



team.work() //conference 2018



Tenant Context Service

Case study of a tier 0, multi-region, HA service

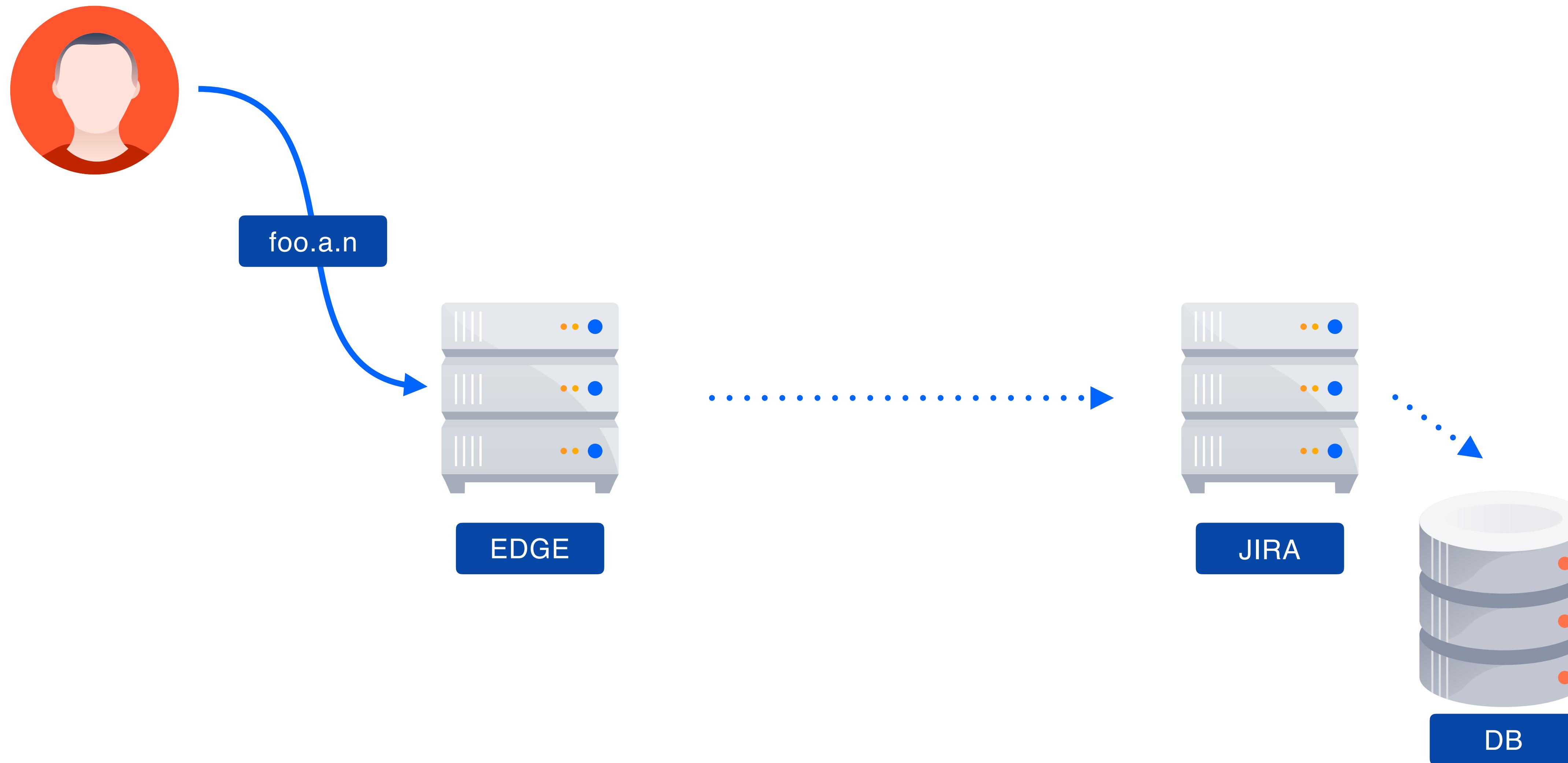


JACKSON MOES | SENIOR DEVELOPER - CLOUD PROVISIONING | @JSONMOES

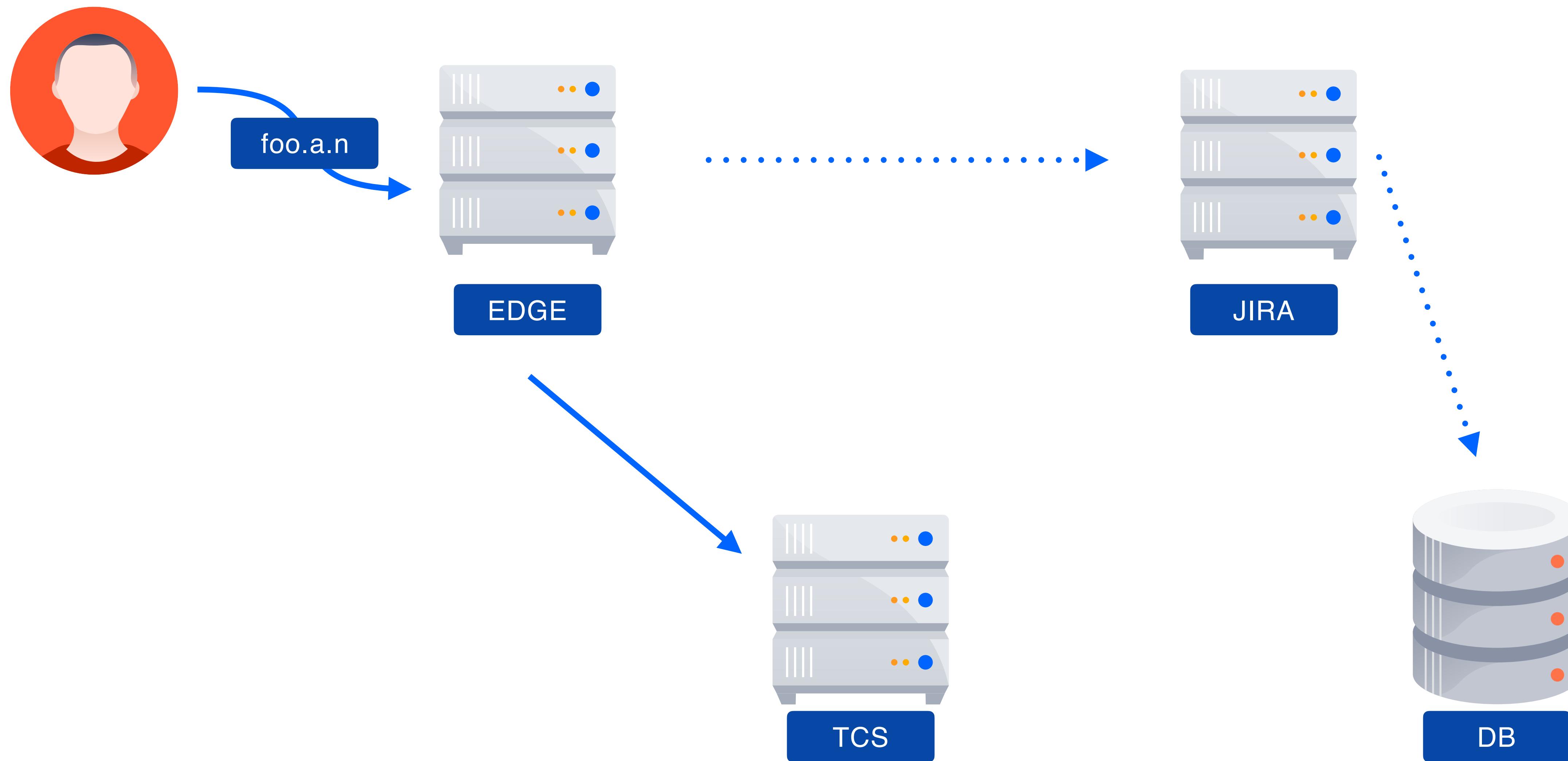
What even is TCS?

