

# A Netflix case study: Edge Service Architecture

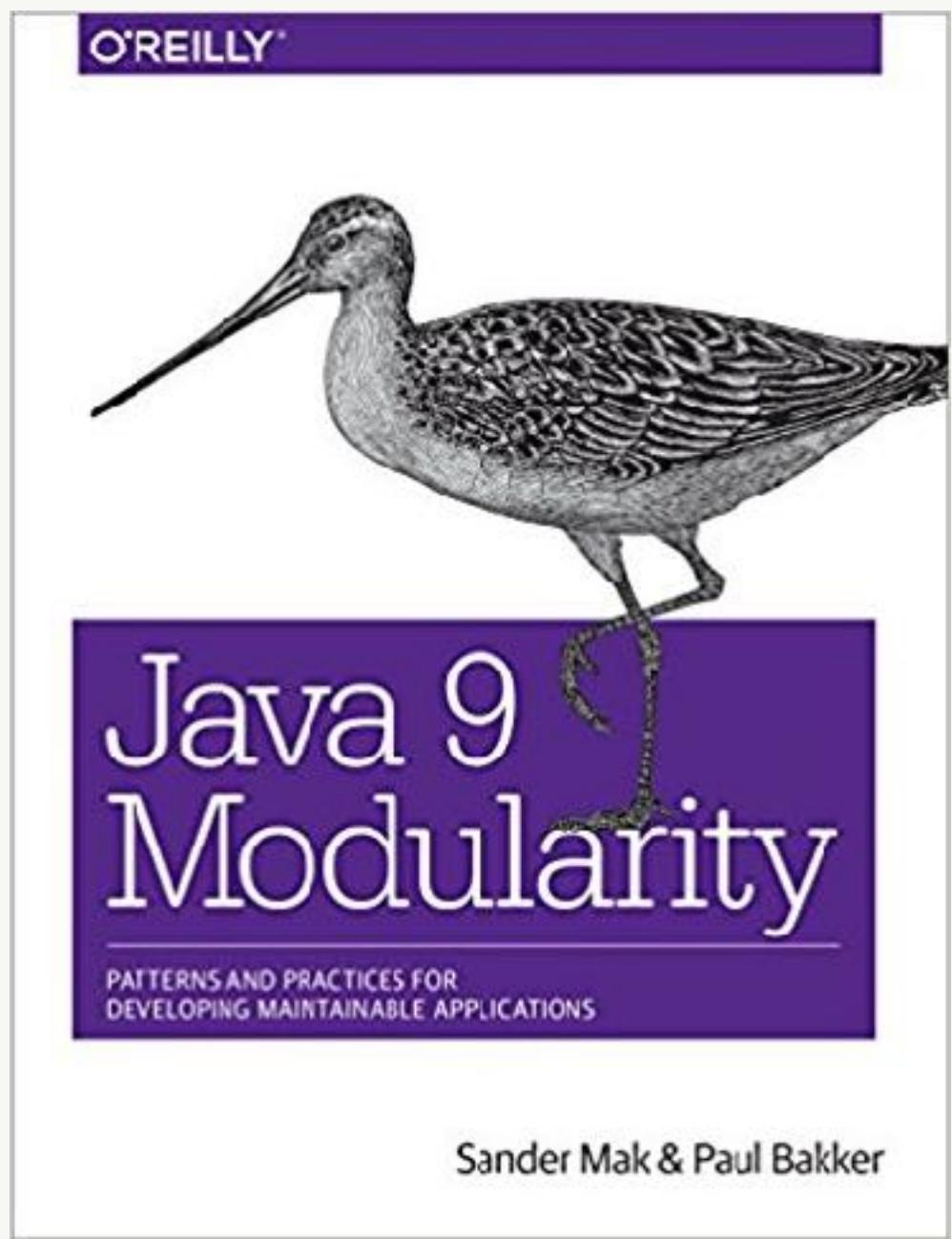
Paul Bakker

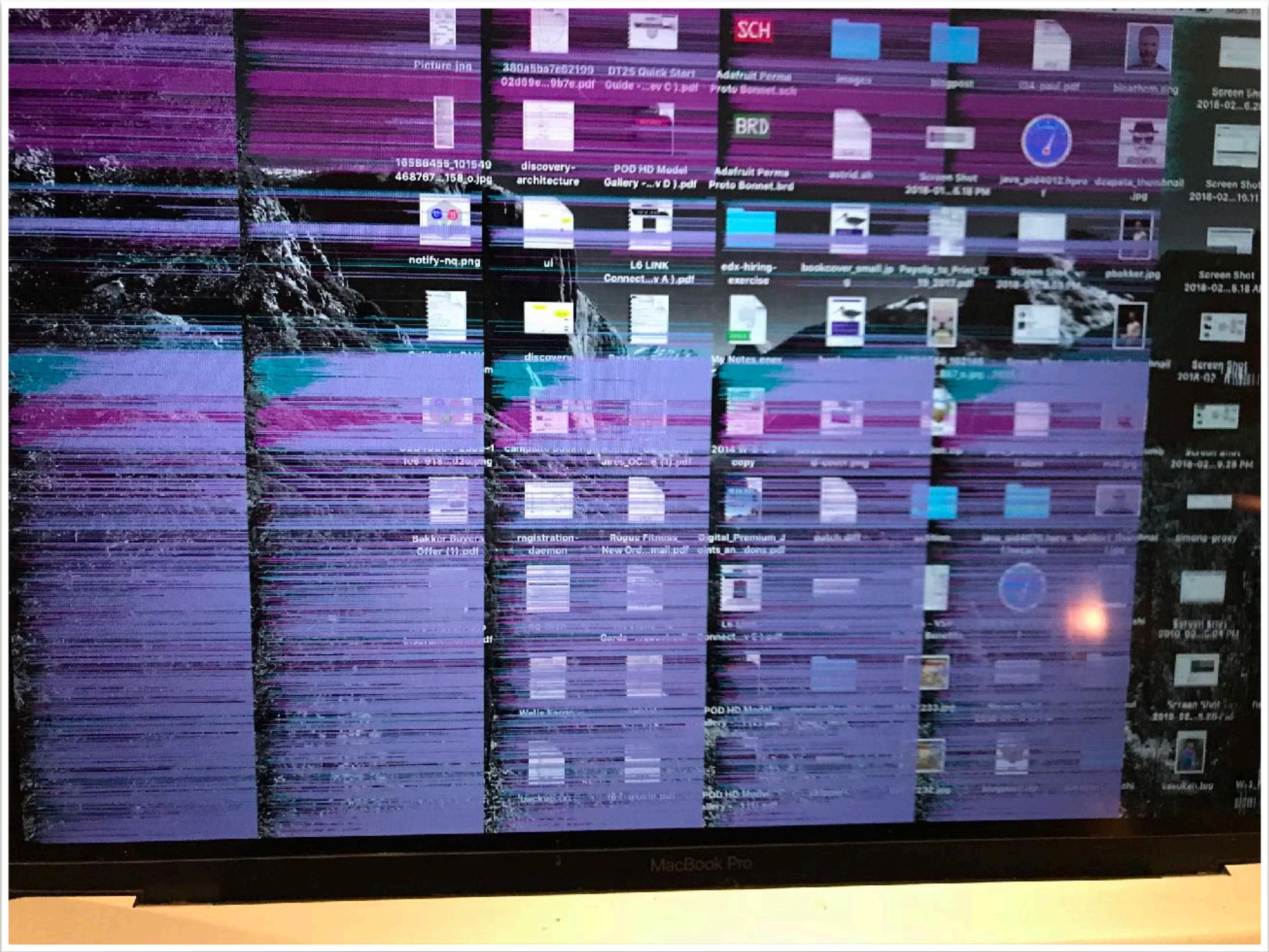


# Paul Bakker



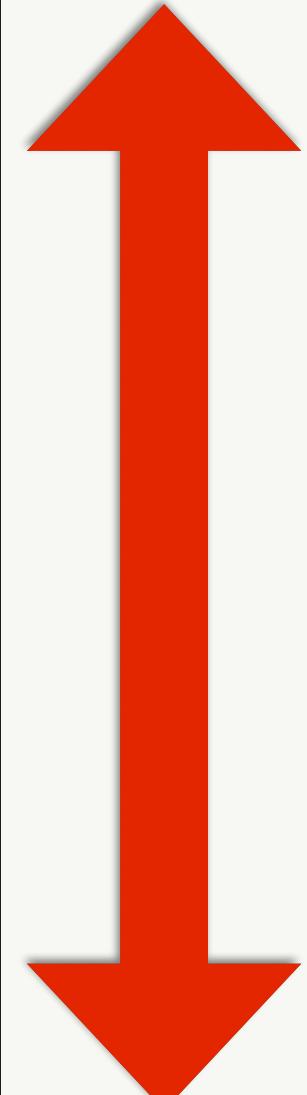
Edge Developer Productivity @ Netflix





# Contents.

## Architecture patterns

- 
- What and why Edge?
  - Edge architectures and their tradeoffs
  - API data access patterns
  - Developer experience

