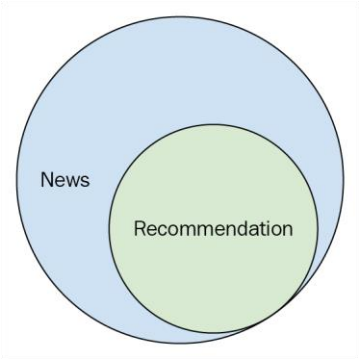
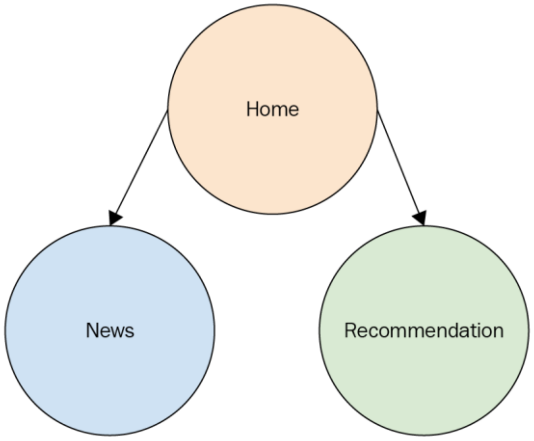
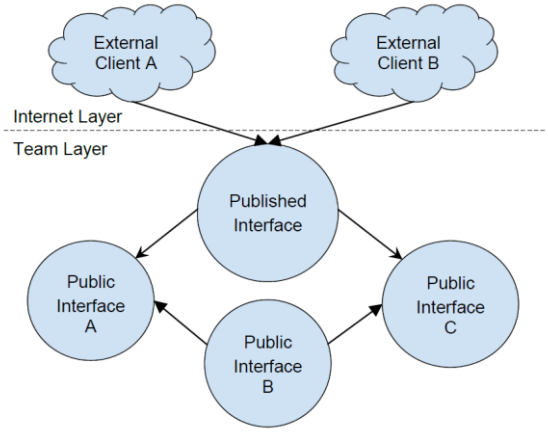
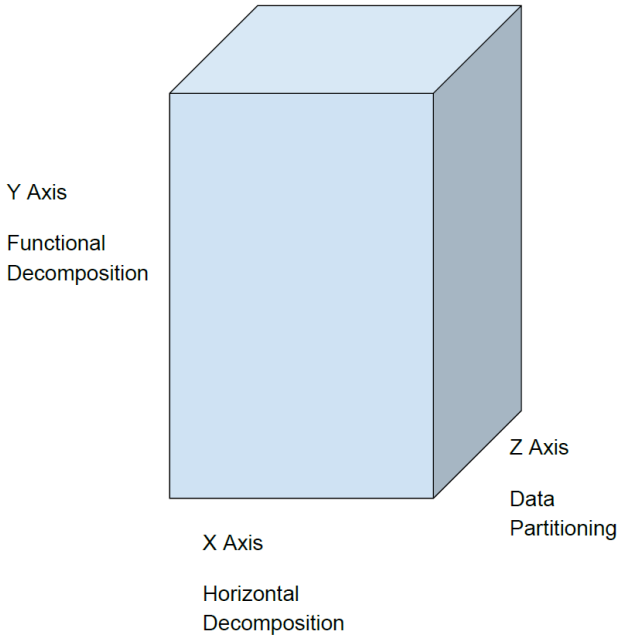
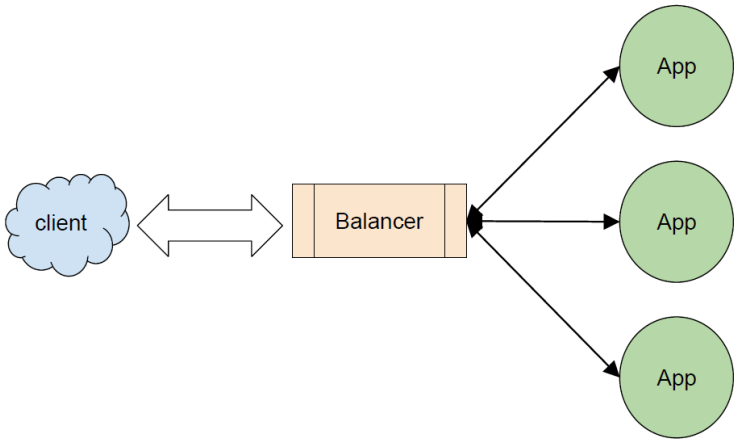
Chapter 1: Understanding the Microservices Concepts

