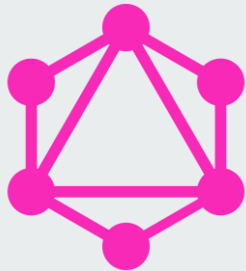# GraphQL
For Network and Security Engineers

Damien Garros
Infrastructure Automation Architect

FRNOG 36, September 2022

Agenda

What is GraphQL

GraphQL vs REST

Use Cases
for Network and Security Engineers

- **Securing GraphQL applications**

- **Infrastructure Automation**

Best Practices

Integration with Ansible (demo)

Damien Garros

Freelance Automation Architect

Infrastructure as Code, Automation & Observability for 10+ years,

Previously leading Technical Architecture at >>> network.toCode()

Spent 9 years in the US, recently moved back to France

damiengarros   @dgarros   @damgarros

GraphQL

- Data **query and manipulation language** for APIs

- Designed to make APIs fast, flexible, and developer-friendly.

- Move some complexity out of the client, on the server side.

- Complementary / Alternative to REST API
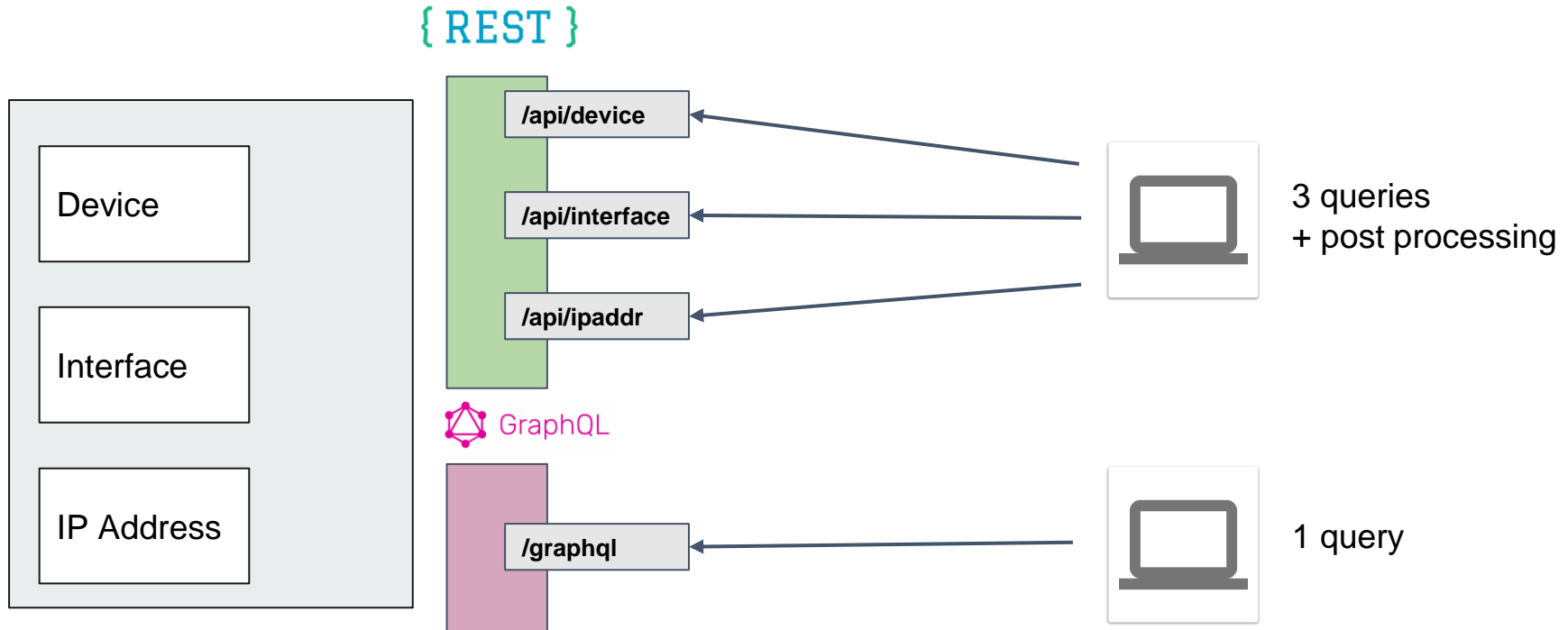
- Developed at Facebook, released in 2015

GraphQL

GraphQL

GraphQL vs REST

GraphQL Components

Query	Mutation	Subscription
Query data	Manipulate Data Create/Update/Delete Action	Real Time Update (Over WebSocket)