Read optimised view of entity metadata

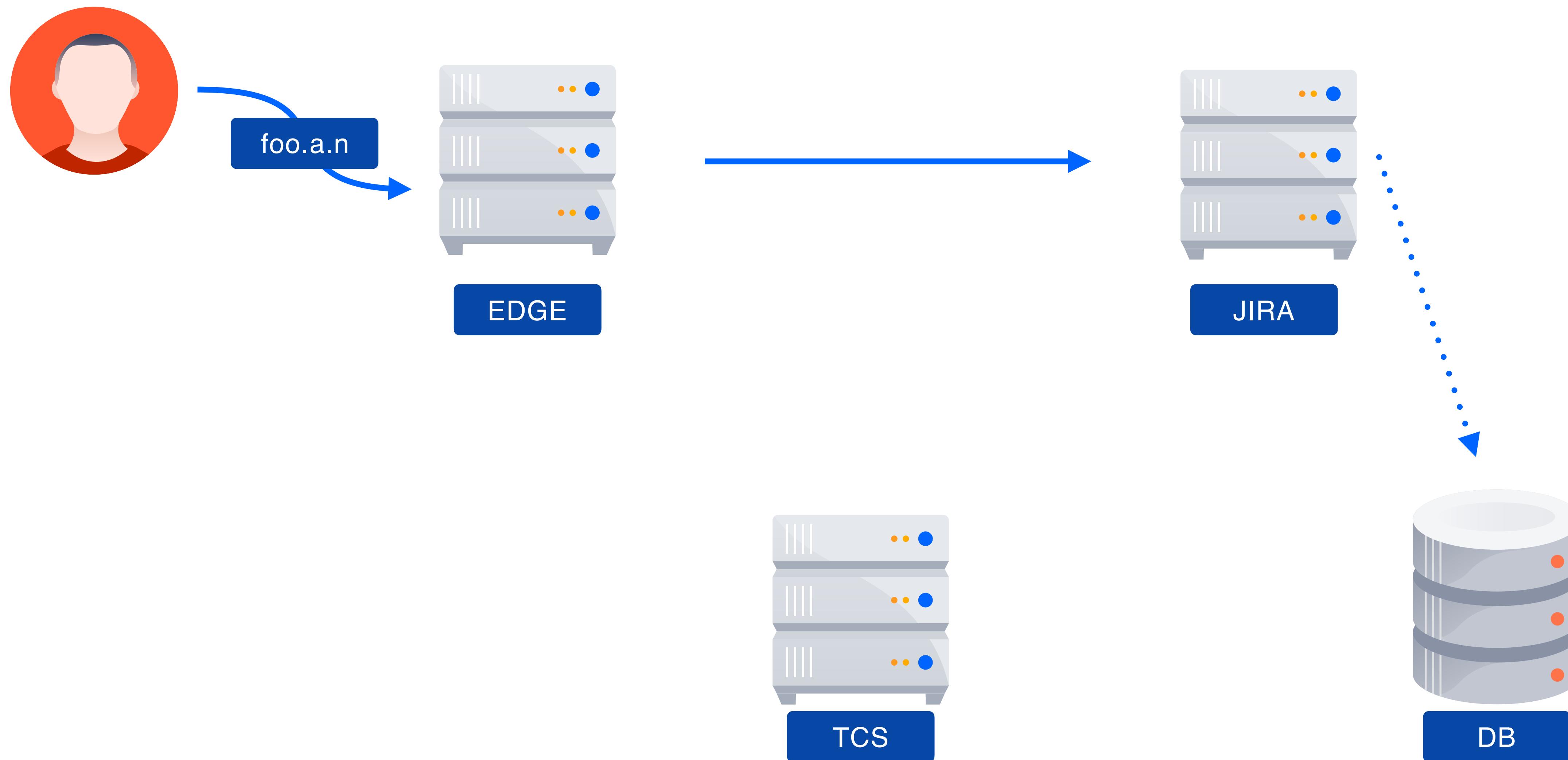
Vertigo architecture



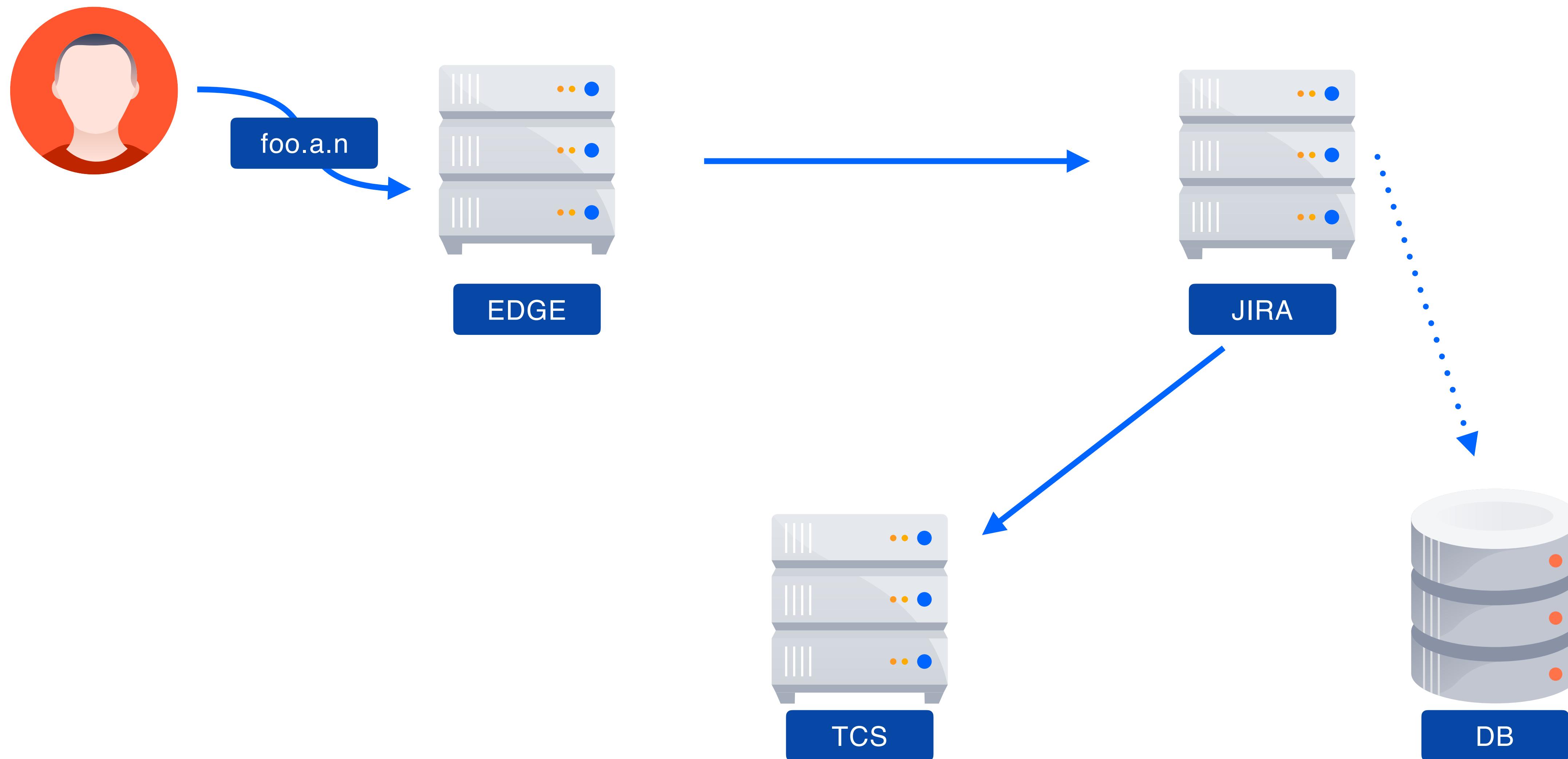
Vertigo architecture



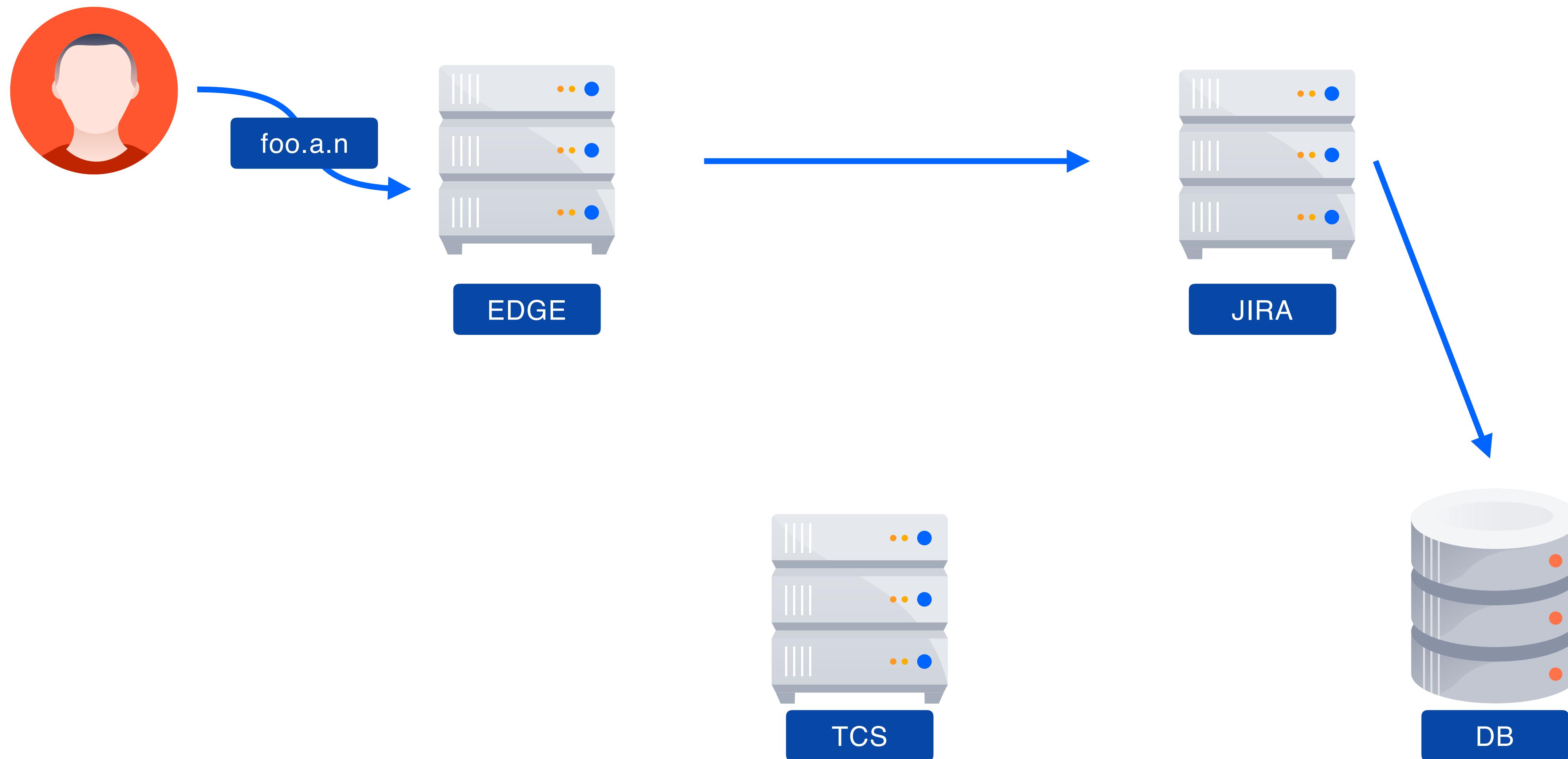
Vertigo architecture



Vertigo architecture



Vertigo architecture



Requirements



Fast

Called multiple times
per user request



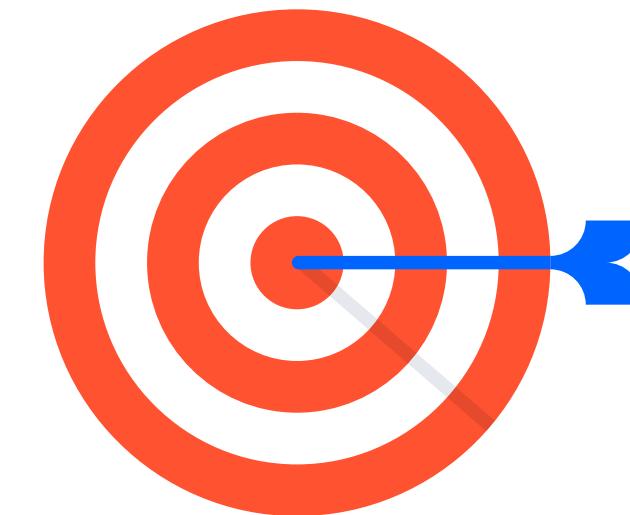
Secure

Services can only
access their info



Multi Region

Regions can fail
Cross region network
is slow



Tier 0

Can't fail, or else our
cloud is down

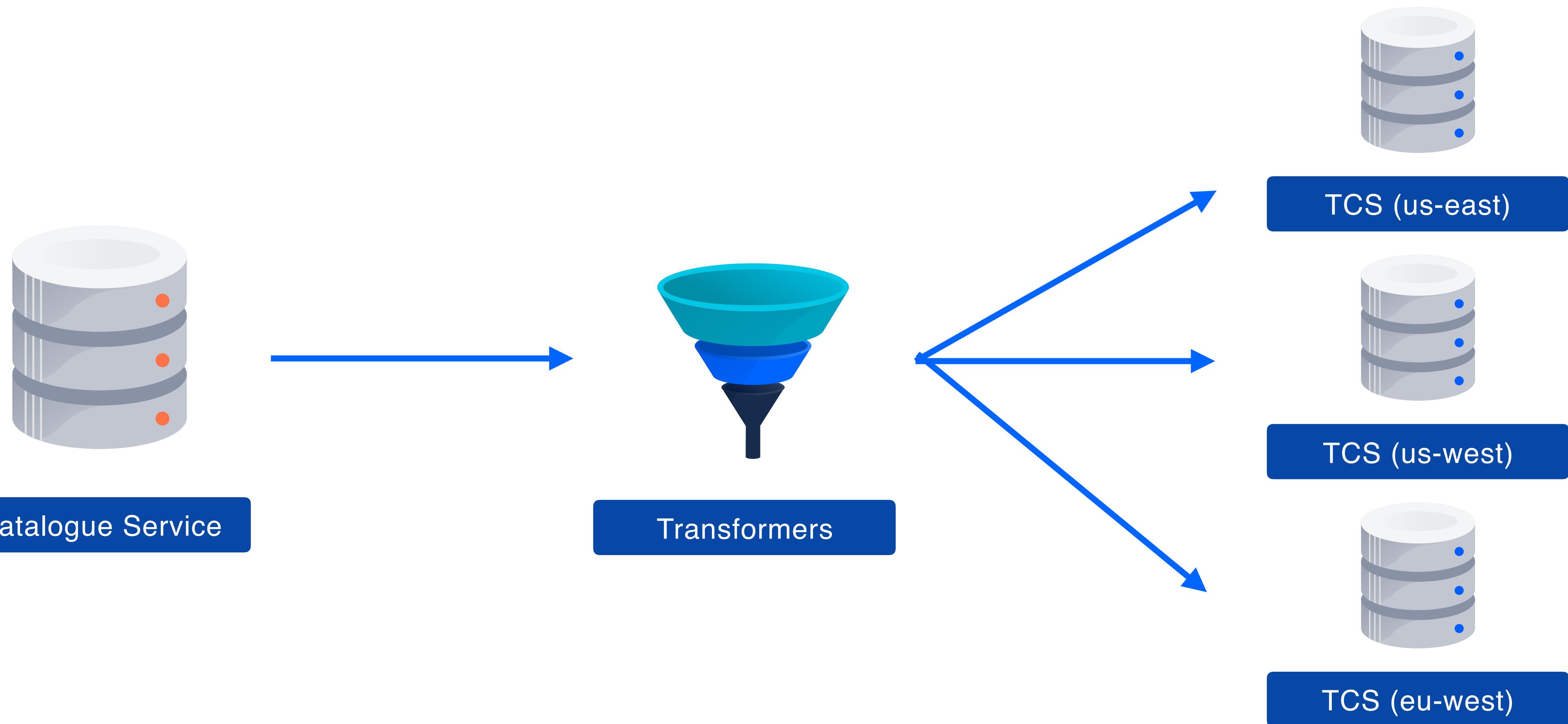
READS

Fast,
HA,
Eventually
Consistent

WRITES

Slow (ish),
One Region,
Strongly
Consistent

Decoupled Read / Write Pipeline



Requirements for TCS



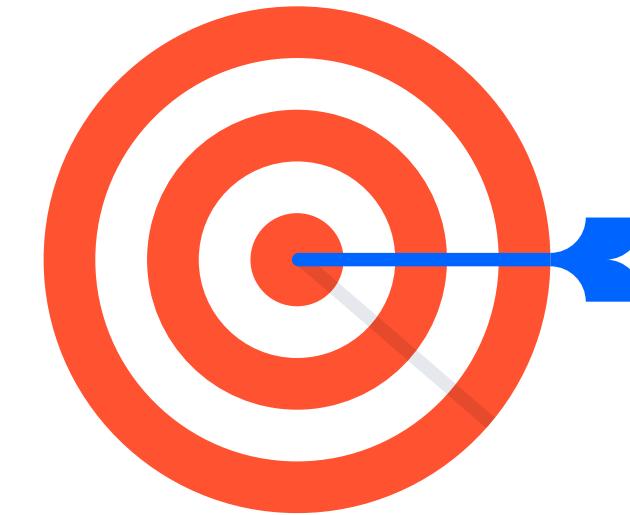
Fast *Reads*
Called multiple times
per user request



Secure
Services can only
access their output



Multi Region
Regions can fail
Cross region network
is slow



Tier 0
Can't fail, or else our
cloud is down

Catalogue Record

```
1  {
2    "CloudId": "DUMMY-158c8204-ff3b-47c2-adbb-a0906ccc722b",
3    "EntityId": "DUMMY-158c8204-ff3b-47c2-adbb-a0906ccc722b",
4    "Content": {
5      "license-data": {
6        "maintenanceEndDate": "2026-03-07",
7        "maintenanceStartDate": "2026-02-07",
8        "sen": "SEN-5283380",
9        "products": {
10          "confluence.ondemand": {
11            "pricingPlanId": "c1d6f853-c64a-489d-95b8-5b065a81983e",
12            "billingPeriod": "MONTHLY",
13            "unitCount": 5000,
14            "active": true
15          },
16          "jira-software.ondemand": {
17            "pricingPlanId": "65b27220-c20d-496d-acbf-3e644827412e",
18            "billingPeriod": "MONTHLY",
19            "unitCount": 5000,
20            "active": true
21          },
22          "jira-servicedesk.ondemand": {
23            "pricingPlanId": "fc29c8a0-0ba8-443e-aed9-55ddd30d5eb8",
```

```
20         "active": true
21     },
22     "jira-servicedesk.ondemand": {
23         "pricingPlanId": "fc29c8a0-0ba8-443e-aed9-55ddd30d5eb8",
24         "billingPeriod": "MONTHLY",
25         "unitCount": 5000,
26         "active": true
27     },
28     "com.atlassian.confluence.plugins.confluence-questions.ondemand": {
29         "pricingPlanId": "c564ae39-343d-4f36-9691-6e9d9a330fea",
30         "billingPeriod": "MONTHLY",
31         "unitCount": 5000,
32         "active": true
33     },
34     "com.radiantminds.roadmaps-jira.ondemand": {
35         "pricingPlanId": "bca4830d-bc70-4470-b128-46dee753bdb5",
36         "billingPeriod": "MONTHLY",
37         "unitCount": 5000,
38         "active": true
39     },
40     "hipchat.cloud": {
41         "billingPeriod": "MONTHLY",
42         "unitCount": 5000
43     },
44     "com.gliffy.integration.confluence.ondemand": {
45         "billingPeriod": "MONTHLY",
46         "unitCount": 5000,
47         "active": true
48     }
49   }
50 },
51 "siteadmin": {},  
52 "sf": {}
```

```
52 "confluence": {
53     "vcache.globalCacheIdentifier": "*****",
54     "applinksServerId": "*****",
55     "vcache.globalDBRollbackIdentifier": "*****",
56     "applinksType": "conf",
57     "applinksConsumerKey": "Confluence:*****",
58     "provisionedDate": "2016-07-21T03:48:51.360Z",
59     "applinksPublicKey": "*****",
60     "sen": "SEN-*****"
61 },
62 "poppy": {},
63 "user-management": {
64     "aa-on-startup": true,
65     "throngEnabled": true,
66     "firstActivationDate": 0,
67     "version": 1520230111033,
68     "products": [
69         {
70             "billingType": "MONTHLY",
71             "unitCount": 5000,
72             "productKey": "jira-servicedesk.ondemand"
73         },
74         {
75             "billingType": "MONTHLY",
76             "unitCount": 5000,
77             "productKey": "confluence.ondemand"
78         },
79         {
80             "billingType": "MONTHLY",
81             "unitCount": 5000,
82             "productKey": "jira-software.ondemand"
83         }
84     ]
85 }
```

```
71     "unitCount": 5000,
72     "productKey": "jira-servicedesk.ondemand"
73   },
74   {
75     "billingType": "MONTHLY",
76     "unitCount": 5000,
77     "productKey": "confluence.ondemand"
78   },
79   {
80     "billingType": "MONTHLY",
81     "unitCount": 5000,
82     "productKey": "jira-software.ondemand"
83   }
84 ]
85 },
86 "hipchat": {
87   "previouslyDeactivated": true,
88   "webClientURL": "https://app.stride.com"
89 },
90 "cloud-provisioner": {
91   "timezone": "Australia/Sydney",
92   "cloudName": "product-fabric",
93   "productSpecs": {
94     "confluence.ondemand": {},
95     "jira-software.ondemand": {},
96     "jira-servicedesk.ondemand": {},
97     "com.atlassian.confluence.plugins.confluence-questions.ondemand": {
98       "parentProductKeys": [
99         "confluence.ondemand"
100      ]
101    },
102    "com.radiantminds.roadmaps-jira.ondemand": {},
103    "com.gliffy.integration.confluence.ondemand": {}
104  },
105  "atlassian.jira.ondemand": {
106    "parentProductKeys": [
107      "confluence.ondemand"
108    ]
109  }
110 }
```

```
71     "unitCount": 5000,
72     "productKey": "jira-servicedesk.ondemand"
73   },
74   {
75     "billingType": "MONTHLY",
76     "unitCount": 5000,
77     "productKey": "confluence.ondemand"
78   },
79   {
80     "billingType": "MONTHLY",
81     "unitCount": 5000,
82     "productKey": "jira-software.ondemand"
83   }
84 ]
85 },
86 "hipchat": {
87   "previouslyDeactivated": true,
88   "webClientURL": "https://app.stride.com"
89 },
90 "cloud-provisioner": {
91   "timezone": "Australia/Sydney",
92   "cloudName": "product-fabric",
93   "productSpecs": {
94     "confluence.ondemand": {},
95     "jira-software.ondemand": {},
96     "jira-servicedesk.ondemand": {},
97     "com.atlassian.confluence.plugins.confluence-questions.ondemand": {
98       "parentProductKeys": [
99         "confluence.ondemand"
100      ]
101    },
102    "com.radiantminds.roadmaps-jira.ondemand": {},
103    "com.gliffy.integration.confluence.ondemand": {}
104  },
105  "atlassian.jira.ondemand": {
106    "parentProductKeys": [
107      "confluence.ondemand"
108    ]
109  }
110 }
```

```
108 "jira-shard": {
109     "bucket": "3",
110     "productHost": "jira-prod-us-7-3.prod.atl-paas.net",
111     "providerHost": "jira-prod-us-7.prod.atl-paas.net",
112     "graphqlHost": "jira-prod-us-7-graphql.prod.atl-paas.net",
113     "shard": "jira-prod-us-7"
114 },
115 "confluence-shard": {
116     "bucket": "3",
117     "productHost": "confluence-prod-us-5-3.prod.atl-paas.net",
118     "providerHost": "confluence-prod-us-5.prod.atl-paas.net",
119     "graphqlHost": "confluence-prod-us-5-graphql.prod.atl-paas.net",
120     "shard": "confluence-prod-us-5"
121 },
122 "monarch_jira": {
123     "created": 1502291393,
124     "pool": "jira-prod-us-7",
125     "handle": {
126         "resource_guid": "DUMMY-158c8204-ff3b-47c2-adbb-a0906ccc722b.monarch.jira",
127         "user_conn_limit": 50,
128         "readonly_user": {
129             "password": "*****",
130             "name": "*****"
131         },
132         "database_size": "regular",
133         "type": "postgres",
134         "database": "*****",
135         "instance_id": "*****",
136         "instance_resource_guid": "*****",
137         "port": 5432,
138         "cloudid": "DUMMY-158c8204-ff3b-47c2-adbb-a0906ccc722b",
139         "context": "jira",
140         "host": "*****.*****.us-east-1.rds.amazonaws.com",
141         "instance_region": "us-east-1"
142     }
143 }
```

```
127
128 "user_conn_limit": 50,
129 "readonly_user": {
130     "password": "*****",
131     "name": "*****"
132 },
133 "database_size": "regular",
134 "type": "postgres",
135 "database": "*****",
136 "instance_id": "*****",
137 "instance_resource_guid": "*****",
138 "port": 5432,
139 "cloudid": "DUMMY-158c8204-ff3b-47c2-adbb-a0906ccc722b",
140 "context": "jira",
141 "host": "*****.*****.us-east-1.rds.amazonaws.com",
142 "instance_region": "us-east-1",
143 "user": {
144     "password": "*****",
145     "name": "*****"
146 },
147 "status": "success"
148 },
149 "monarch_confluence": {
150     "created": 1502291393,
151     "pool": "confluence-prod-us-5",
152     "handle": {
153         "resource_guid": "DUMMY-158c8204-ff3b-47c2-adbb-a0906ccc722b.monarch.confluence",
154         "user_conn_limit": 50,
155         "readonly_user": {
156             "password": "*****",
157             "name": "*****"
158 },
159         "database_size": "large",
160         "type": "postgres",
```