**How we do things**



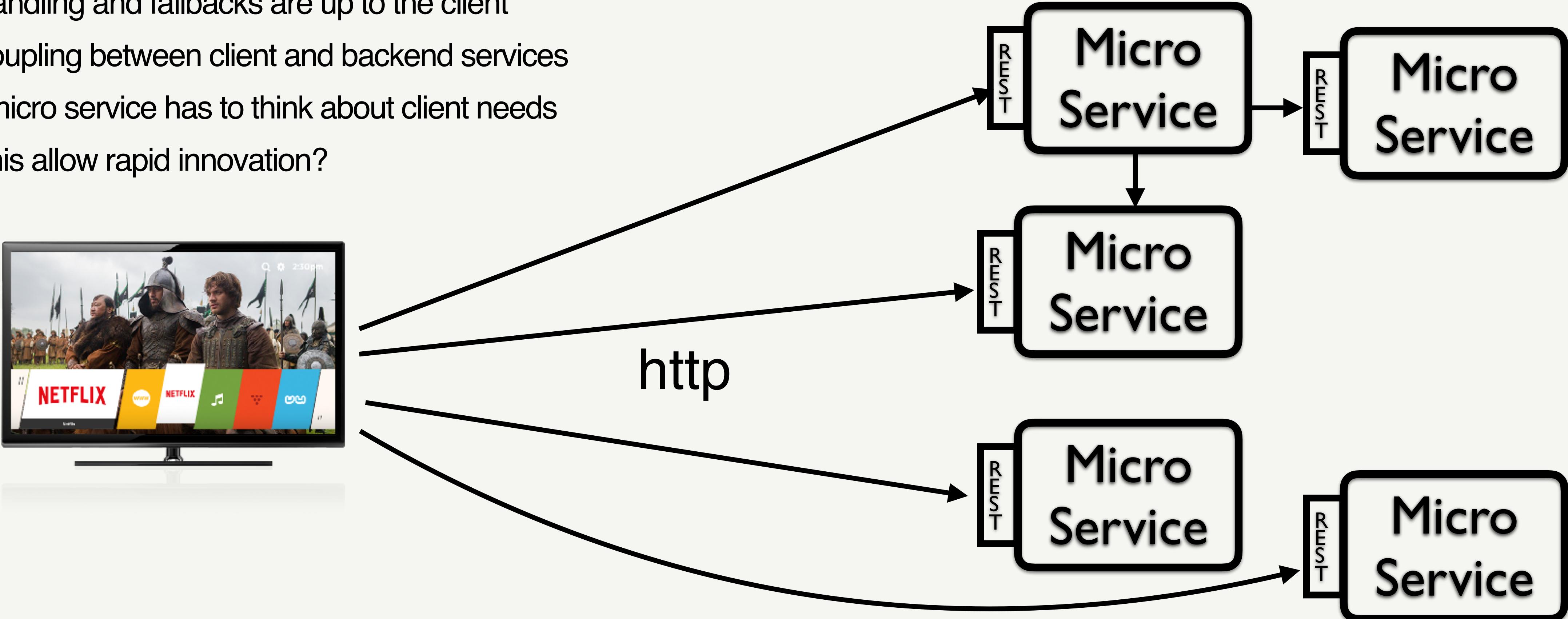
This talk is **not** a template telling  
you how to do micro services.

We want an architecture that  
allows us to **innovate**.

**What is *Edge Service  
Architecture*?**

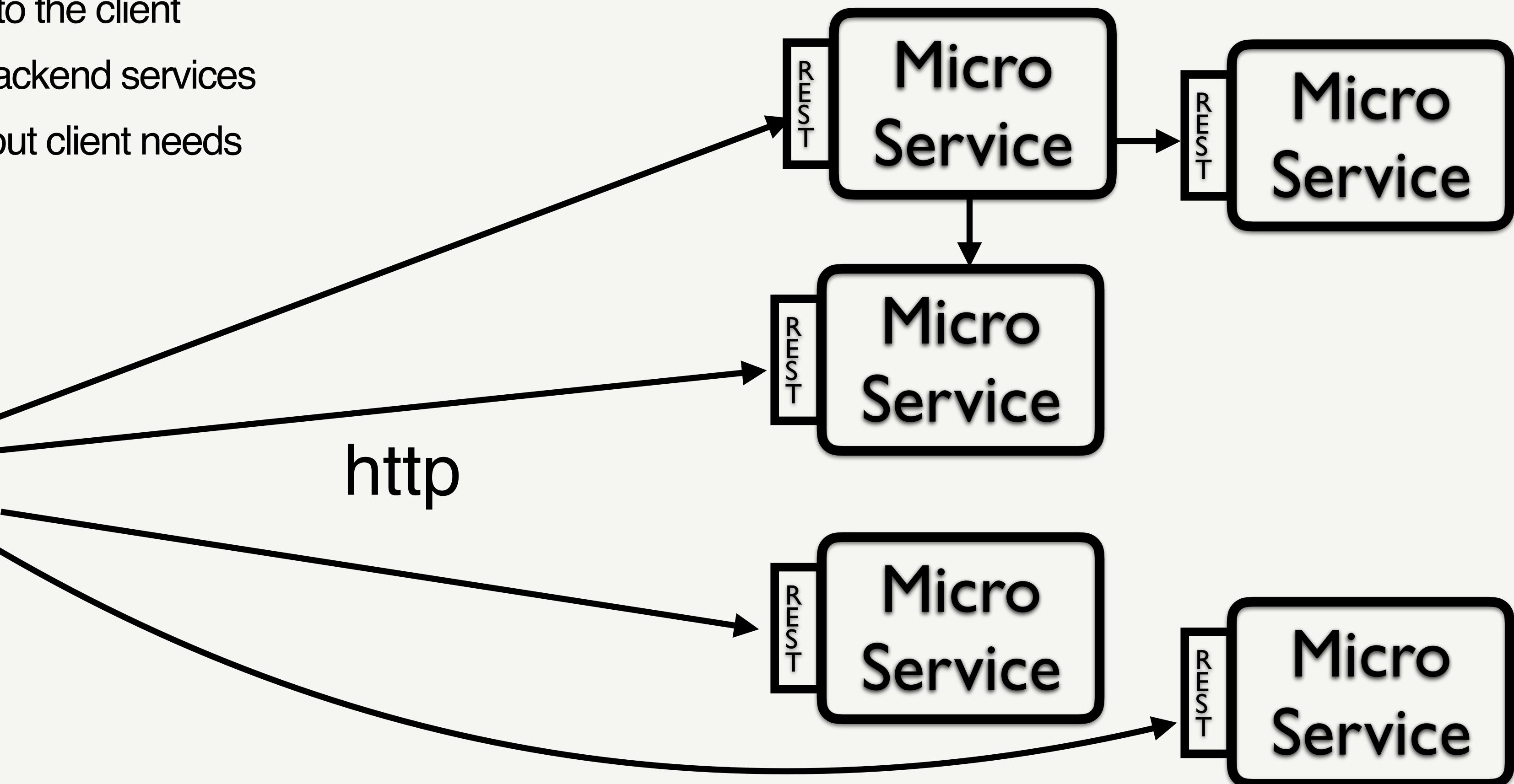
Lets start with *How do we  
expose Micro Services to  
devices?*

- Multiple calls from the client
- Possibly fetching duplicate data
- Error handling and fallbacks are up to the client
- Tight coupling between client and backend services
- Every micro service has to think about client needs
- Does this allow rapid innovation?