GraphQL Schema

Resolvers

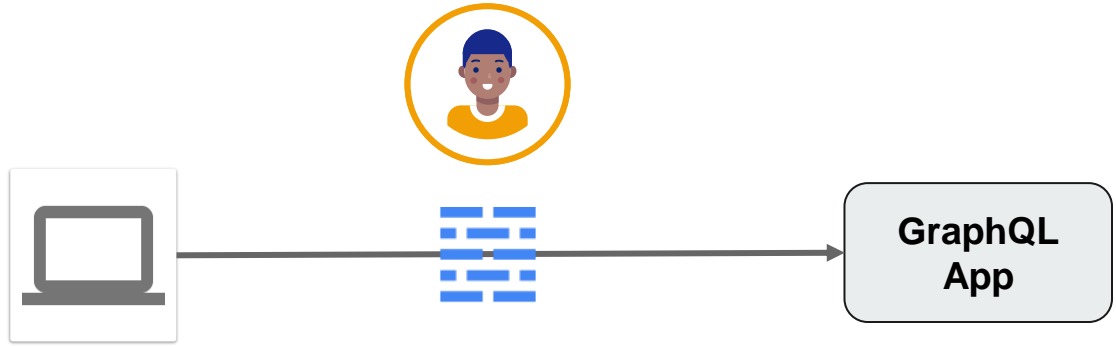| SQL Database	Graph Database	No-SQL Database
