Dynamic Discovery (D2)

Service Discovery and Load Balancing

Shivam Gupta
Sr Software Engineer

Sanjay Singh
Sr Software Engineer

D2 – Dynamic Discovery

- Part of the open source Rest.li framework
- Translates a REST resource or endpoint to an IP-address/hostname

```
d2://service-name/123 → http://my.hostname.biz:9520/context/123
```

- A library that uses Zookeeper as the registry store
  - Implementations include Java, C++, and Python

D2 – Responsibilities

Service Discovery

Maintain a registry of online hosts for each microservice

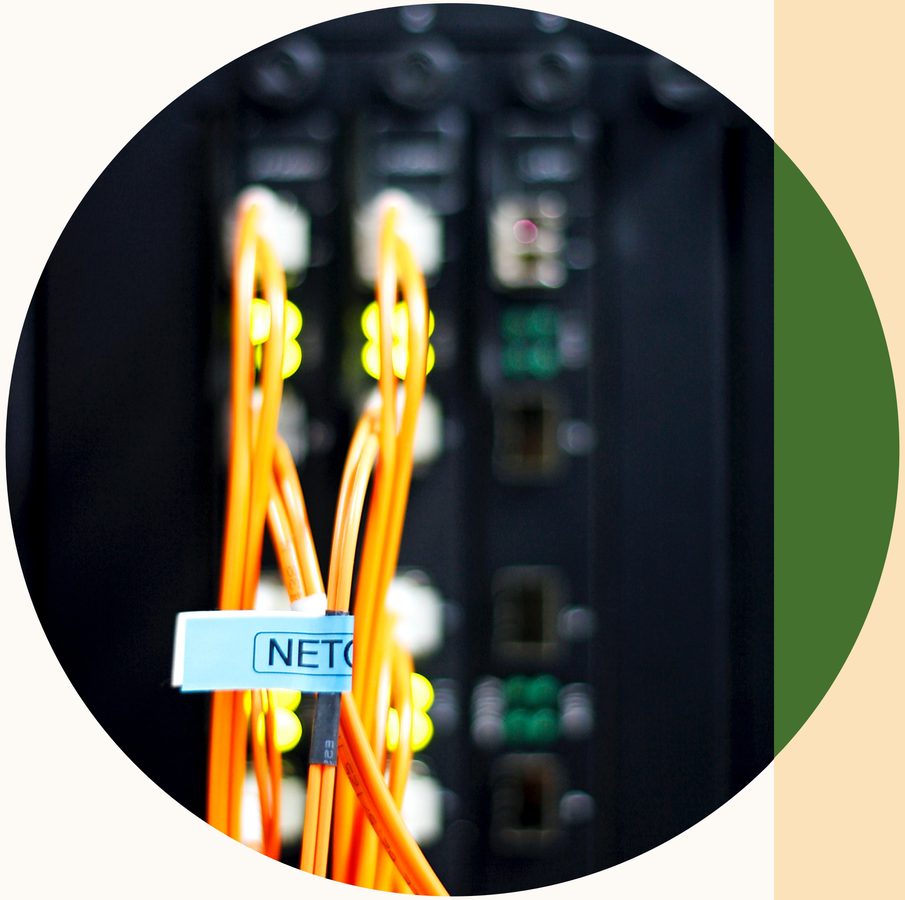
Load Balancing

Ensure fair distribution of traffic to available hosts for optimal performance

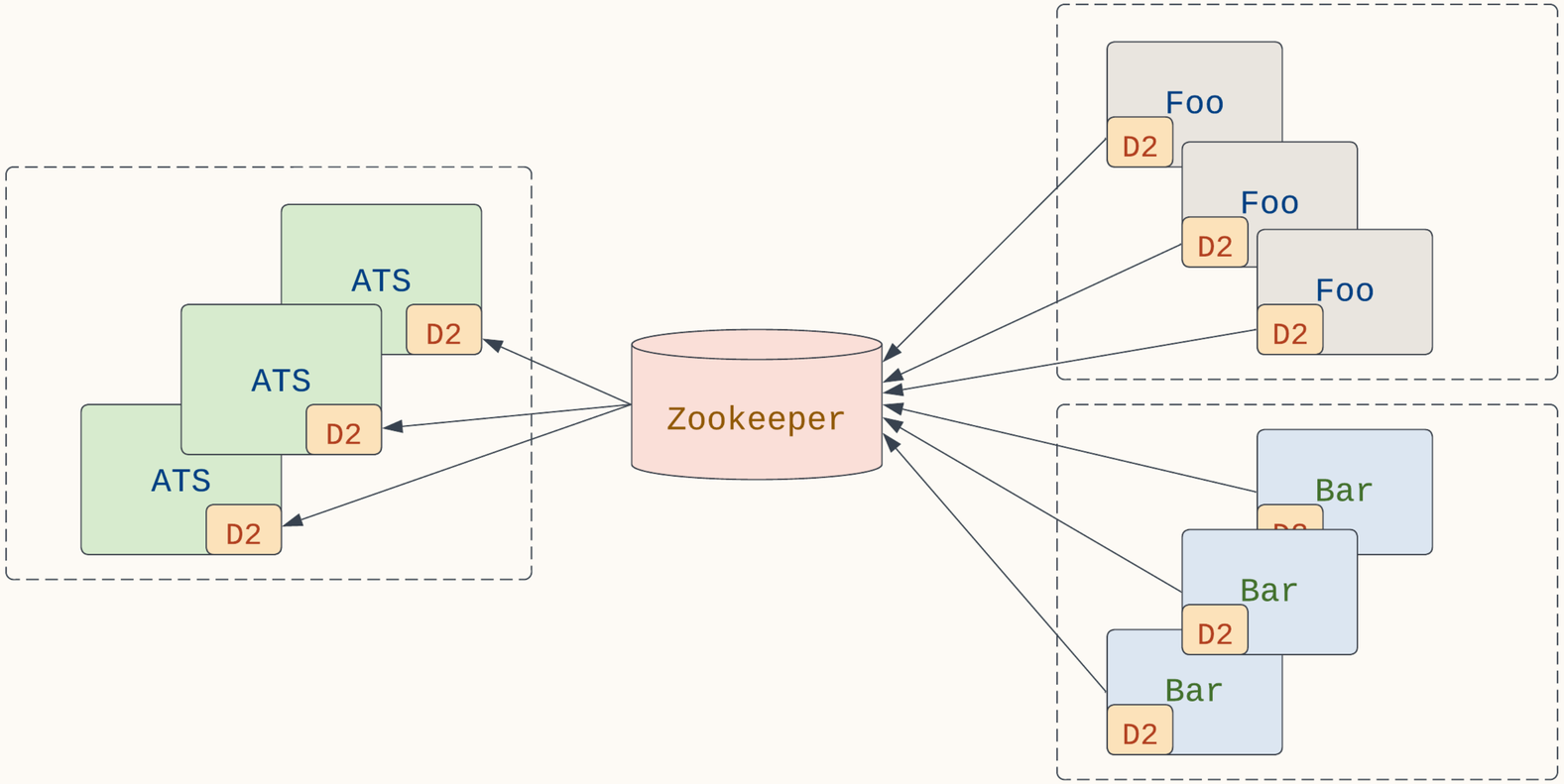
Why Not DNS?

Domain Name System (DNS) has several limitations:

- Slow update propagation and less reliability
- Basic load balancing – inability to incorporate parameters like server load or latency
- Can’t support advanced use-cases like load tests