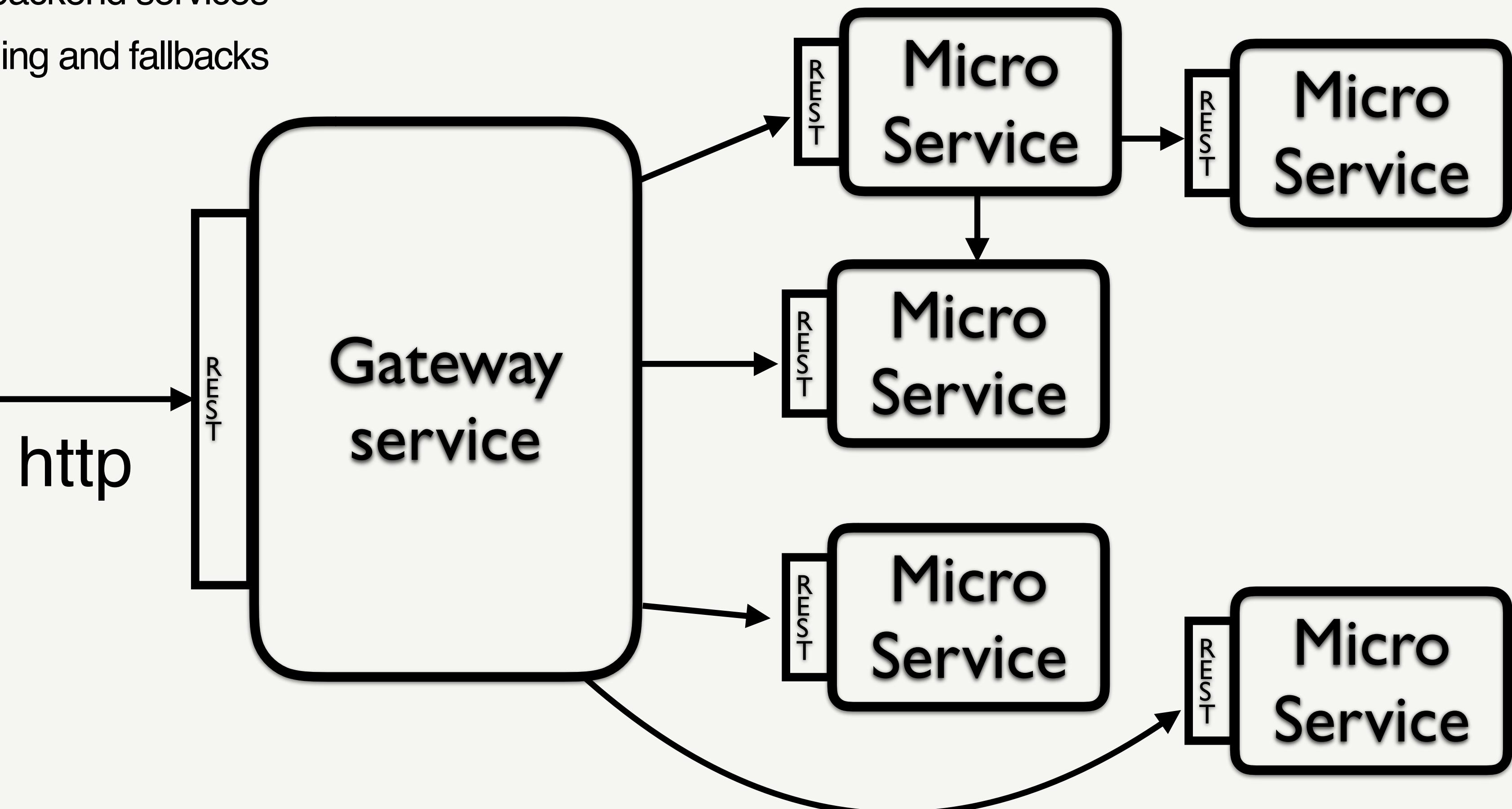
This is bad, but not horrible

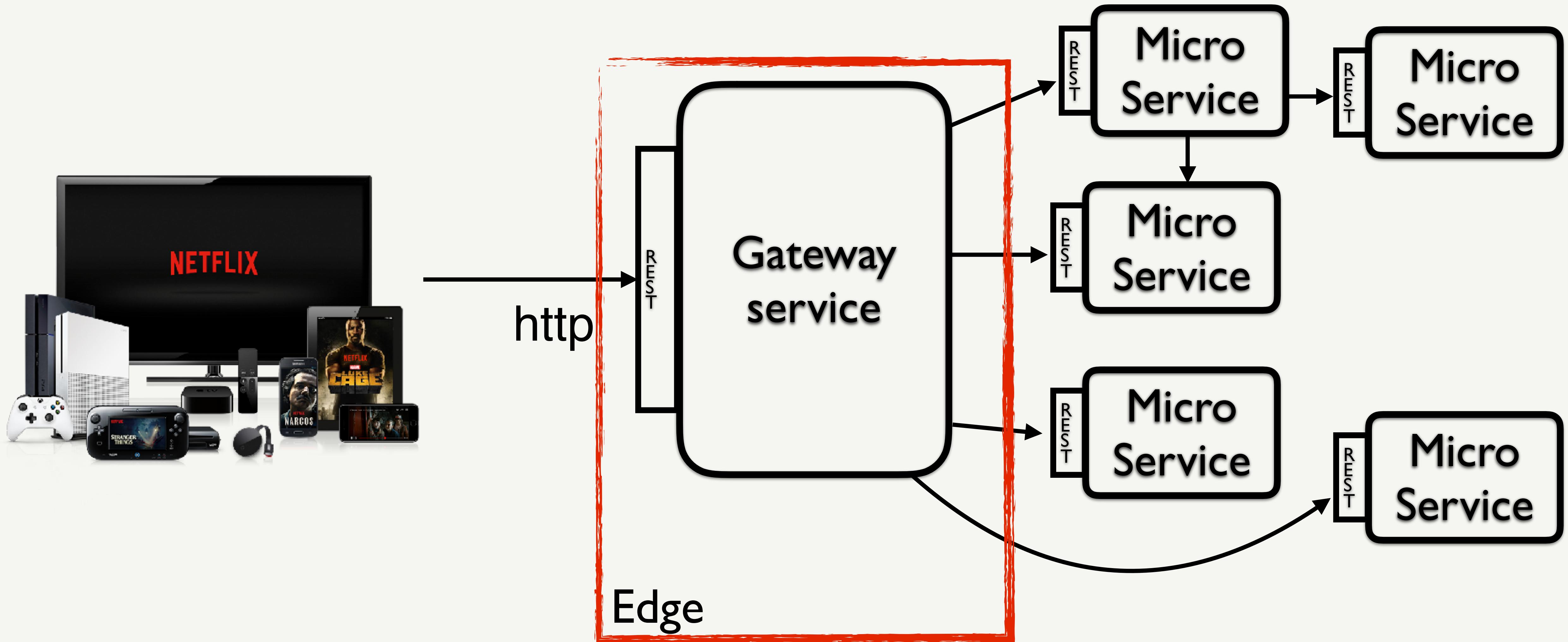
- Multiple calls from the client
- Possibly fetching duplicate data
- Error handling and fallbacks are up to the client
- Tight coupling between client and backend services
- Every micro service has to think about client needs
- Does this allow rapid innovation?

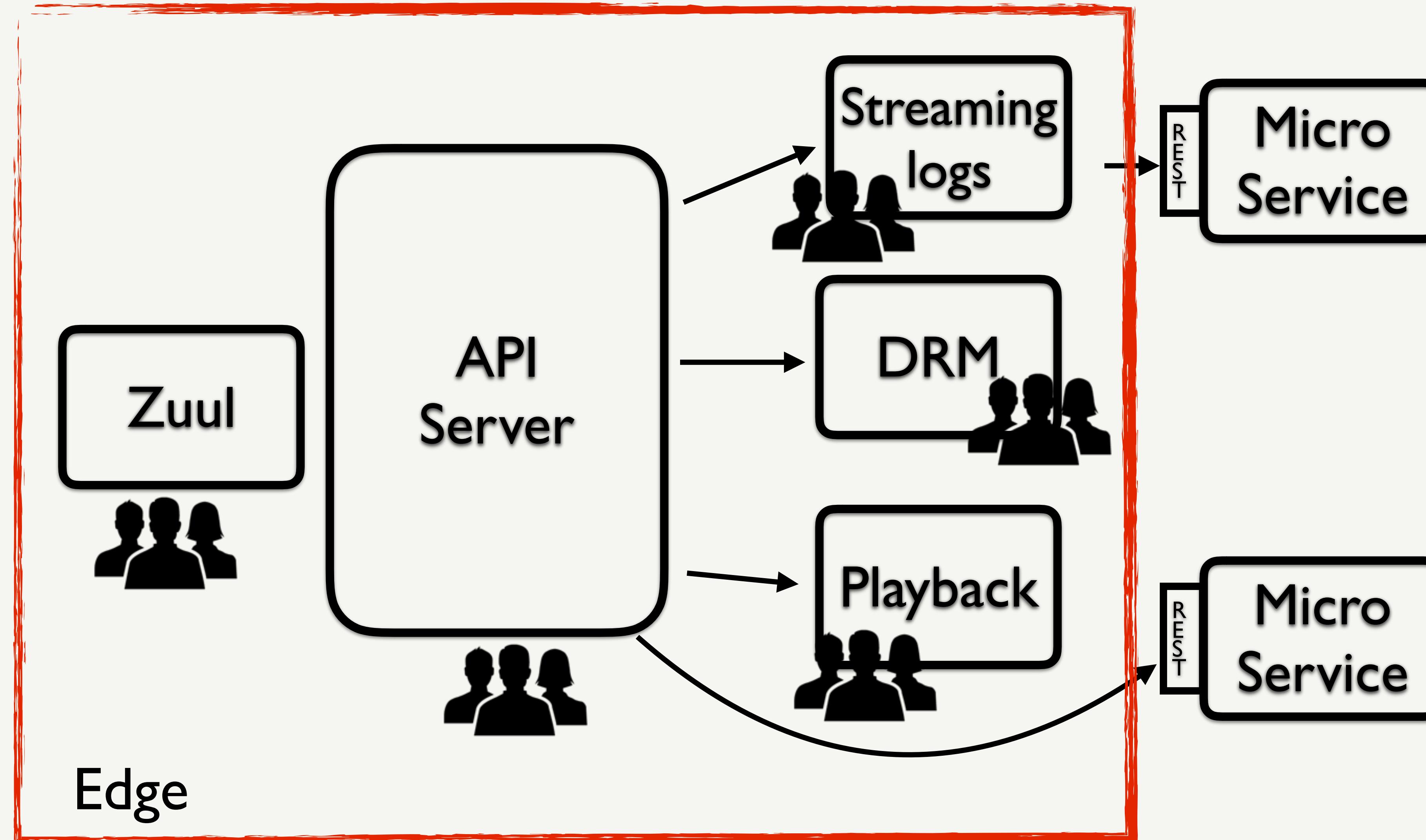


Now it's horrible

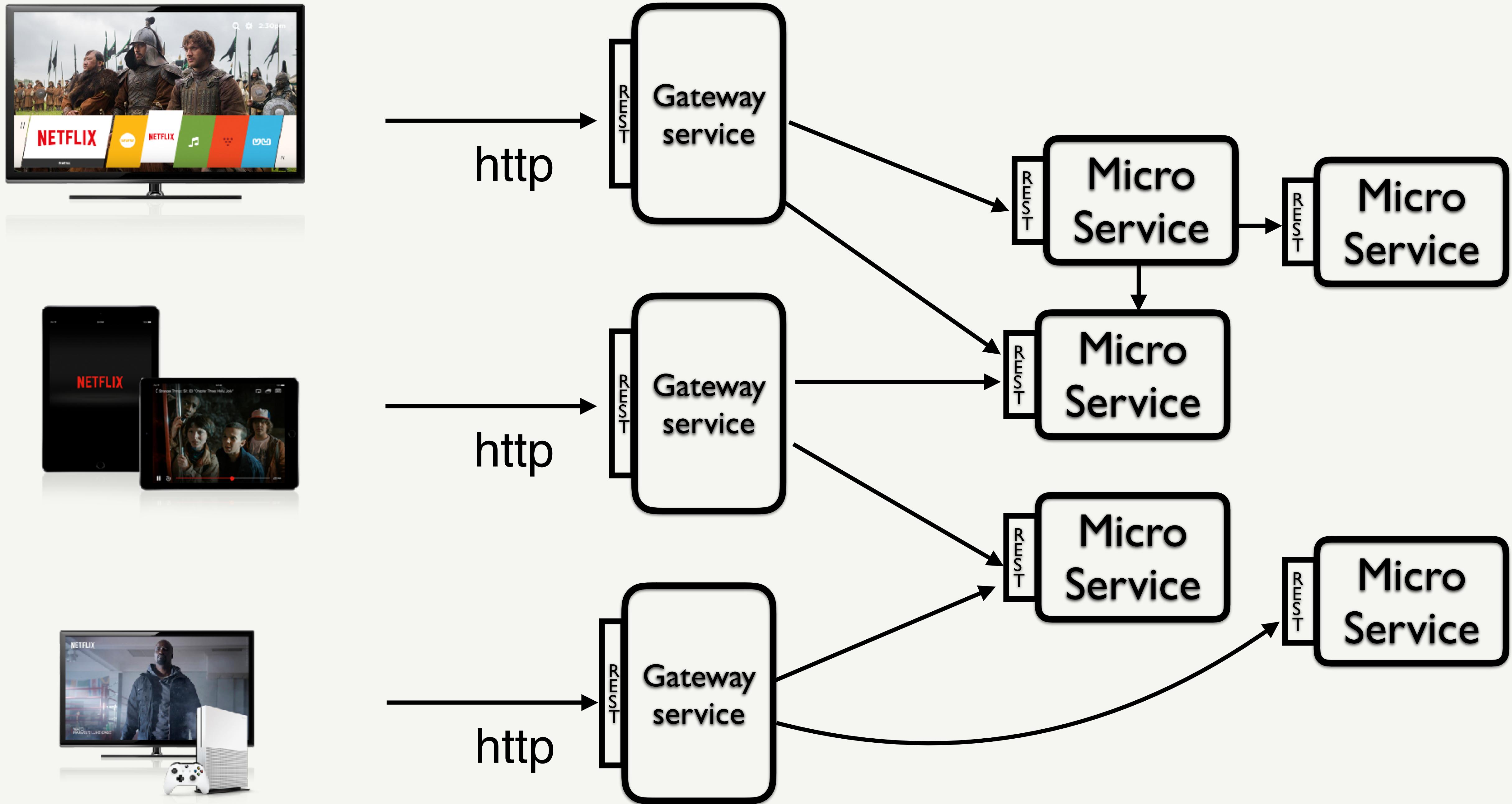
- Clients only need a single call
- No coupling between clients and backend services
- Gateway takes care of error handling and fallbacks
- Still a one size fits all API
- Does this allow rapid innovation?