SLOW!

Don't download and transform
that every time

Confluence Transformer Output

| | |
|---------------|--|
| Source | ConfluenceTransformer 🔗 |
| Key | DUMMY-158c8204-ff3b-47c2-adbb-a0906ccc722b-confluence |
| Record | <pre>{ "applinks": "{\"hostId\": \"4df4196b-09e4-3e26-9a29-bb127ae0d3c6\", \"system\": [{\"appId\": \"70d83bc8-0aff-3fa5-8121-5ae90121f5fc\", \"type\": \"jira\", \"baseUrl\": \"https://product-fabri", "cloudId": "DUMMY-158c8204-ff3b-47c2-adbb-a0906ccc722b", "db-database-name": "D6gYmD8hzz26PgdxSVkcXm", "db-host": "monarch-confluence-prod-us-5-6a3p5.ci94gmxuq4iv.us-east-1.rds.amazonaws.com", "db-port": 5432, "db-type": "postgres", "db-user-name": "XeiWZEKkC7xuQYX2WuAQQZ", "db-user-password": "*****", "firstActivationDate": 0, "isAlacarte": true, "product.com.atlassian.confluence.plugins.confluence-questions.active": "true", "product.com.atlassian.confluence.plugins.confluence-questions.billingPeriod": "MONTHLY", "product.com.atlassian.confluence.plugins.confluence-questions.pricingPlanId": "c564ae39-343d-4f36-9691-6e9d9a330fea", "product.com.atlassian.confluence.plugins.confluence-questions.unitCount": "5000", "product.com.gliffy.integration.confluence.active": "true", "product.com.gliffy.integration.confluence.billingPeriod": "MONTHLY", "product.com.gliffy.integration.confluence.unitCount": "5000", "product.com.radiantminds.roadmaps-jira.active": "true", "product.com.radiantminds.roadmaps-jira.billingPeriod": "MONTHLY", "product.com.radiantminds.roadmaps-jira.pricingPlanId": "bca4830d-bc70-4470-b128-46dee753bdb5", "product.com.radiantminds.roadmaps-jira.unitCount": "5000", "product.confluence.active": "true", "product.confluence.billingPeriod": "MONTHLY", "product.confluence.pricingPlanId": "c1d6f853-c64a-489d-95b8-5b065a81983e", "product.confluence.unitCount": "5000", "product.jira-servicedesk.active": "true", "product.jira-servicedesk.billingPeriod": "MONTHLY", "product.jira-servicedesk.pricingPlanId": "fc29c8a0-0ba8-443e-aed9-55ddd30d5eb8", "product.jira-servicedesk.unitCount": "5000", "product.jira-software.active": "true", "product.jira-software.billingPeriod": "MONTHLY", "product.jira-software.pricingPlanId": "65b27220-c20d-496d-acbf-3e644827412e", "product.jira-software.unitCount": "5000", "provisionedDate": "2016-07-21T03:48:51.360Z", "sen": "SEN-5283380", "vcache.globalCacheIdentifier": "1524479999957", "vcache.globalDBRollbackIdentifier": "1524479999958" }</pre> |

```
{  
    "applinks": "{\"hostId\": \"4df4196b-09e4-3e26-9a29-bb127ae0d3c6\", \"system\": [ {\"appId\": \"70d83bc8-0aff-3fa5-8  
    \"cloudId\": \"DUMMY-158c8204-ff3b-47c2-adbb-a0906ccc722b\",  
    \"db-database-name\": \"D6gYmD8hzz26PgdxSVkcXm\",  
    \"db-host\": \"monarch-confluence-prod-us-5-6a3p5.ci94gmxuq4iv.us-east-1.rds.amazonaws.com\",  
    \"db-port\": 5432,  
    \"db-type\": \"postgres\",  
    \"db-user-name\": \"XeiWZEKkC7xuQYX2WuAQQZ\",  
    \"db-user-password\": \"*****\",  
    \"firstActivationDate\": 0,  
    \"isAlacarte\": true,  
    \"product.com.atlassian.confluence.plugins.confluence-questions.active\": \"true\",  
    \"product.com.atlassian.confluence.plugins.confluence-questions.billingPeriod\": \"MONTHLY\",  
    \"product.com.atlassian.confluence.plugins.confluence-questions.pricingPlanId\": \"c564ae39-343d-4f36-9691-6e9d9a3  
    \"product.com.atlassian.confluence.plugins.confluence-questions.unitCount\": \"5000\",  
    \"product.com.gliffy.integration.confluence.active\": \"true\",  
    \"product.com.gliffy.integration.confluence.billingPeriod\": \"MONTHLY\",  
    \"product.com.gliffy.integration.confluence.unitCount\": \"5000\",  
    \"product.com.radiantminds.roadmaps-jira.active\": \"true\",  
    \"product.com.radiantminds.roadmaps-jira.billingPeriod\": \"MONTHLY\",  
    \"product.com.radiantminds.roadmaps-jira.pricingPlanId\": \"bca4830d-bc70-4470-b128-46dee753bdb5\",  
    \"product.com.radiantminds.roadmaps-jira.unitCount\": \"5000\",  
    \"product.confluence.active\": \"true\",  
    \"product.confluence.billingPeriod\": \"MONTHLY\",  
    \"product.confluence.pricingPlanId\": \"c1d6f853-c64a-489d-95b8-5b065a81983e\",  
    \"product.confluence.unitCount\": \"5000\",  
    \"product.jira-servicedesk.active\": \"true\",  
    \"product.jira-servicedesk.billingPeriod\": \"MONTHLY\",  
    \"product.jira-servicedesk.pricingPlanId\": \"fc29c8a0-0ba8-443e-aed9-55ddd30d5eb8\",  
    \"product.jira-servicedesk.unitCount\": \"5000\",  
    \"product.jira-software.active\": \"true\",  
    \"product.jira-software.billingPeriod\": \"MONTHLY\",  
    \"product.jira-software.pricingPlanId\": \"65b27220-c20d-496d-acbf-3e644827412e\",  
    \"product.jira-software.unitCount\": \"5000\",  
    \"provisionedDate\": \"2016-07-21T03:48:51.360Z\",  
    \"sen\": \"SEN-5283380\",  
    \"vcache.globalCacheIdentifier\": \"1524479999957\",  
    \"vcache.globalDBRollbackIdentifier\": \"1524479999958\"  
}
```

TCS in practice

Requests

US-EAST / WEST: 4k rps each

EU: 4.5k rps

Consumers

Jira, Confluence, Identity, Stride, Edge,
Permissions (and more!)

Planned Consumers

Trello, Media, Stargate, Chilli, Org Service (also
more!)

TCS in practice

Requests

US-EAST / WEST: 4k rps each
EU: 4.5k rps

Consumers

Jira, Confluence, Identity, Stride, Edge,
Permissions (and more!)

Planned Consumers

Trello, Media, Stargate, Chilli, Org Service (also
more!)

TCS in practice

Requests

US-EAST / WEST: 4k rps each
EU: 4.5k rps

Consumers

Jira, Confluence, Identity, Stride, Edge,
Permissions (and more!)

Planned Consumers

Trello, Media, Stargate, Chilli, Org Service (also
more!)

“

Easy, just write an ASAP
wrapper on Dynamo

OUR TEAM - 2016 (NAÏVE)

Dynamo isn't magic

Can fail

Sometimes AWS is just down
DNS resolution to a specific AZ can fail

Can ‘succeed’

Very slow response times randomly

Increases system load

As requests increase, increasing connections
need to be held open / made to dynamo

Dynamo isn't magic

Can fail

Sometimes AWS is just down
DNS resolution to a specific AZ can fail

Can 'succeed'

Very slow response times randomly

Increases system load

As requests increase, increasing connections
need to be held open / made to dynamo

Dynamo isn't magic

Can fail

Sometimes AWS is just down
DNS resolution to a specific AZ can fail

Can 'succeed'

Very slow response times randomly

Increases system load

As requests increase, increasing connections
need to be held open / made to dynamo

“

So just use DAX?

EVERYONE POST MAY 2018

DAX isn't perfect

Still uses network

You still have to request DAX (it's just very fast)
All network / AWS issues can still apply

Expensive

Would double the cost of running the TCS in each region

Fairly recent

Full Micros support for DAX was only announced May 2018

DAX isn't perfect

Still uses network

You still have to request DAX (it's just very fast)
All network / AWS issues can still apply

Expensive

Would double the cost of running the TCS in each region

Fairly recent

Full Micros support for DAX was only announced May 2018

DAX isn't perfect

Still uses network

You still have to request DAX (it's just very fast)
All network / AWS issues can still apply

Expensive

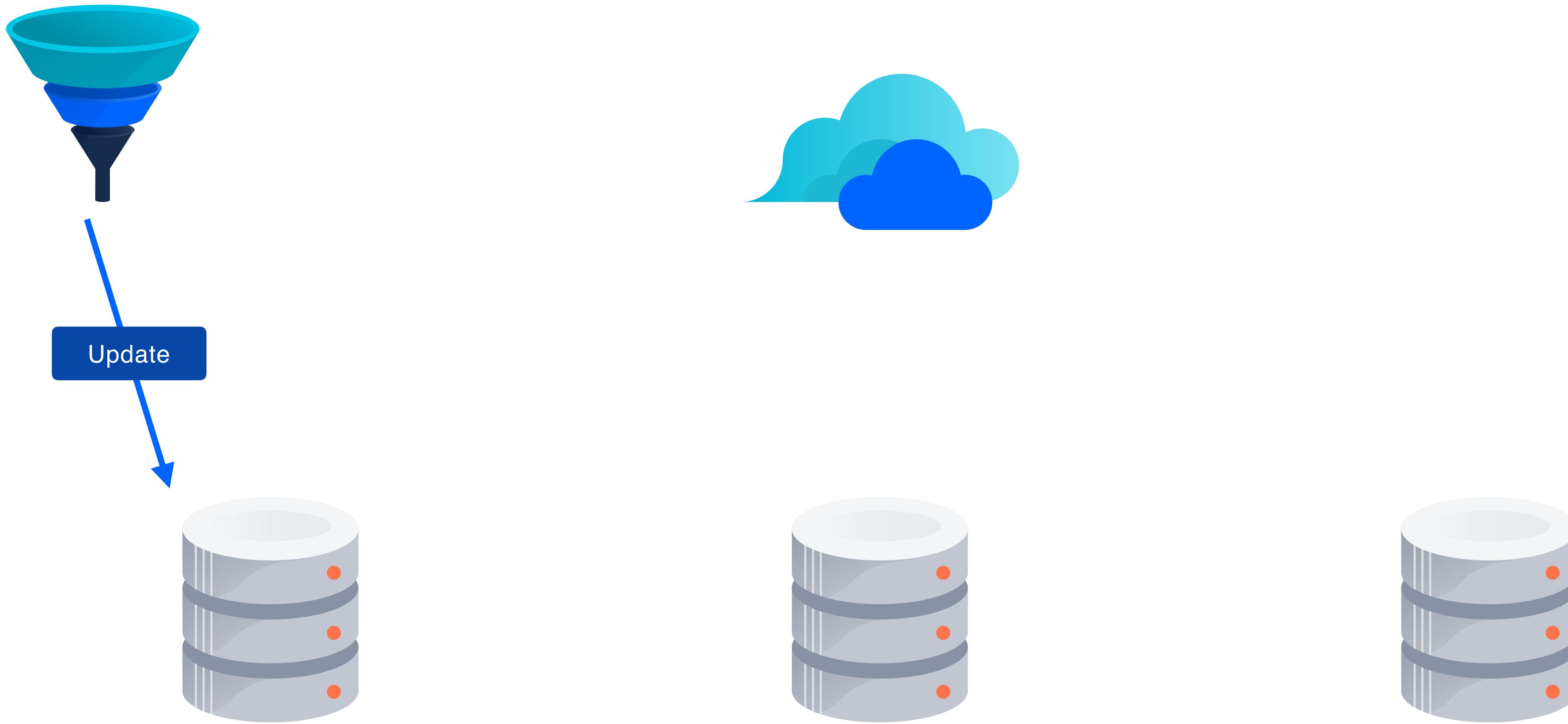
Would double the cost of running the TCS in each region

