
        msg.addSection(message_section)
```

Evolution of teams and notes updates

Problem Statement:

Management: A highlight of activities to post to upstream management to show opportunity engagement and work tracking.

Evolution of teams and notes updates

Problem Statement:

Management: A highlight of activities to post to upstream management to show opportunity engagement and work tracking.

Conflict:

Management: I need you to do this extra work that has ambiguous intentions and no feedback loop.
Me: I don’t want to do that.

Evolution of teams and notes updates

Problem Statement:

Management: A highlight of activities to post to upstream management to show opportunity engagement and work tracking.

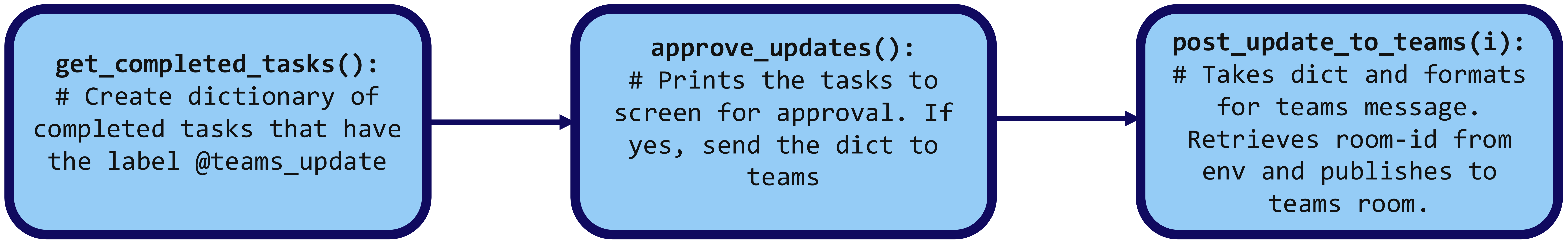
Conflict:

Management: I need you to do this extra work that has ambiguous intentions and no feedback loop.
Me: I don’t want to do that.

Resolution:

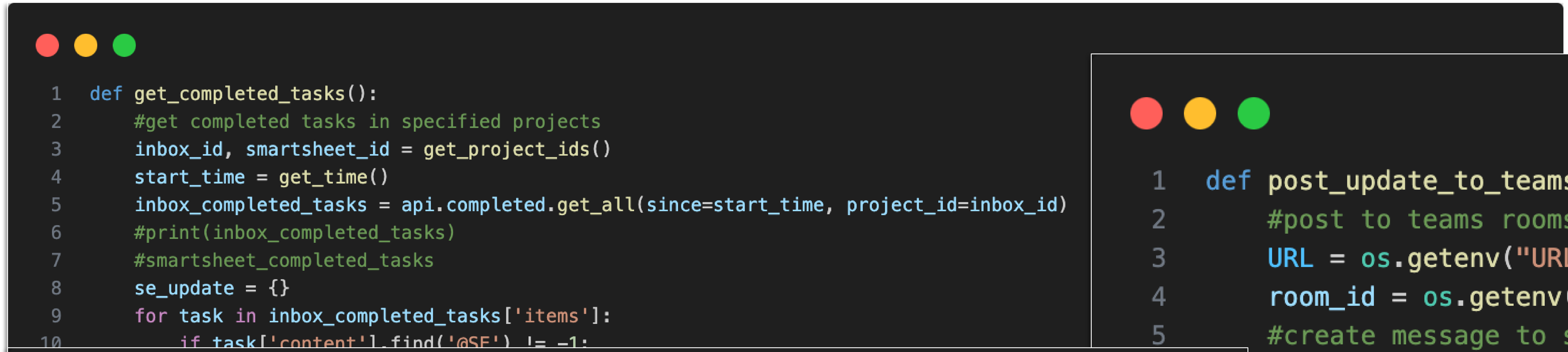
Me: I will build something that minimally complies, but requires little effort on my part.

Teams Update Workflow

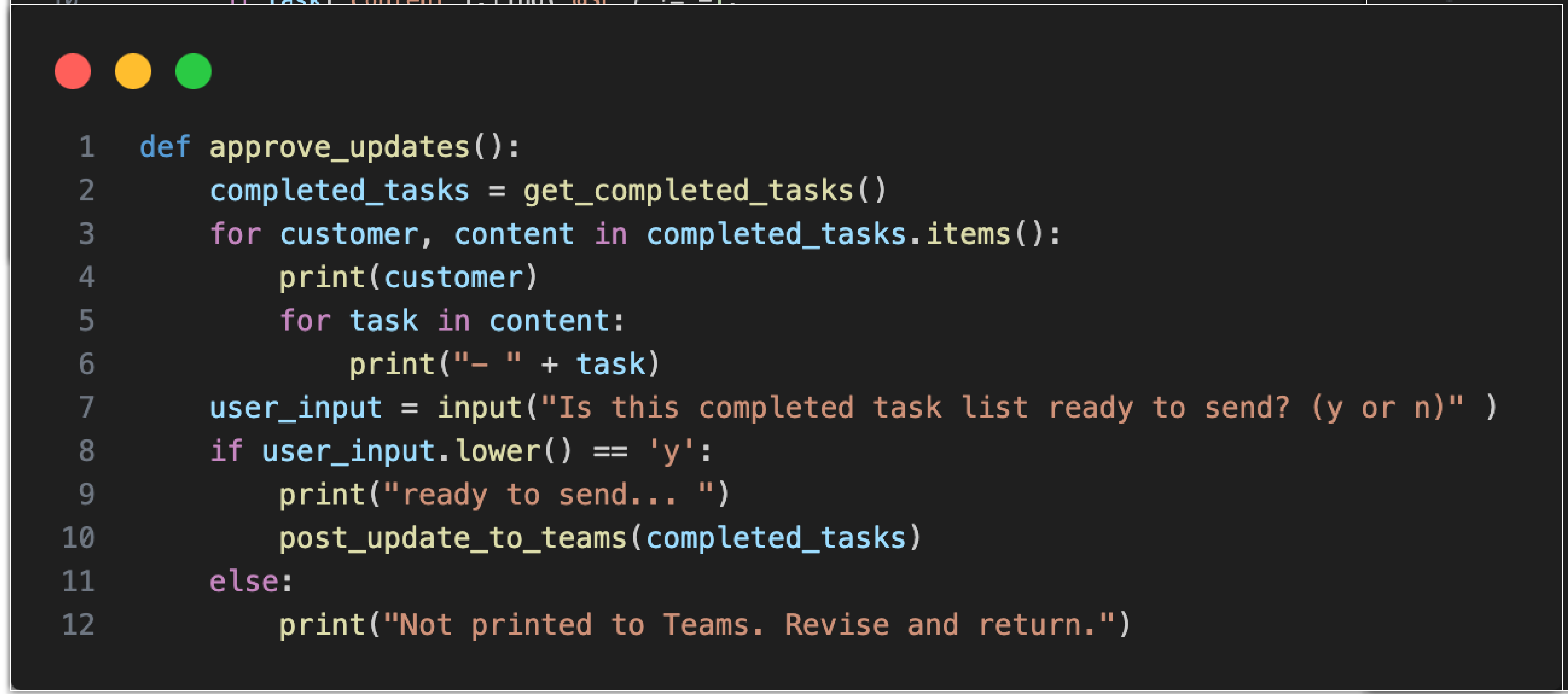


get_completed_tasks():
# Create dictionary of completed tasks that have the label @teams_update
approve_updates():
# Prints the tasks to screen for approval. If yes, send the dict to teams
post_update_to_teams(i):
# Takes dict and formats for teams message. Retrieves room-id from env and publishes to teams room.

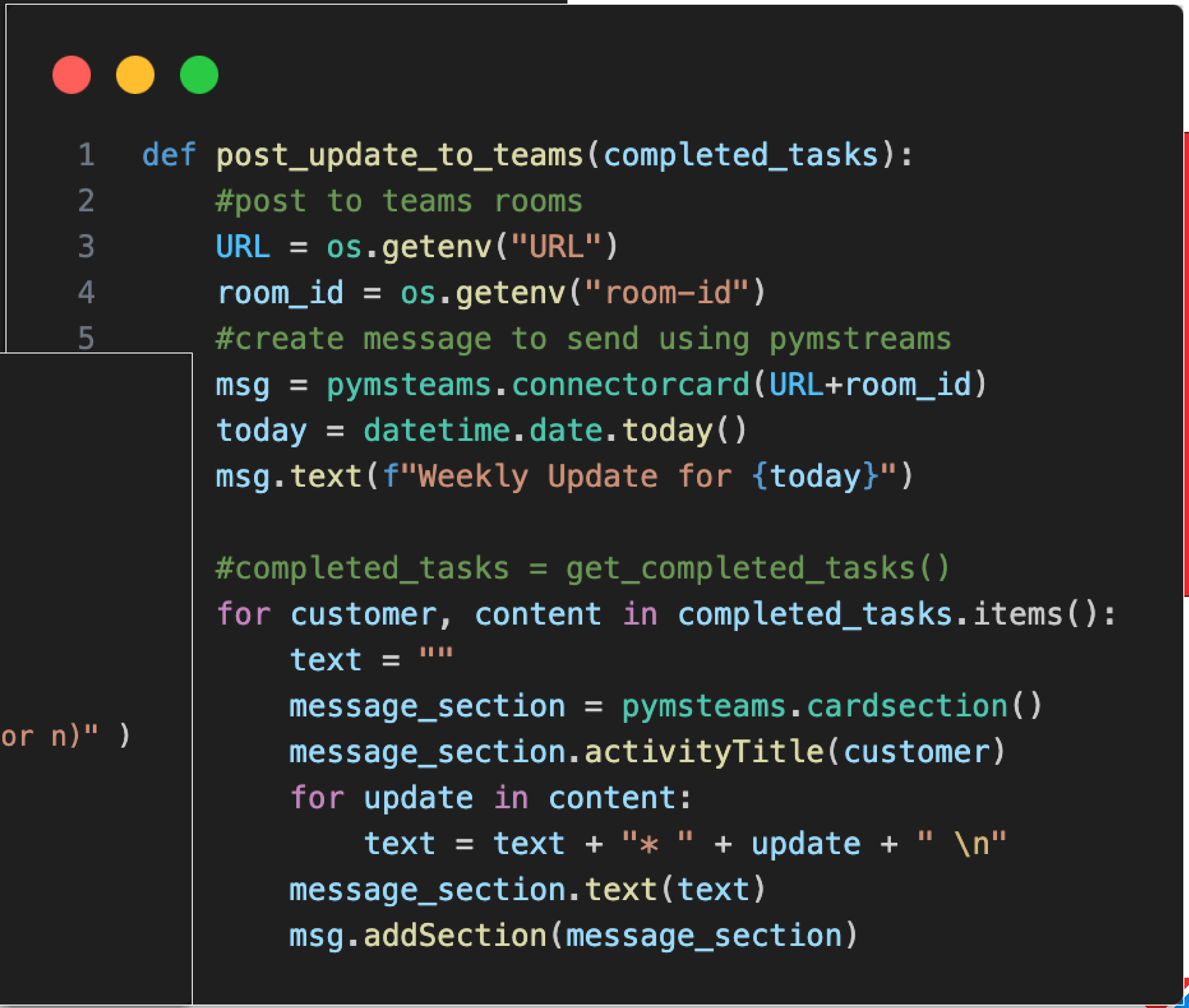
Teams Update Python App