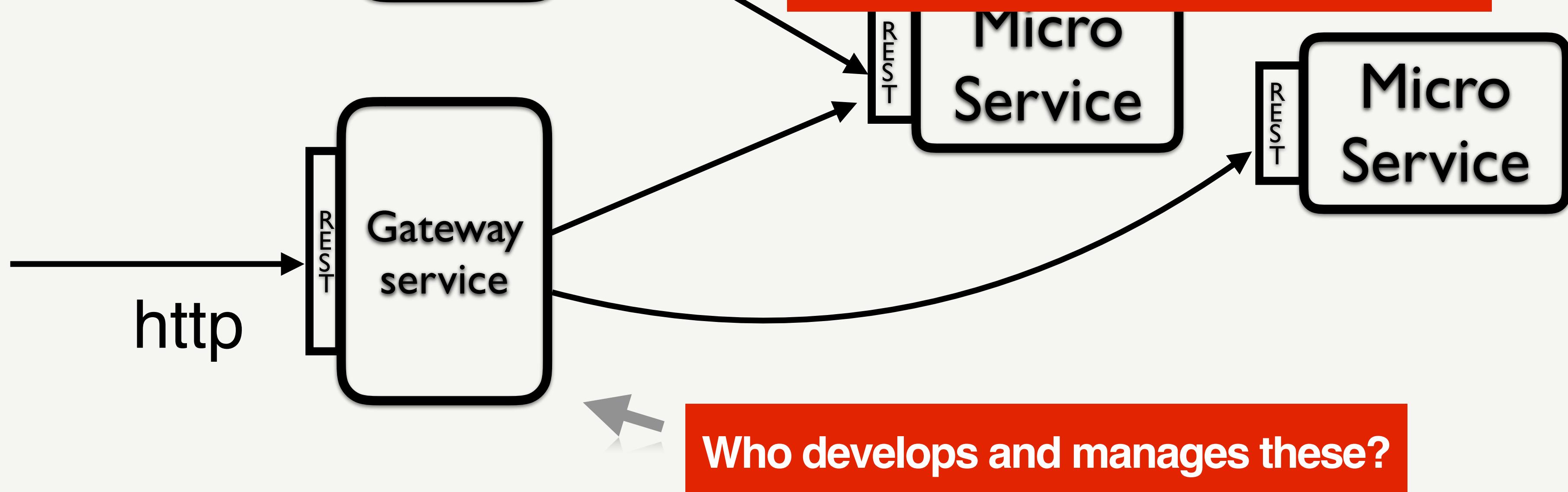
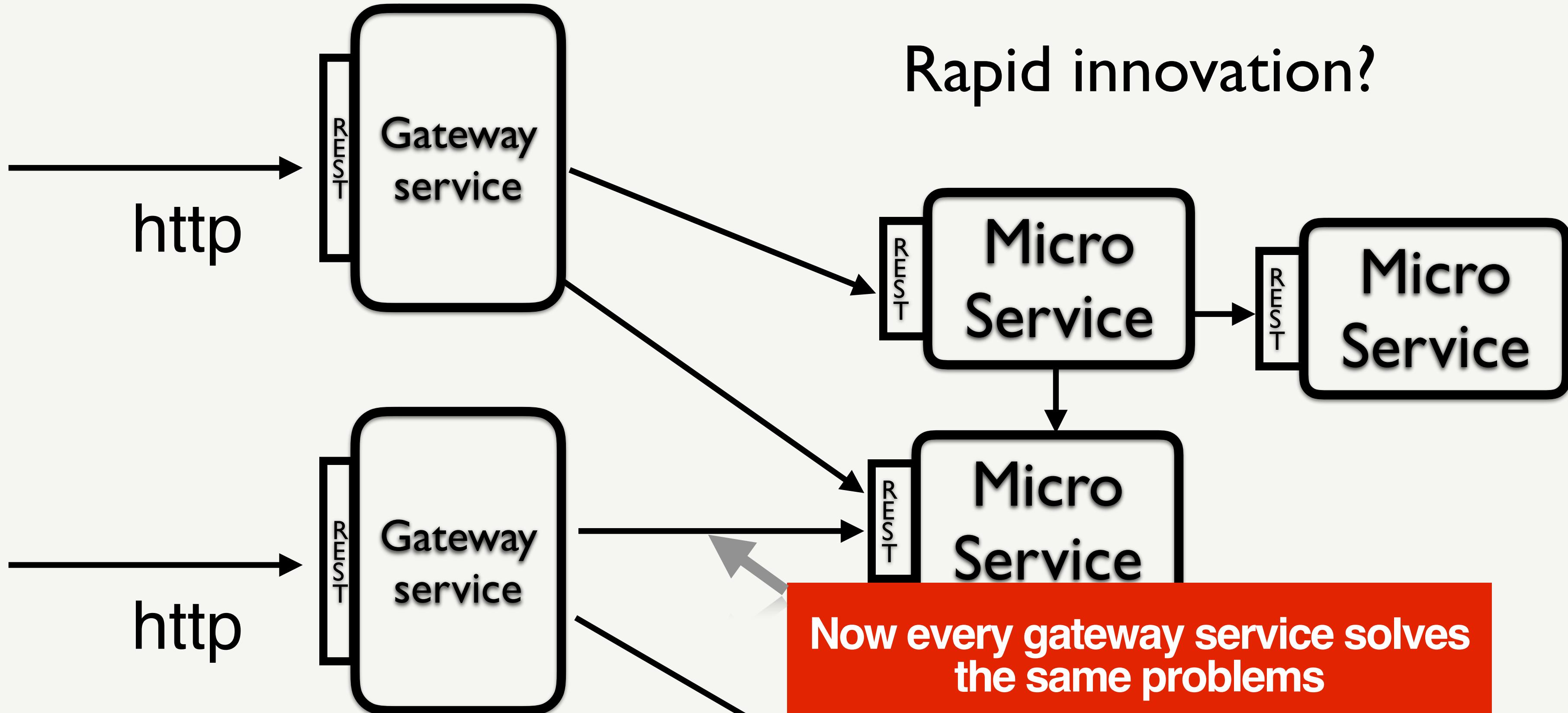


Edge Developer  
Experience



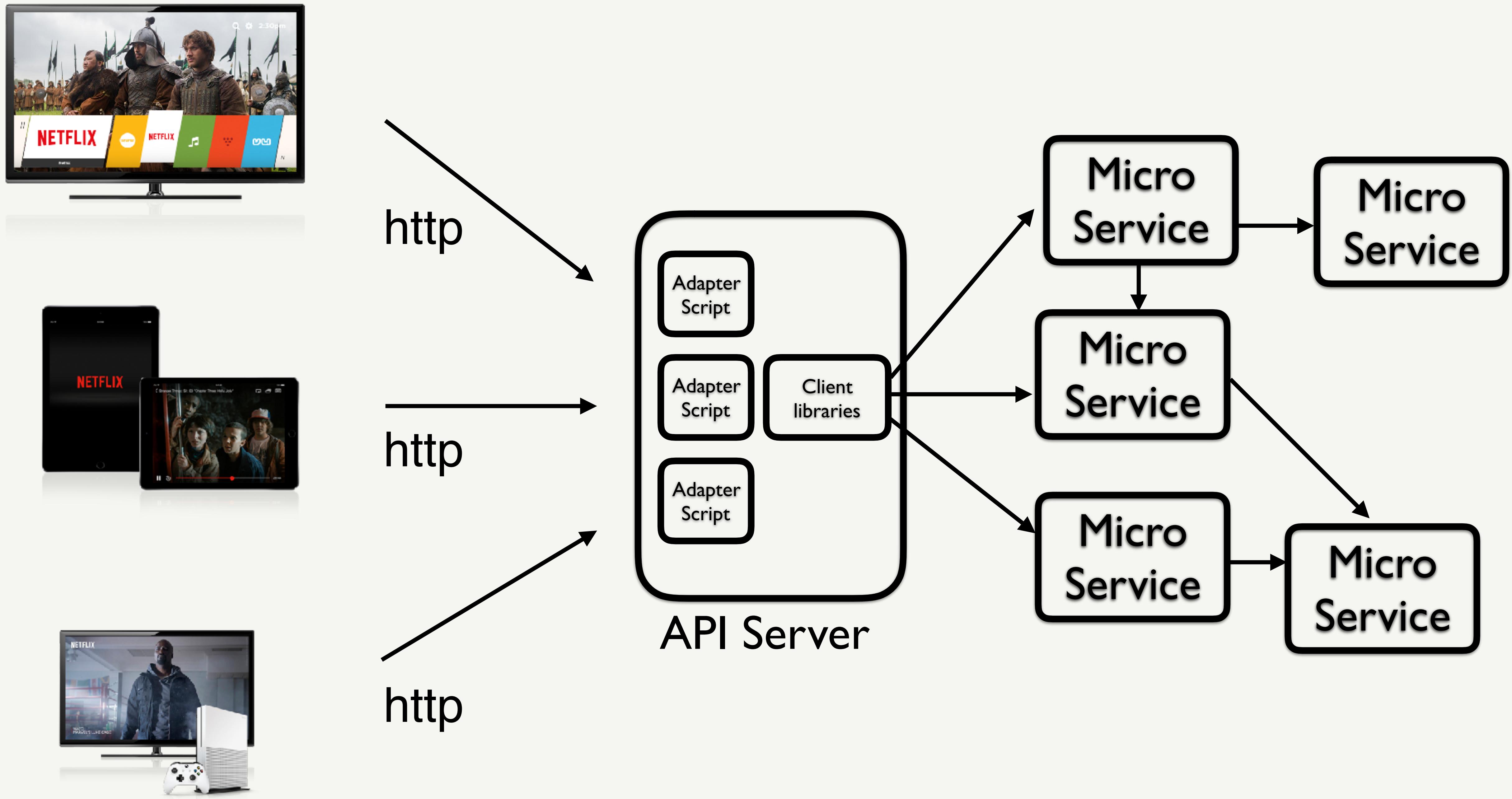


Rapid innovation?