Fairly recent

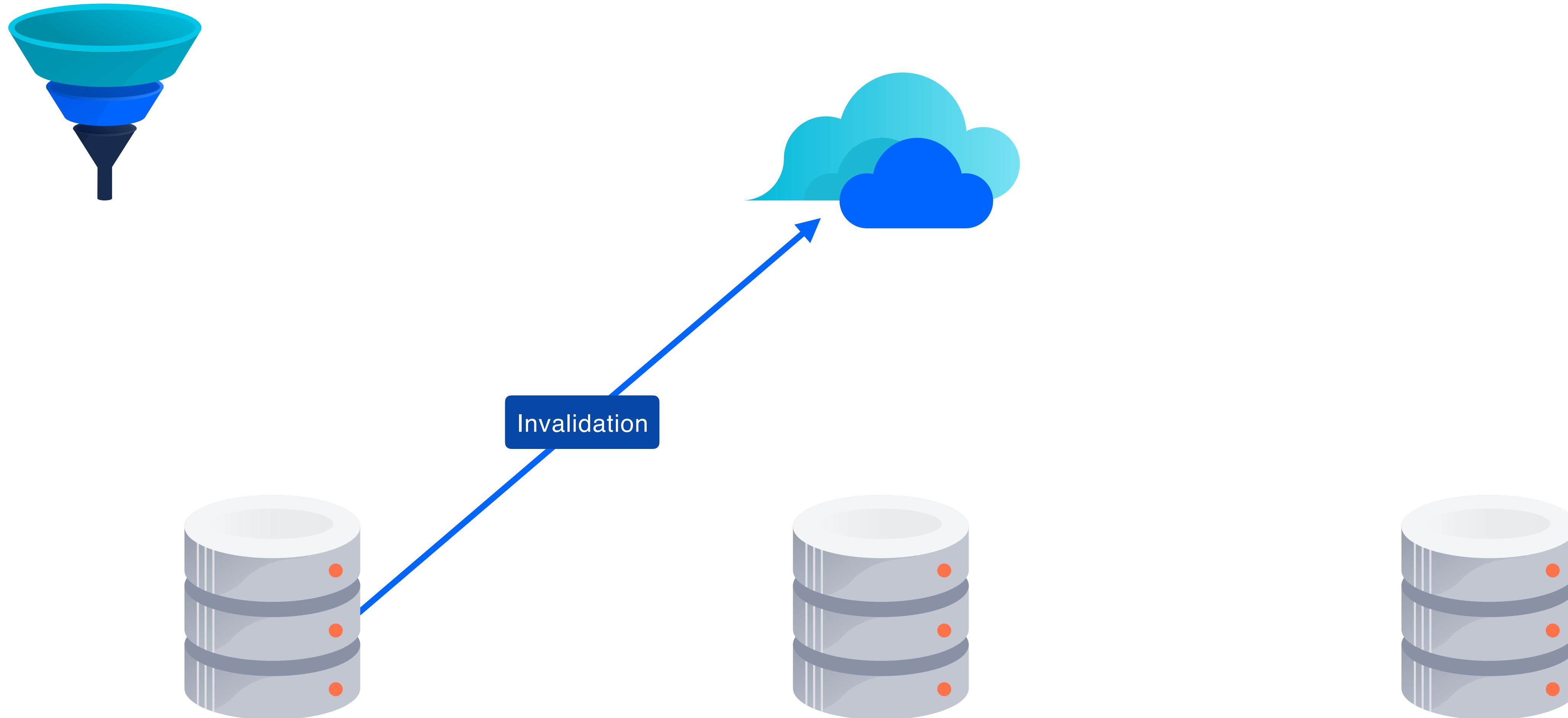
Full Micros support for DAX was only announced May 2018

If network calls are the issue,
can we optimise such that the
hot path doesn't hit network?

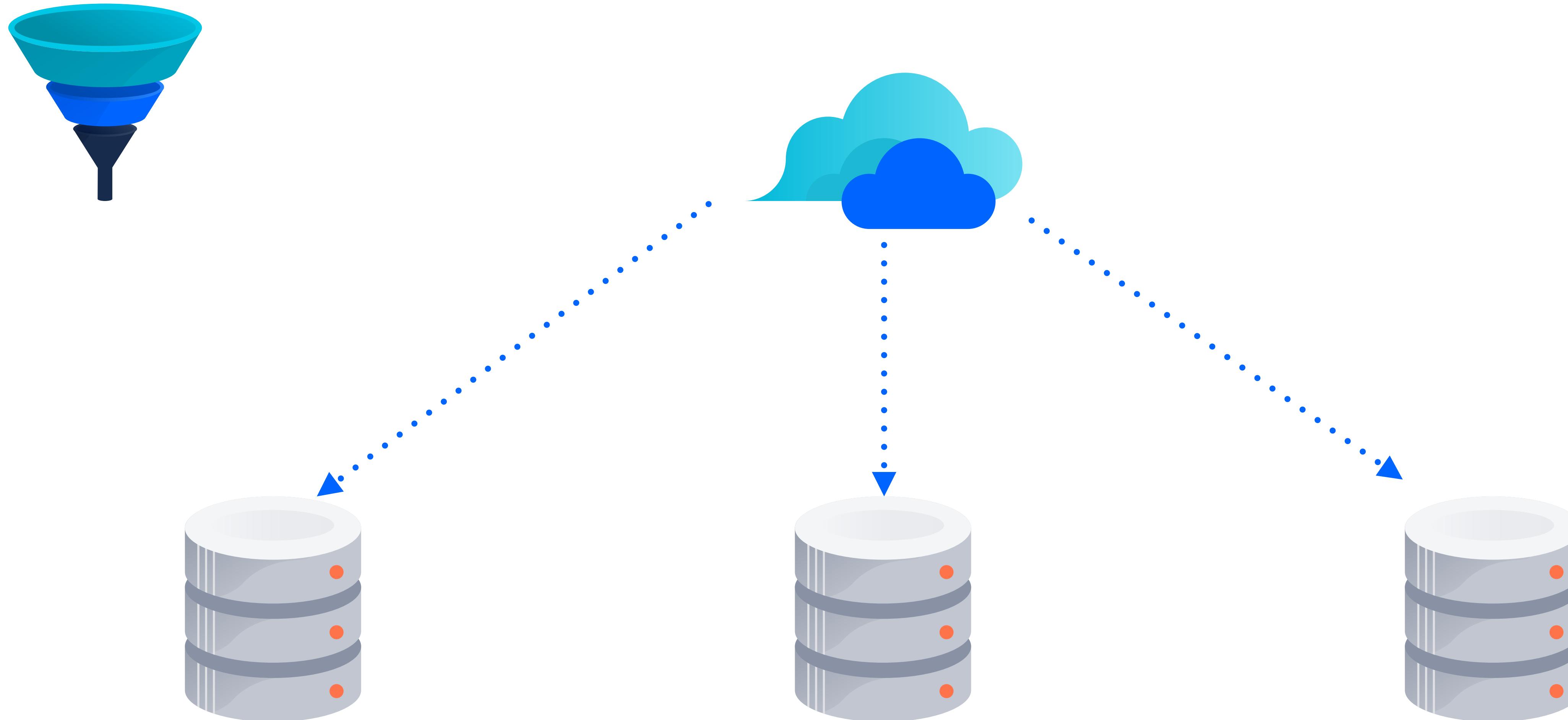
Server Node Caches (SNS based)



Server Node Caches (SNS based)



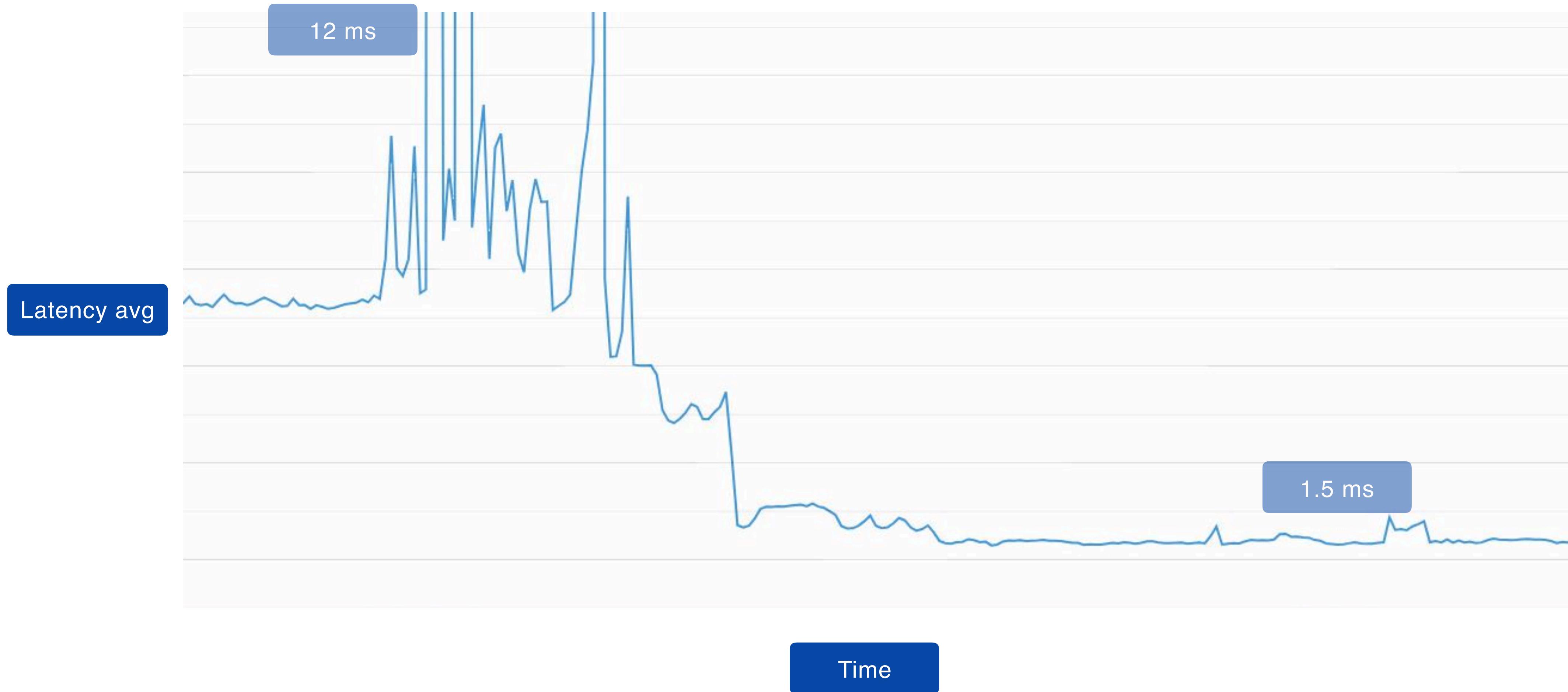
Server Node Caches (SNS based)



Dynamo Read Units



ELB Latency



Conversations you never want

-  Joshua Batchelor Feb 21, 9:32 AM
We're seeing failures in both regions
-  Frederik Happel Feb 21, 9:33 AM
@JsonMoes
-  Jackson Moes Feb 21, 9:33 AM
Yeah
-  Frederik Happel Feb 21, 9:33 AM
@here war room on level 2
-  Jackson Moes Feb 21, 9:33 AM
Sorry couldn't access stride
Cause TCS was down

- Where are y'all at?
-  Frederik Happel Feb 21, 9:33 AM
not there
organising
-  Rob Kelley Feb 21, 9:35 AM
@JsonMoes this seems roughly 30 minutes after a deployment again...

Graphs you never want





To Support

We pay for AWS high priority support.
When you're unsure in an incident
about if it's AWS, use it!

“

The time required for Elastic Load Balancing to scale can range from 1 to 7 minutes

ELBs aren't magic

Doesn't handle spikes

Time to scale up means that region failover case
is not properly supported

Are not prewarmed

A new deployment will give you a nice small ELB
which will almost immediately need to scale

Can just die

Instead of graceful degradation and 503ing some
requests, actual ELB nodes can just fail

ELBs aren't magic

Doesn't handle spikes

Time to scale up means that region failover case
is not properly supported

Are not prewarmed

A new deployment will give you a nice small ELB
which will almost immediately need to scale

Can just die

Instead of graceful degradation and 503ing some
requests, actual ELB nodes can just fail

ELBs aren't magic

Doesn't handle spikes

Time to scale up means that region failover case
is not properly supported

Are not prewarmed

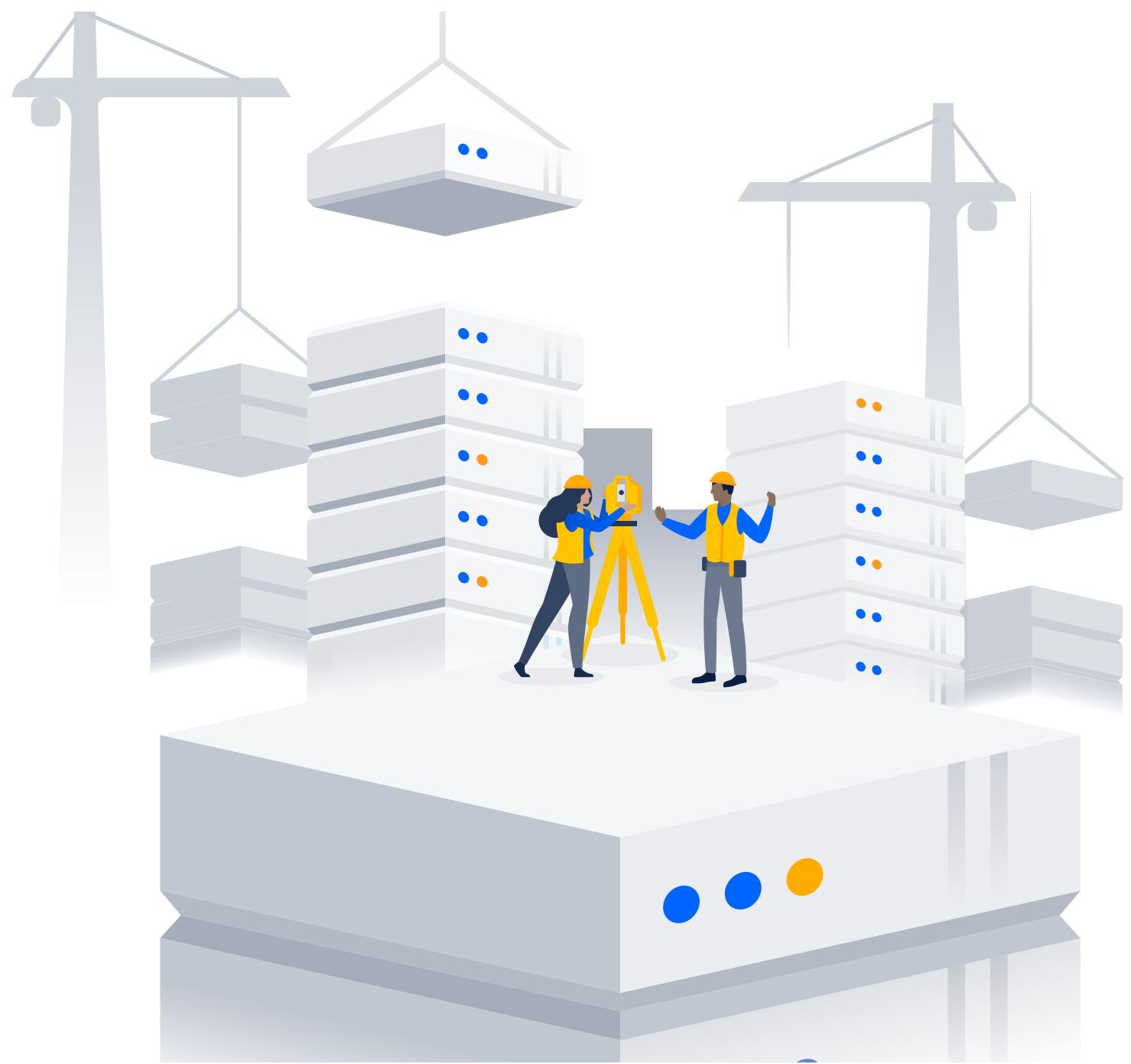
A new deployment will give you a nice small ELB
which will almost immediately need to scale