```python
def get_completed_tasks():
    #get completed tasks in specified projects
    inbox_id, smartsheet_id = get_project_ids()
    start_time = get_time()
    inbox_completed_tasks = api.completed.get_all(since=start_time, project_id=inbox_id)
    #print(inbox_completed_tasks)
    #smartsheet_completed_tasks
    se_update = {}
    for task in inbox_completed_tasks['items']:
        if task['content'].find('@SE') != -1:
```



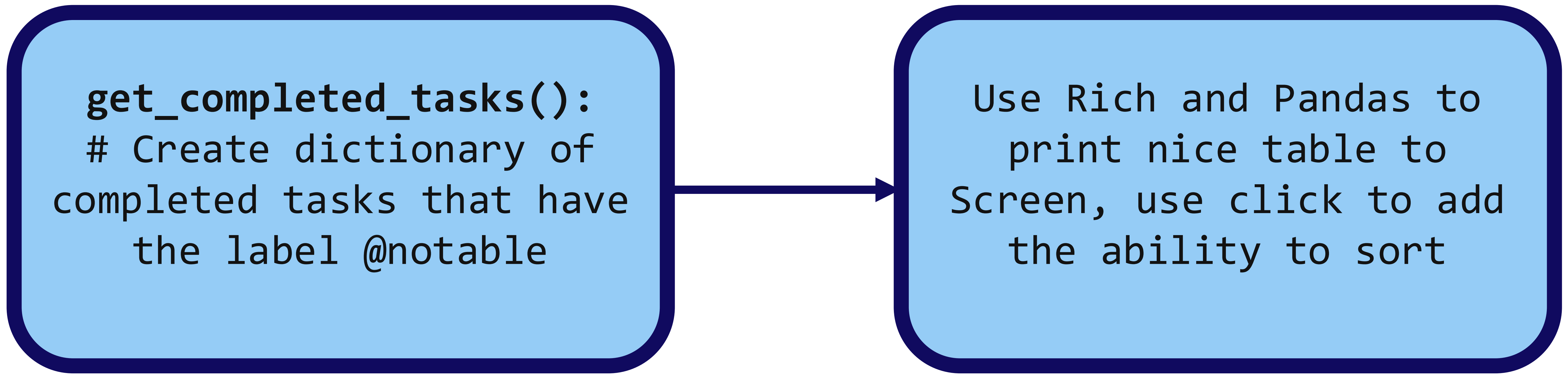
```python
def approve_updates():
    completed_tasks = get_completed_tasks()
    for customer, content in completed_tasks.items():
        print(customer)
        for task in content:
            print("- " + task)
    user_input = input("Is this completed task list ready to send? (y or n)" )
    if user_input.lower() == 'y':
        print("ready to send... ")
        post_update_to_teams(completed_tasks)
    else:
        print("Not printed to Teams. Revise and return.")
```



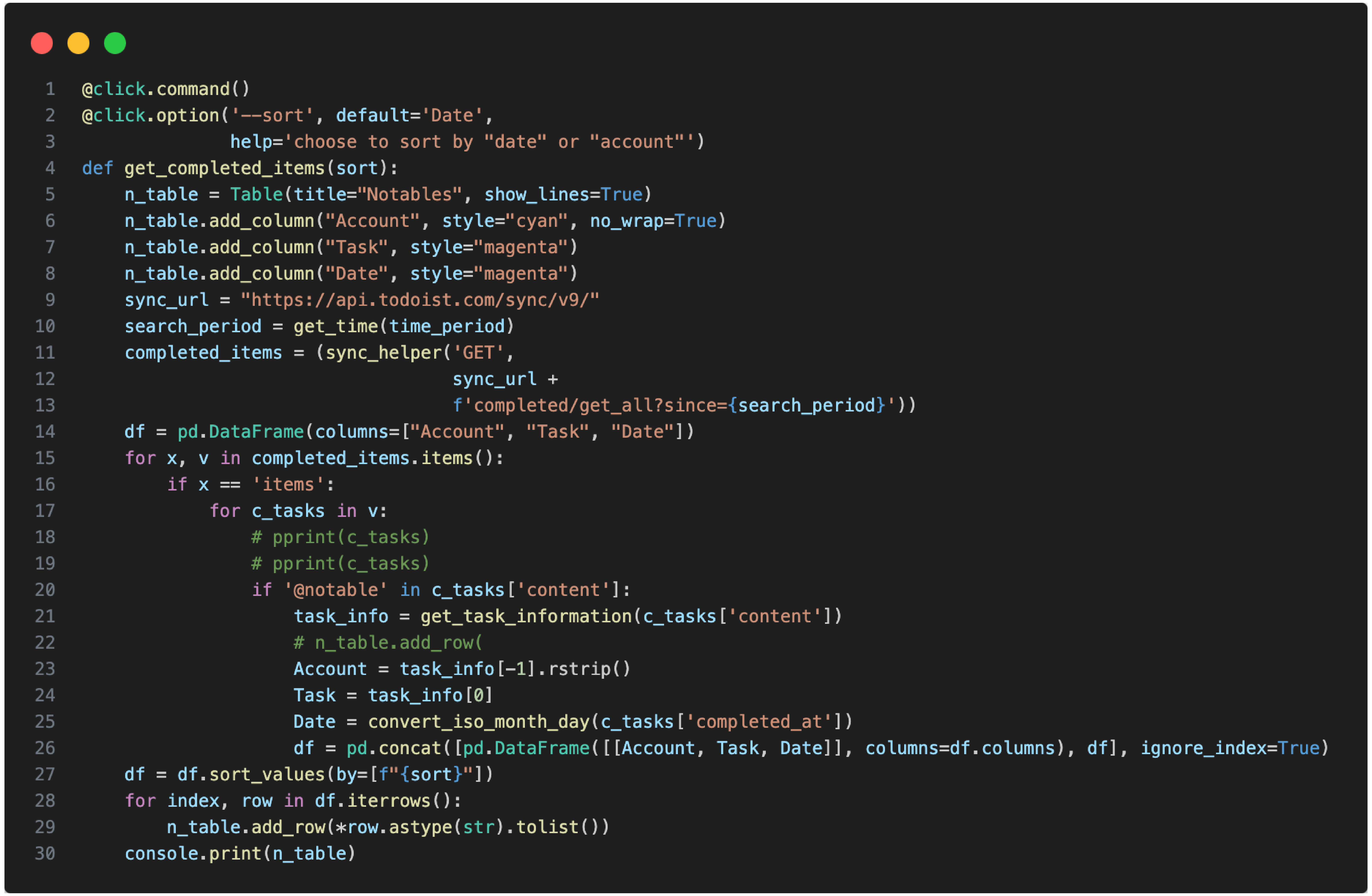
```python
def post_update_to_teams(completed_tasks):
    #post to teams rooms
    URL = os.getenv("URL")
    room_id = os.getenv("room-id")
    #create message to send using pymstreams
    msg = pymsteams.connectorcard(URL+room_id)
    today = datetime.date.today()
    msg.text(f"Weekly Update for {today}")

    #completed_tasks = get_completed_tasks()
    for customer, content in completed_tasks.items():
        text = ""
        message_section = pymsteams.cardsection()
        message_section.activityTitle(customer)
        for update in content:
            text = text + "* " + update + " \n"
        message_section.text(text)
        msg.addSection(message_section)
```

Notes Workflow

Requirements change. Post same info to Salesforce. Existing friction, can’t get API access to Salesforce.

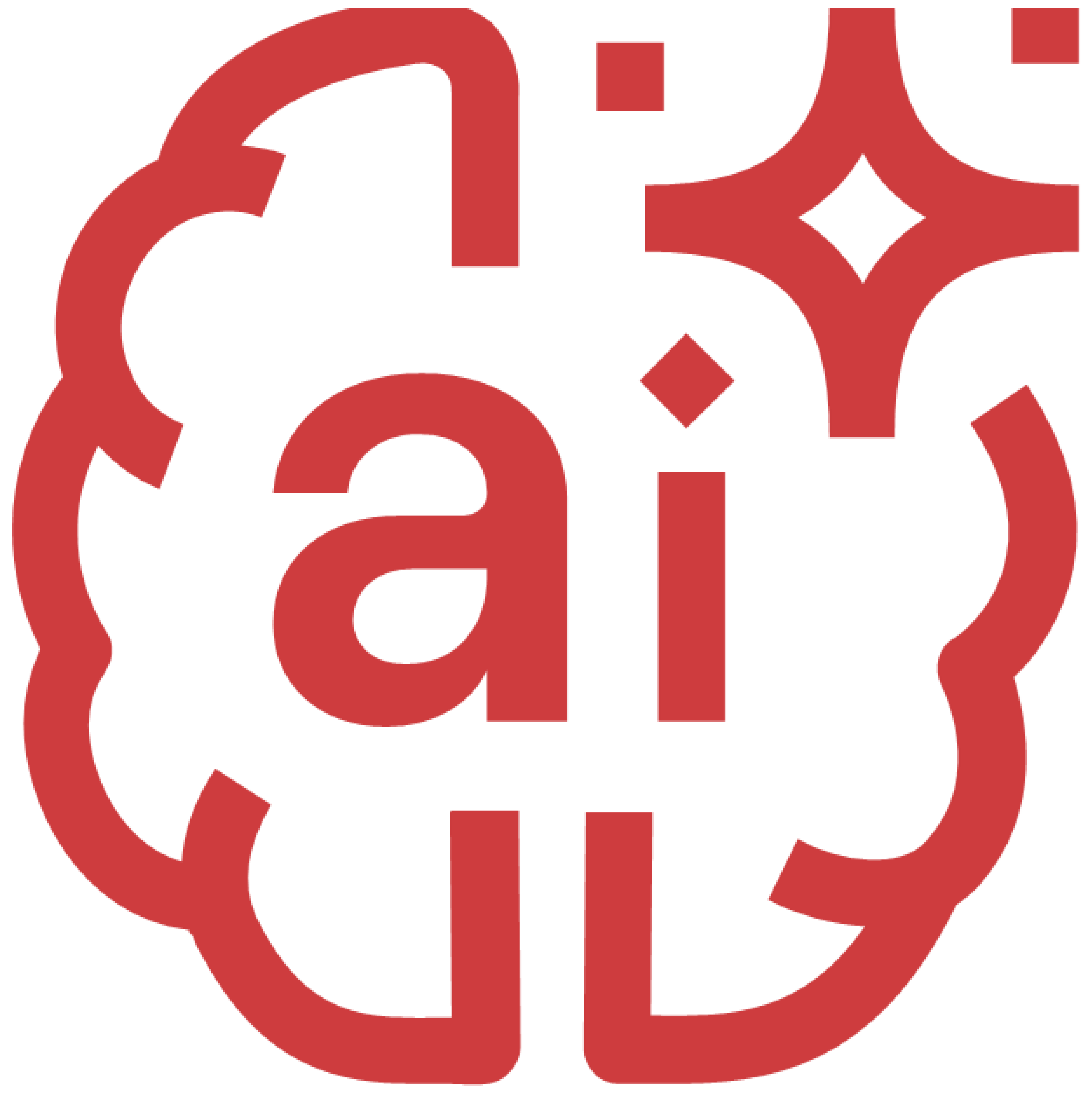

get_completed_tasks():
# Create dictionary of completed tasks that have the label @notable
Use Rich and Pandas to print nice table to Screen, use click to add the ability to sort

Example Code