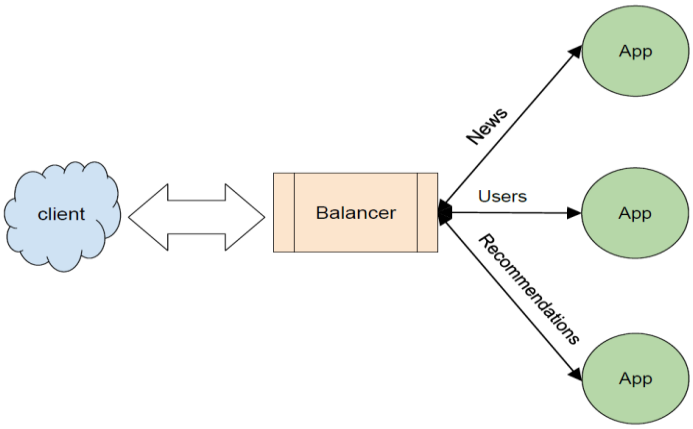


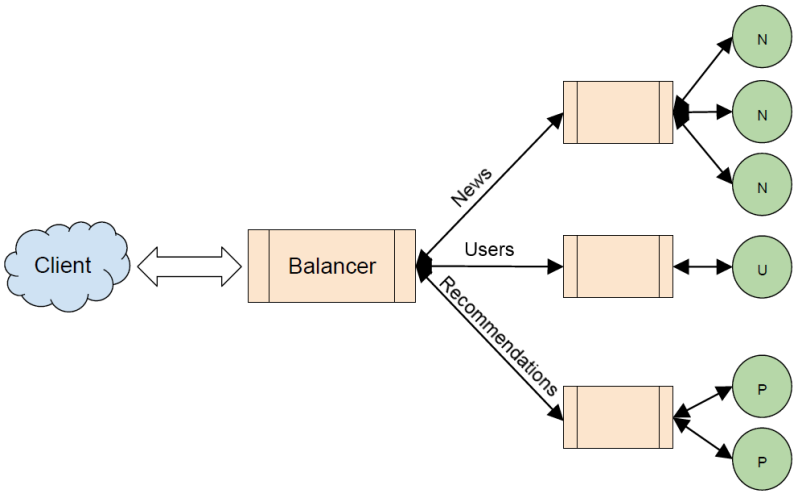


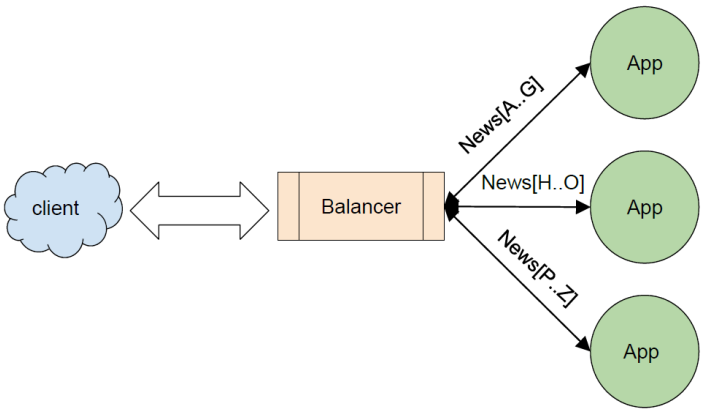


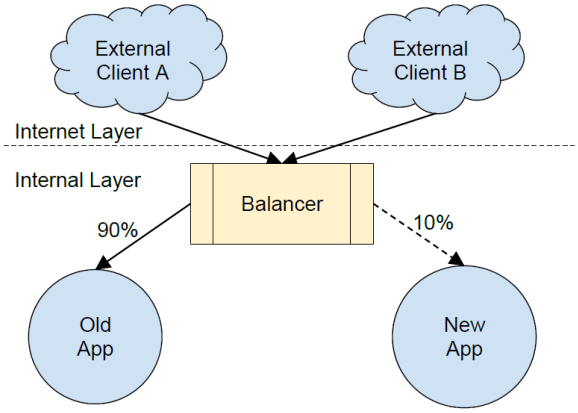

Y Axis
Functional
Decomposition
Z Axis
Data
Partitioning
X Axis
Horizontal
Decomposition

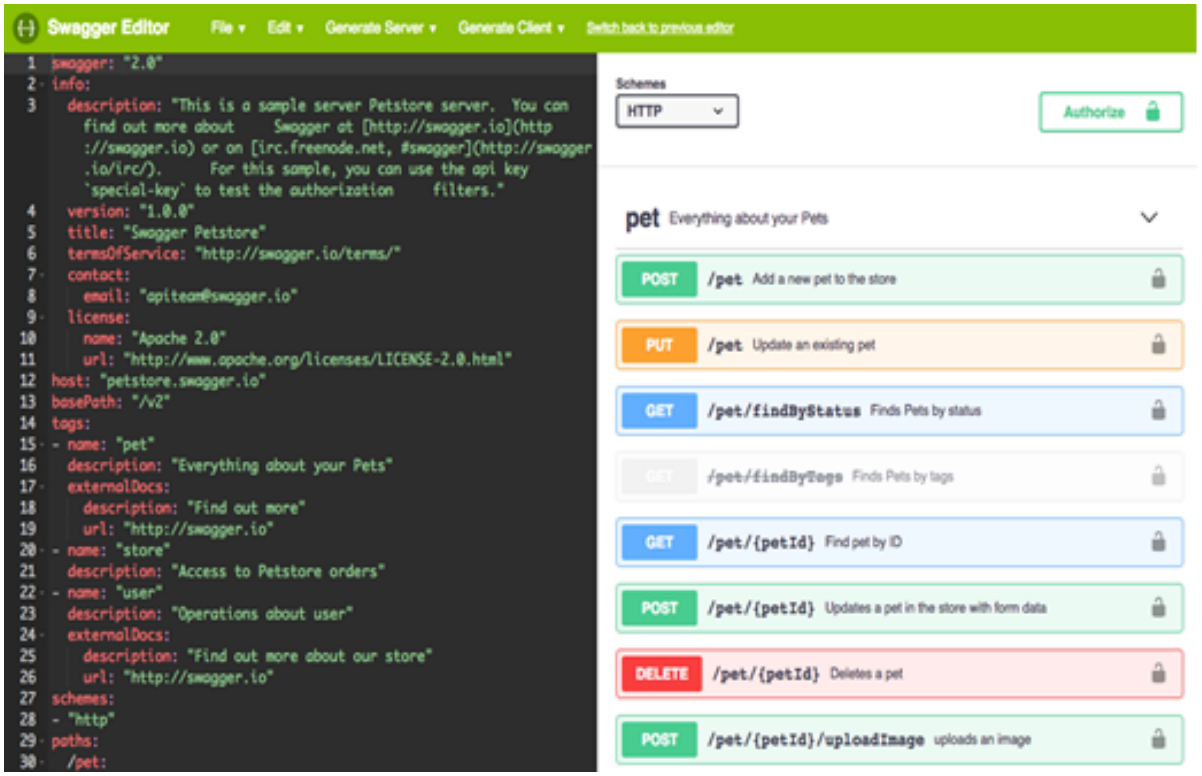

client
Balancer
App
App
App

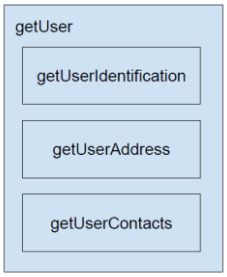

client
Balancer
News
Users
Recommendations
App
App
App

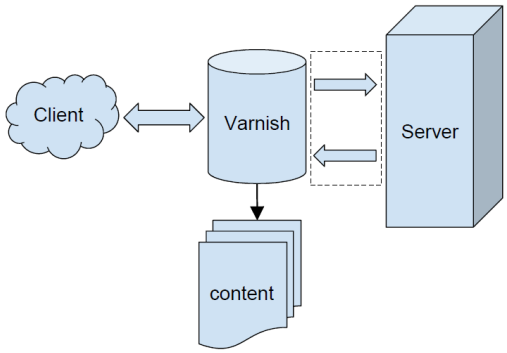

Client
Balancer
News
Users
Recommendations
N
N
N
U
P
P


client
Balancer
News[A..G]
News[H..O]
News[P..Z]
App
App
App

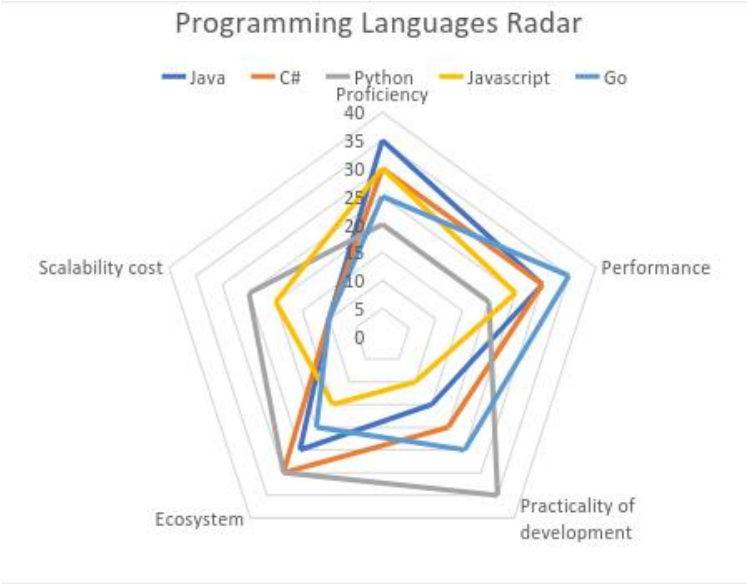

External Client A
External Client B
Internet Layer
Internal Layer
Balancer
90%
10%
Old App
New App