Can just die

Instead of graceful degradation and 503ing some
requests, actual ELB nodes can just fail

To ALBs

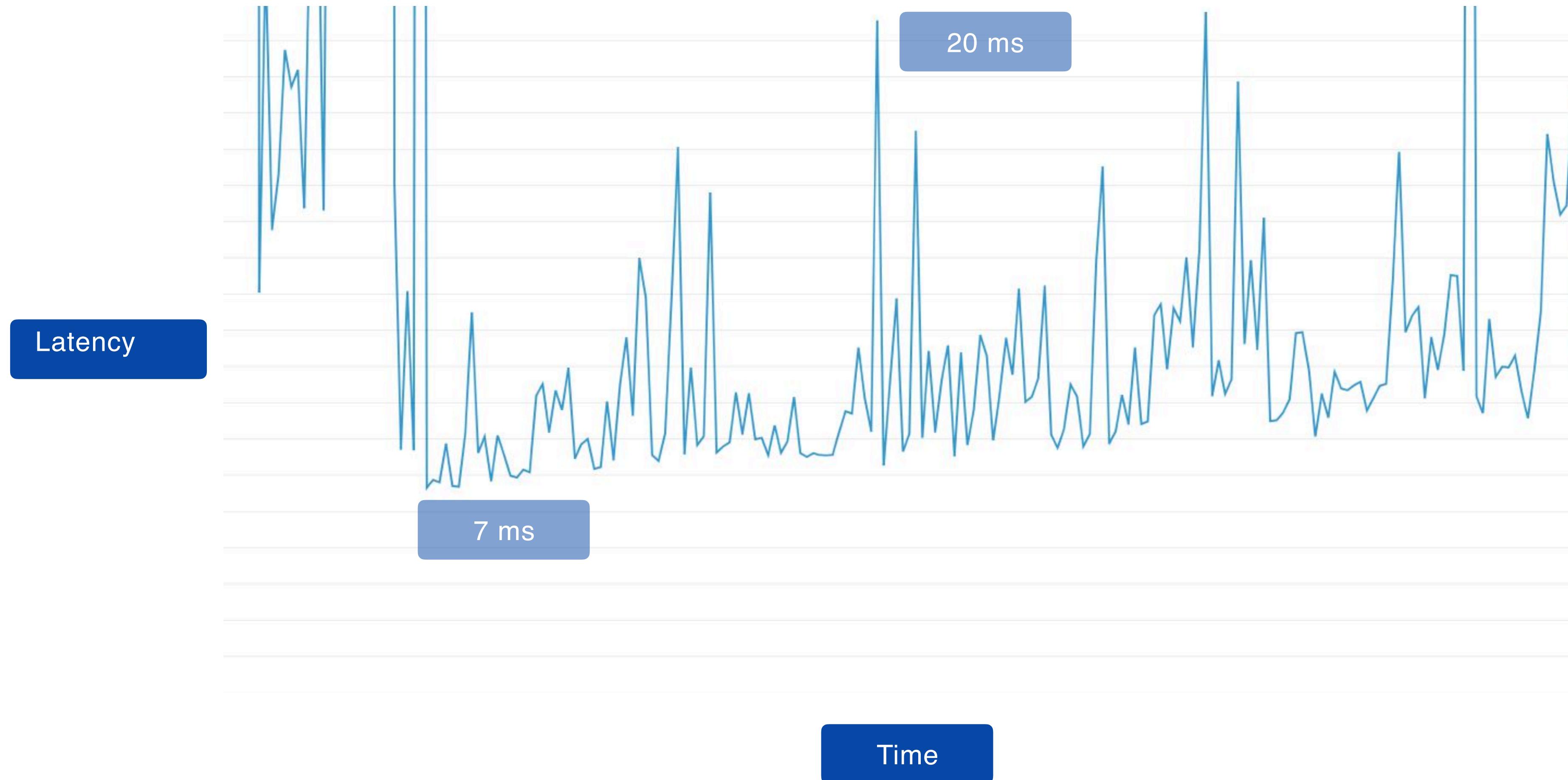
```
55      55    loadBalancer:  
56 +    type: ALB  
57 +    single: true  
56      58    connectionTimeout: 300  
57      59    scaling:  
58      60    metrics:  
59      61    CPUUtilization:  
60      62    Threshold:  
61      63    Lower: 20  
62      64    # yet another rule of thumb  
63      65    # 1 core for background tasks  
64      66    # 2 cores for main application
```



... With Capacity

AWS beta allows preprovisioning ALB units so it won't scale down below the given amount

Graphs you never want v2



ELB

Least
outstanding
requests
routing

ALB

Round robin
routing

LOAD BALANCING

ALBs make slow nodes worse
and direct more traffic there
than ELBs

But why are the nodes slow?

Isn't the ALB

We just fixed that problem and have additional capacity

Probably isn't Dynamo

We have a ~90%-95% cache hit ratio on each node, so we barely hit dynamo

The Caches I guess?

Pretty much the only other part of the request flow that isn't SpringBoot

But why are the nodes slow?

Isn't the ALB

We just fixed that problem and have additional capacity

Probably isn't Dynamo

We have a ~90%-95% cache hit ratio on each node, so we barely hit dynamo

The Caches I guess?

Pretty much the only other part of the request flow that isn't SpringBoot

But why are the nodes slow?

Isn't the ALB

We just fixed that problem and have additional capacity

Probably isn't Dynamo

We have a ~90%-95% cache hit ratio on each node, so we barely hit dynamo

The Caches I guess?

Pretty much the only other part of the request flow that isn't SpringBoot

Caches aren't magic

Request amortisation
Requests to the cache will often incur maintenance penalties

Go cold

After the TTL a request will have to do a cold lookup again, which adds a lot of time

Have no throttling

If we get multiple requests at a time for a key, we hit the backing store multiple times

Caches aren't magic

Request amortisation
Requests to the cache will often incur
maintenance penalties

Go cold

After the TTL a request will have to do a cold
lookup again, which adds a lot of time

Have no throttling

If we get multiple requests at a time for a key, we
hit the backing store multiple times

Caches aren't magic

Request amortisation
Requests to the cache will often incur maintenance penalties

Go cold

After the TTL a request will have to do a cold lookup again, which adds a lot of time

Have no throttling

If we get multiple requests at a time for a key, we hit the backing store multiple times

How About a Better Cache



David Connard

HOT-83092_caffeinate_our_item_cache → master

MERGED

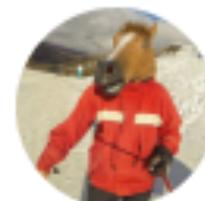
Use Caffeine cache instead of Guava cache

[Overview](#)

[Diff](#)

[Commits](#)

Details



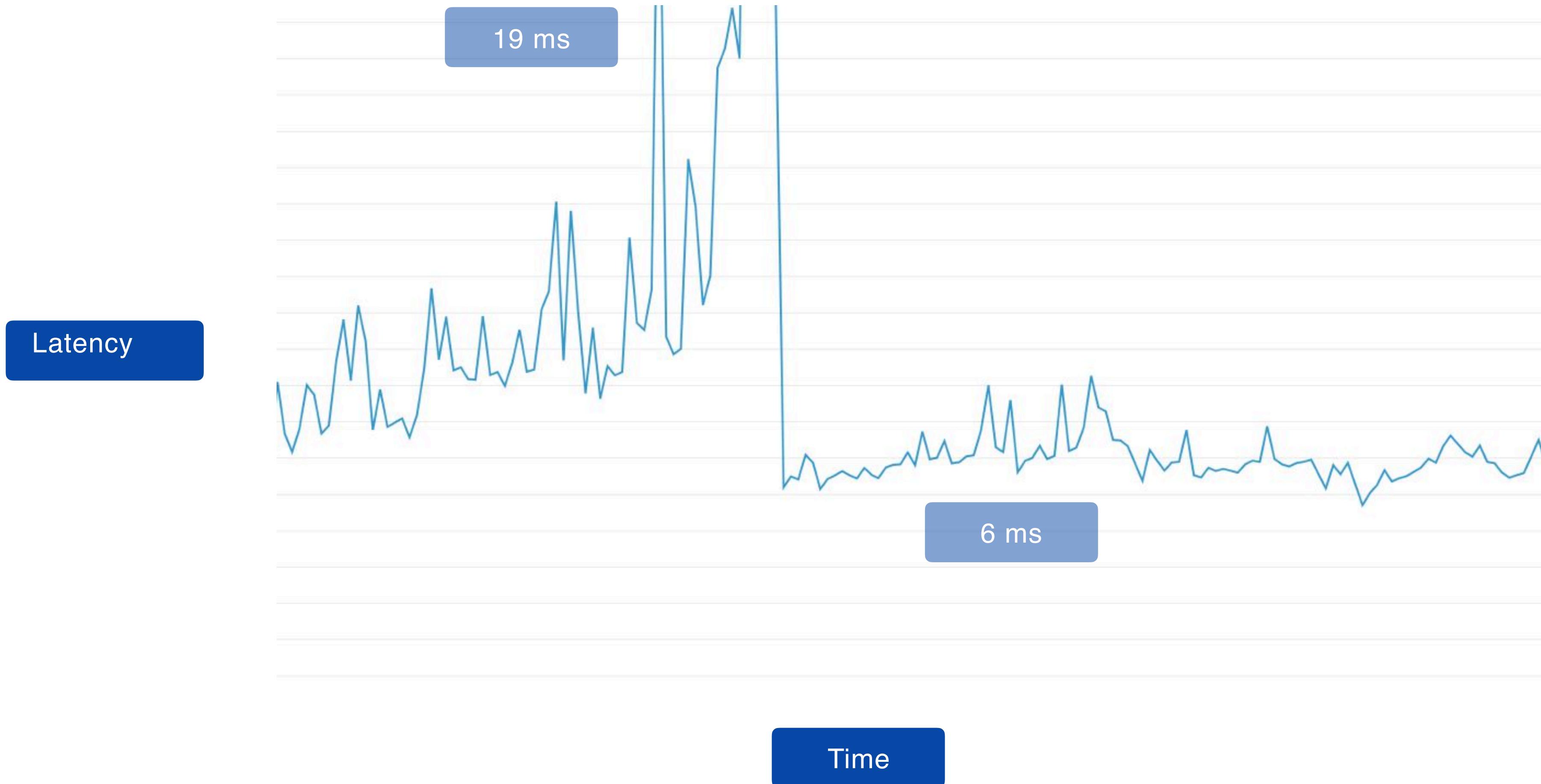
David Connard created a pull request 26 Feb 2018

An embarrassingly small change here appears to improve the performance of our Item cache.

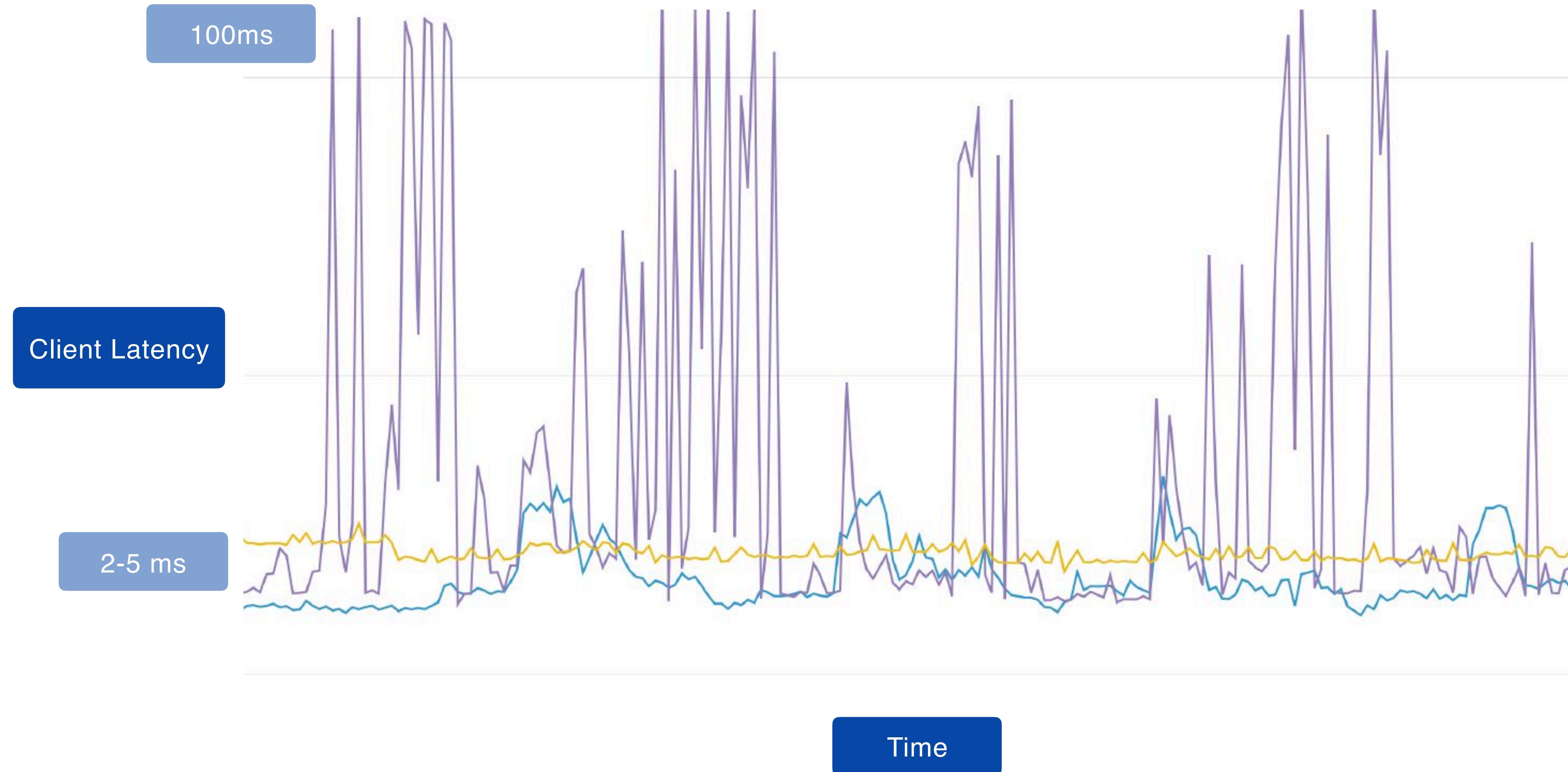
How About a Better Cache

```
-      cache = CacheBuilder.newBuilder()
+      cache = Caffeine.newBuilder()
        .maximumWeight(cacheSizeMB * 1024 * 1024)
        .weigher(contentLengthWeigher)
        .expireAfterWrite(cacheTimeSecsPlusJitter, TimeUnit.SECONDS)
-
-      .concurrencyLevel(concurrencyLevel)
-      .build();
```

How About a Better Cache



Graphs you never want v3



Graphs you never want v3

TCS response p99 



Clients aren't magic

Short cache times

Can't have a long TTL or else risk caching incorrect information

Suboptimal fallback

Typically have a static primary / secondary with requests going sequentially to both (or nothing!)