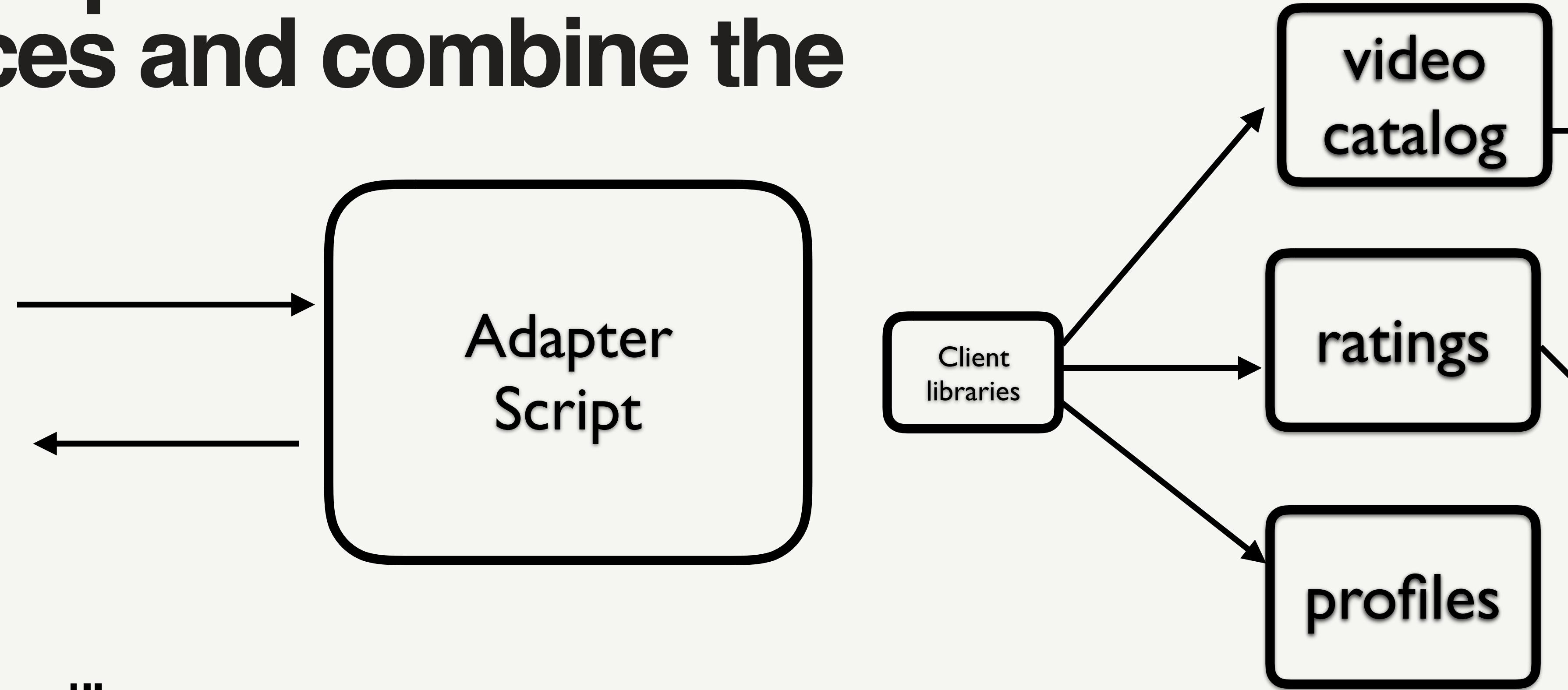
# Current Edge architecture(s)



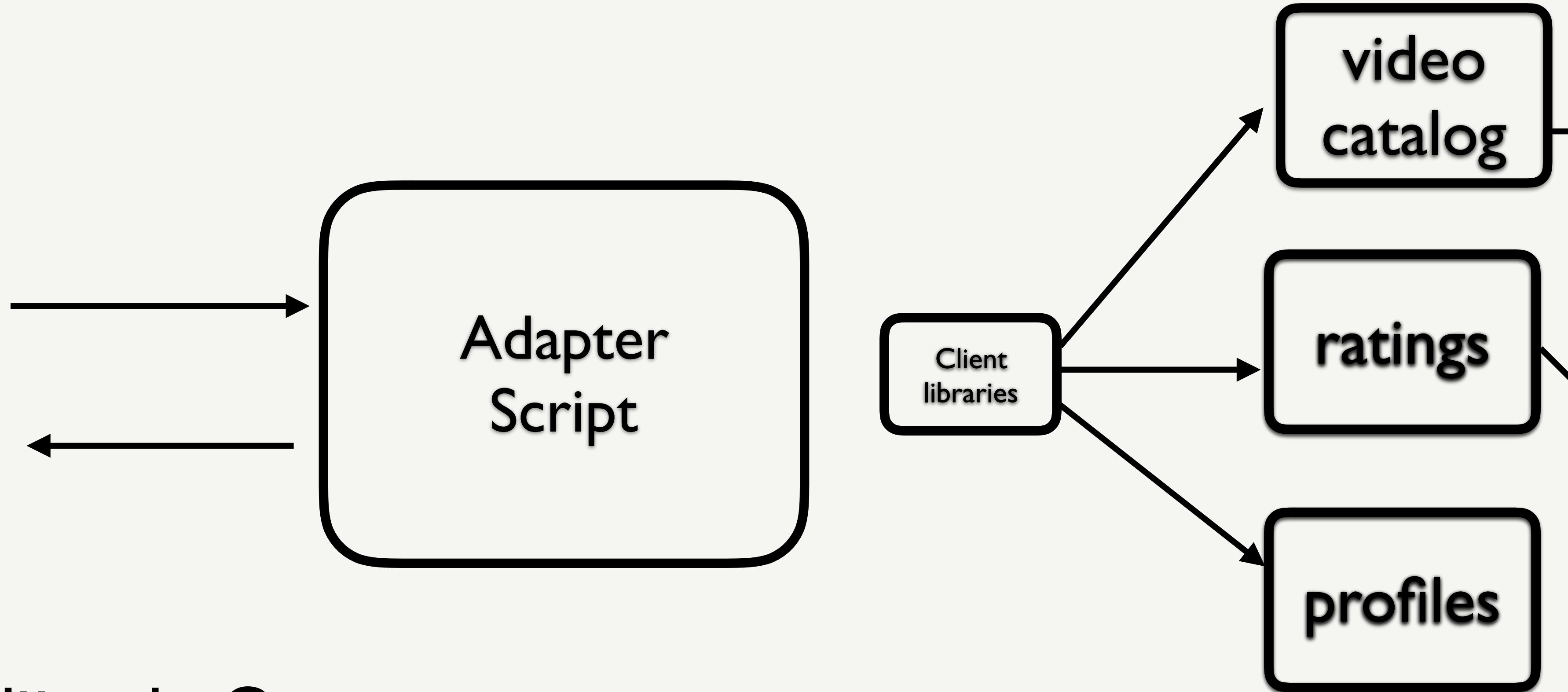


# The adapter scripts fetch data from multiple services and combine the results

```
{  
  "userProfile": "Paul",  
  "videoTitle": "Narcos",  
  "rating": "THUMBS_UP"  
}
```



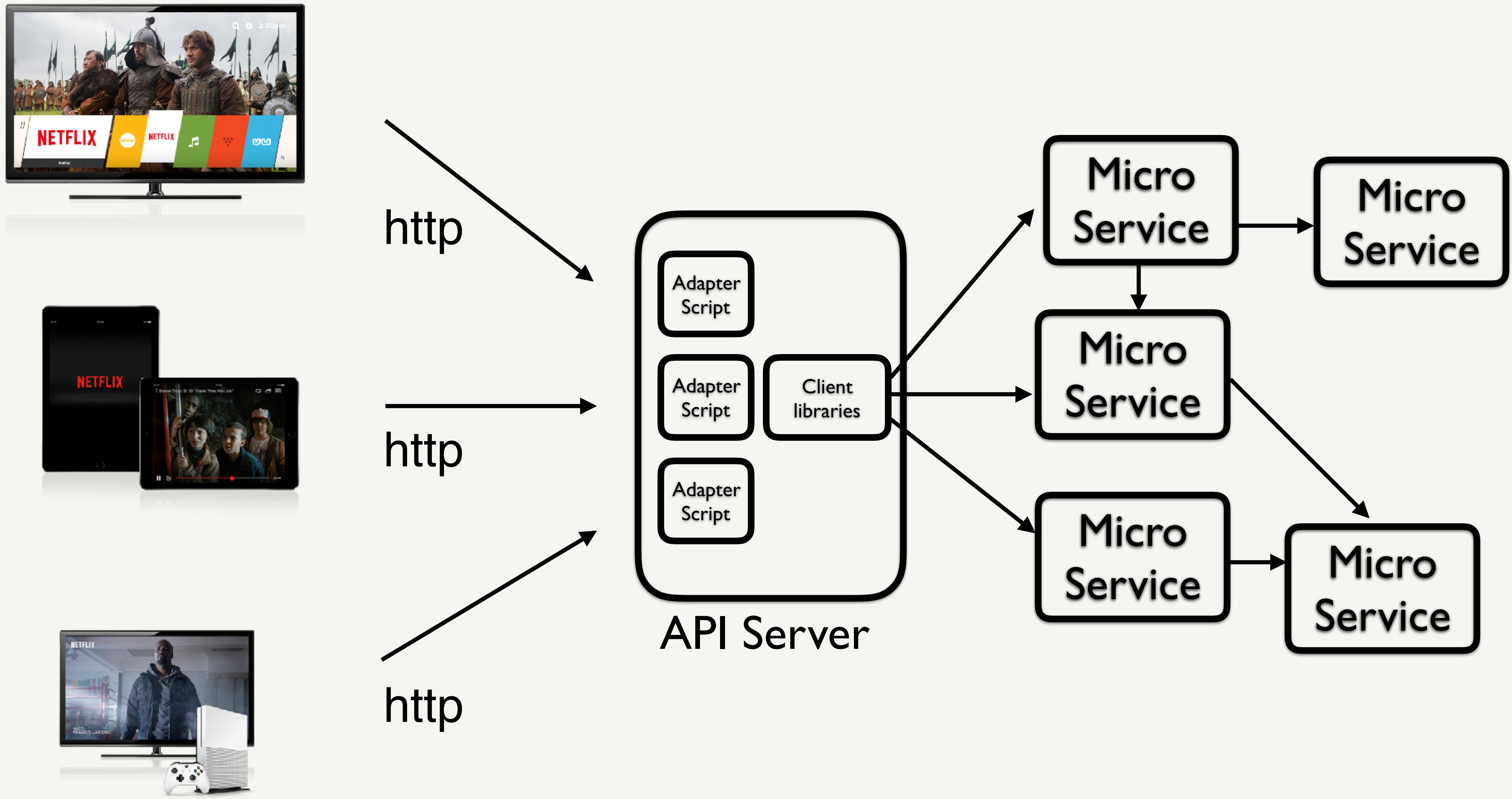
# The Netflix API server



- Adapter scripts written in Groovy
- Uploaded to the API server
- FaaS from the developer point of view?