```python
@click.command()
@click.option('--sort', default='Date',
              help='choose to sort by "date" or "account"')
def get_completed_items(sort):
    n_table = Table(title="Notables", show_lines=True)
    n_table.add_column("Account", style="cyan", no_wrap=True)
    n_table.add_column("Task", style="magenta")
    n_table.add_column("Date", style="magenta")
    sync_url = "https://api.todoist.com/sync/v9/"
    search_period = get_time(time_period)
    completed_items = (sync_helper('GET',
                                   sync_url +
                                   f'completed/get_all?since={search_period}'))
    df = pd.DataFrame(columns=["Account", "Task", "Date"])
    for x, v in completed_items.items():
        if x == 'items':
            for c_tasks in v:
                # pprint(c_tasks)
                # pprint(c_tasks)
                if '@notable' in c_tasks['content']:
                    task_info = get_task_information(c_tasks['content'])
                    # n_table.add_row(
                    Account = task_info[-1].rstrip()
                    Task = task_info[0]
                    Date = convert_iso_month_day(c_tasks['completed_at'])
                    df = pd.concat([pd.DataFrame([[Account, Task, Date]], columns=df.columns), df], ignore_index=True)
    df = df.sort_values(by=[f"{sort}"])
    for index, row in df.iterrows():
        n_table.add_row(*row.astype(str).tolist())
    console.print(n_table)
```

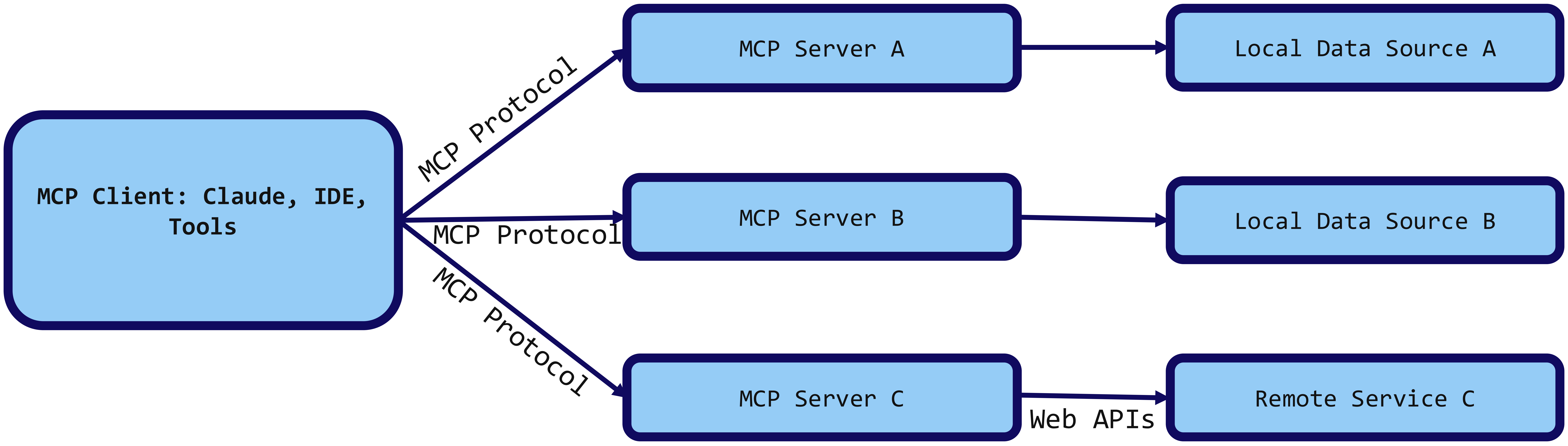
… And then there was AI



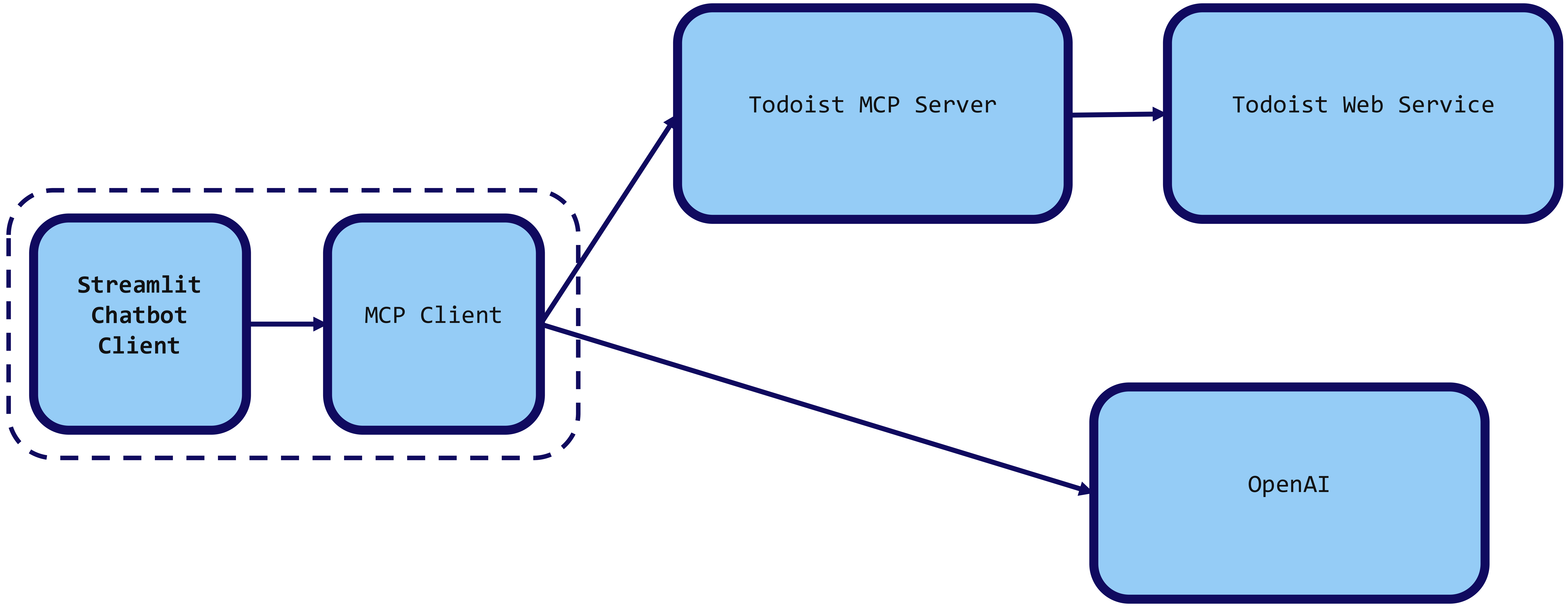
What is MCP?

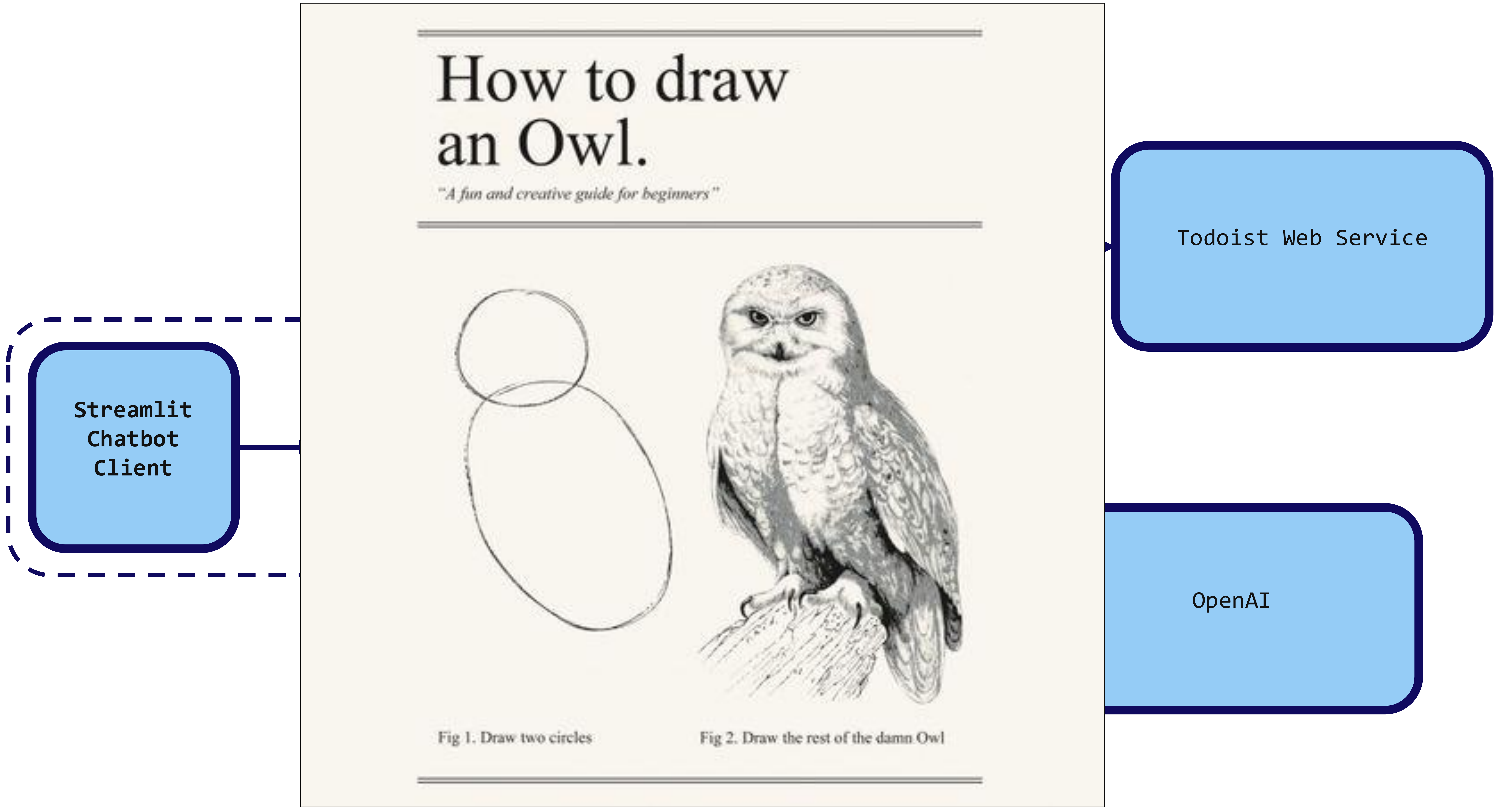
MCP Provides a standardized way to connect AI models to different data sources and tools.
MCP helps build agents and complex workflows on top of LLMs.

- modelcontextprotocol.io

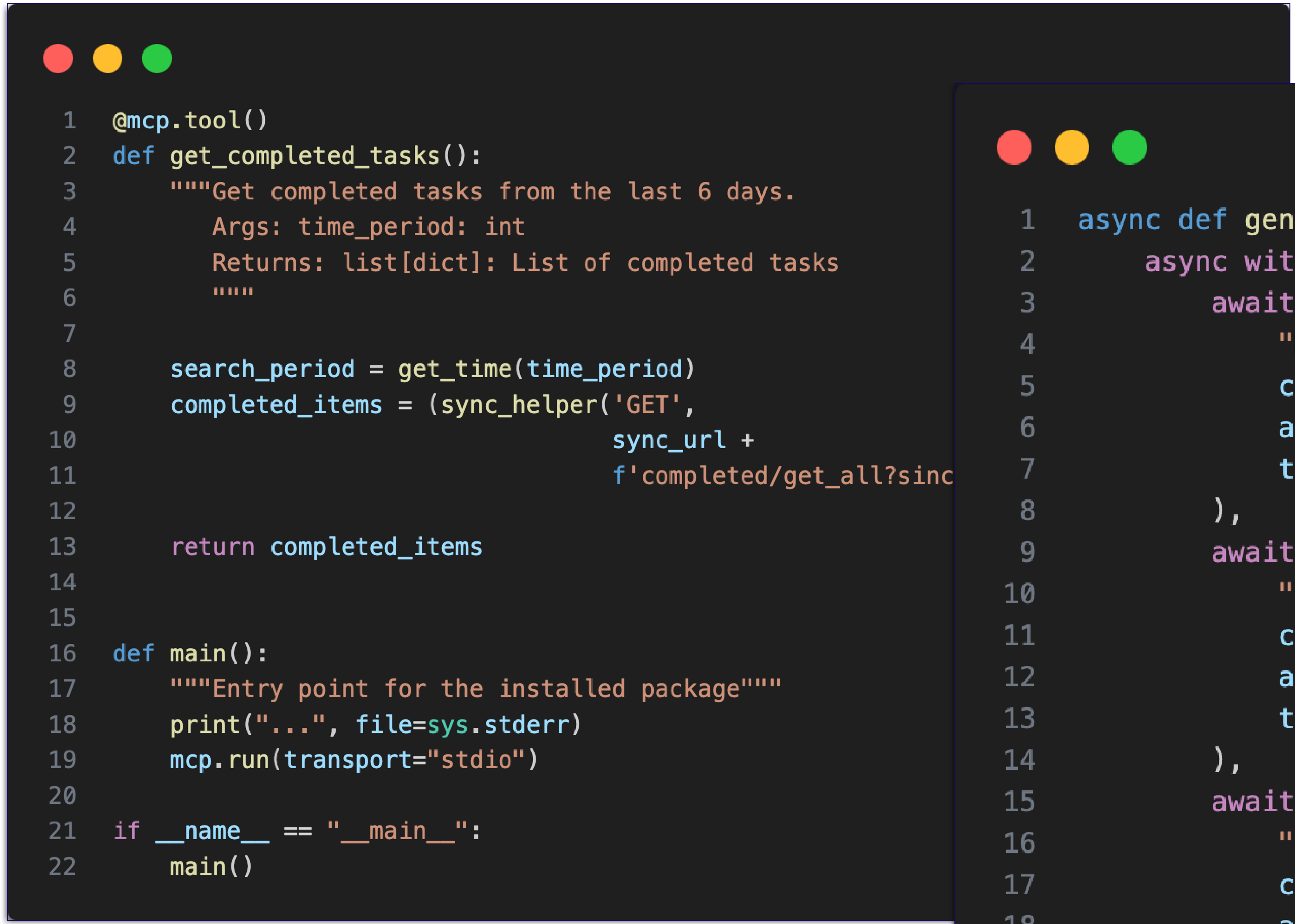


MCP Workflow

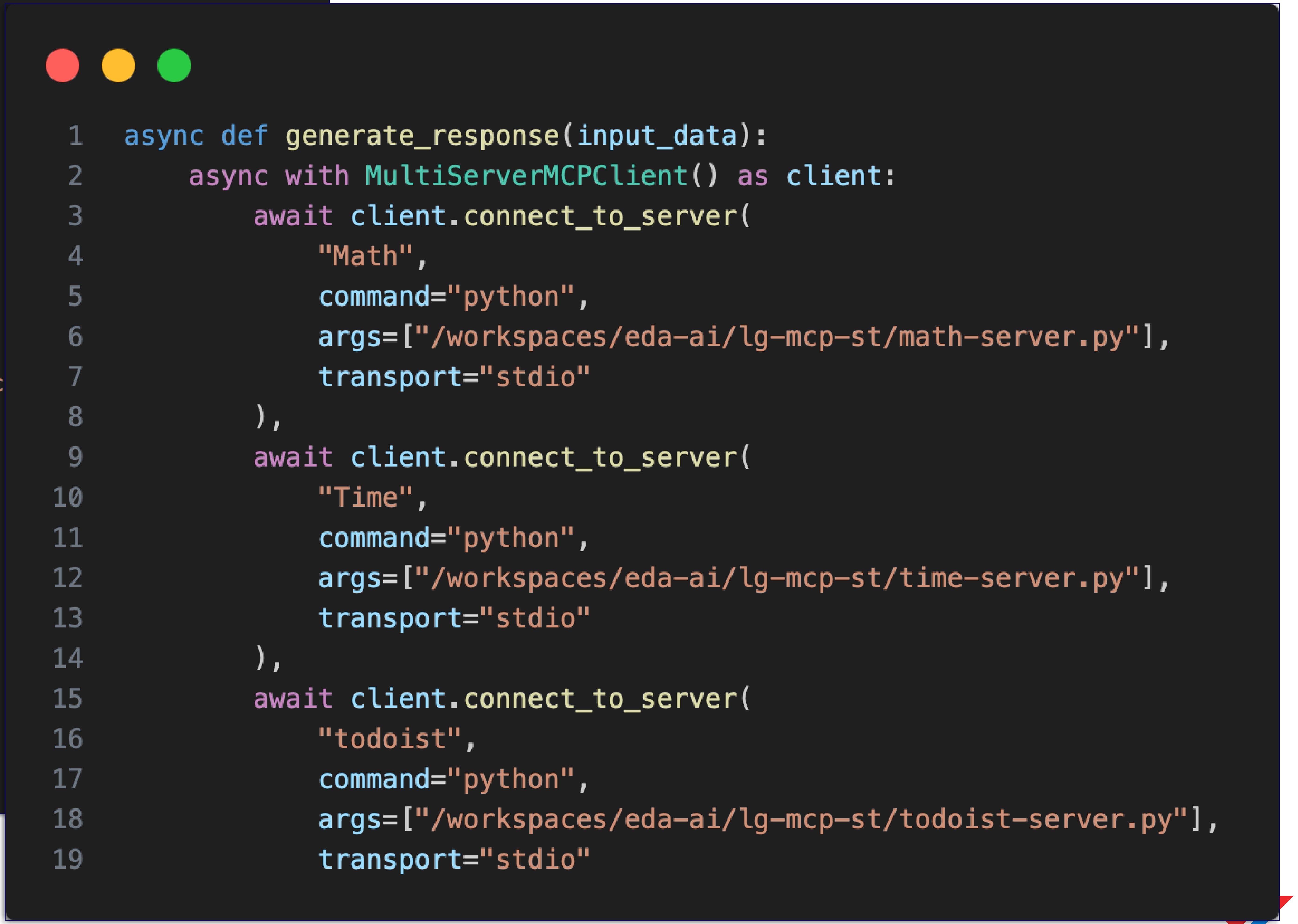



Streamlit Chatbot Client
How to draw an Owl.
"A fun and creative guide for beginners"
Fig 1. Draw two circles
Fig 2. Draw the rest of the damn Owl
Todoist Web Service
OpenAI

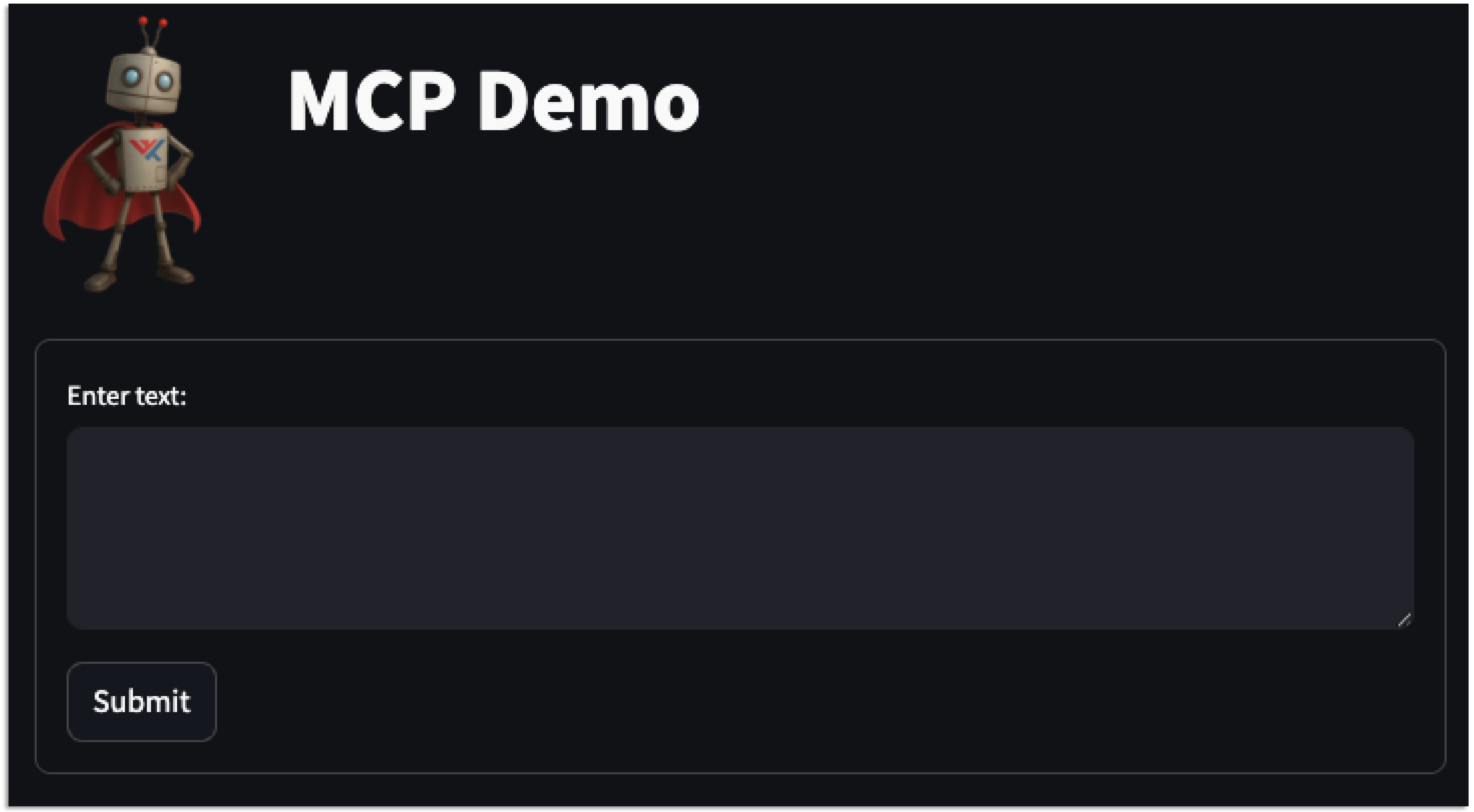
Todoist MCP Example