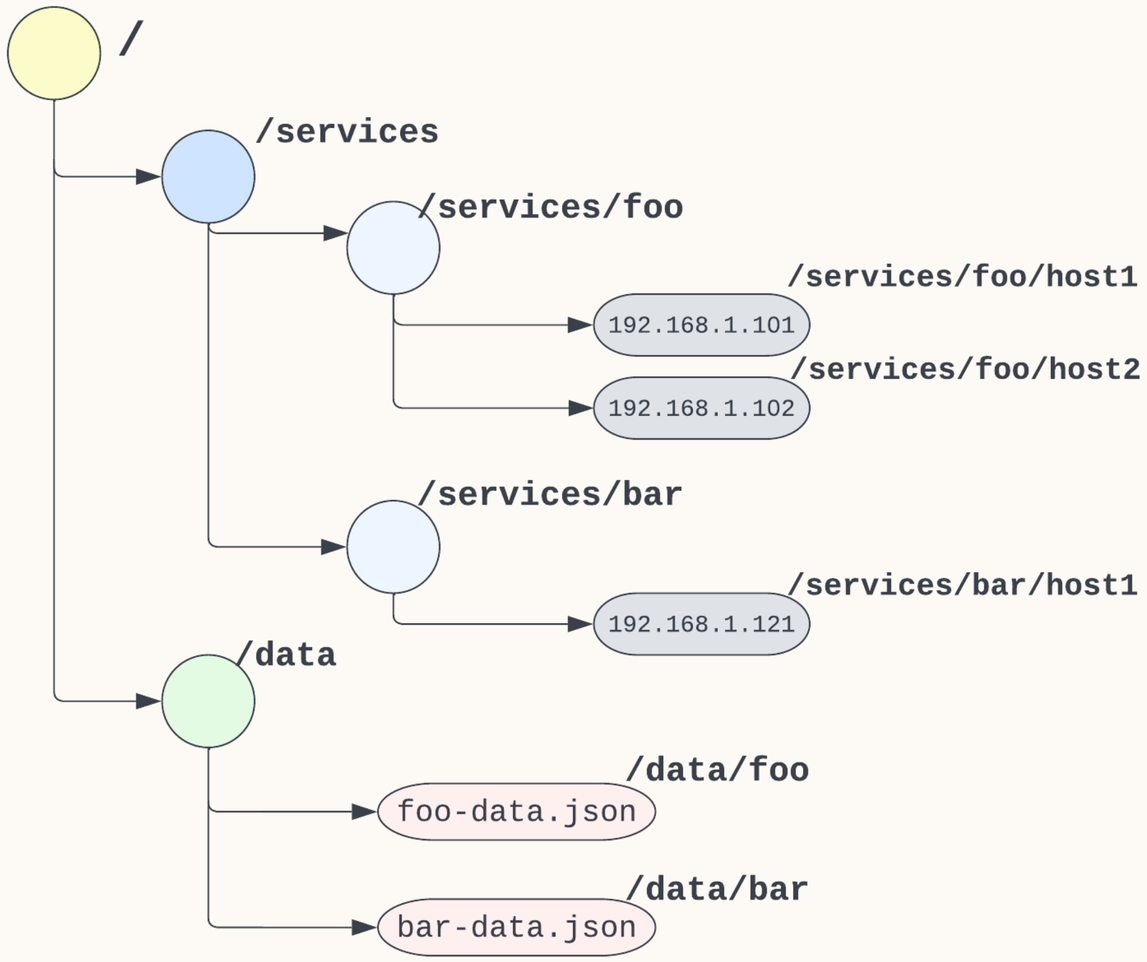


D2 – Service Discovery



D2 and Zookeeper

- ZK for highly reliable coordination of distributed applications
  - Faster updates
- ZK functionality well-suited for service-discovery
  - Ephemeral data
  - Watches
- D2 library builds an in-memory cache of ZK service registry and listens for updates



D2 – Load Balancing

- Two modes – *random*, *degrader*
- *Random*: picks a random host
- *Degrader*: passive health checks
  - Tracks calls to monitor health
  - Cluster-level and host-level tracking
  - Active health checks done by ZK

D2 – Load Balancing

- Cluster-level health tracking
  - Tracks calls to monitor health
  - Drops traffic if threshold is exceeded
- Host-level health tracking
  - Tracks latencies and error-rates for each host
  - Assigns a *weight* to each host in the cluster
  - *weight* == probability of selection

