

WHISKEY  
SAYS  
HI!





# TRAFFIC OPS: FROM MONOLITH TO SOA

Srijeet Chatterjee,  
Community Over Code, October 2023,  
Halifax, CA

# A LITTLE BIT ABOUT ME...

- Principal Engineer, CDN, Comcast
- Been with Comcast for 7 years
- Committer on Apache Traffic Control
- Live in Denver, Colorado
- Originally from (close to) Mumbai, India
- Likes: Solving challenging problems, hiking, running, dogs, food, beer, cricket, football (a.k.a soccer)
- Dislikes: Cold weather/ snow (Why am I in Canada?!)
- LinkedIn: <https://www.linkedin.com/in/srijeet-chatterjee-74809731/>
- Email: [srijeet0406@gmail.com](mailto:srijeet0406@gmail.com)
- Slack: Apache Traffic Control slack

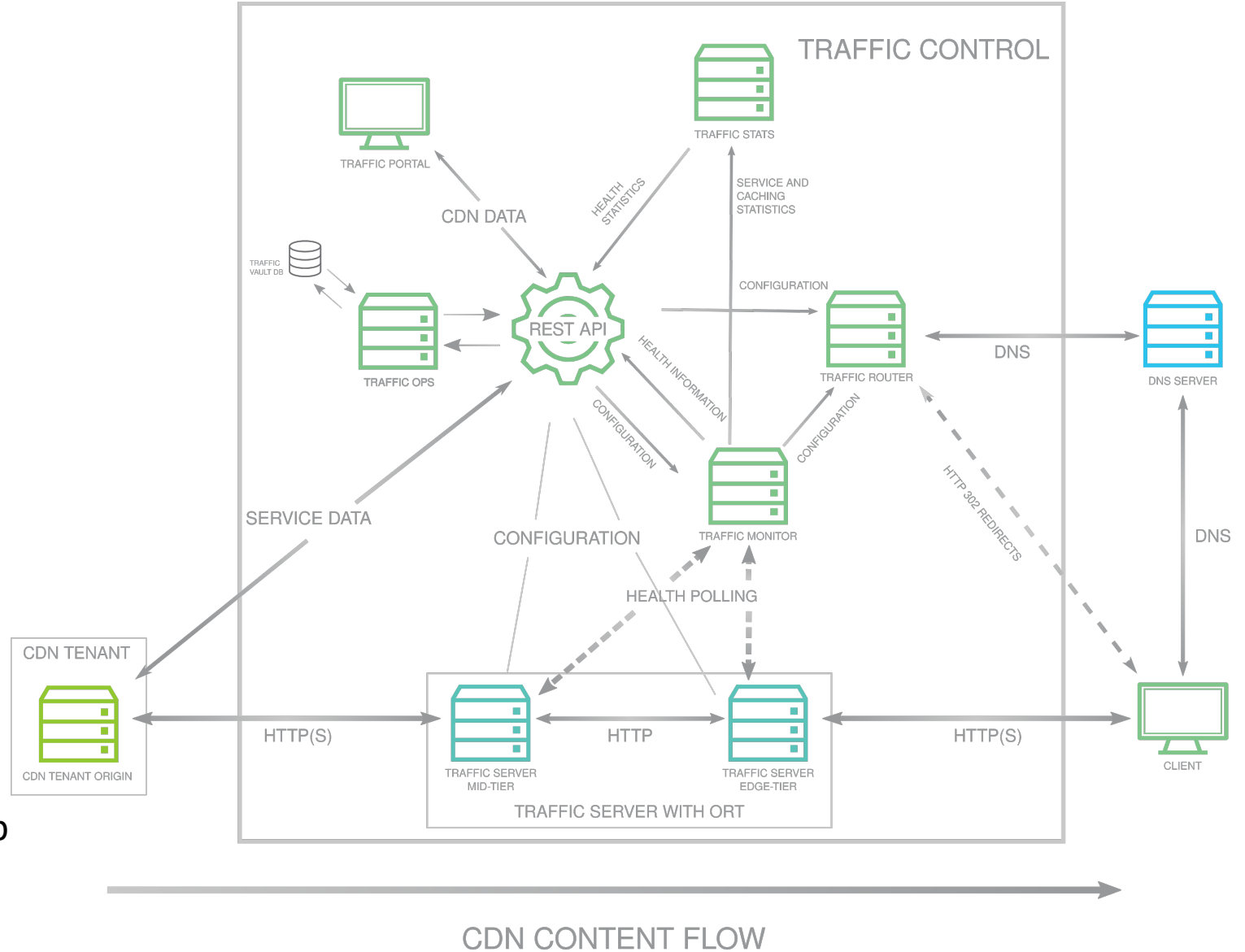