```python
@mcp.tool()
def get_completed_tasks():
    """Get completed tasks from the last 6 days.
       Args: time_period: int
       Returns: list[dict]: List of completed tasks
       """

    search_period = get_time(time_period)
    completed_items = (sync_helper('GET',
                                   sync_url +
                                   f'completed/get_all?sinc


    return completed_items


def main():
    """Entry point for the installed package"""
    print("...", file=sys.stderr)
    mcp.run(transport="stdio")

if __name__ == "__main__":
    main()
```

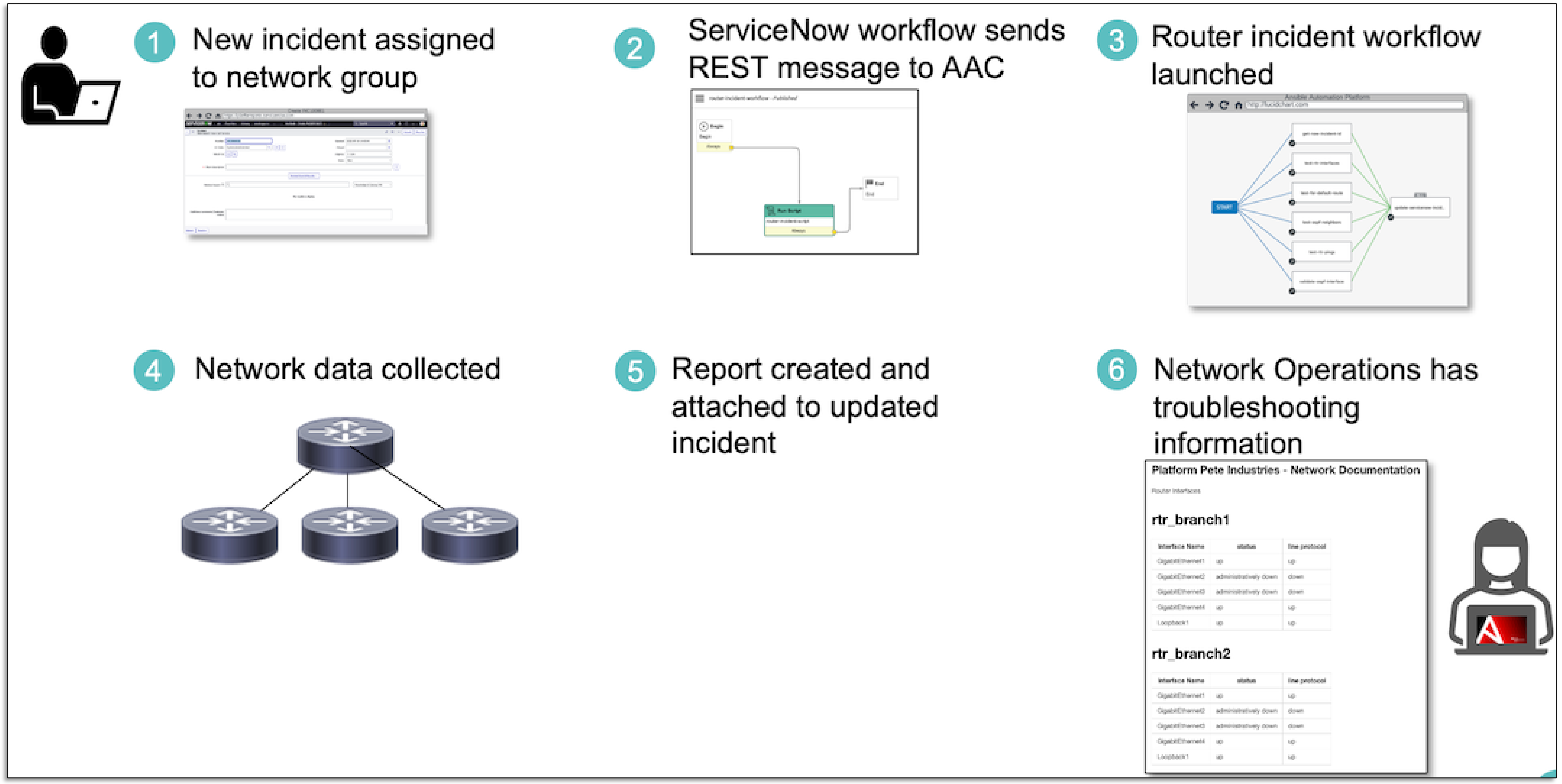