A man with a long, dark beard and mustache is shown from the waist up, wearing detailed, ornate armor. He is holding a large, gold-colored sword hilted with intricate designs. The background is a warm, golden-yellow color.

The micro service monolith



# Monoliths are scary.

- Single point of failure
- The API server is BIG
- One bad script, config or client library can affect availability



# Did 'Luke Cage' Break Netflix? Outage Leaves Saturday Bingers In Dark

by Ross A. Lincoln  
and Dominic Patten

October 1, 2016 2:59pm



O'REILLY®

Software  
Architecture  
CONFERENCE

February 25-28 • New York

Engineering the  
future of software



▶ TV

- ▶ NEWS
- ▶ LUKE CAGE
- ▶ MARVEL
- ▶ NETFLIX

TECH

TECH | MOBILE | SOCIAL MEDIA | ENTERPRISE | CYBERSECURITY | TECH GUIDE

## 'Luke Cage' overload? Netflix viewers report outage amid binge of new show

Javier E. David | @TeflonGeek

Published 3:54 PM ET Sat, 1 Oct 2016 | Updated 2:40 PM ET Sun, 2 Oct 2016



Related

[STX To Distribute Martin Scorsese's 'The Irishman'](#)  
[After All With China Deal: The Director's Next Project](#)

While Netflix did not respond to requests for comment,



Jim Spellman | Getty Images

Does Power Man have the ability to fix a broken streaming platform?

# Netflix is down this weekend, inter Cage binge (Update: It's back)

By Andrew Liptak | @AndrewLiptak | Oct 1, 2016, 4:26pm EDT



SHARE

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

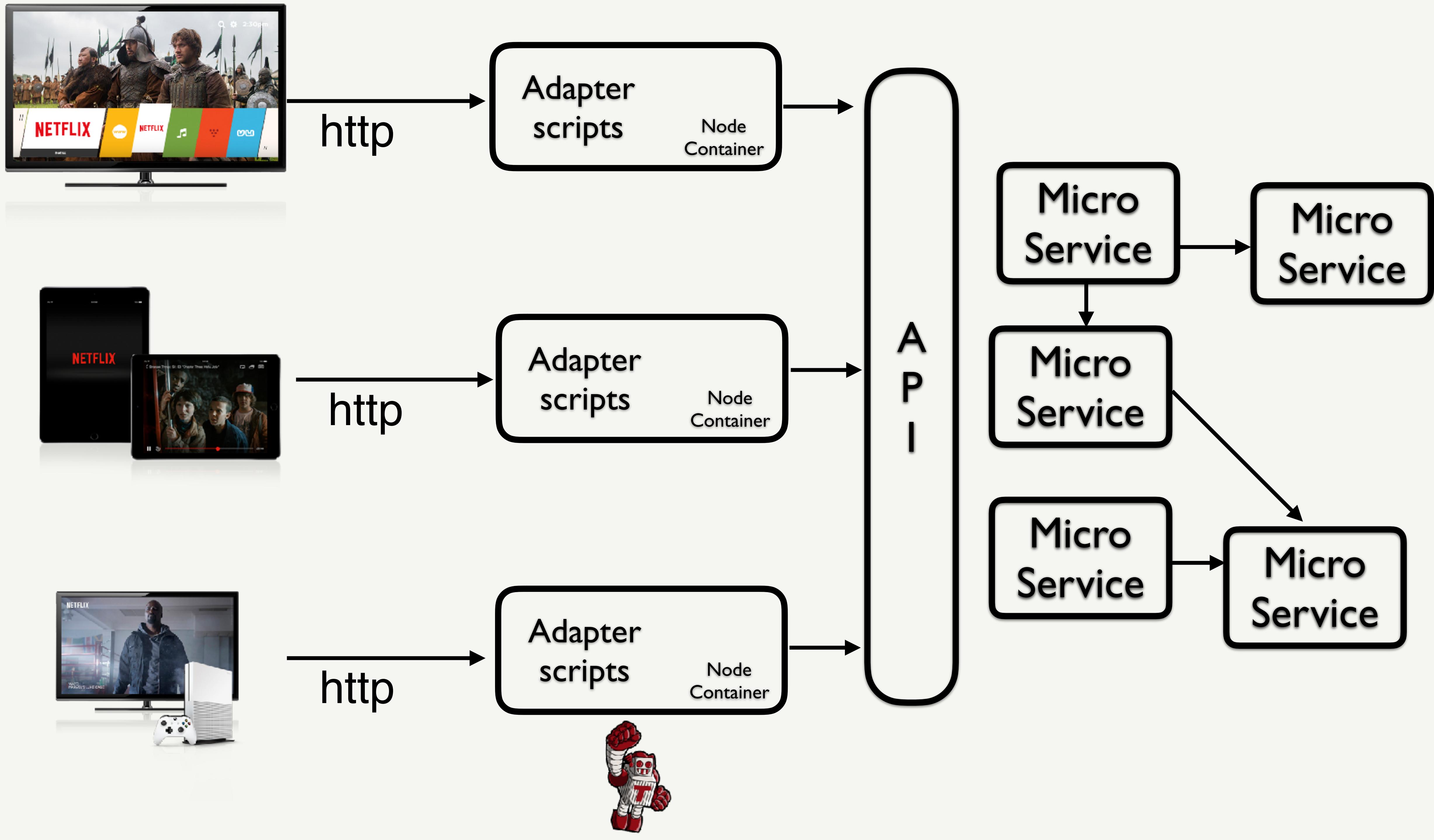
▼

▼

▼

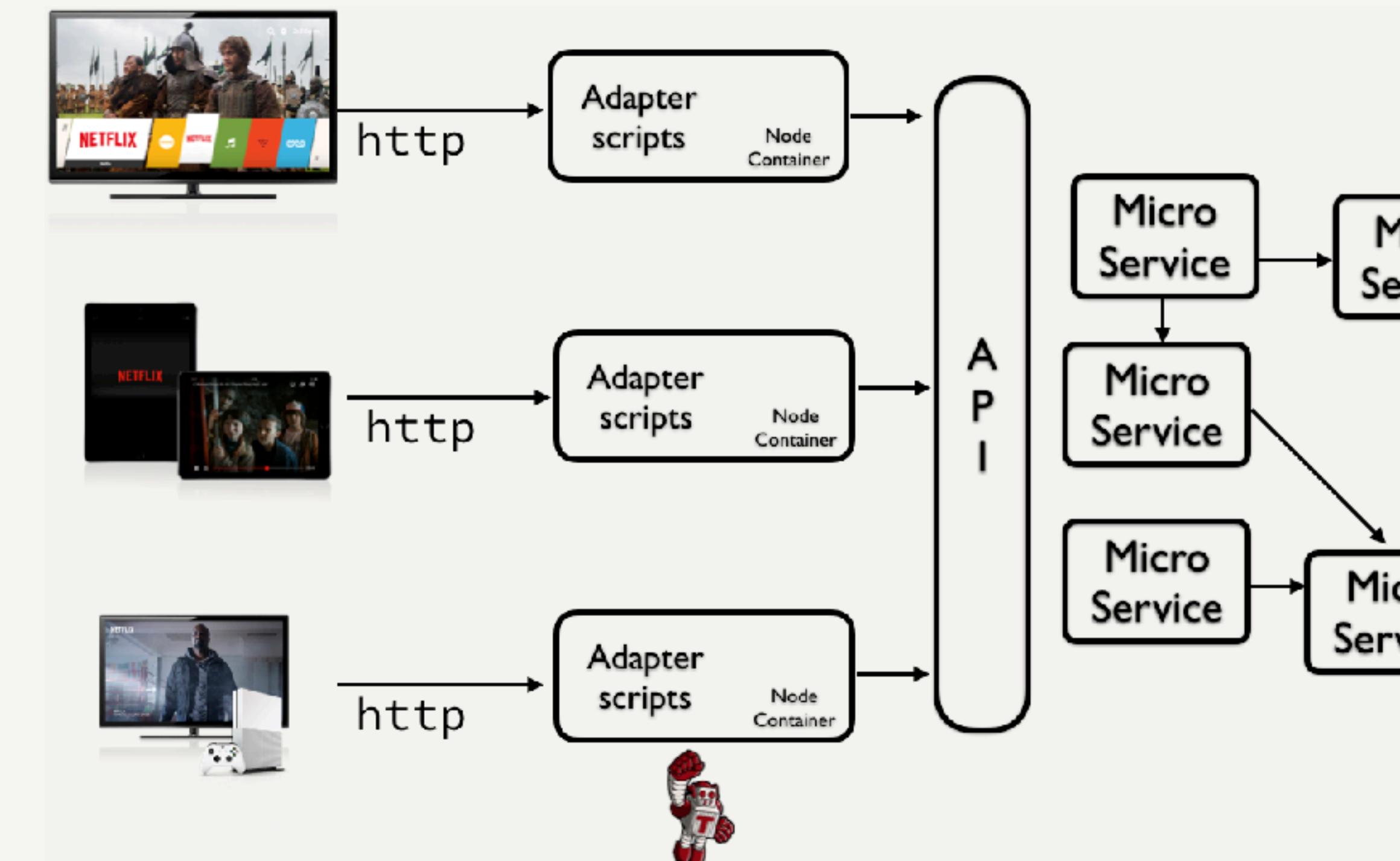
▼

▼



# Breaking things up

- Each set of endpoint scripts is deployed as a lightweight service
- Using containers to decrease deployment overhead
- Better deployment isolation
- Better Developer Experience!