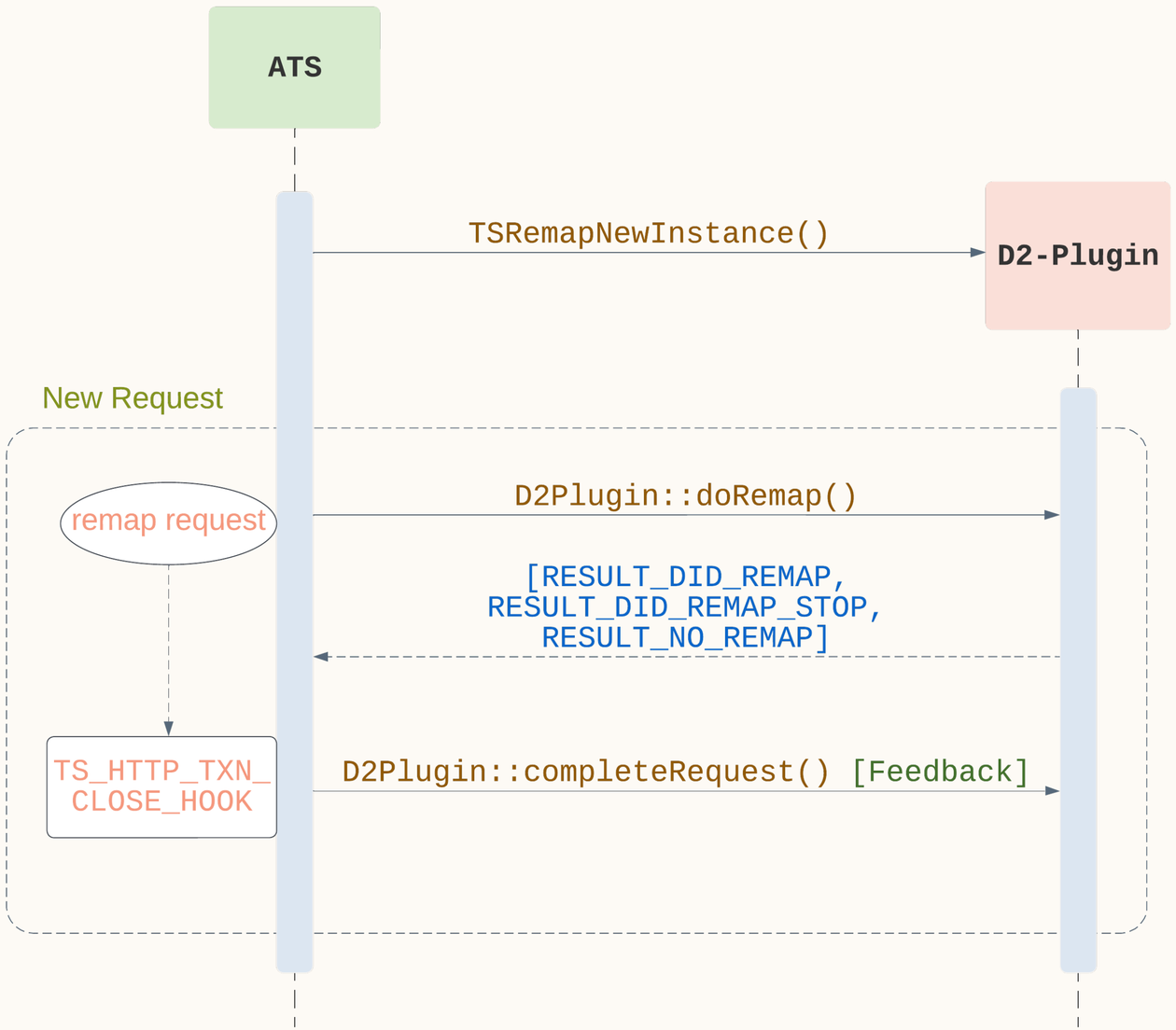
ATS-D2 Implementation

```
map  /feed/
http://feed.dns.disco.linkedin.com:1234/feed/
@plugin=d2-plugin.so
@pparam=d2://feedService/
```

- Global + Remap plugin
- `http://feed.dns.disco.linkedin.com:1234/feed/`: Fallback "map_to" URL
- `d2-plugin.so`: D2 plugin shared object file
- `d2://feedService/`: D2 service name to be used for routing

ATS-D2 Implementation

- Subclasses “atscppapi::RemapPlugin”
- Registers for “TS_HTTP_TXN_CLOSE_HOOK” to get feedback