```python
async def generate_response(input_data):
    async with MultiServerMCPClient() as client:
        await client.connect_to_server(
            "Math",
            command="python",
            args=["/workspaces/eda-ai/lg-mcp-st/math-server.py"],
            transport="stdio"
        ),
        await client.connect_to_server(
            "Time",
            command="python",
            args=["/workspaces/eda-ai/lg-mcp-st/time-server.py"],
            transport="stdio"
        ),
        await client.connect_to_server(
            "todoist",
            command="python",
            args=["/workspaces/eda-ai/lg-mcp-st/todoist-server.py"],
            transport="stdio"
```


MCP Demo
Enter text:
Submit

Enterprise Examples

Enterprise solutions


1 New incident assigned to network group
2 ServiceNow workflow sends REST message to AAC
3 Router incident workflow launched
START
4 Network data collected
5 Report created and attached to updated incident
6 Network Operations has troubleshooting information

Platform Pete Industries - Network Documentation

Router interfaces

rtr_branch1

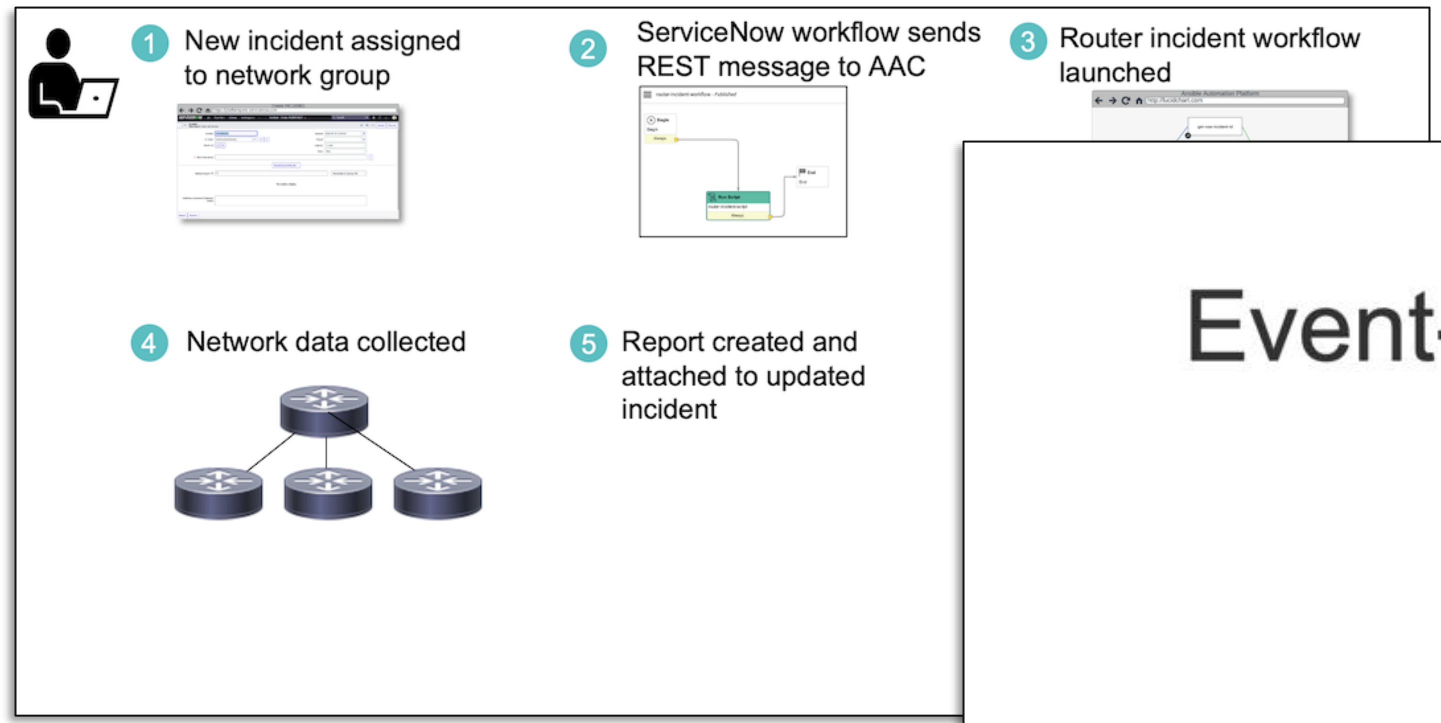
Interface Name	status	line protocol
GigabitEthernet1	up	up
GigabitEthernet2	administratively down	down
GigabitEthernet3	administratively down	down
GigabitEthernet4	up	up
Loopback1	up	up

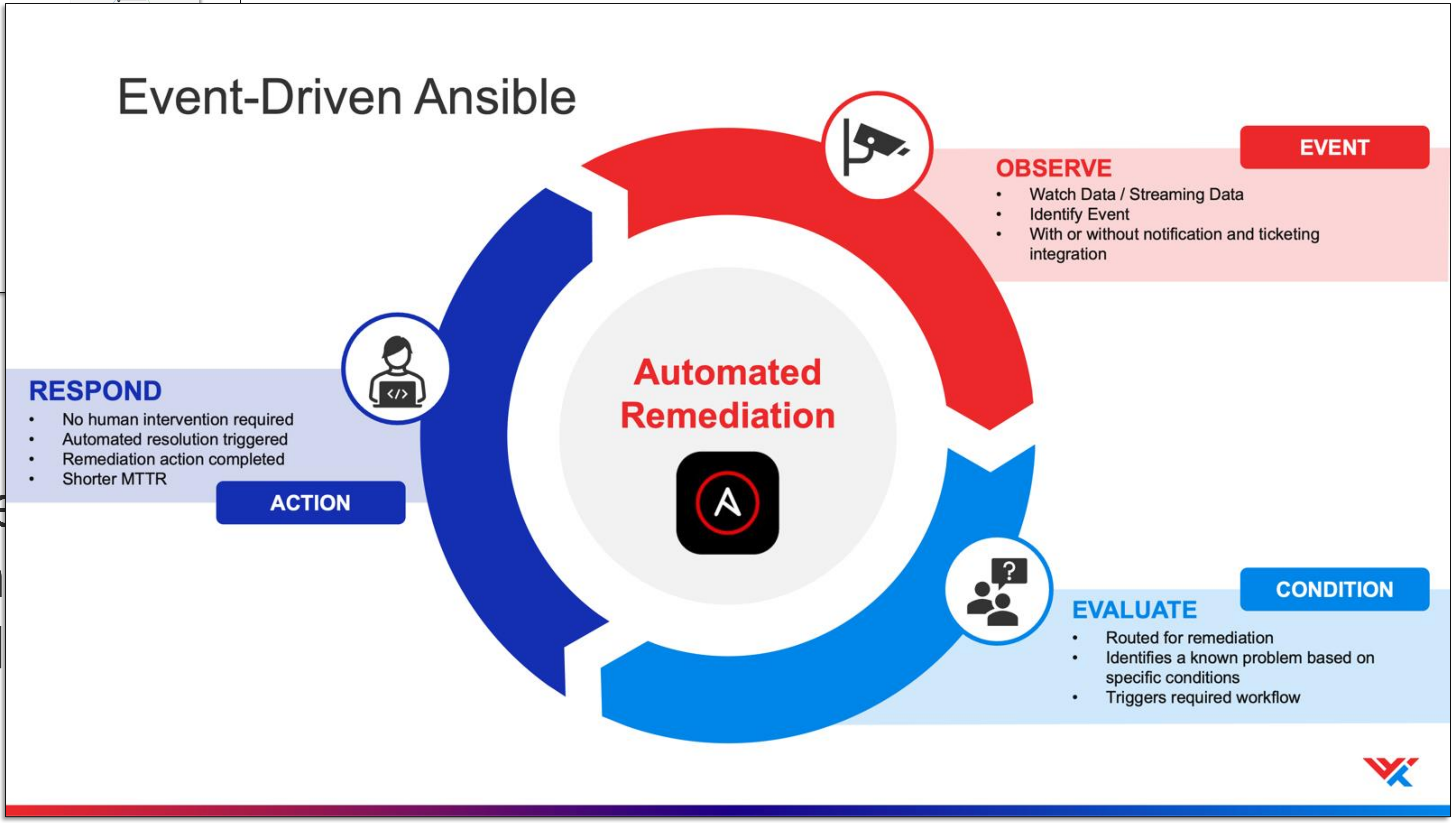
rtr_branch2

Interface Name	status	line protocol
GigabitEthernet1	up	up
GigabitEthernet2	administratively down	down
GigabitEthernet3	administratively down	down
GigabitEthernet4	up	up
Loopback1	up	up

Automated First-level actions (troubleshooting, escalation, remediation)

Enterprise solutions


1 New incident assigned to network group
2 ServiceNow workflow sends REST message to AAC
3 Router incident workflow launched
4 Network data collected
5 Report created and attached to updated incident


Event-Driven Ansible

OBSERVE
- Watch Data / Streaming Data
- Identify Event
- With or without notification and ticketing integration

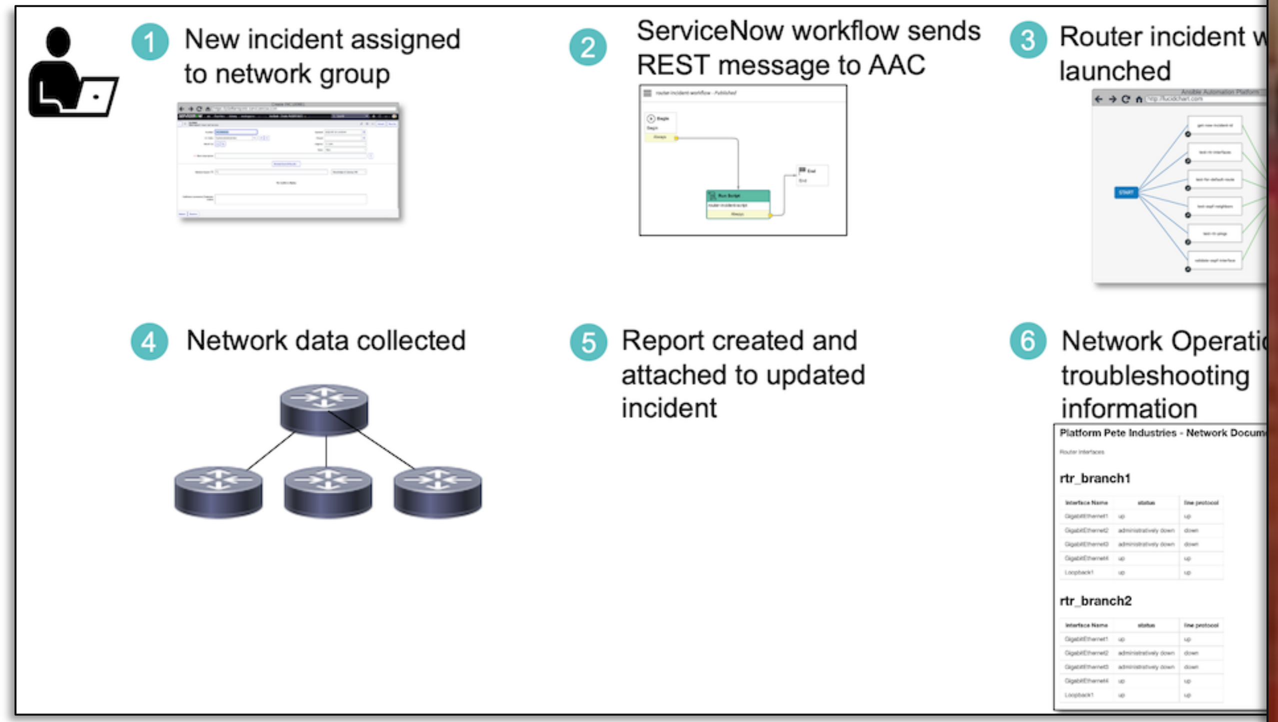
EVALUATE
- Routed for remediation
- Identifies a known problem based on specific conditions
- Triggers required workflow

RESPOND
- No human intervention required
- Automated resolution triggered
- Remediation action completed
- Shorter MTTR

Automated First-le
actions (troublesh
escalation, remed

Event-based automation.
The ability to take action
based on external data.

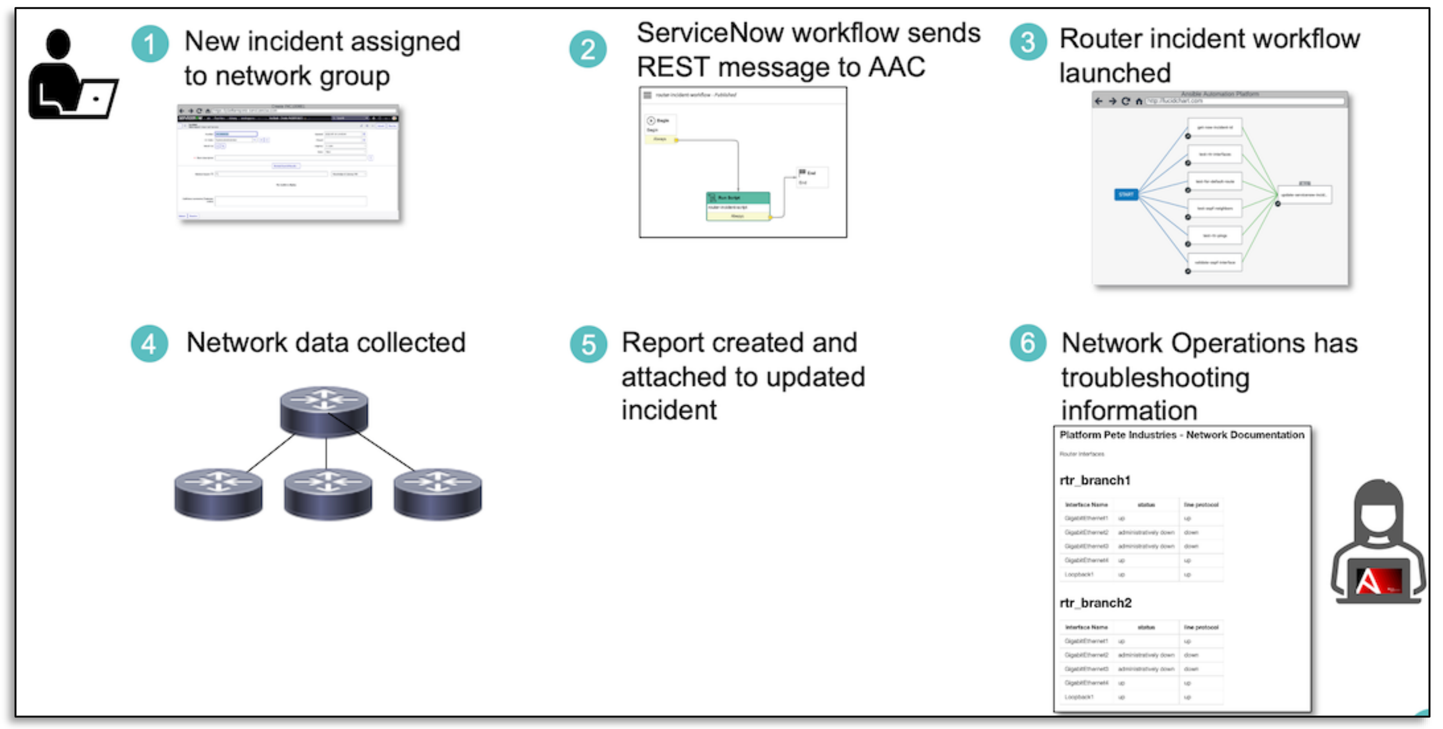
Enterprise solutions

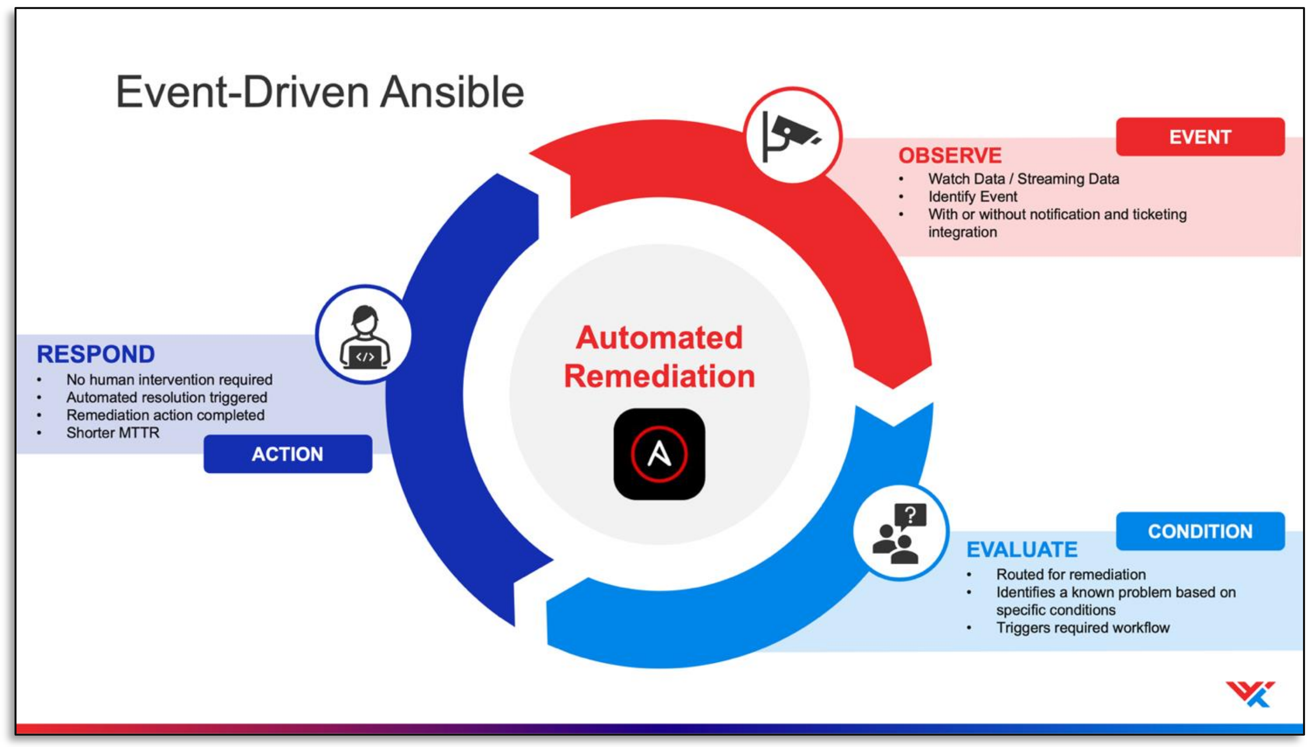

1 New incident assigned to network group
2 ServiceNow workflow sends REST message to AAC
3 Router incident w launched
4 Network data collected
5 Report created and attached to updated incident
6 Network Operati troubleshooting information
rtr_branch1
rtr_branch2

Automated First-level actions (troubleshooting escalation, remediation)

Answering questions from end-users about the actions of a ruleset

Enterprise solutions


1 New incident assigned to network group
2 ServiceNow workflow sends REST message to AAC
3 Router incident workflow launched
4 Network data collected
5 Report created and attached to updated incident
6 Network Operations has troubleshooting information
rtr_branch1
rtr_branch2

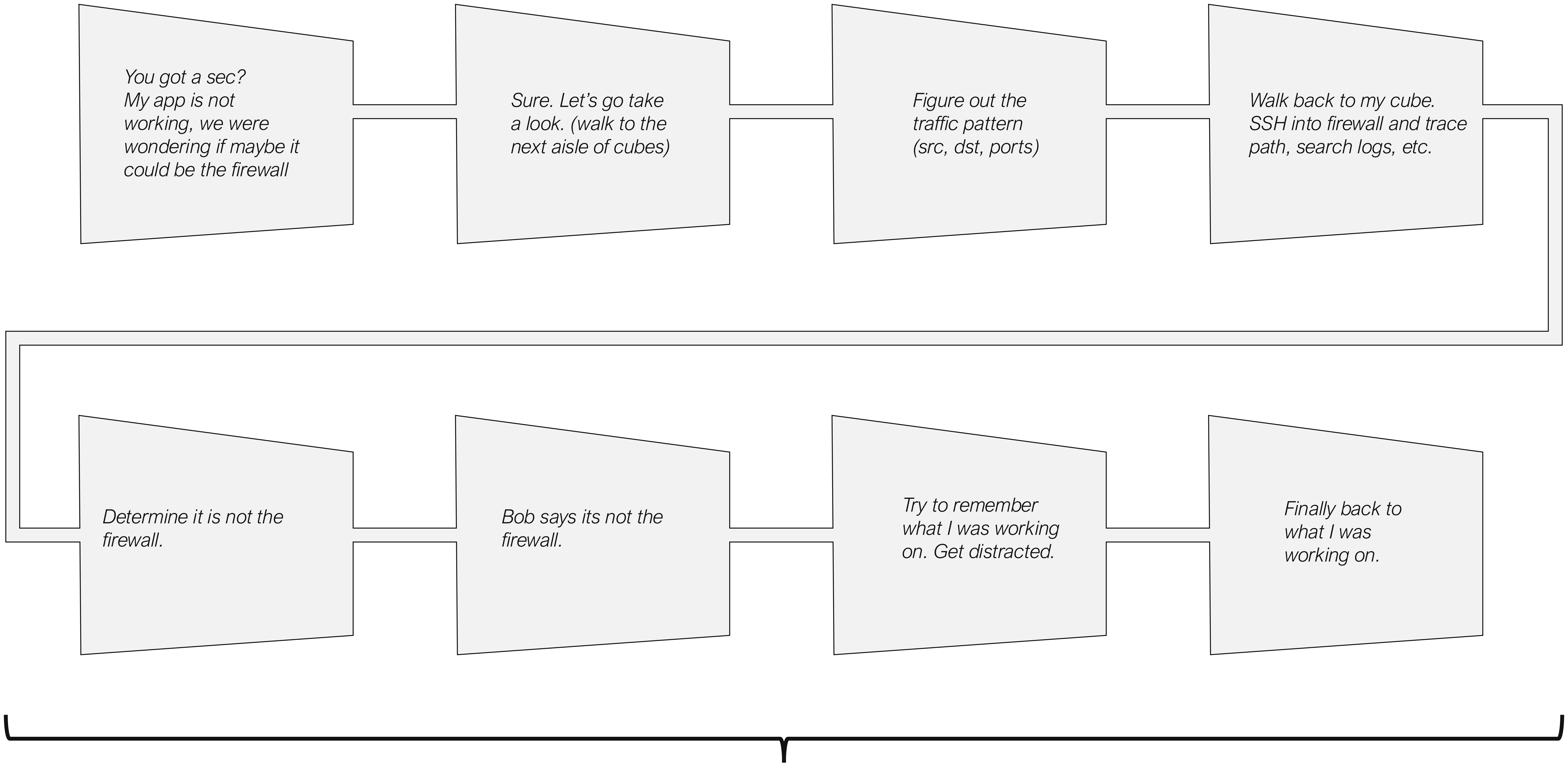

Event-Driven Ansible
Automated Remediation
EVENT
OBSERVE
Watch Data / Streaming Data
Identify Event
With or without notification and ticketing integration
CONDITION
EVALUATE
Routed for remediation
Identifies a known problem based on specific conditions
Triggers required workflow
RESPOND
No human intervention required
Automated resolution triggered
Remediation action completed
Shorter MTTR
ACTION

Automated First-level actions (troubleshooting, escalation, remediation)

Event-based automation. The ability to take action based on external data.

Answering questions from end-users about the actions of a ruleset

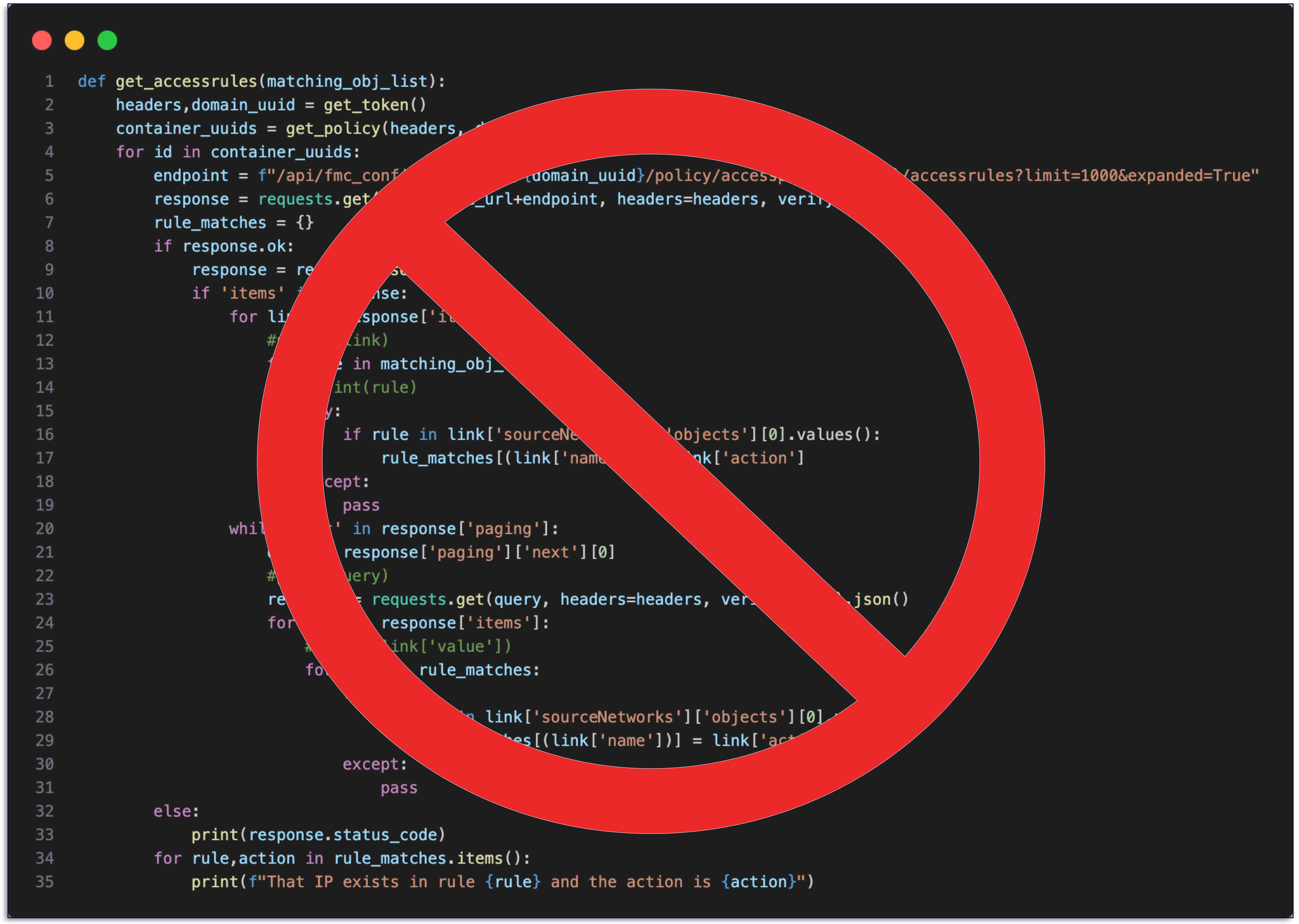
Answering Questions About Firewalls


You got a sec? My app is not working, we were wondering if maybe it could be the firewall
Sure. Let's go take a look. (walk to the next aisle of cubes)
Figure out the traffic pattern (src, dst, ports)
Walk back to my cube. SSH into firewall and trace path, search logs, etc.
Determine it is not the firewall.
Bob says its not the firewall.
Try to remember what I was working on. Get distracted.
Finally back to what I was working on.

One Hour

Evolution of It’s Not The Firewall

Brute Force Coding



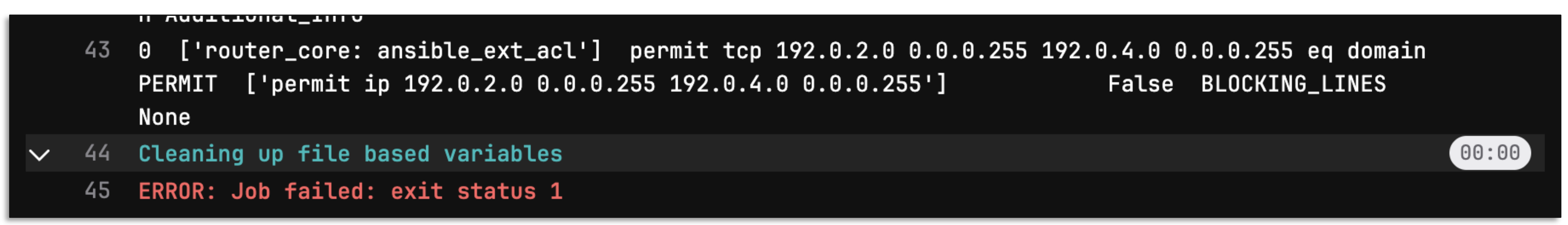
```
def get_accessrules(matching_obj_list):
    headers,domain_uuid = get_token()
    container_uuids = get_policy(headers,
    for id in container_uuids:
        endpoint = f"/api/fmc_conf          {domain_uuid}/policy/access          /accessrules?limit=1000&expanded=True"
        response = requests.get          _url+endpoint, headers=headers, verif
        rule_matches = {}
        if response.ok:
            response = re
            if 'items'        se:
                for li      sponse['i
                    #        ink)
                            in matching_obj_
                            int(rule)
                           :
                            if rule in link['sourceNe          'objects'][0].values():
                                rule_matches[(link['name          nk['action']
                           cept:
                            pass
                whil         ' in response['paging']:
                            response['paging']['next'][0]
                    #        ery)
                    re         = requests.get(query, headers=headers, ver          .json()
                    for            response['items']:
                                 ink['value'])
                        fo           rule_matches:

                                  link['sourceNetworks']['objects'][0]
                                 es[(link['name'])] = link['ac
                            except:
                                pass
        else:
            print(response.status_code)
        for rule,action in rule_matches.items():
            print(f"That IP exists in rule {rule} and the action is {action}")
```

- Took A Long Time
- Fragile Code
- Ultimately Abandoned

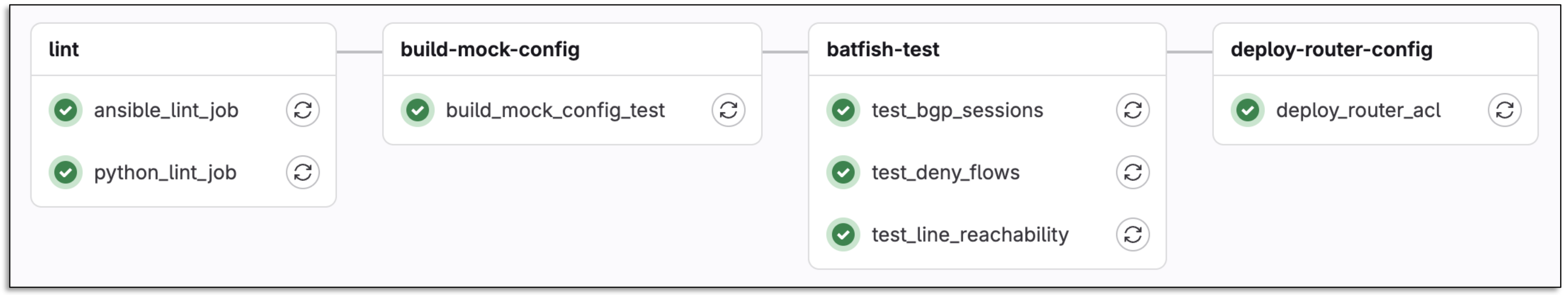
Evolution of It's Not The firewall

Batfish

CI/CD



```
0  ['router_core: ansible_ext_acl']  permit tcp 192.0.2.0 0.0.0.255 192.0.4.0 0.0.0.255 eq domain
PERMIT   ['permit ip 192.0.2.0 0.0.0.255 192.0.4.0 0.0.0.255']            False  BLOCKING_LINES
None
Cleaning up file based variables                                                          00:00
ERROR: Job failed: exit status 1
```



lint	build-mock-config	batfish-test	deploy-router-config
ansible_lint_job	build_mock_config_test	test_bgp_sessions	deploy_router_acl
python_lint_job		test_deny_flows	
		test_line_reachability	

Slackbot

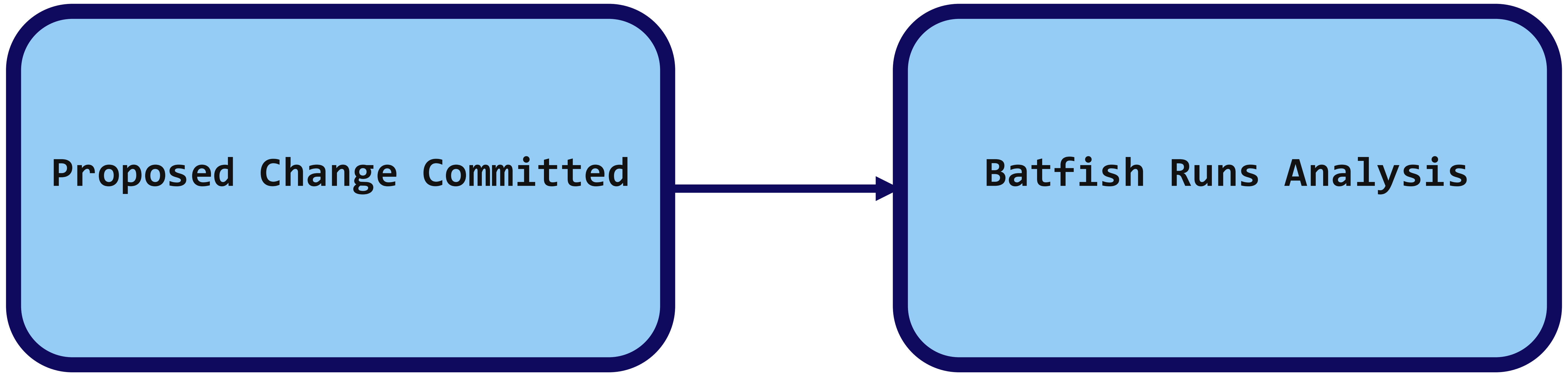
What is Batfish?

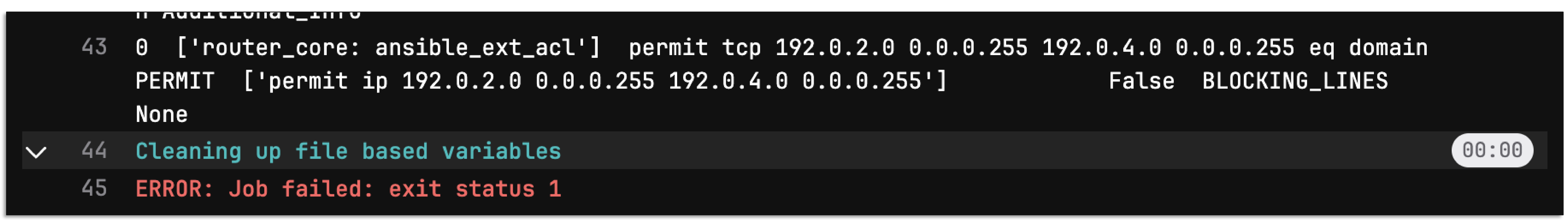
AWS-managed open source configuration analysis tool

Allows you to programmatically ask questions of a particular network configuration

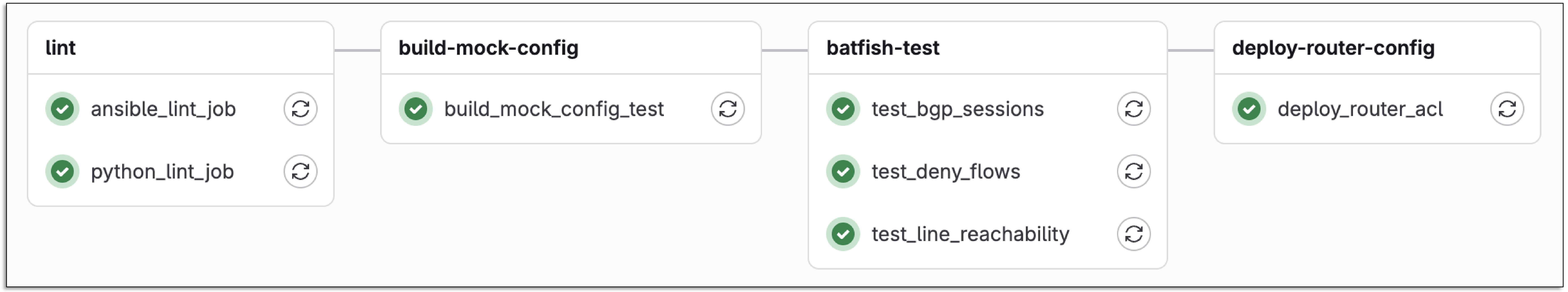
- Will the BGP sessions be established correctly?
- Is a certain network reachable?
- Does an ACL allow certain traffic?

Gitlab CI/CD Workflow

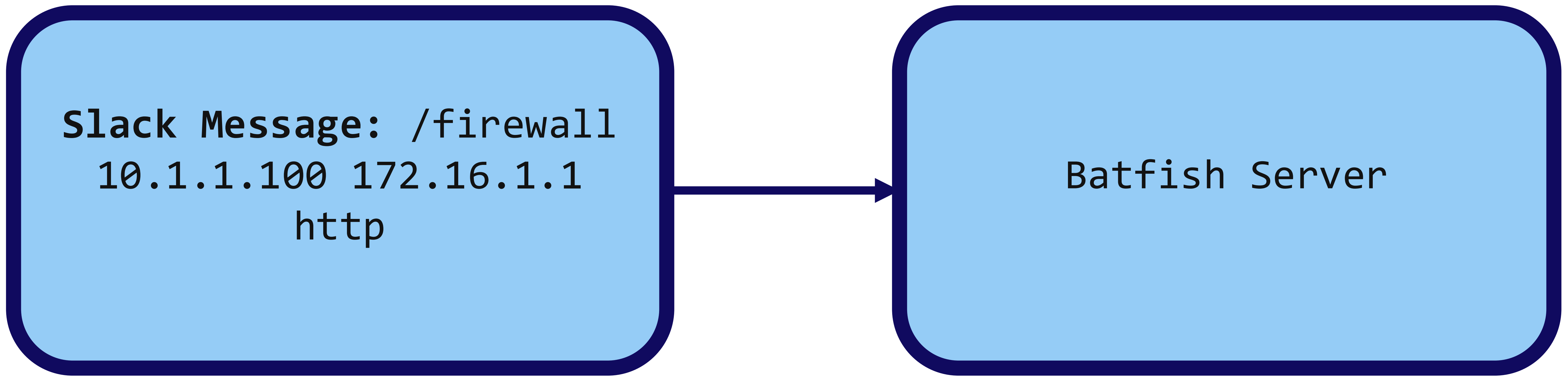