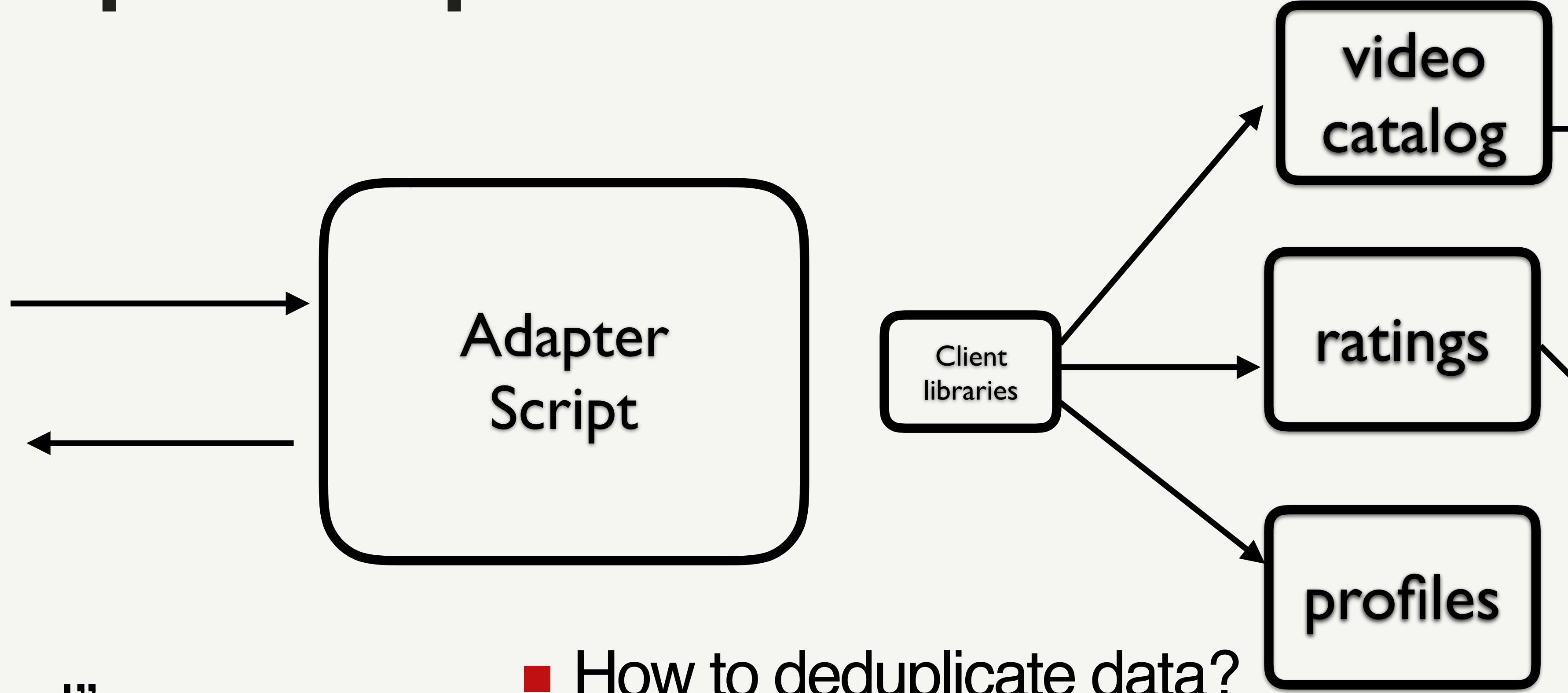


# Adapter scripts

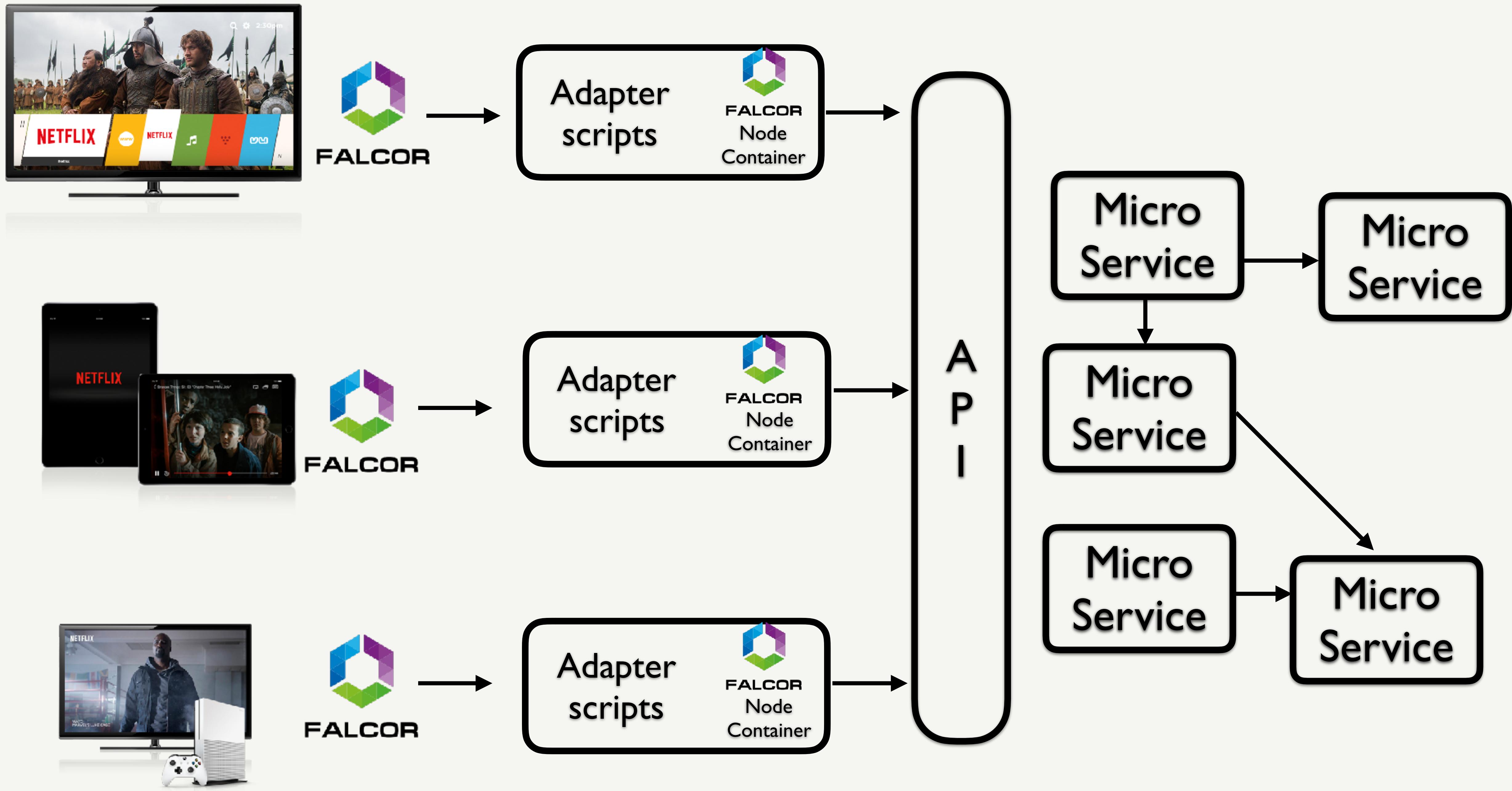


# How to write adapter scripts?

```
{  
  "userProfile": "Paul",  
  "videoTitle": "Narcos",  
  "rating": "THUMBS_UP"  
}
```



- How to deduplicate data?
- How to represent downstream errors and timeouts?
- Is JSON ideal from the client's perspective?



# There is no API

## Just declare what data you need.

```
getVideoTitleById(...)  
getProfile(...).getName()  
getRatingForVideoId(...)
```

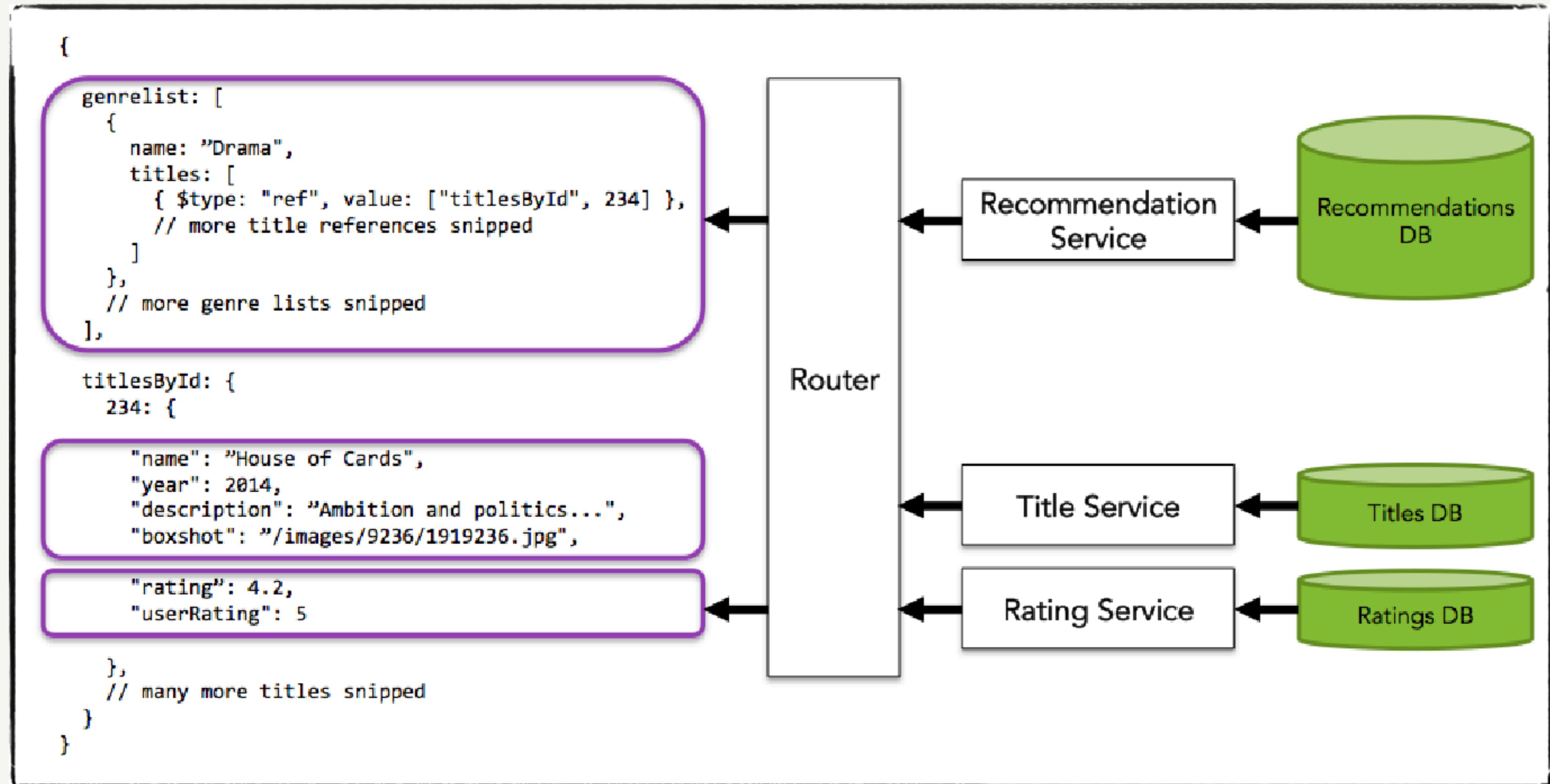


```
[  
  ["currentProfile", "allVideosRated", 0, ["title", "thumbRating"]]  
  ["currentProfile", "name"]  
]
```