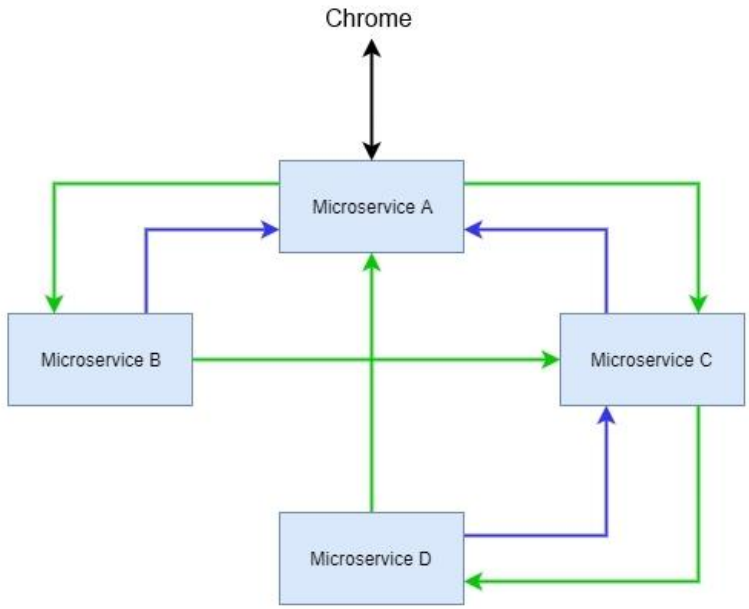

Swagger Editor
File
Edit
Generate Server
Generate Client
Switch back to previous editor
swagger: "2.0"
info:
description: "This is a sample server Petstore server. You can find out more about Swagger at http://swagger.io or on [irc.freenode.net, #swagger](http://swagger.io/irc/). For this sample, you can use the api key `special-key` to test the authorization filters."
version: "1.0.0"
title: "Swagger Petstore"
termsOfService: "http://swagger.io/terms/"
contact:
email: "apiteam@swagger.io"
license:
name: "Apache 2.0"
url: "http://www.apache.org/licenses/LICENSE-2.0.html"
host: "petstore.swagger.io"
basePath: "/v2"
tags:
- name: "pet"
description: "Everything about your Pets"
externalDocs:
description: "Find out more"
url: "http://swagger.io"
- name: "store"
description: "Access to Petstore orders"
- name: "user"
description: "Operations about user"
externalDocs:
description: "Find out more about our store"
url: "http://swagger.io"
schemes:
- "http"
paths:
/pet:
Schemes
HTTP
Authorize
pet Everything about your Pets
POST /pet Add a new pet to the store
PUT /pet Update an existing pet
GET /pet/findByStatus Finds Pets by status
GET /pet/findByTags Finds Pets by tags
GET /pet/{petId} Find pet by ID
POST /pet/{petId} Updates a pet in the store with form data
DELETE /pet/{petId} Deletes a pet
POST /pet/{petId}/uploadImage uploads an image