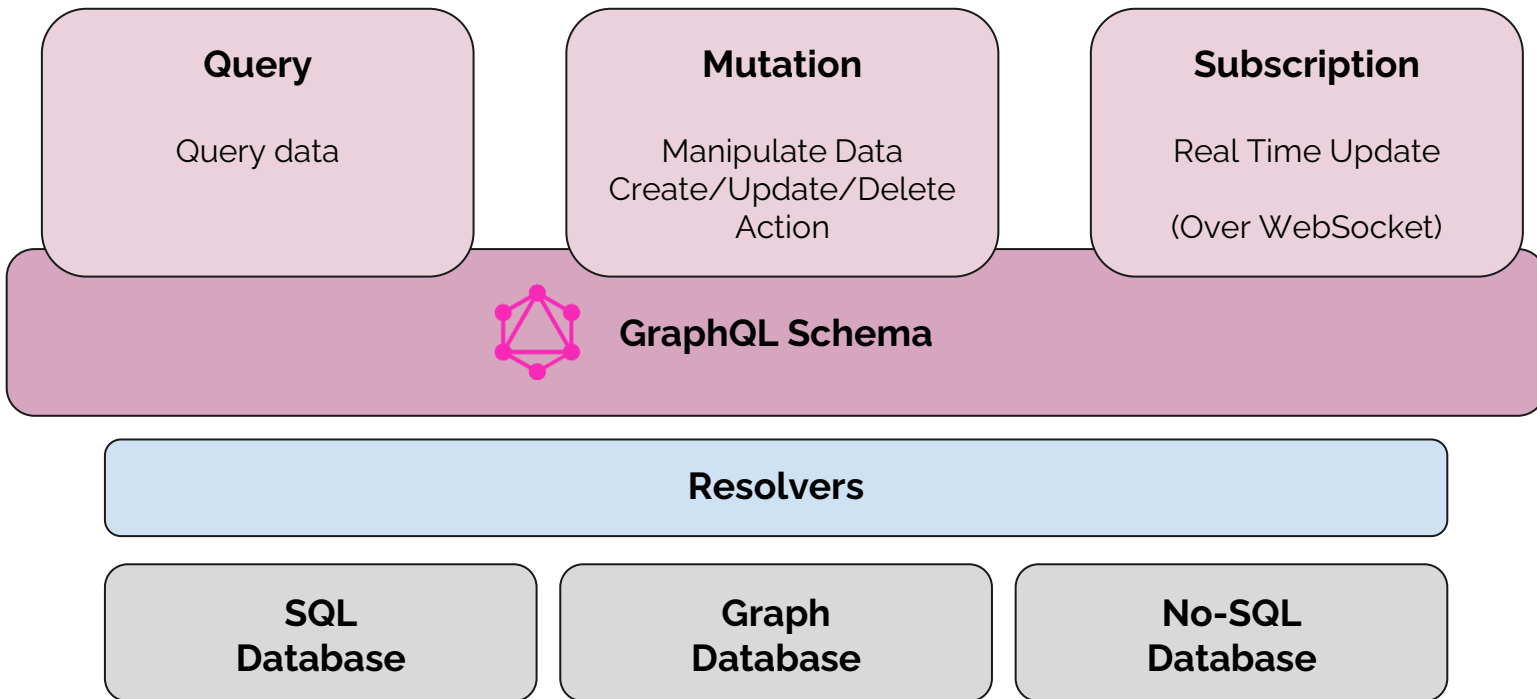
Query Cost
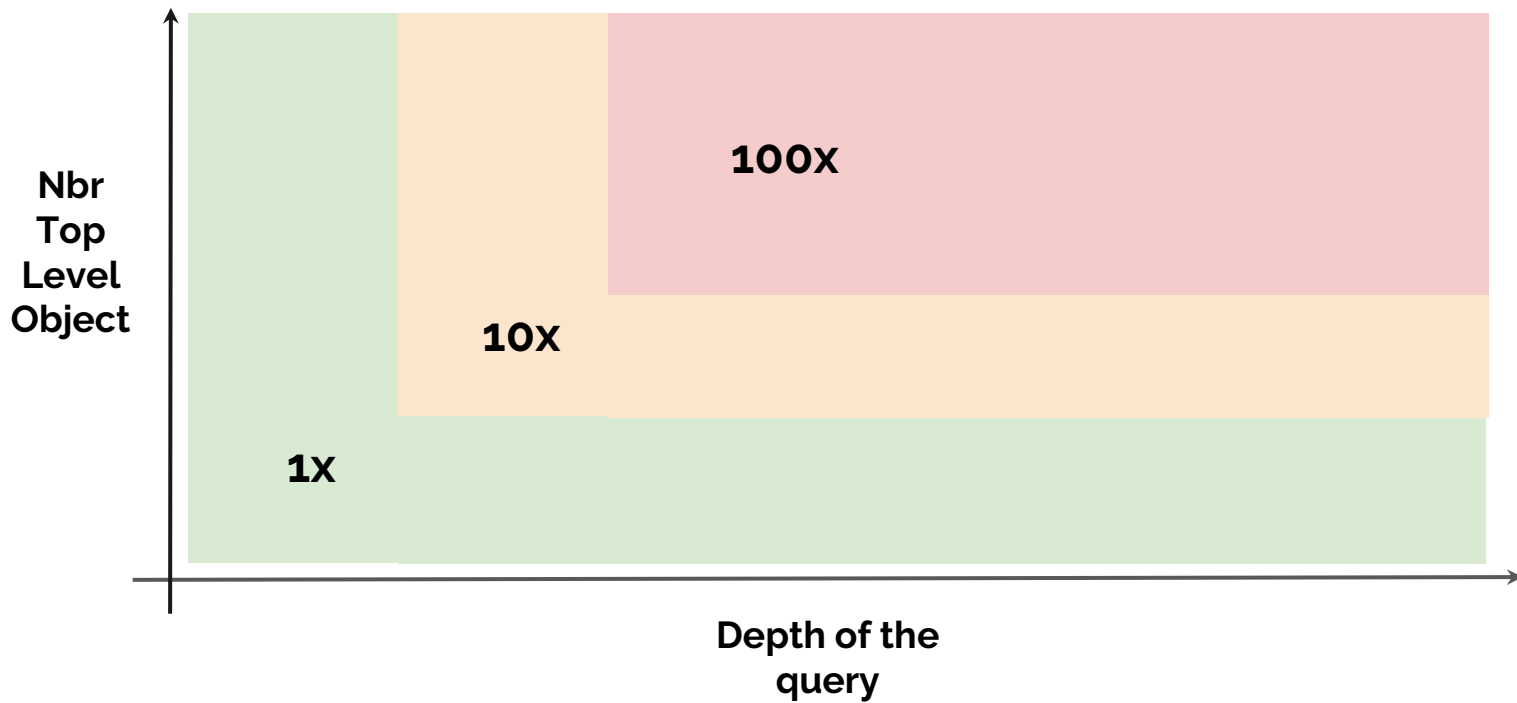
Other Features