Are out of sync

It's hard to do synchronised improvements that span client and TCS code

Clients aren't magic

Short cache times

Can't have a long TTL or else risk caching
incorrect information

Suboptimal fallback

Typically have a static primary / secondary with
requests going sequentially to both (or nothing!)

Are out of sync

It's hard to do synchronised improvements that
span client and TCS code

Clients aren't magic

Short cache times

Can't have a long TTL or else risk caching incorrect information

Suboptimal fallback

Typically have a static primary / secondary with requests going sequentially to both (or nothing!)

Are out of sync

It's hard to do synchronised improvements that span client and TCS code

“

I'll make my own client, with
redundancy and long lived
caches

BENDER - PROBABLY

Micros supported Sidecars

```
41  41    compose:
42  42      jira:
43  43        image: docker.atl-paas.net/sox/jira
44  44        tag: latest
45  45        ports:
46  46          - 8090:8090
47  47          - 9999:9999 # JMX port
Q   48 +
      depends_on:
      - tcs-sidecar
      links:
      - tcs-sidecar
48  52    reverse_proxy:
49  53      # Source code of this jira-nginx-sidecar is here: https://stash.atlassian.com/projects/JSCRE/repos/jira-nginx
50  54      # When you update tag here, most probably you will need to update tag in docker-compose.yml file.
51  55      image: docker.atl-paas.net/sox/jira-nginx-sidecar
52  56      tag: 2018-05-01_8a09293
53  57      ports:
54  58        - 8080:8080
55  59        - 10128:10128 #nginx status consumed by datadog
56  60      links:
57  61        - jira
58  62    s3_blackhole:
59  63      image: docker.atl-paas.net/sox/s3-blackhole-sidecar
60  64      tag: 0.7.3_b6845d6
61  65      command: ['--enable-compression', '--enable-datadog', '--cleanup', '--path', '/opt/micros/serviceVolume/s3-k
66 +
      tcs-sidecar:
67 +
        image: docker.atl-paas.net/sox/tcs-sidecar-application
68 +
        tag: 1.0-stable-release
```

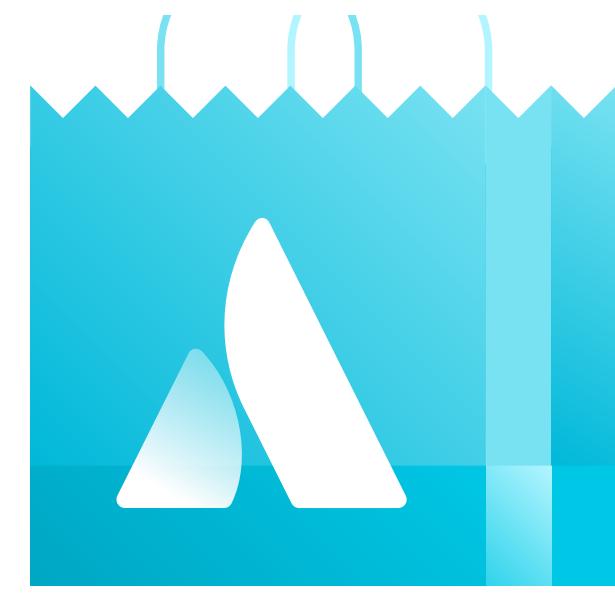
TCS sidecar



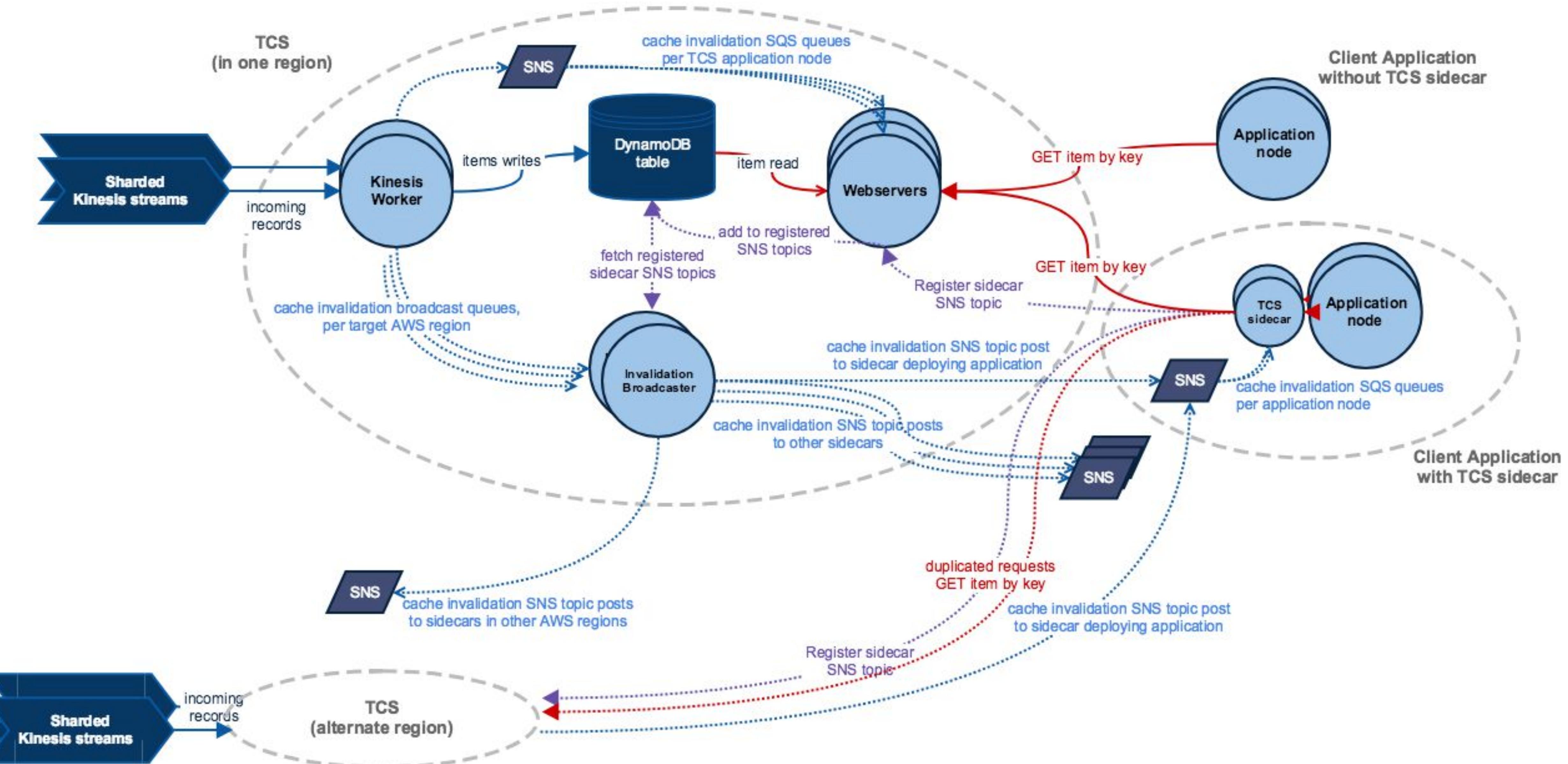
Long caches
Hook into parent TCS
invalidations



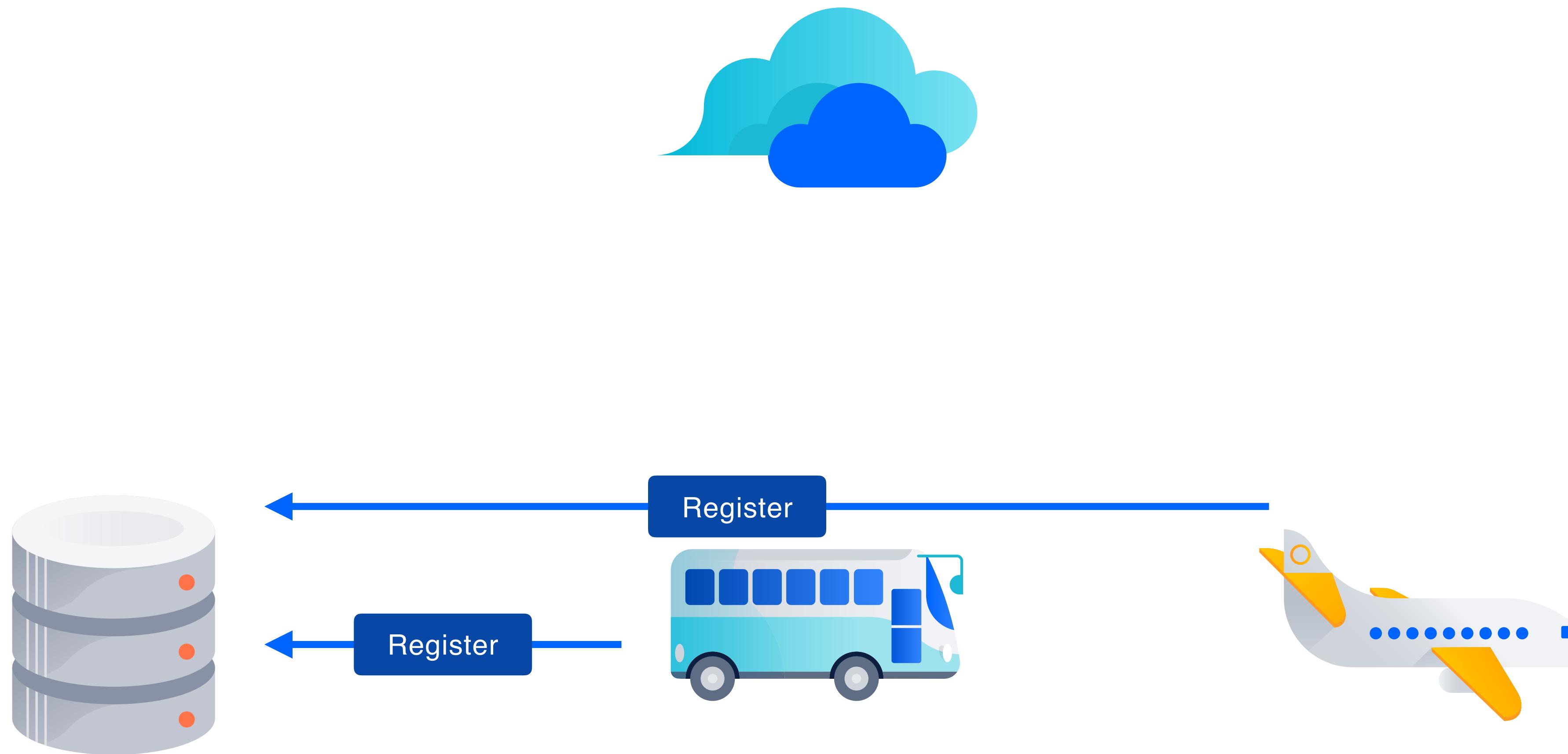
Multi requests
Send requests to multiple
TCSs



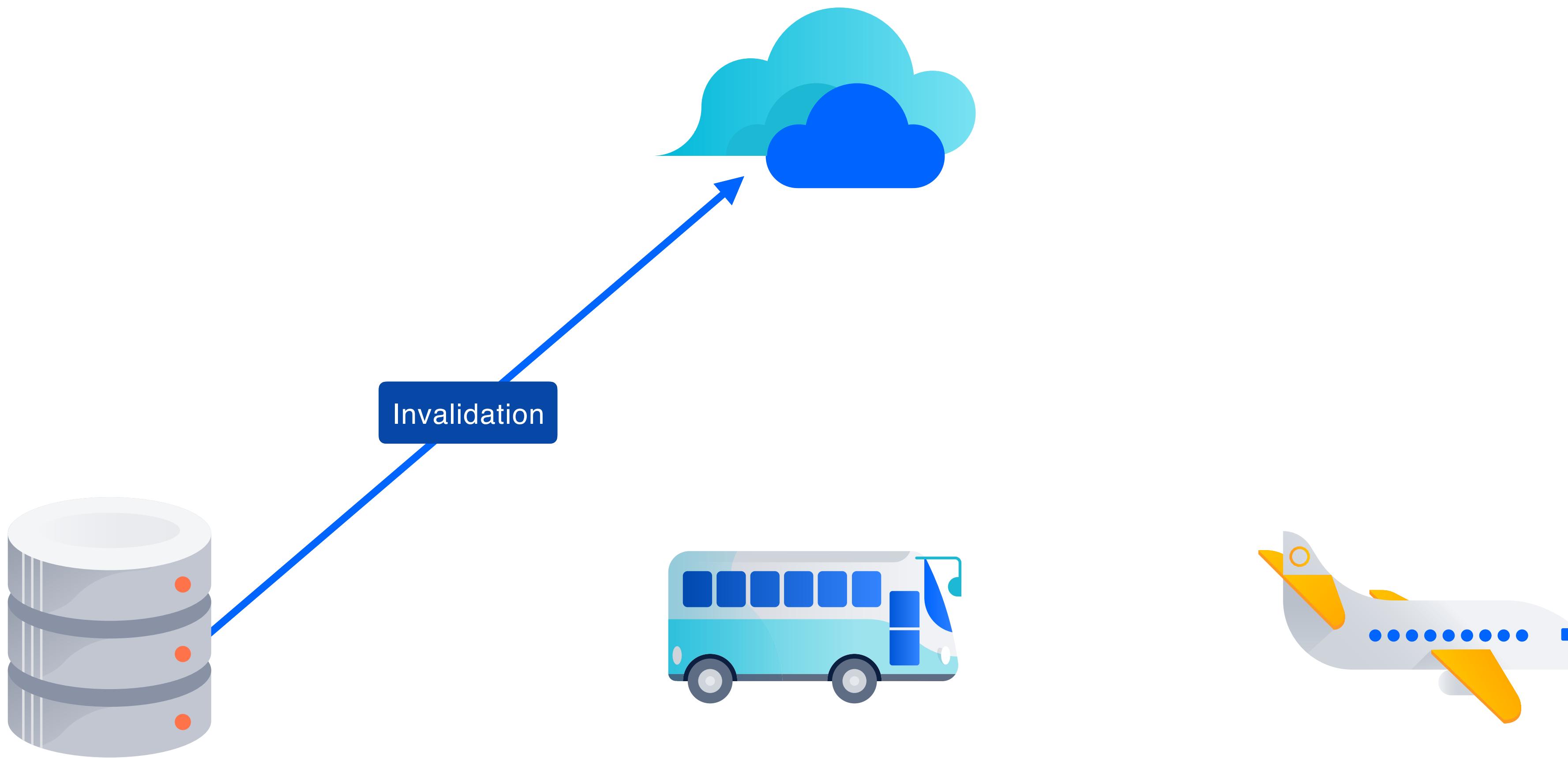
Easy Adoption
Exact same API, just on
localhost