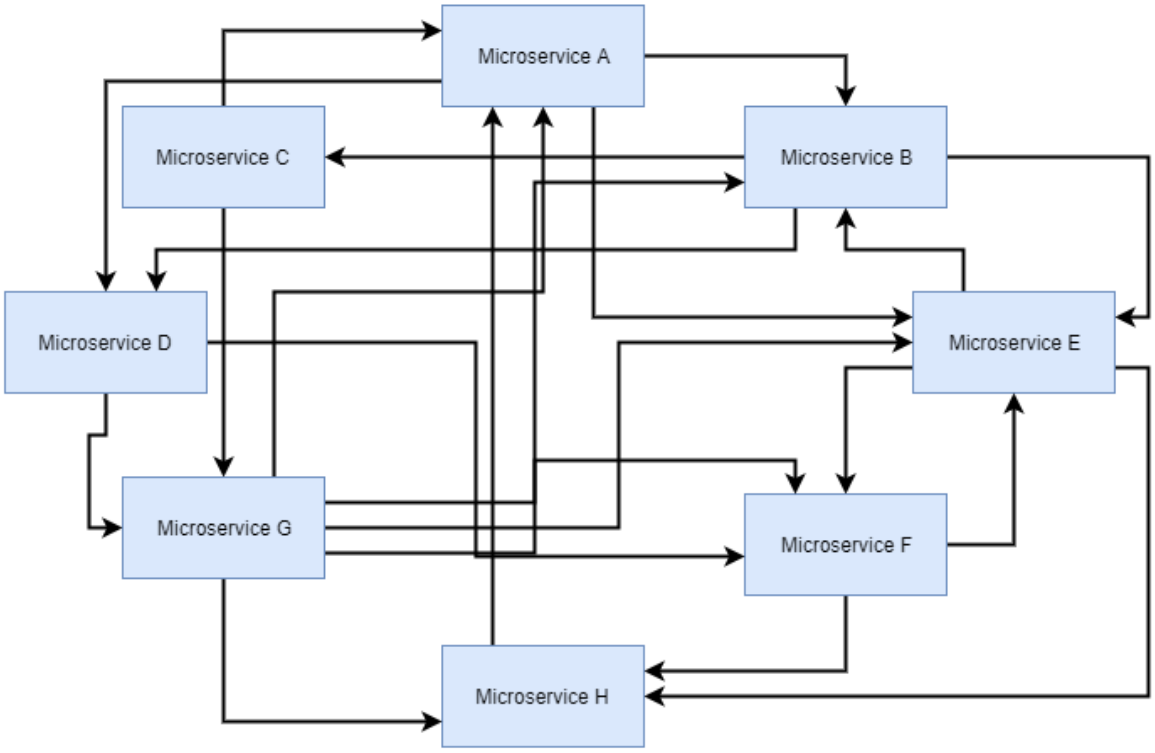

getUser
getUserIdentification
getUserAddress
getUserContacts

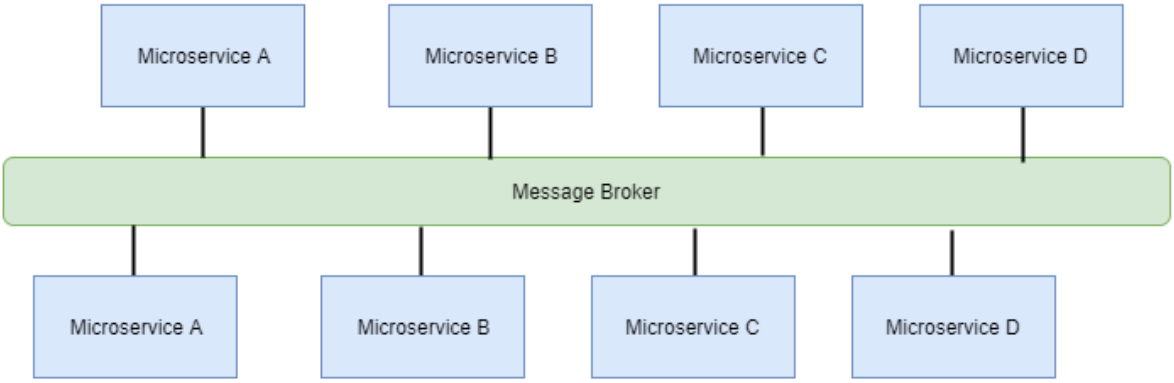


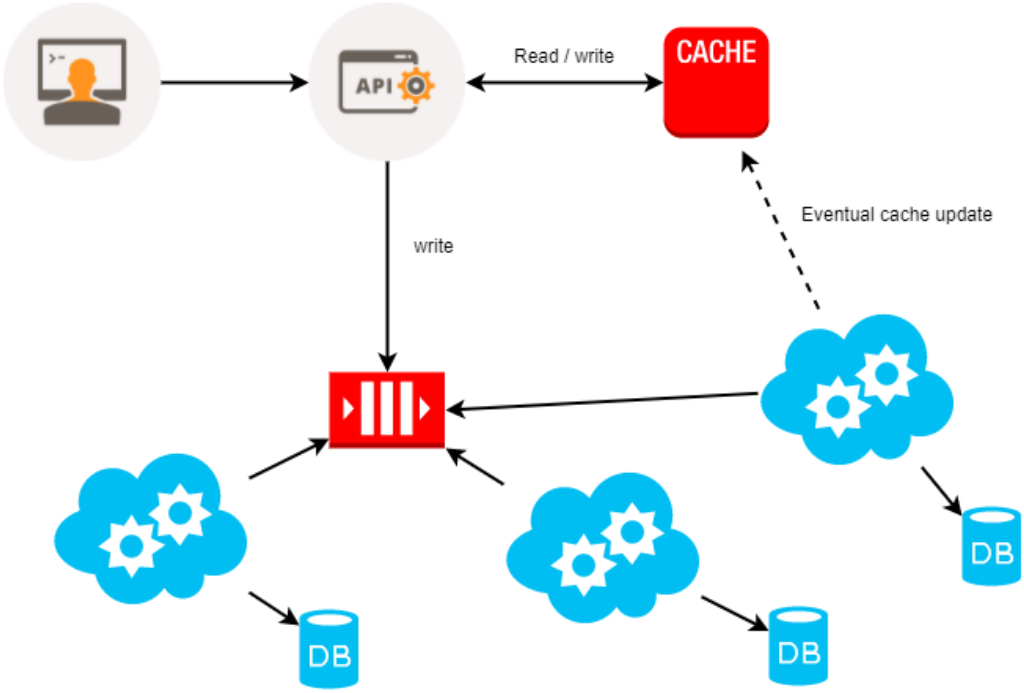
Chapter 2: The Microservice Tools

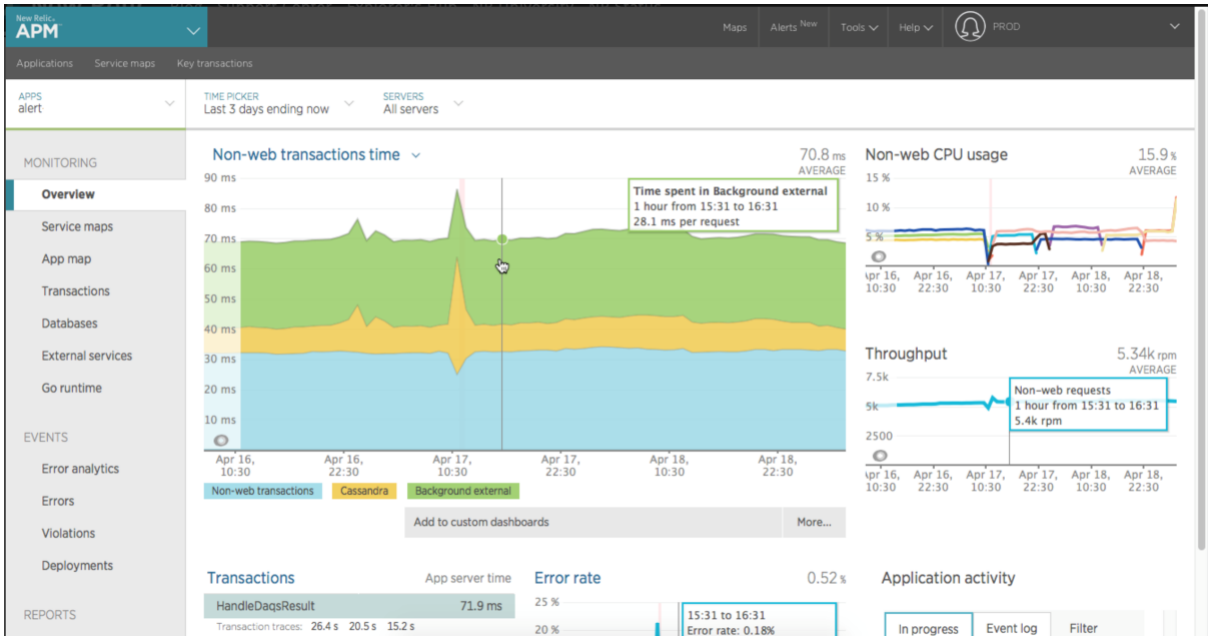


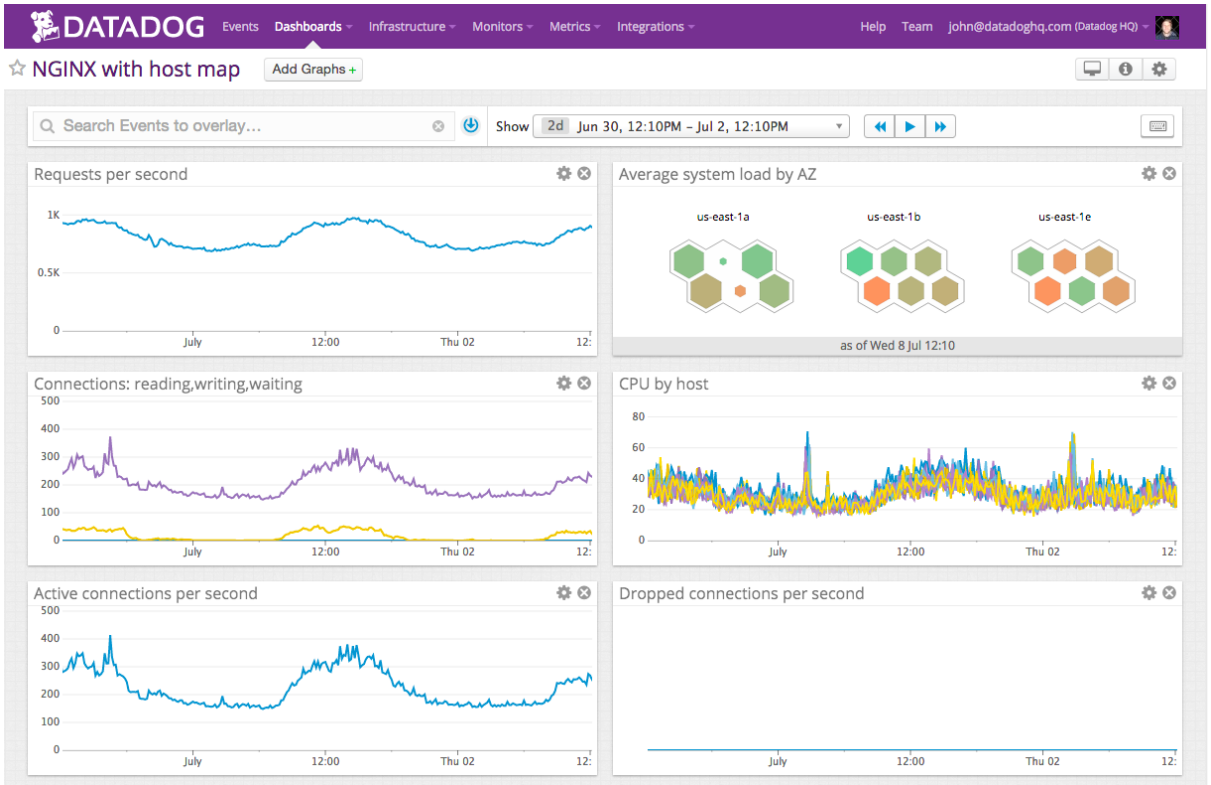

Chrome
Microservice A
Microservice B
Microservice C
Microservice D

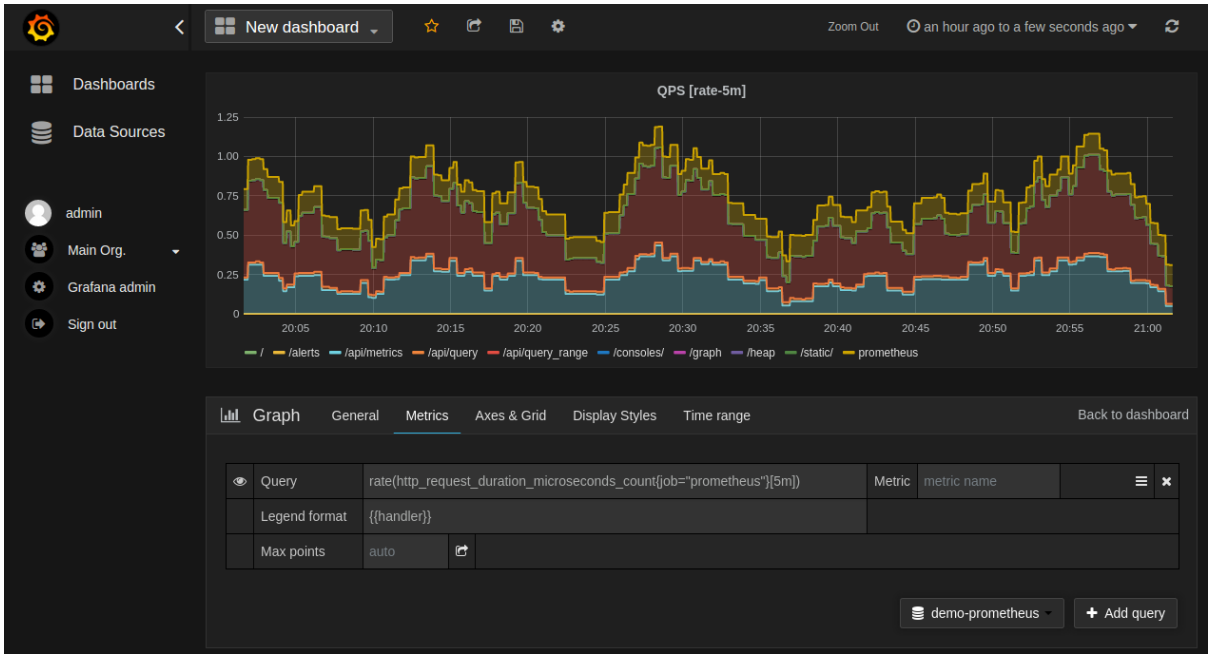

Microservice A
Microservice C
Microservice B
Microservice D
Microservice E
Microservice G
Microservice F
Microservice H


Microservice A
Microservice B
Microservice C
Microservice D
Message Broker
Microservice A
Microservice B
Microservice C
Microservice D


API
Read / write
CACHE
write
Eventual cache update
DB
DB
DB


New Relic.
APM
Maps
Alerts New
Tools
Help
PROD
Applications
Service maps
Key transactions
APPS
alert
TIME PICKER
Last 3 days ending now
SERVERS
All servers
MONITORING
Overview
Service maps
App map
Transactions
Databases
External services
Go runtime
EVENTS
Error analytics
Errors
Violations
Deployments
REPORTS
Non-web transactions time
70.8 ms
AVERAGE
Time spent in Background external
1 hour from 15:31 to 16:31
28.1 ms per request
Non-web transactions
Cassandra
Background external
Add to custom dashboards
More...
Non-web CPU usage
15.9 %
AVERAGE
Throughput
5.34k rpm
AVERAGE
Non-web requests
1 hour from 15:31 to 16:31
5.4k rpm
Transactions
App server time
HandleDaqsResult
71.9 ms
Transaction traces: 26.4 s 20.5 s 15.2 s
Error rate
0.52 %
15:31 to 16:31
Error rate: 0.18%
Application activity
In progress
Event log
Filter