# WHAT IS SOA?

- Service Oriented architecture (SOA) is a method of software development that uses software components called services to create business applications<sup>[1]</sup>.
- Each service provides a business capability, and services can communicate with each other across platforms and languages.
- Seamless from end user perspective.

# OVERVIEW OF THE APACHE TRAFFIC CONTROL SYSTEM

**Traffic Monitor:**  
 Manages and configures the CDNs for traffic for Apache. Traffic Ops interacts and setup the CDNs on the edge nodes of the hashing hashes to the TO database



# MOTIVATIONS

- Traffic Ops (TO) was previously a monolithic application
- Converting TO to an SOA product would help us move faster in every way
- Smaller releases => Eliminating restarting TO, lesser bugs
- Quicker to production
- Quicker resolution of bugs and customer requested features
- Flexibility: Leave the TO “core” as it is, develop plugins/ services for other requirements
- Why not just use plugins? -> This is currently already integrated with TO => Requires restarting TO.

# GOALS

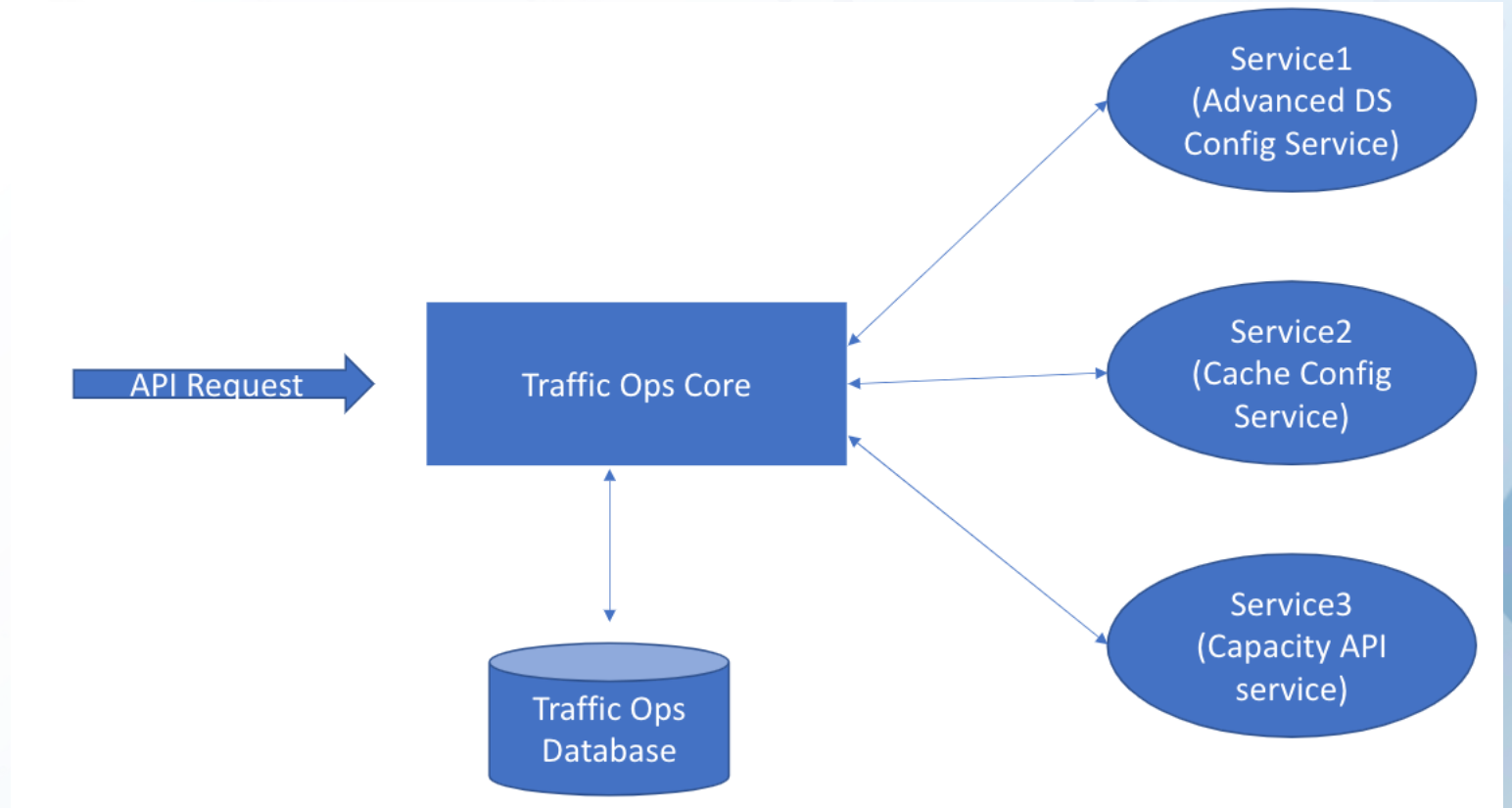
- Quicker deployments – Deploying a single service vs a whole application => Happier Ops team
- De-coupled architecture – Don't need to upgrade the whole product
- Scalability – Ability to scale individual services independently of TO Core
- TO remains the proxy to the database (TODB)
- Testing and security built into the services; can be exercised outside the scope of ATC/TO
- Increased ease of use for new contributors; don't need to learn the whole TO code base