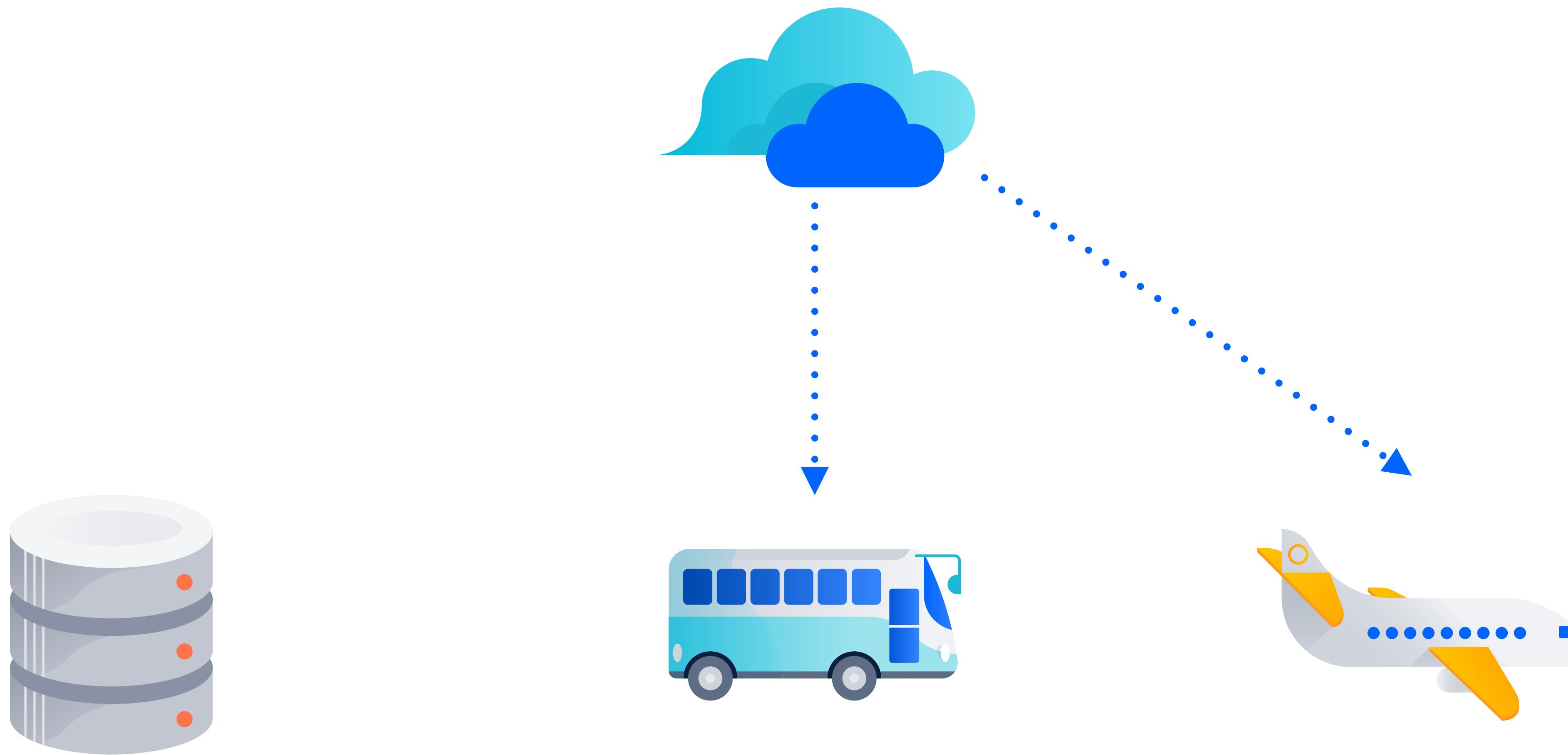
Sidecar caches (SNS based)



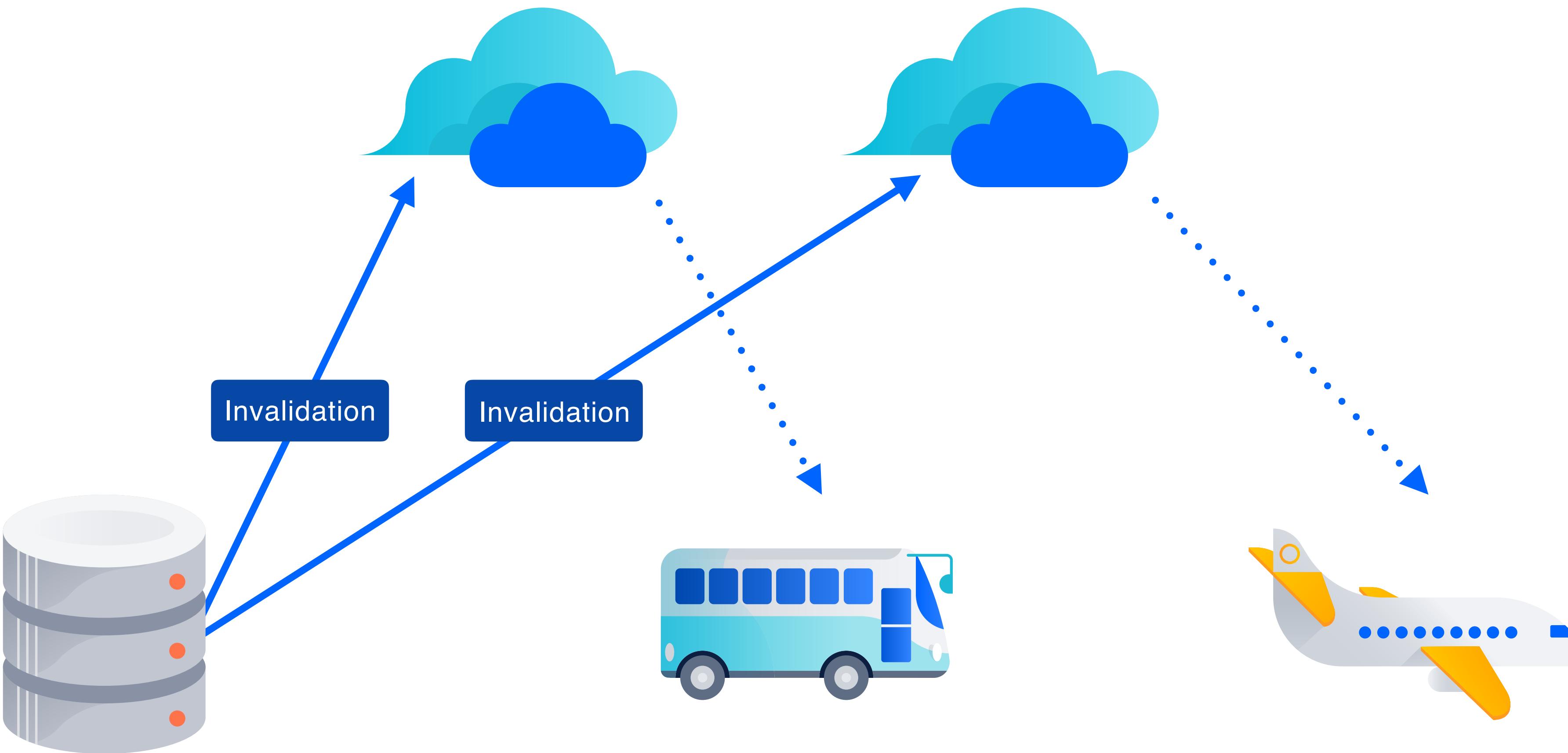
Sidecar caches (SNS based)



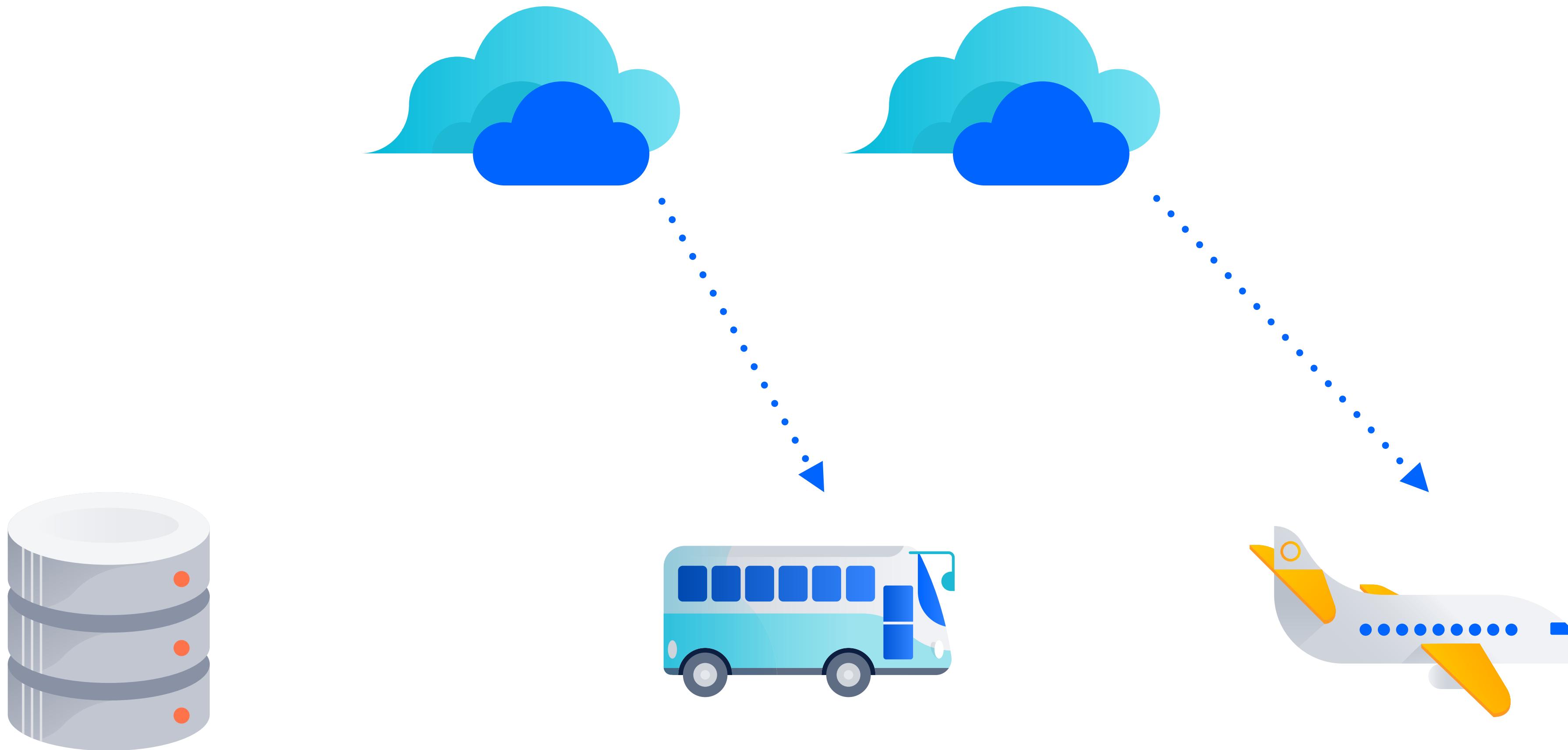
Sidecar caches (SNS based)



Sidecar caches - Micros Limitations



Sidecar caches - Micros Limitations



Working Around Limitations

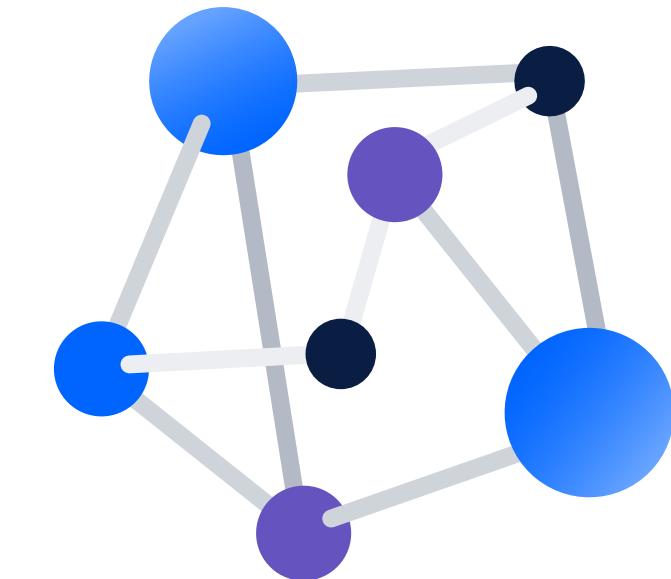
```
staging:
  resources:
    # allow staging TCSs to publish to our SNS topic for tcs-sidecar-cache-validation
    - type: sns-notification
      name: tcs-sidecar-cache-validation
      lifecycle: eternal
    attributes:
      Policy:
        Statement:
          - Sid: Allow-Writer-Writes
            Effect: Allow
            Action: "sns:Publish"
            Principal:
              AWS:
                - "arn:aws:iam::752631980301:role/tenant-context-service" # stg-us
                - "arn:aws:iam::831214378661:role/tenant-context-service" # stg-eu
```