DATADOG
Events
Dashboards
Infrastructure
Monitors
Metrics
Integrations
Help
Team
john@datadoghq.com (Datadog HQ)
NGINX with host map
Add Graphs +
Search Events to overlay...
Show
2d
Jun 30, 12:10PM – Jul 2, 12:10PM
Requests per second
Average system load by AZ
us-east-1a
us-east-1b
us-east-1e
as of Wed 8 Jul 12:10
Connections: reading,writing,waiting
CPU by host
Active connections per second
Dropped connections per second
July
12:00
Thu 02




```
Server Software:        Werkzeug/0.12.2
Server Hostname:        localhost
Server Port:            5000

Document Path:          /
Document Length:        11 bytes

Concurrency Level:      100
Time taken for tests:   2.247 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      165000 bytes
HTML transferred:       11000 bytes
Requests per second:    444.99 [#/sec] (mean)
Time per request:       224.722 [ms] (mean)
Time per request:       2.247 [ms] (mean, across all concurrent requests)
Transfer rate:          71.70 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.1      0       1
Processing:    59  210  37.5    221     231
Waiting:       53  210  38.2    221     231
Total:         59  210  37.5    221     232
```

wrk - a HTTP benchmarking tool

wrk is a modern HTTP benchmarking tool capable of generating significant load when run on a single multi-core CPU. It combines a multithreaded design with scalable event notification systems such as epoll and kqueue.

An optional LuaJIT script can perform HTTP request generation, response processing, and custom reporting. Details are available in SCRIPTING and several examples are located in scripts/.

Basic Usage

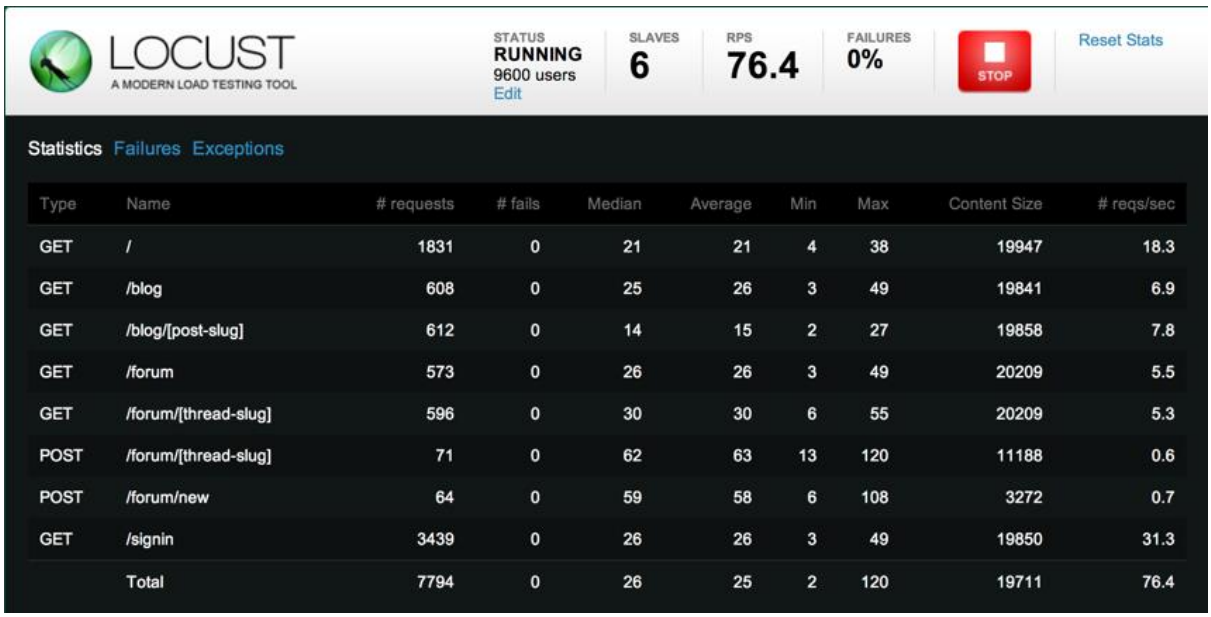
```
wrk -t12 -c400 -d30s http://127.0.0.1:8080/index.html
```

This runs a benchmark for 30 seconds, using 12 threads, and keeping 400 HTTP connections open.

Output:

```
Running 30s test @ http://127.0.0.1:8080/index.html
  12 threads and 400 connections
  Thread Stats   Avg      Stdev     Max   +/- Stdev
    Latency   635.91us    0.89ms  12.92ms   93.69%
    Req/Sec    56.20k     8.07k   62.00k    86.54%
  22464657 requests in 30.00s, 17.76GB read
Requests/sec: 748868.53
Transfer/sec:    606.33MB
```

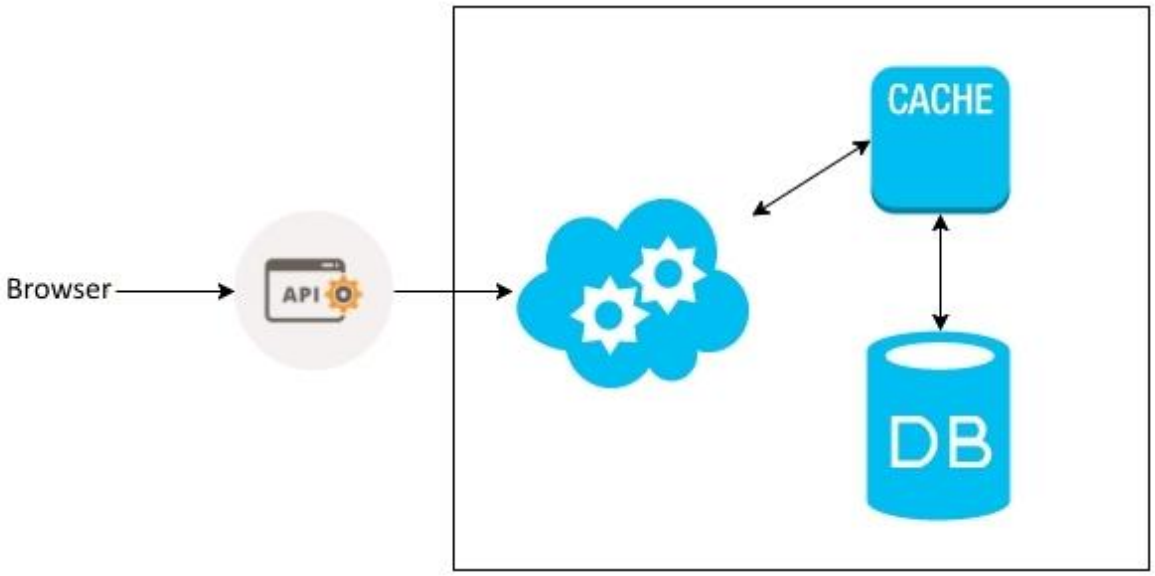
```
Running 10s test @ http://localhost:5000/
  4 threads and 100 connections
  Thread Stats   Avg      Stdev     Max   +/- Stdev
    Latency   268.68ms   73.98ms 678.53ms   89.14%
    Req/Sec    92.41     25.42   141.00     69.17%
  3688 requests in 10.09s, 594.26KB read
Requests/sec:    365.55
Transfer/sec:     58.90KB
```