- Variables

- Fragments

- Multiple queries

- Introspection Query

GraphQL

Use Cases for Network and Security Engineer

Use Cases for Network and Security Engineer

Manage / Secure
a GraphQL Application

GraphQL App

Network Automation

python™

ANSIBLE
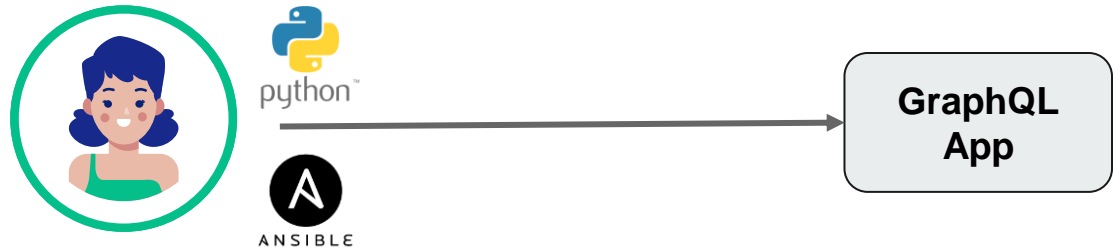
GraphQL App

Applications supporting GraphQL today

List non exhaustive

Applications that COULD support GraphQL tomorrow ??

FORTINET®

paloalto® NETWORKS

PeeringDB

Anyone
exposing relational data
via a REST API

zayo®

IX API

EQUINIX

PACKETFABRIC

PEERING MANAGER

Some might already support it but I couldn't find it in my research

Manage / Secure GraphQL Applications

- All requests goes to the same endpoint, HTTP status code are not relevant (always returns 200)

- Potential vector of attack

  - DOS / Query are very powerful (too powerful ?)

  - Batch of queries

  - Full schema available,

Network Automation with GraphQL

- Easier to query and consume data.