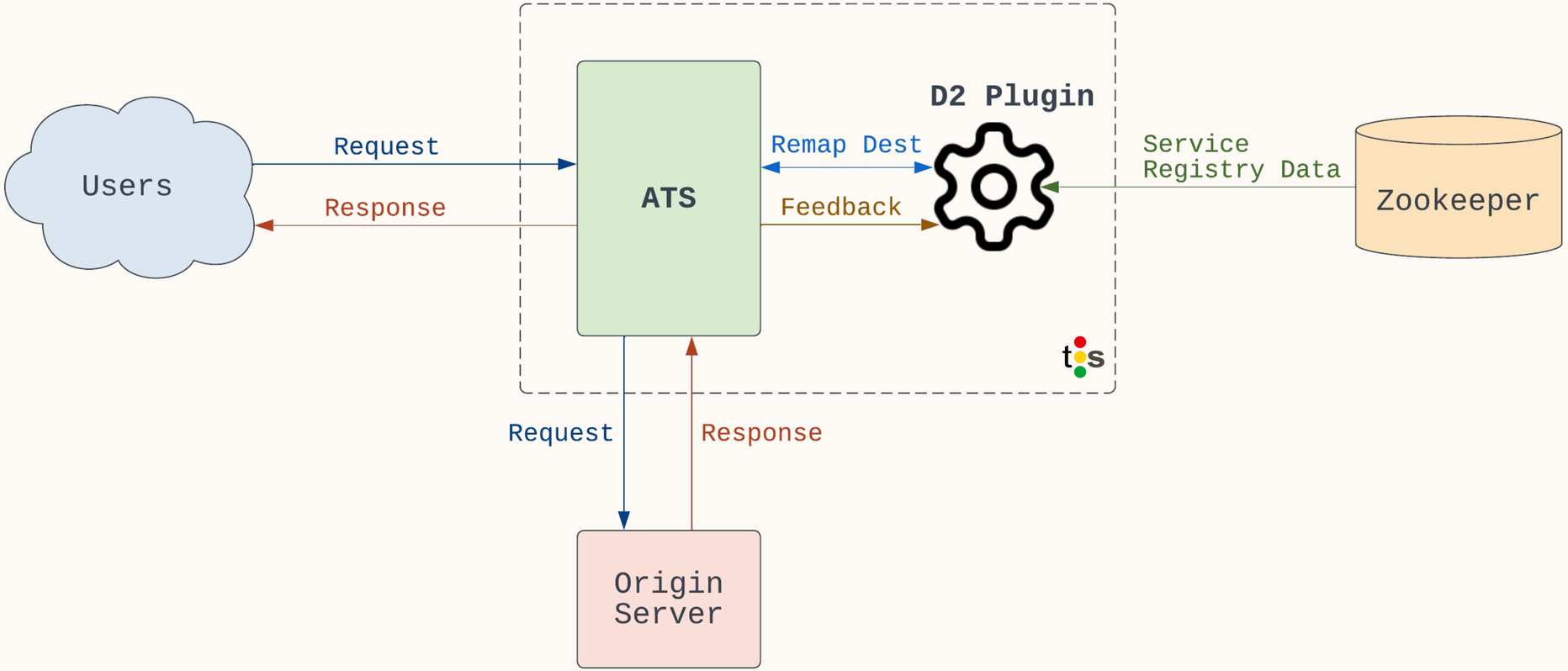


ATS-D2 – Advanced Features

```
map  /feed/  http://127.0.0.1:1234/feed/
@plugin=d2-plugin.so
@pparam=`{
            "d2_service": "feed",
            "abtest": {
              "testkey”: "ats.abtest.key",
              "treatments":{
                "canary":{"d2_service:"feedNew"}
              }
            }
          }’
```

- Service load tests by modifying D2 weights
- A/B testing and ramping new services
- Quarantine feature
- Backup of service registry data to flat files

Summary


Users
Request
Response
ATS
D2 Plugin
Remap Dest
Feedback
Service
Registry Data
Zookeeper
Request
Response
Origin
Server

D2 – Challenges

… and Future Scope

- Problems due to a large ATS fleet
  - Herd behavior
  - Ineffective degrader load balancing, especially for low QPS services
- Fine grained load balancing state
  - Maintained at individual REST resource level
  - Performance constraints
- Programming language dependence

Thank you

LinkedIn