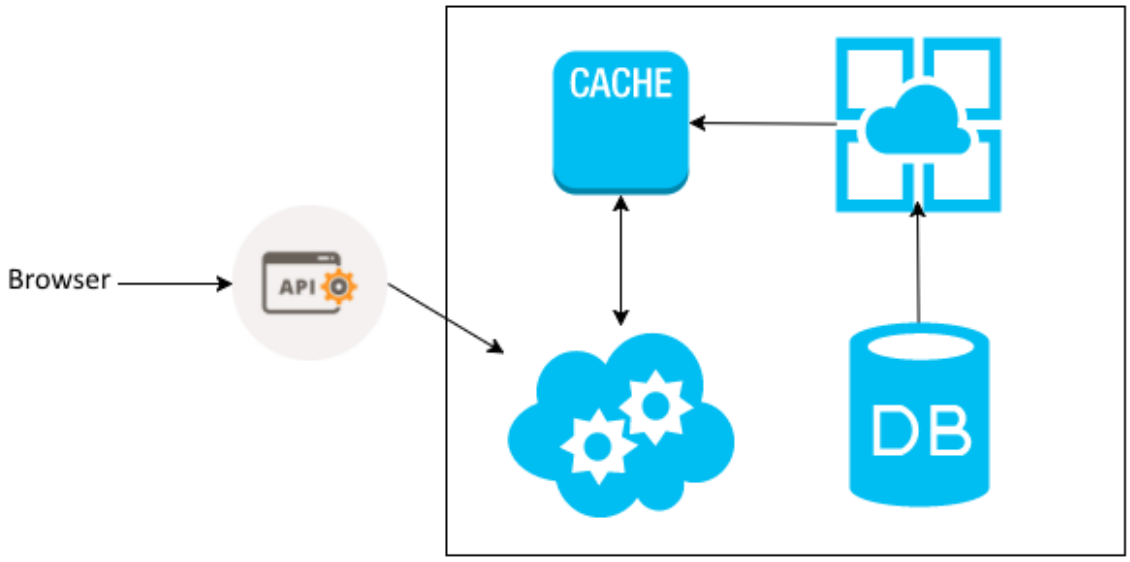


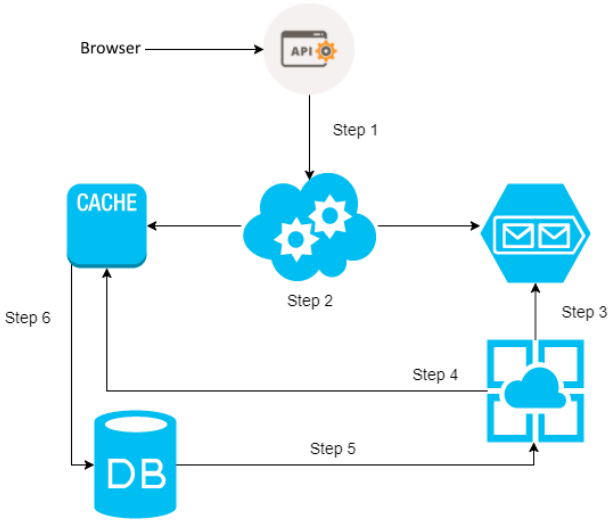
Type	Name	# requests	# fails	Median	Average	Min	Max	Content Size	# reqs/sec
GET	/	1831	0	21	21	4	38	19947	18.3
GET	/blog	608	0	25	26	3	49	19841	6.9
GET	/blog/[post-slug]	612	0	14	15	2	27	19858	7.8
GET	/forum	573	0	26	26	3	49	20209	5.5
GET	/forum/[thread-slug]	596	0	30	30	6	55	20209	5.3
POST	/forum/[thread-slug]	71	0	62	63	13	120	11188	0.6
POST	/forum/new	64	0	59	58	6	108	3272	0.7
GET	/signin	3439	0	26	26	3	49	19850	31.3
	Total	7794	0	26	25	2	120	19711	76.4

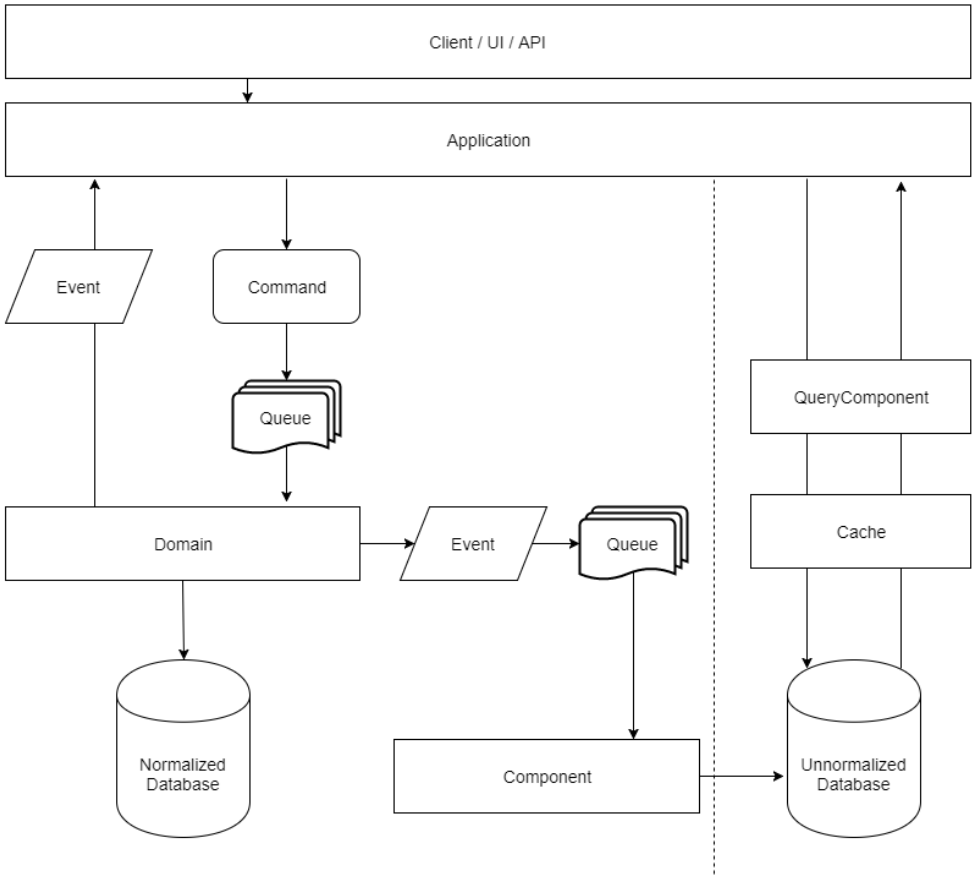
Chapter 3: Internal Patterns

```
                                        Table "public.users"
  Column  |  Type   |                     Modifiers                      | Storage  | Stats target | Description
----------+---------+----------------------------------------------------+----------+--------------+-------------
 id       | integer | not null default nextval('users_id_seq'::regclass) | plain    |              |
 name     | text    | not null                                           | extended |              |
 email    | text    | not null                                           | extended |              |
 password | text    | not null                                           | extended |              |
Indexes:
    "users_pkey" PRIMARY KEY, btree (id)
```