```
0  ['router_core: ansible_ext_acl']  permit tcp 192.0.2.0 0.0.0.255 192.0.4.0 0.0.0.255 eq domain
PERMIT  ['permit ip 192.0.2.0 0.0.0.255 192.0.4.0 0.0.0.255']            False  BLOCKING_LINES
None
Cleaning up file based variables                                                              00:00
ERROR: Job failed: exit status 1
```



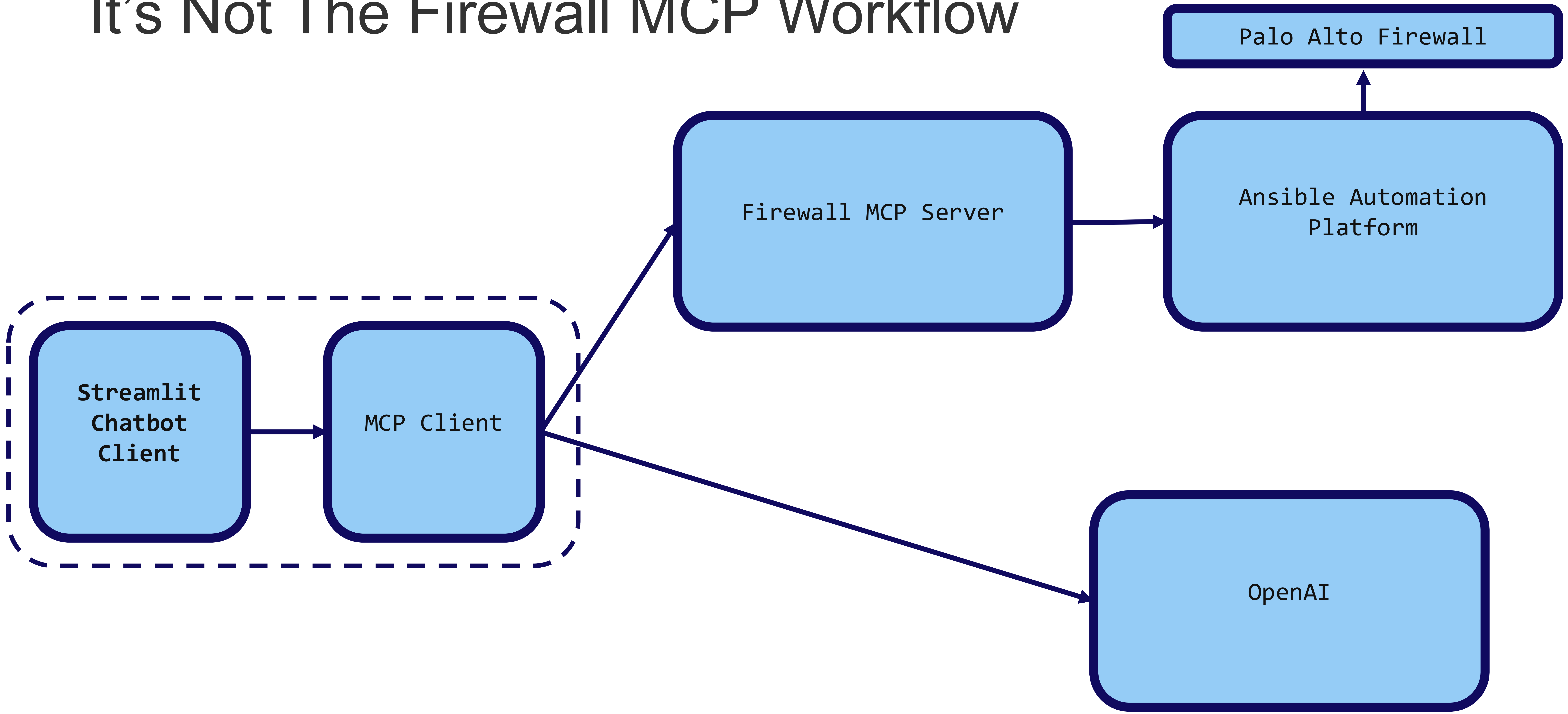
It's Not The Firewall Slackbot Workflow



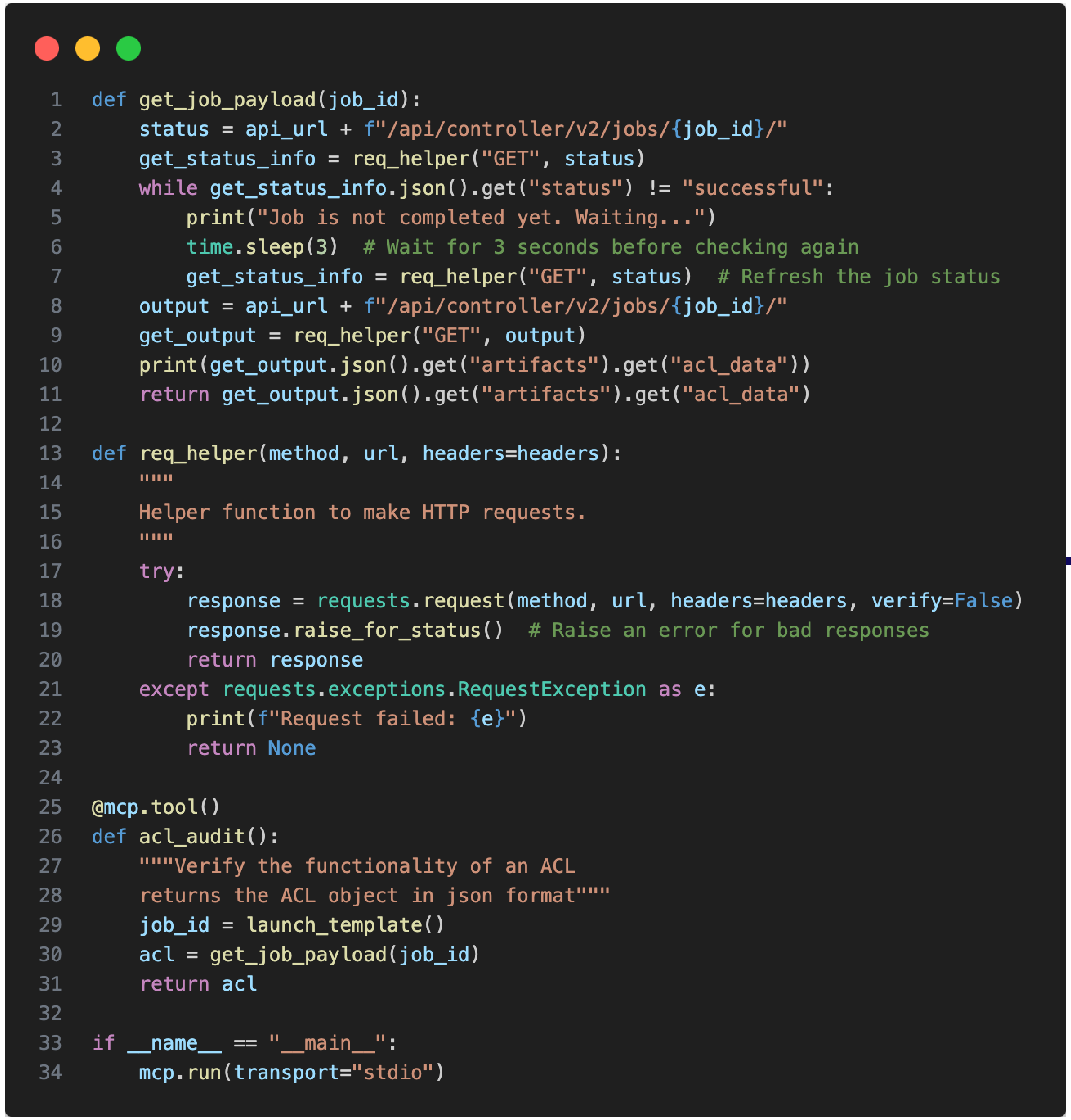
It’s Not The Firewall Slackbot Workflow

Slack Message: /firewall 10.1.1.100 172.16.1.1 http
Batfish Server

It’s Not The Firewall MCP Workflow

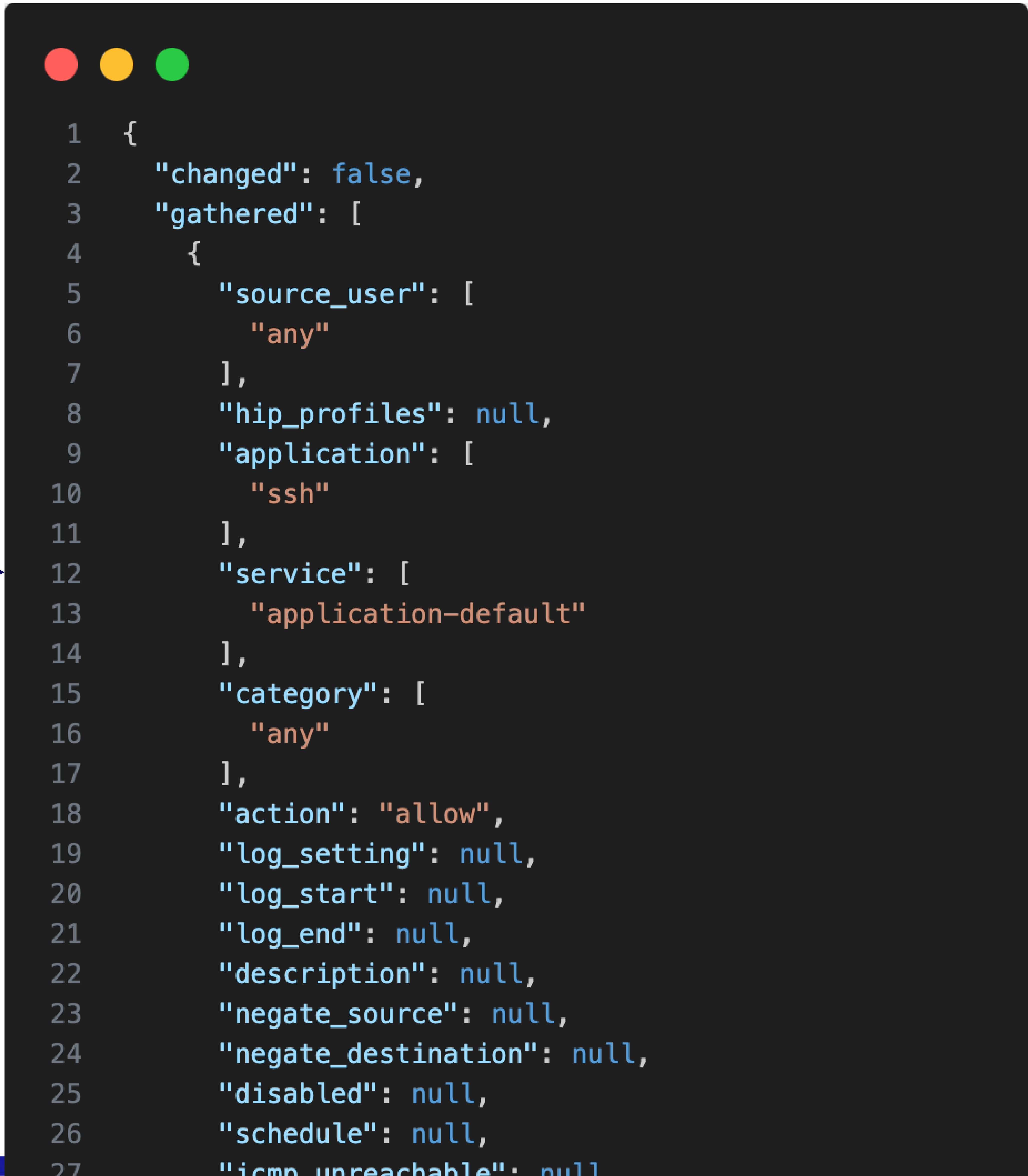


Firewall MCP Code Example

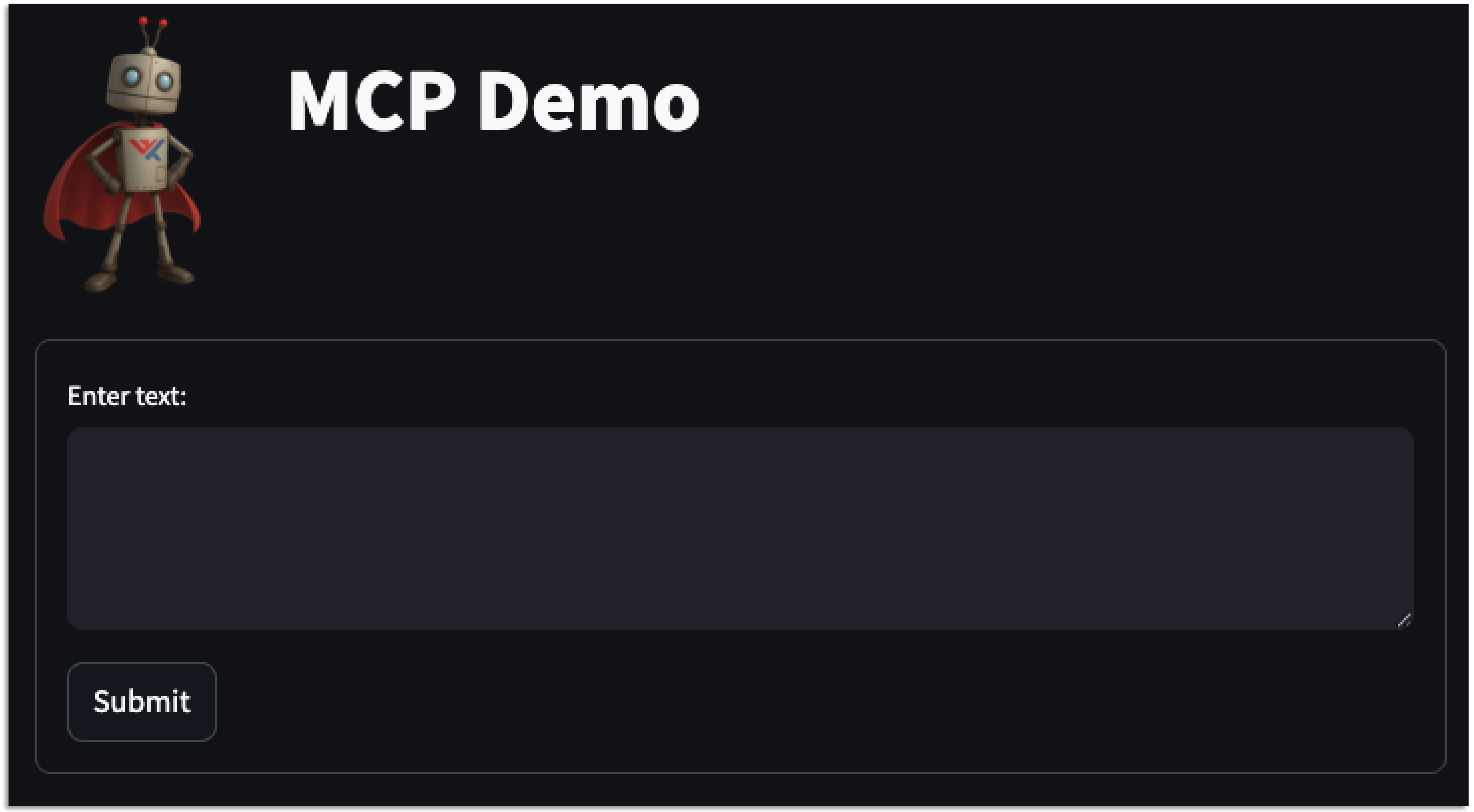