# GOALS (2)

- Language agnostic; backend services are not limited to Golang => everyone is welcome!
- Backend services should have the ability to incorporate their own databases and/ or UI.

# PROPOSED MODEL

**SERVICE  
ORIENTED  
ARCHITECTURE**



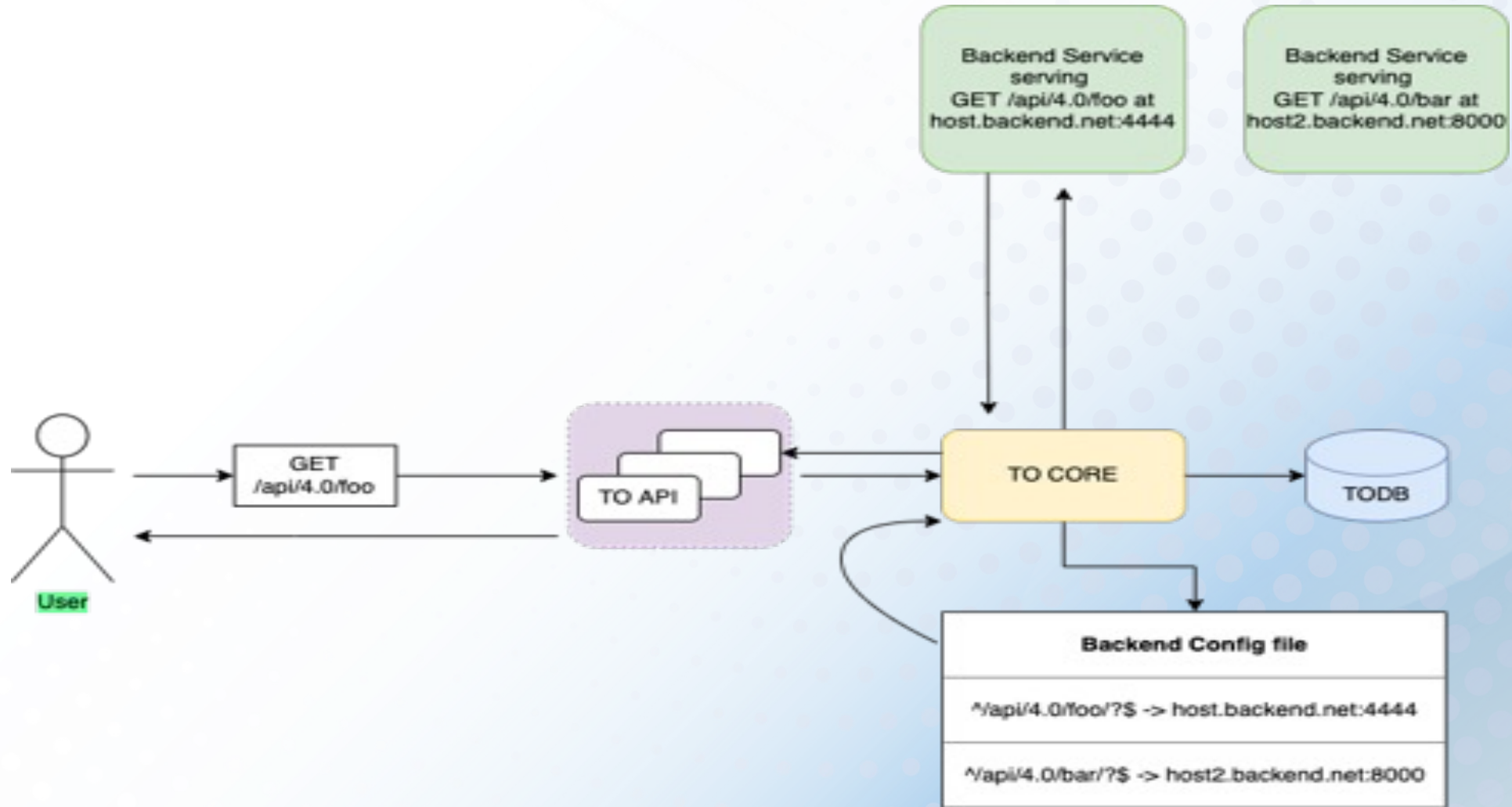
# SOA REFACTORING

- New config option to TO: **-backendcfg**

```
{
  "routes": [
    {
      "path": "api/4.0/foo",
      "method": "GET",
      "hosts": [
        {
          "protocol": "https",
          "hostname": "localhost",
          "port": 8444
        }
      ],
      "insecure": true,
      "permissions": [
        "CDN:READ"
      ],
      "routeId": 123456,
      "opts": {
        "alg": "roundrobin"
      }
    }
  ]
}
```

How do you want to distribute traffic among the backend hosts for this route?

# NEW TRAFFIC OPS ARCHITECTURE



# BACKEND SERVICE REQUIREMENTS

- Can be as simple as a HTTP server, with one endpoint
- Can be written in any language
- Can have its own database if needed (separate from TODB)
- Security – hosted on the same server as TO (for now), can share the same certificates
- Unit/ Integration test – responsibility of the backend service
- UI considerations – can have its own UI, or a separate module in Traffic Portal (TP)
- TO should not have to be restarted/ redeployed if backend services are added/ deleted/ modified -> SIGHUP

# DEMO



# USEFUL RESOURCES

- [Blueprint](#)
- [Pull Request](#)

# ACKNOWLEDGEMENTS

- Dave Neuman, who had initially proposed this idea.
- Rawlin Peters, for his guidance and Traffic Ops expertise.
- The entire ATC community for their continuous feedback.

# FUTURE WORK

- Supporting more than “round-robin” in the “options” section -> weights?
- Security: considering not sharing certificates with TO.
- Convert some of the existing functionality into backend services.

QUESTIONS?



Image courtesy: <https://unsplash.com/photos/AoqgGAqrLpU>

# Thank you!

