



StreamlitはNWエンジニアにとって Web開発の銀の弾丸となり得るか？（仮）

CTC
渡部 友也

NW自動化

Webを作るFrontエンジニアとツールやインフラを作るBackendエンジニアは
スキルセットも文化も完全に別物である



Frontendエンジニア

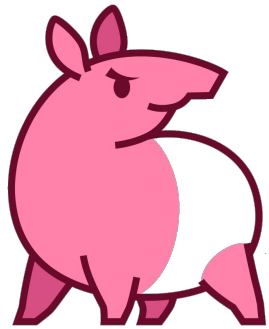


Backendエンジニア

画像はイメージです

NW自動化

NWエンジニアが扱う対象はBackendシステム(インフラ)なので
言語的にもPython/Go langのLibraryが身近にある
あとはCLIが結局好き



GoBGP



kubernetes



NWエンジニア
(almost Backendエンジニア)



python™

NETMKO



Requests

NW自動化

だけど、ちょっとWeb画面が欲しい時もあるの

UIがCLIだったり、入力用のyamlを作る様な形になってしまいがち・・・
自分で使う時はいいけど、引き渡す時はGUIをさくっとつけたい時がある

だけどWeb画面の開発は得意じゃない

```
import sys
from jira_manager import JiraManager
import yaml
import requests

def get_context(task):
    del task["cron"]
    return task.pop("type"), task

if __name__ == '__main__':
    with open("yaml/data.yaml", "r", encoding="utf-8_sig") as yml:
        ticket_data = yaml.safe_load(yml)
    if sys.argv[1] == "get_tasks":
        for task in ticket_data:
            print(task)
    else:
        uri, payload = get_context(ticket_data[sys.argv[1]])
        if "onboarding_tasks" == sys.argv[1]:
            onboardee = input("オンボードするメンバーの名前を入力してください。(本ツールにIDと名前が登録されていることも確認してください):")
            epic_id = input("EpicIDを入力してください:")
            payload["account_name"] = onboardee
            payload["epicId"] = epic_id
        elif "onboarding_support" == sys.argv[1]:
            supporter = input("サポーターの名前を入力してください:")
            epic_id = input("EpicIDを入力してください:")
            payload["account_name"] = supporter
            payload["epicId"] = epic_id

    response = requests.post(f"http://localhost:8000/ticket/{uri}/", json=payload)
```



```
DUT:
  host: 192.168.0.126
  port: 22
  alias: CA120-2
  <<: *credential_CA120
  device_type: apresia_amios
DUT_via_console:
  host: 192.168.0.144
  port: 8104
  alias: CA120-2_serial
  <<: *credential_CA120
  device_type: apresia_amios_telnet
CentOS9_0:
  host: 192.168.1.141
  port: 22
  alias: CentOS9_0
  username: admin
  password:
  device_type: linux
CentOS9_1:
  host: 192.168.1.142
  port: 22
  alias: CentOS9_1
  username: admin
  password:
  device_type: linux
```