## ■ Treat multiple data sources as one *logical* model

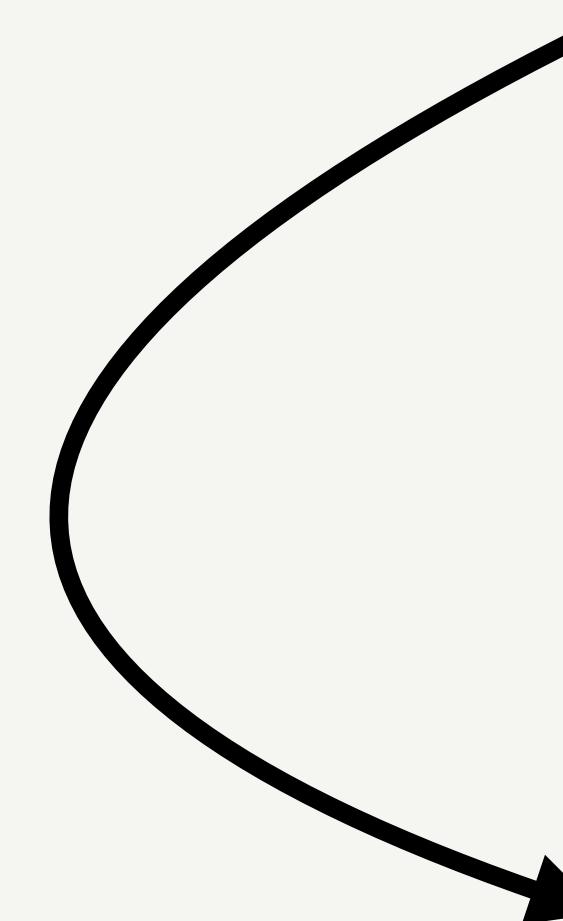


# FALCOR



# JSON Graph - refs.

- Deduplicate data
- Access data as a logical model



```
{  
  "allVideosRated": {  
    "28": {  
      "$type": "ref",  
      "value": [  
        "profilesByGuid",  
        "UXXGMA0J3VGGLGWTQ6NMN70PGU",  
        "videosById",  
        80190279  
      ]  
    }  
  },  
  "videosById": {  
    "80190279": {  
      "thumbRating": {  
        "$type": "atom",  
        "value": {  
          "rating": "THUMBS_UP",  
          "timestamp": 1514346641221  
        }  
      },  
      "title": "The Indian Detective"  
    }  
  }  
}
```

```
get( ["allVideosRated", 28, ["title", "thumbRating"]]);
```

# API Discovery



**If “there is no API”....**

**How do I know what to do?**

# Discovering *paths* and putting them together

The screenshot shows the SKIPPER interface for the DNA service. The top navigation bar includes links for SKIPPER, SERVICES, API, DOCUMENTATION, SUPPORT, and phdker. The main content area is titled "DNA (Discovery, Non Member, Account)".

The configuration form contains the following fields:

- Customer:** tester\_bjacobs@netflix.com (selected)
- Email:** tester\_bjacobs@netflix.com
- Name:** Ben
- Country:** US
- Status:** CURRENT\_MEMBER
- Delete from cache:** (button)
- Profile:** Ben
- Request Context:** {"obelixNamespace": "Default", "obelixBundle": "tvui-eureka", "uiFlavor": "darwin"}
- ESN:** NFXONE-02-10UVXBIFPEUJIB3997GBJP92D4GNXGBABISAYVBXAFORWP4U93E4
- Country override:** (empty field)

Below the form, there are tabs for EXPLORER, REPL, and Mock generator. A message indicates the available routes for the DNA API service in TEST.

The routes listed are:

- > CALL ["authenticate"]
- > CALL ["updateIdentityInfo"]
- > GET ["email"]
- > GET ["isTestAccount"]
- > GET ["hasAlerts"]

On the right side, there are sections for ABTESTING and ACCOUNT.

# Edge PaaS - Developer Experience

The right balance between  
*serverless* and *developer experience*?

# “API Script” developers

## Disappointed

- Can't run code locally
- No step debugging
- No unit testing

## But spoiled

- Deployment is trivial
- Easy to support many versions
- Nothing to manage!



# Developer Experience

- Projects runs in container on developer machine
- Local editing and debugging
- External routing to developer machine
- Run locally what runs in production

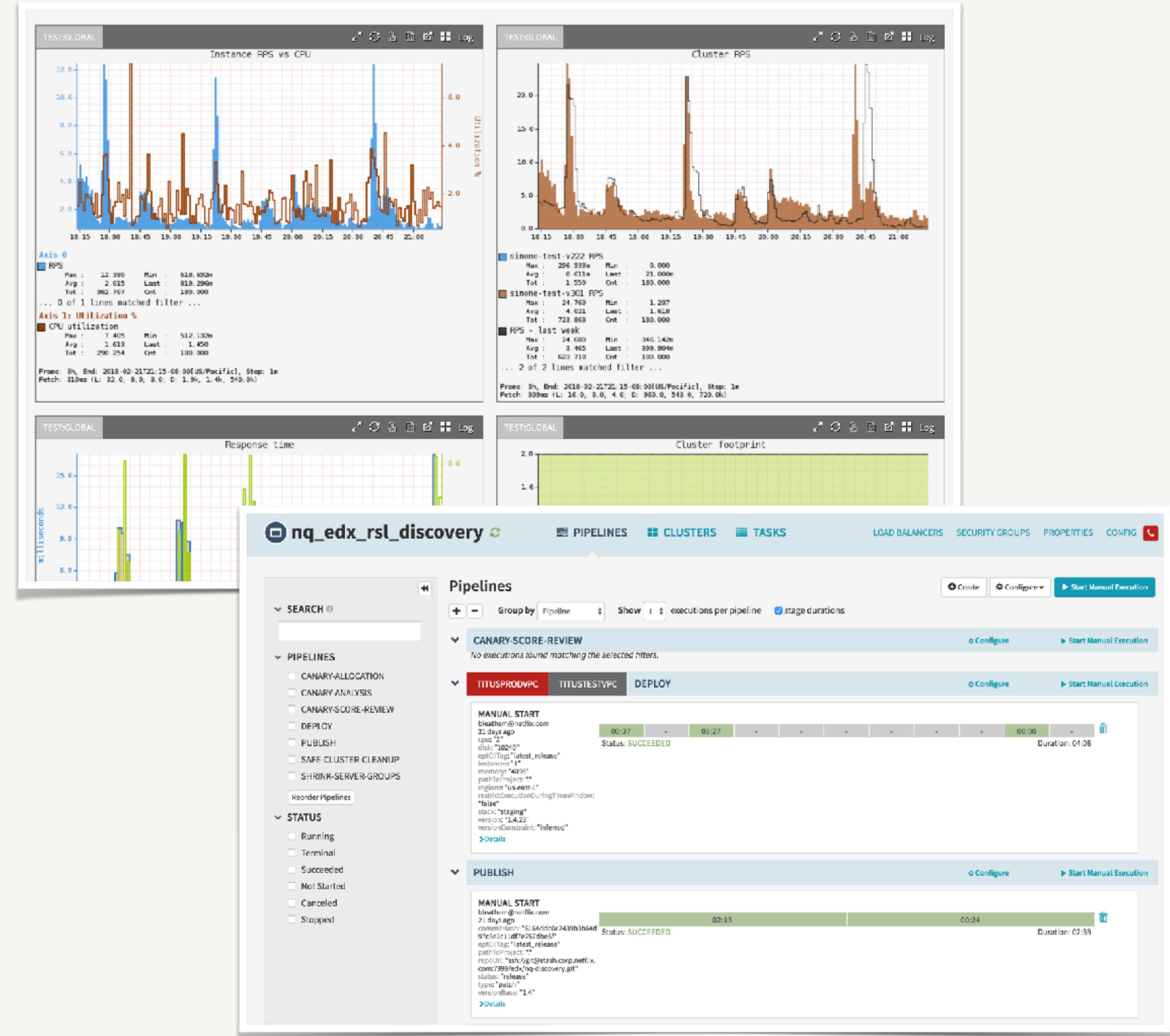


# NodeQuark demo



# Out of the box.

- Metrics
- Alerts
- Streaming logging
- Deployment pipelines
- Canary analysis



# Thank you.

Paul Bakker  
@pbakker

NETFLIX