Sidecar multirequests



Short circuit

Return the first TCS that responds



Dynamic primary

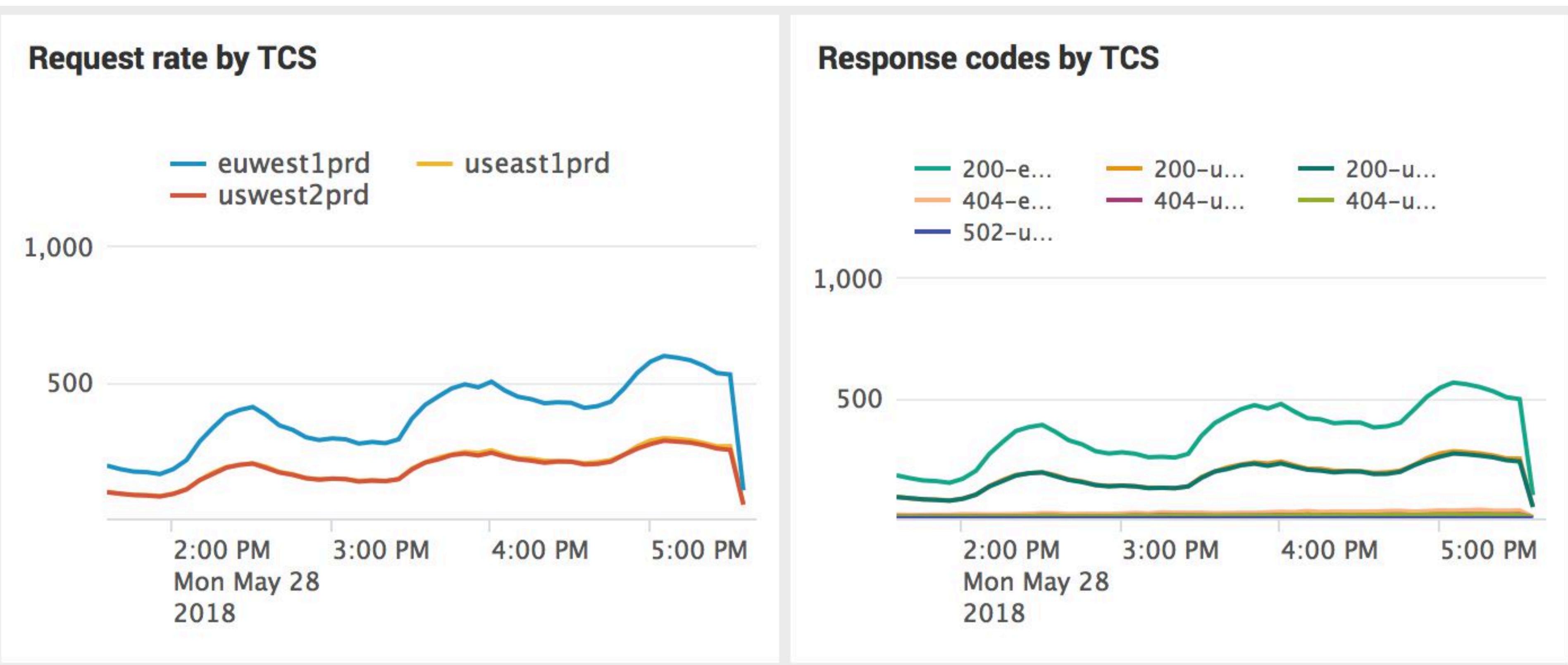
Always request the fastest and most reliable TCS



Stale Key Detection

Can determine when a key is updated in another TCS

Sidecar multirequests



Sidecar Compatibility

```
TCS_URL: 'https://tenant-context-service.us-east-1.staging.atl-paas.net'  
TCS_URL: 'http://tcs-sidecar:24143'
```



Before

Every client has different config with no way of knowing what the TCS is doing



After

Unified way of accessing TCS with best of class implementation

Sidecars require work

YBITRI

You build it... they run it. Be a good citizen and make sure you have good metrics / logs

Resources

Clients will have different amounts of headroom.
Be flexible and have a light footprint

Security

You have access to *all* the main services resources, including ASAP keys.

Sidecars require work

YBITRI

You build it... they run it. Be a good citizen and make sure you have good metrics

Resources

Clients will have different amounts of headroom.
Be flexible and have a light footprint

Security

You have access to *all* the main services resources, including ASAP keys.

Sidecars require work

YBITRI

You build it... they run it. Be a good citizen and
make sure you have good metrics

Resources

Clients will have different amounts of headroom.
Be flexible and have a light footprint

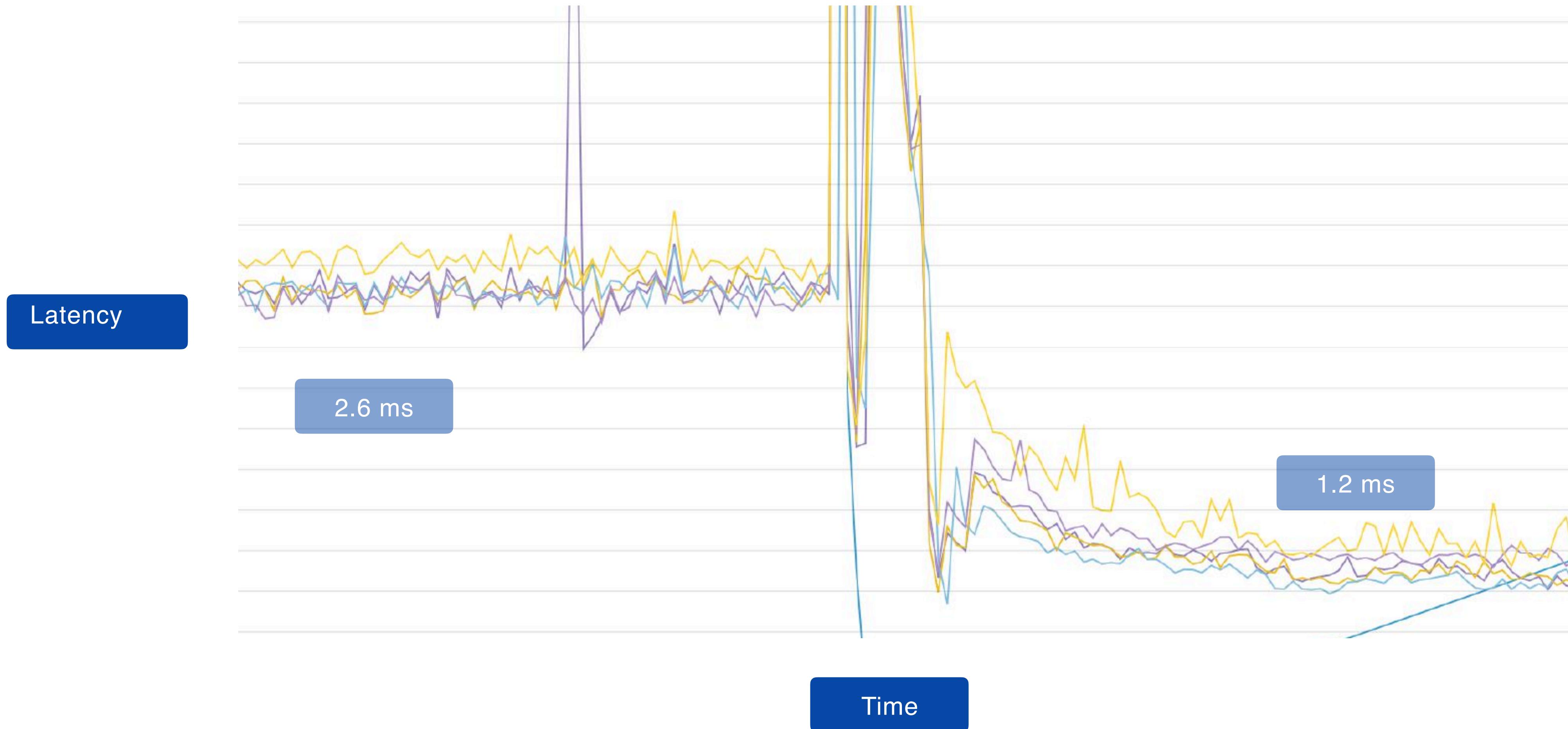
Security

You have access to *all* the main services
resources, including ASAP keys.

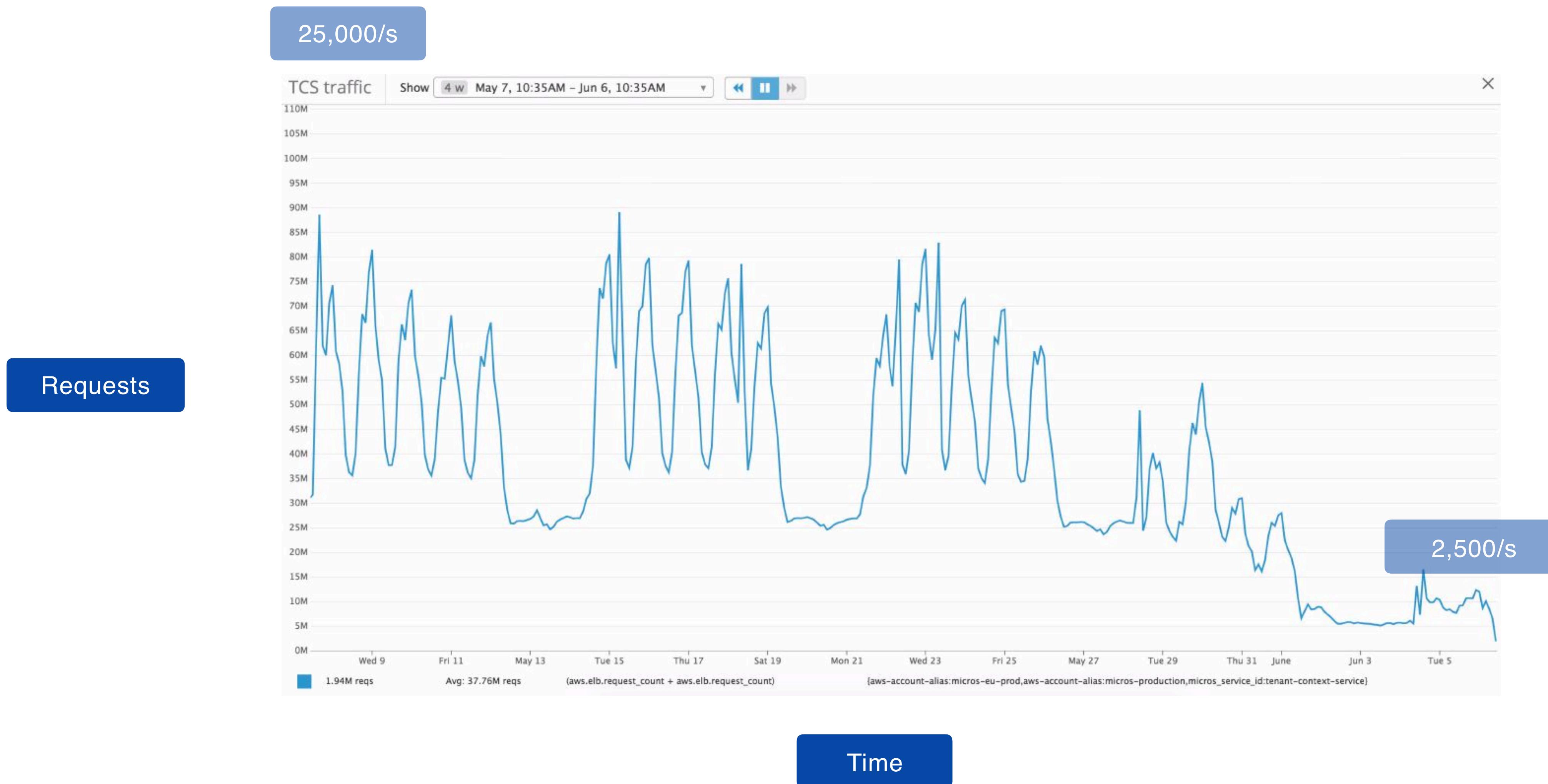
Edge Client Latency



Confluence Client Latency



TCS Request Count (actual service)





Adoption Status

Running fully in production for:
Jira, Confluence, Edge, Demux
(your service?)



Blog

[https://extranet.atlassian.com/pages/
viewpage.action?pageId=3963656059](https://extranet.atlassian.com/pages/viewpage.action?pageId=3963656059)

Where is TCS now?

Cost

\$1000/month/region

Comparison: USS (tier 0) \$5k, Jira (tier 1) \$300k

Edge Latency

Avg: 0.1ms

p75: 0.00ms (we ran out of precision)

p99: 8.0ms

Resources

400 / 800 RW dynamo

6 Webserver nodes

Where is TCS now?

Cost

\$1000/month/region

Comparison: USS (tier 0) \$5k, Jira (tier 1) \$300k

Edge Latency

Avg: 0.1ms

p75: 0.00ms (we ran out of precision)

p99: 8.0ms

Resources

400 / 800 RW dynamo

6 Webserver nodes

Where is TCS now?

Cost

\$1000/month/region

Comparison: USS (tier 0) \$5k, Jira (tier 1) \$300k

Edge Latency

Avg: 0.1ms

p75: 0.00ms (we ran out of precision)

p99: 8.0ms

Resources

400 / 800 RW dynamo

6 Webserver nodes

Summary

Dynamo

[AE]LB

Caches

Clients

Persistence is not magic

1 Make sure it scales

If your service needs to handle X rps / connections, make sure the backing store can actually deal with that load (or what it costs!) or cache around it

2 Make sure it meets SLAs

Do you need to return in X ms? Try abstracting around the backing store to avoid when it slows down

3 It can fail

Your backing store may just fail, should your service also fail for this?

Summary

Dynamo

[AE]LB

Caches

Clients

Load balancing is not magic

1 Routing algorithms

The algorithm used can drastically affect latency and load on your service.

2 It can fail

Load balancers are services too, they won't always just work. Scale your LB appropriately, especially if using failover as a strategy

Summary

Dynamo

[AE]LB

Caches

Clients

Caches are not magic

1 Popularity is not always best

The popular or most common solution may not be the one your service actually needs. Make sure you test alternatives in production style traffic

2 Know how it works

Does your cache amortise load? Does it offload it to a thread? What's its eviction strategy? These things are important to know

Summary

Dynamo

[AE]LB

Caches

Clients

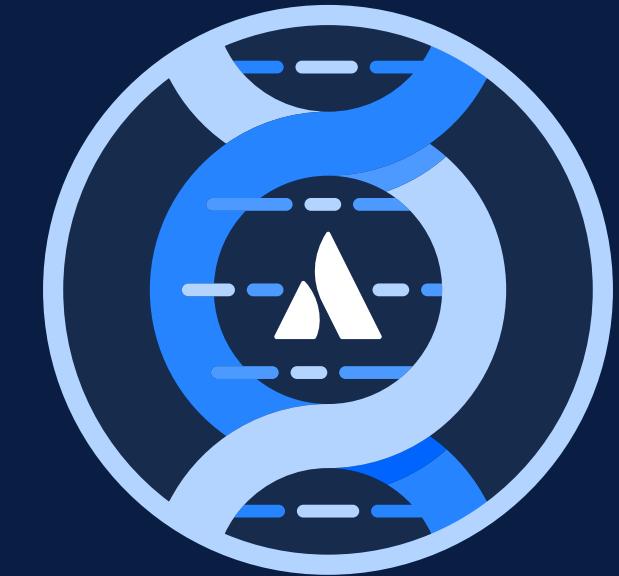
Clients are not magic

1 They can fail

Their node can simply be overloaded, causing it to seem like your service is down from their perspective

2 They can have suboptimal behaviour

Clients do their best to implement what they think they need / want. This might not always be the best way to actually call your service, consider building a sidecar to offload this work



Thank you!

“Cloud Provisioning Team” Room, go/tcs for docs



JACKSON MOES | SENIOR DEVELOPER - CLOUD PROVISIONING | @JSONMOES