NW自動化



Streamlit

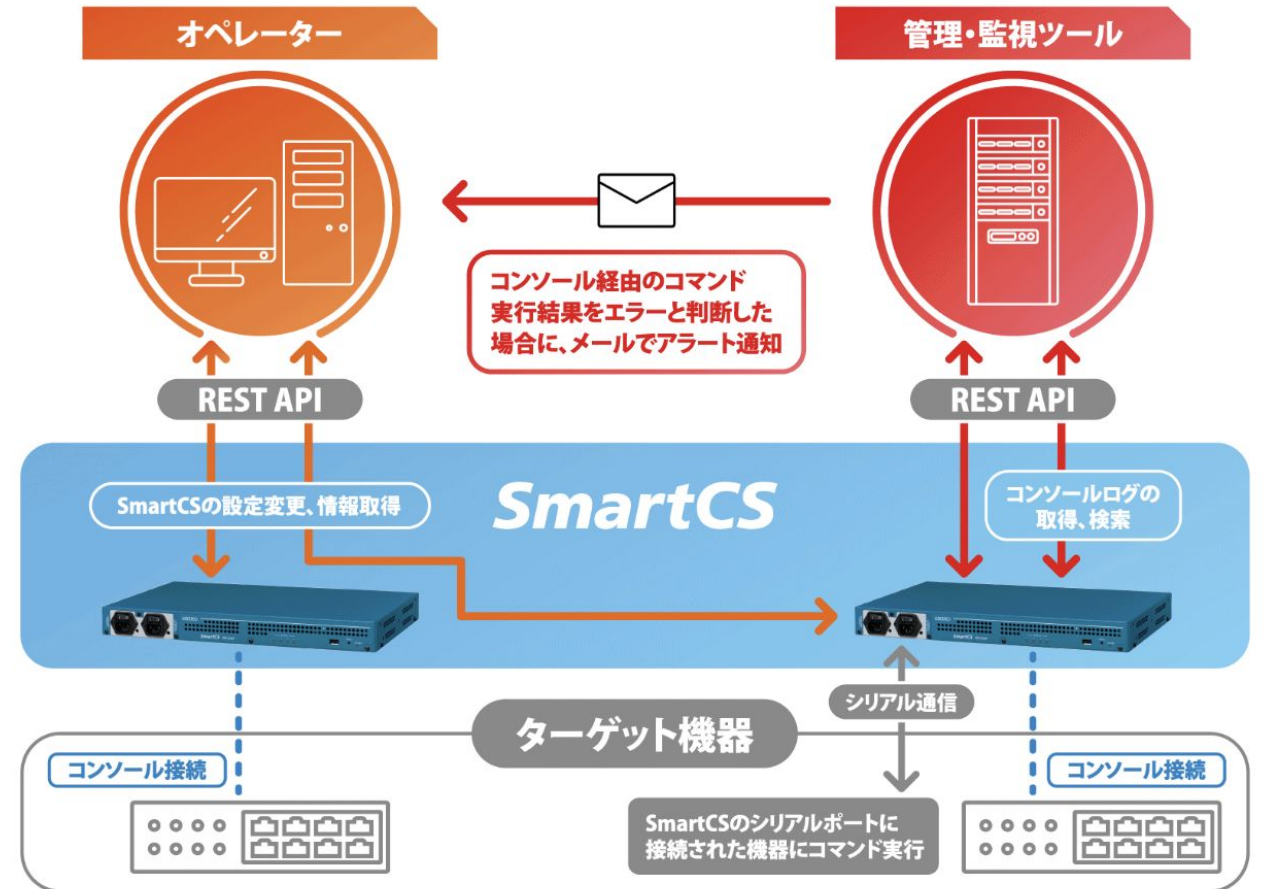
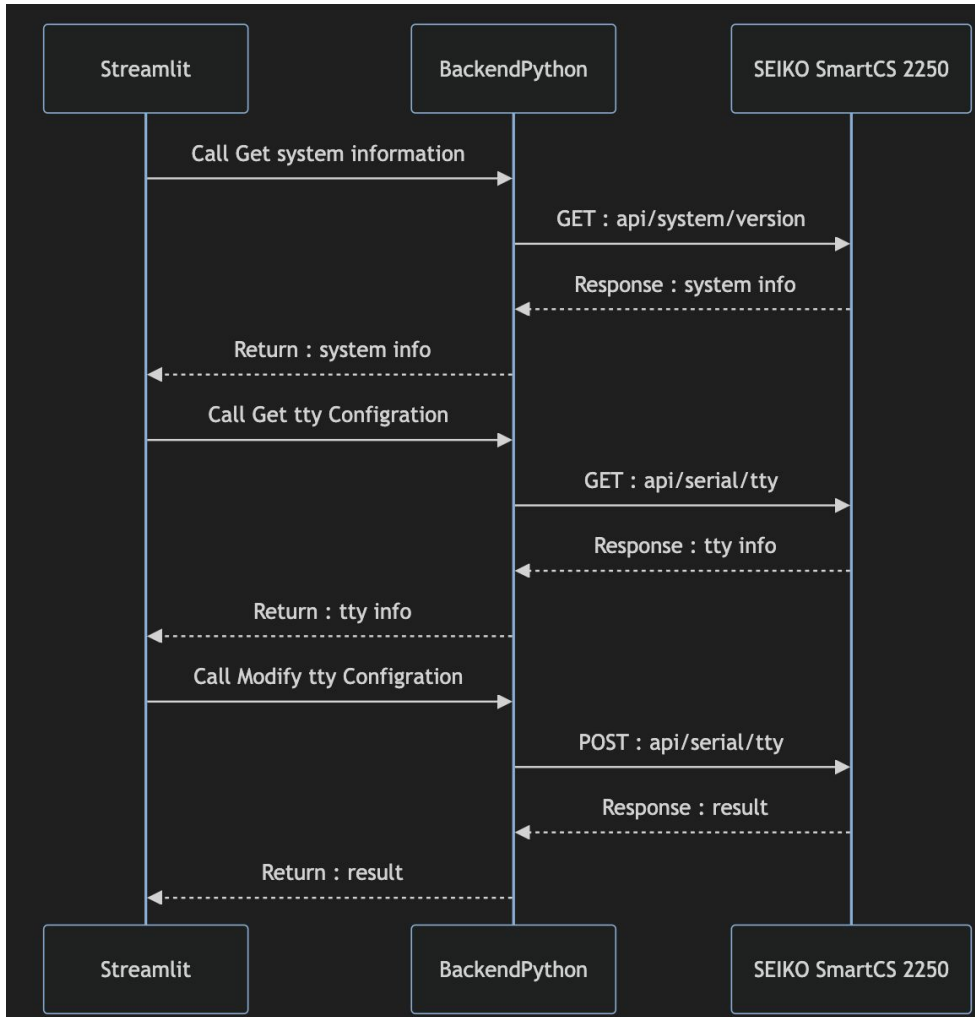
少量のPythonコードだけでInteractiveなWebを作れる

Pythonなので自作Python Libraryをimportするだけで使える

データサイエンティストに重宝されていて、表示のWidgetも豊富

<https://s3-us-west-2.amazonaws.com/assets.streamlit.io/videos/hero-video.mp4>

- 手元にあったSEIKO SmartCS 2250にGUIをつけてみました



NW自動化

GitHub Repository

<https://github.com/ctc-nt/streamlit-with-ne>

Current tty Configuration

	tty	config.baud	config.bitchar	config.parity	config.stop	config.flow	config.detect_dsr	config.
0	1	9,600	8	none	1	none	off	
1	2	115,200	8	none	1	none	off	
2	3	115,200	8	none	1	none	off	
3	4	115,200	8	none	1	none	off	
4	5	115,200	8	none	1	none	off	
5	6	115,200	8	none	1	none	off	
6	7	115,200	8	none	1	none	off	
7	8	115,200	8	none	1	none	off	
8	9	115,200	8	none	1	none	off	
9	10	115,200	8	none	1	none	off	

Modify tty

tty
1

baud
9600

bitchar
8

parity
none

stop
1

flow
none

detect_dsr
on

CTC

▼ *Challenging Tomorrow's Changes*

NW 自動化