```python
def get_job_payload(job_id):
    status = api_url + f"/api/controller/v2/jobs/{job_id}/"
    get_status_info = req_helper("GET", status)
    while get_status_info.json().get("status") != "successful":
        print("Job is not completed yet. Waiting...")
        time.sleep(3)  # Wait for 3 seconds before checking again
        get_status_info = req_helper("GET", status)  # Refresh the job status
    output = api_url + f"/api/controller/v2/jobs/{job_id}/"
    get_output = req_helper("GET", output)
    print(get_output.json().get("artifacts").get("acl_data"))
    return get_output.json().get("artifacts").get("acl_data")

def req_helper(method, url, headers=headers):
    """
    Helper function to make HTTP requests.
    """
    try:
        response = requests.request(method, url, headers=headers, verify=False)
        response.raise_for_status()  # Raise an error for bad responses
        return response
    except requests.exceptions.RequestException as e:
        print(f"Request failed: {e}")
        return None

@mcp.tool()
def acl_audit():
    """Verify the functionality of an ACL
    returns the ACL object in json format"""
    job_id = launch_template()
    acl = get_job_payload(job_id)
    return acl

if __name__ == "__main__":
    mcp.run(transport="stdio")
```

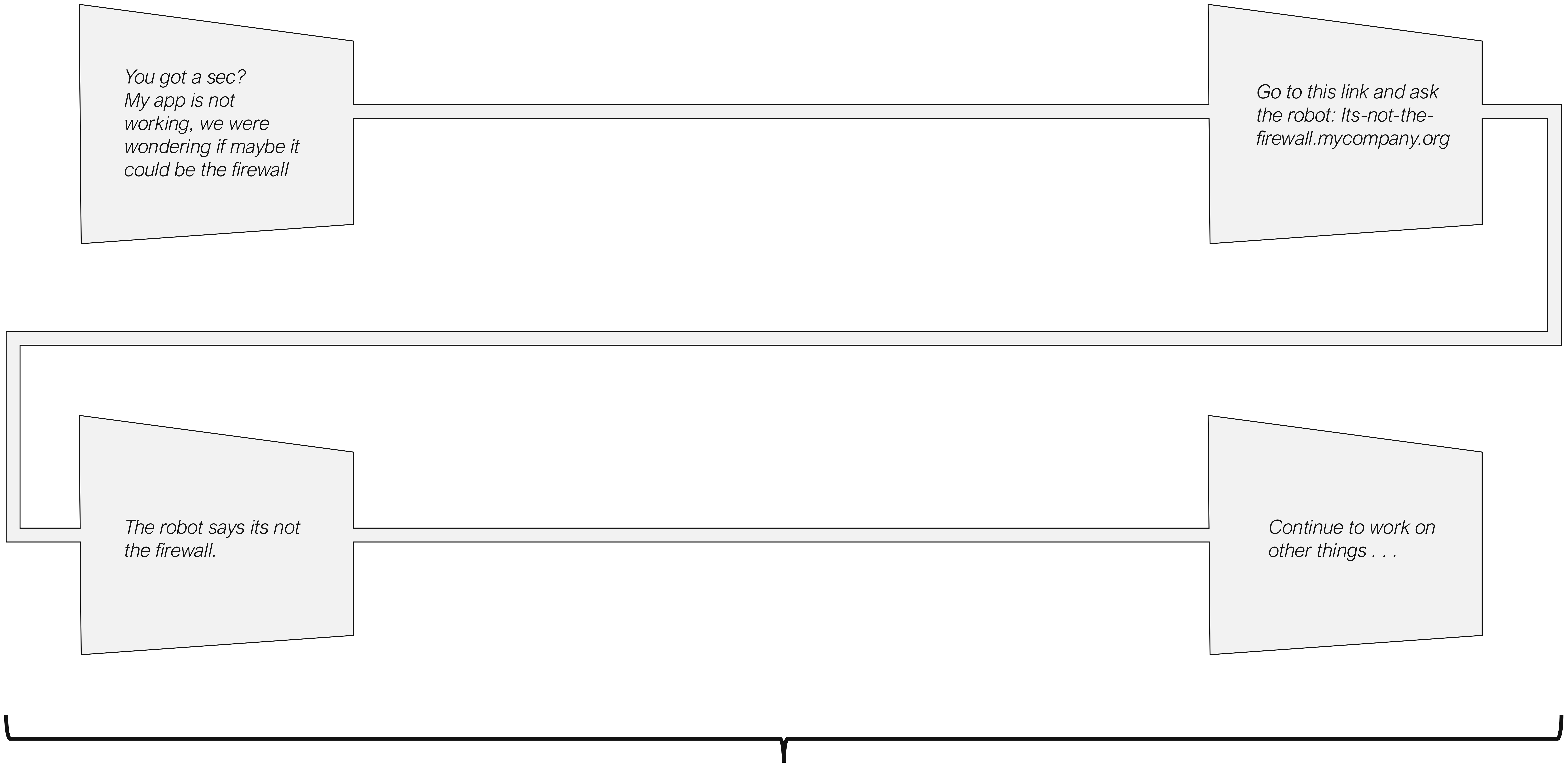
Launches AAP job template to retrieve Firewall Ruleset



```json
{
  "changed": false,
  "gathered": [
    {
      "source_user": [
        "any"
      ],
      "hip_profiles": null,
      "application": [
        "ssh"
      ],
      "service": [
        "application-default"
      ],
      "category": [
        "any"
      ],
      "action": "allow",
      "log_setting": null,
      "log_start": null,
      "log_end": null,
      "description": null,
      "negate_source": null,
      "negate_destination": null,
      "disabled": null,
      "schedule": null,
      "icmp_unreachable": null
```


MCP Demo
Enter text:
Submit

Answering Questions About Firewalls


You got a sec?
My app is not working, we were wondering if maybe it could be the firewall
Go to this link and ask the robot: Its-not-the-firewall.mycompany.org
The robot says its not the firewall.
Continue to work on other things . . .

2 Minutes

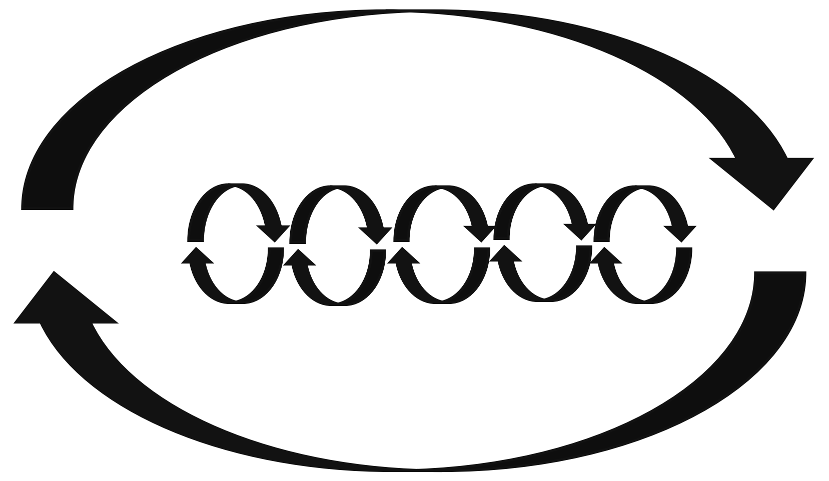
What I’ve Learned

Look for ways to improve your life with automation

What I've Learned

Look for ways to improve your life with automation

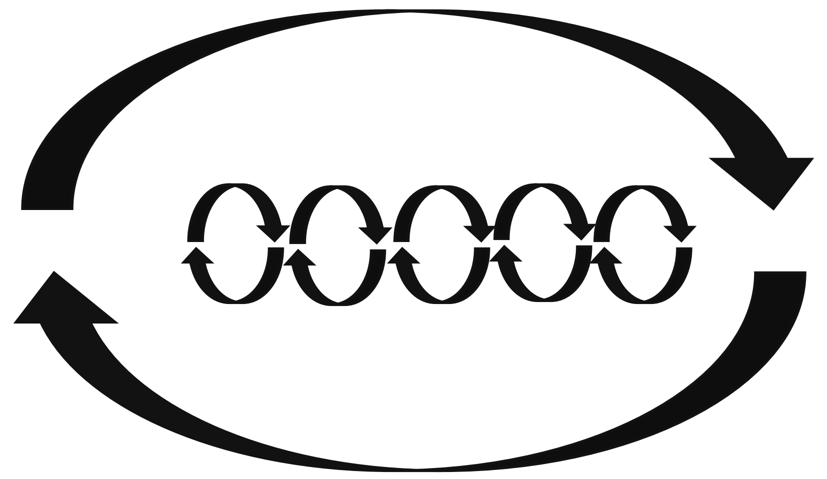
Constant Learning and Iteration



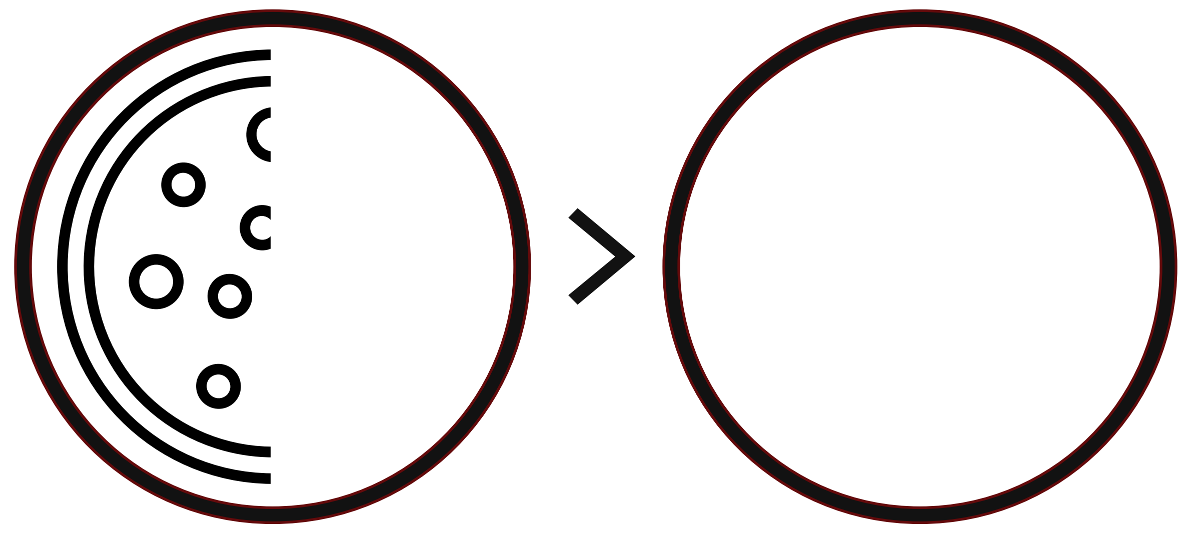
What I’ve Learned

Look for ways to improve your life with automation

Constant Learning and Iteration



50 % of something is better than 100 % of nothing



What I've Learned

Stay Curious!

World Wide Technology

Make a new world happen

boblongmore.com