- Less effort required on the client or in Jinja

- Very easy to pair with Jinja2 to generate files

Configuration File
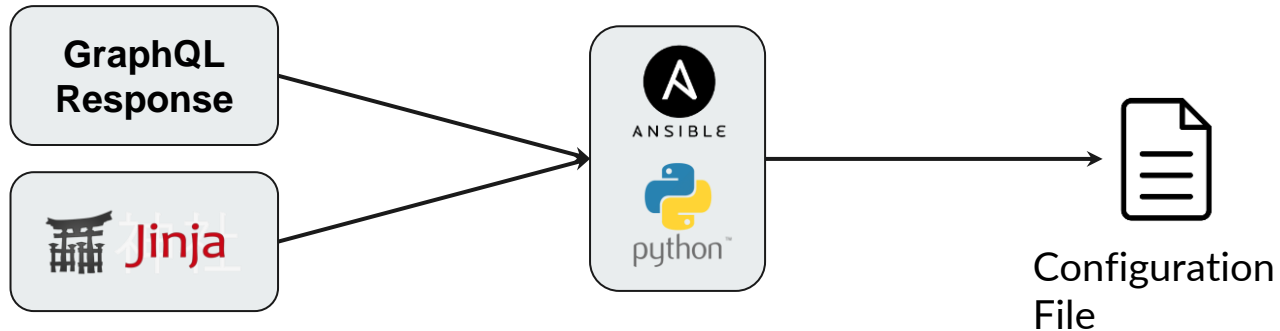
Best Practices

- Be mindful of the "**Cost**" of your queries

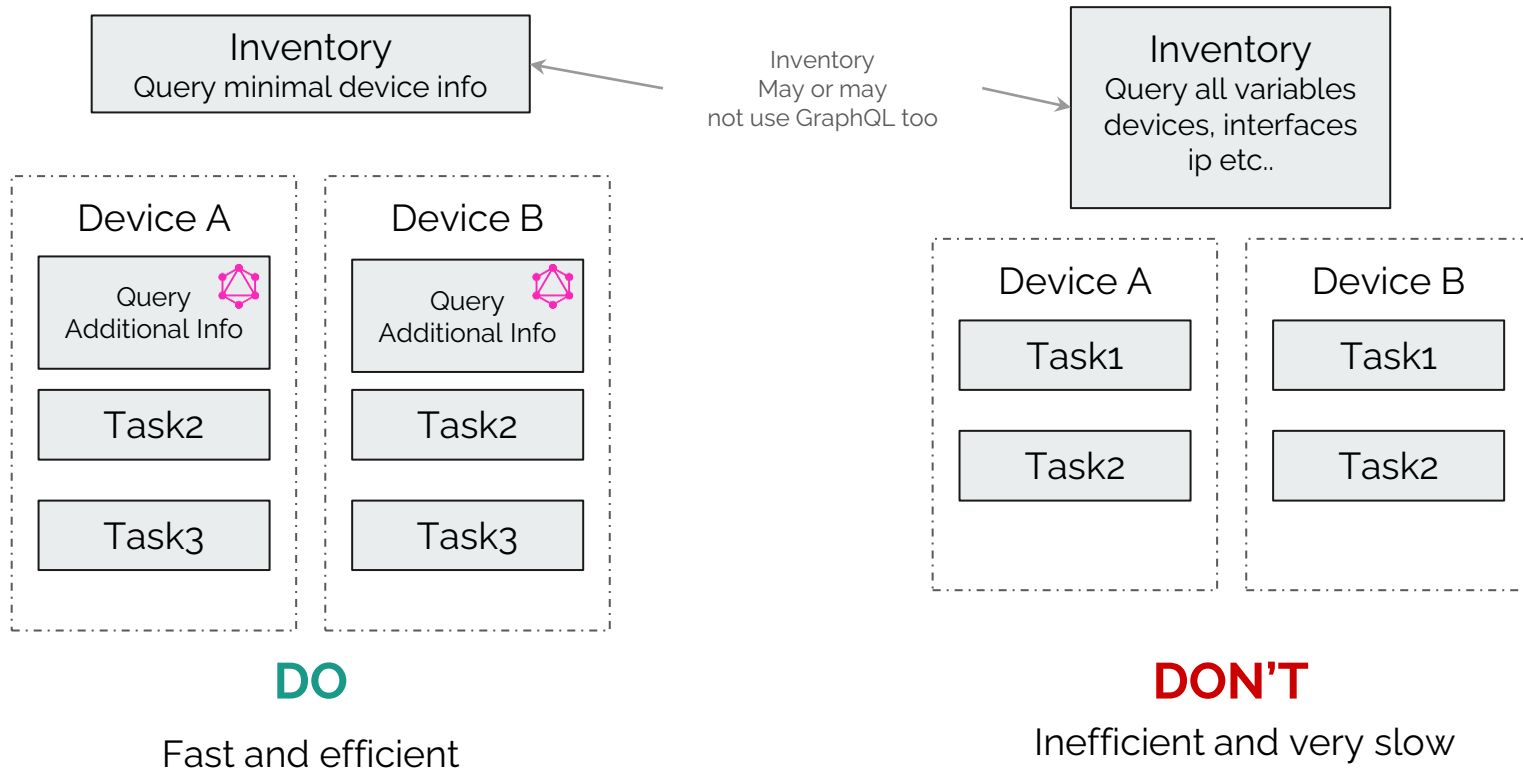- Use filter to narrow down what you want to retrieve

- In Ansible / Nornir, don't query everything in the inventory, query additional information at the beginning of the playbook instead

Integration with Ansible

Leverage GraphQL with Ansible

Inventory
Query minimal device info

Inventory
May or may
not use GraphQL too

Inventory
Query all variables
devices, interfaces
ip etc..

Device A

Query
Additional Info

Task2

Task3

Device B

Query
Additional Info

Task2

Task3

Device A

Task1

Task2

Device B

Task1

Task2

DO

Fast and efficient

DON'T

Inefficient and very slow

Inventory

```yaml
---
plugin: networktocode.nautobot.gql_inventory
api_endpoint: https://demo.nautobot.com/
token: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
query:
 devices:
   device_role: name
   status: slug
   site:
    name:
   filters:
     site: cdg01
group_by:
 - device_role.name
 - status.slug
```

Playbook 1/2

```yaml
---
- name: "Generate Device Configuration with Data Gathered with GraphQL"
  hosts: all
  gather_facts: false
  connection: local
  vars:
    nautobot_token: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
    query_string: |
      query ($device_name: [String!]) {
        devices (name: $device_name) {
          id
          name
          interfaces {
            name
            ip_addresses {
              address
            }
          }
        }
      }
```

Playbook 2/2

```yaml
tasks:
 - name: "Gather Interfaces and IP addresses information per device"
   networktocode.nautobot.query_graphql:
     url: "https://demo.nautobot.com/"
     token: "{{ nautobot_token }}"
     query: "{{ query_string }}"
     graph_variables: "{{ gql_variables }}"
     update_hostvars: yes
   vars:
     gql_variables:
       device_name: "{{ inventory_hostname }}"

 - name: "Generate Configuration"
   ansible.builtin.template:
     src: "config_template.j2"
     dest: "generated_configs/{{ inventory_hostname }}.cfg"
```

Thank you.

GraphQL vs REST

	GraphQL	REST
Schema	Yes	Recommended
Endpoint(s)	Only One	1 per Object Type
Payload	On Demand	Fixed
Read	Query	GET
Create	Mutation	POST
Update	Mutation	PUT
Delete	Mutation	DELETE
Notification	Subscription	With HTTP/2