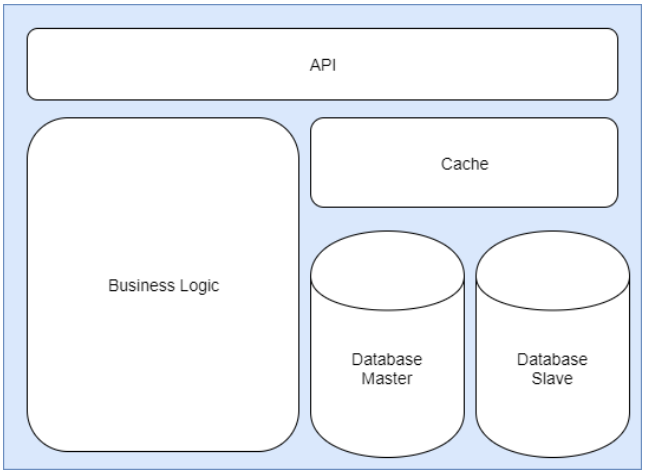


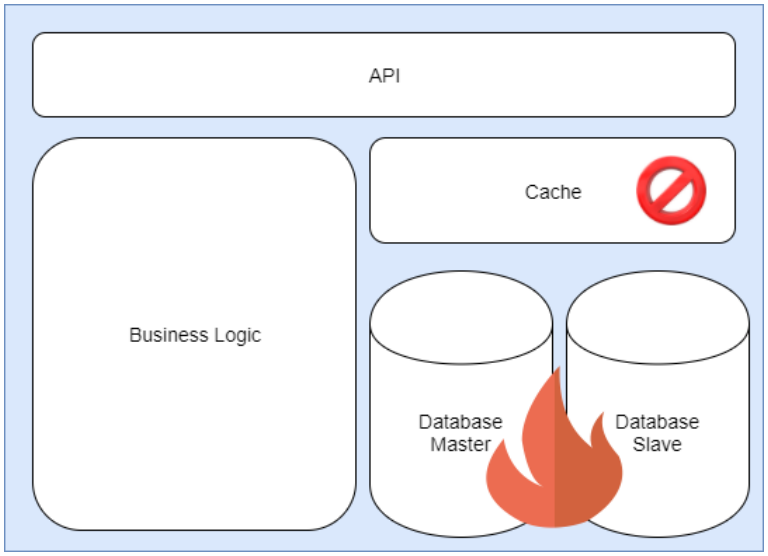

CACHE
Browser
API
DB

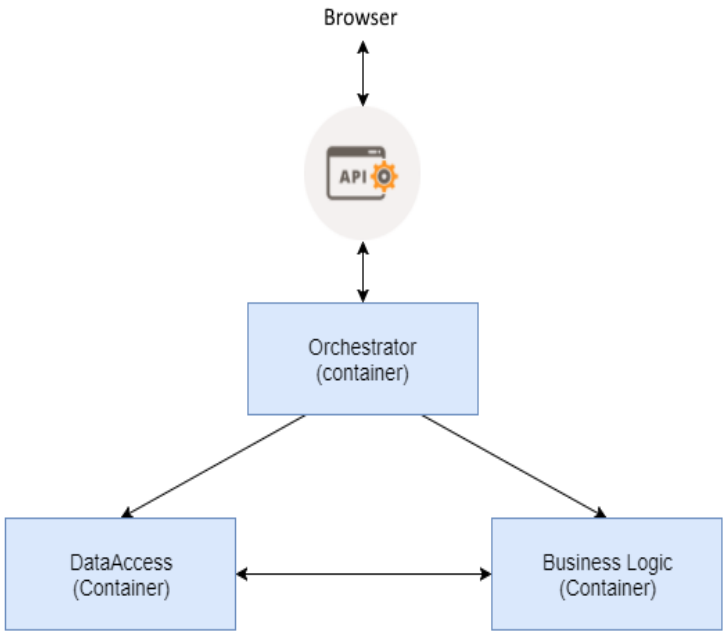

Browser
API
Step 1
CACHE
Step 2
Step 3
Step 6
Step 4
DB
Step 5

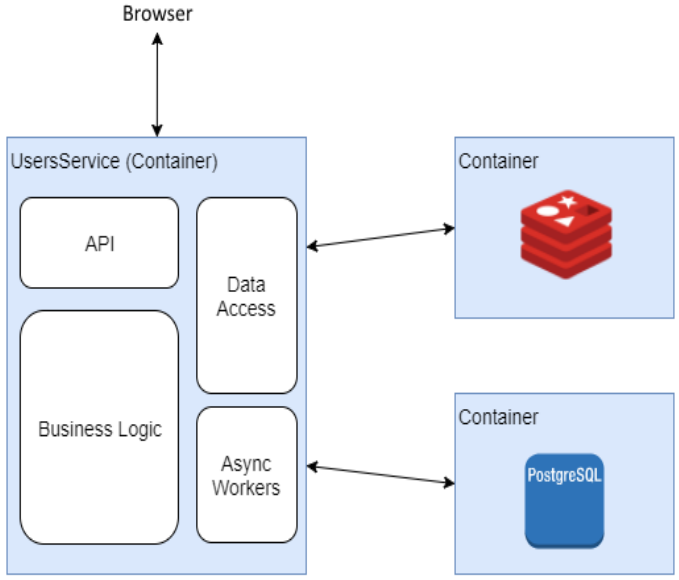

Client / UI / API
Application
Event
Command
Queue
QueryComponent
Cache
Domain
Event
Queue
Normalized Database
Component
Unnormalized Database

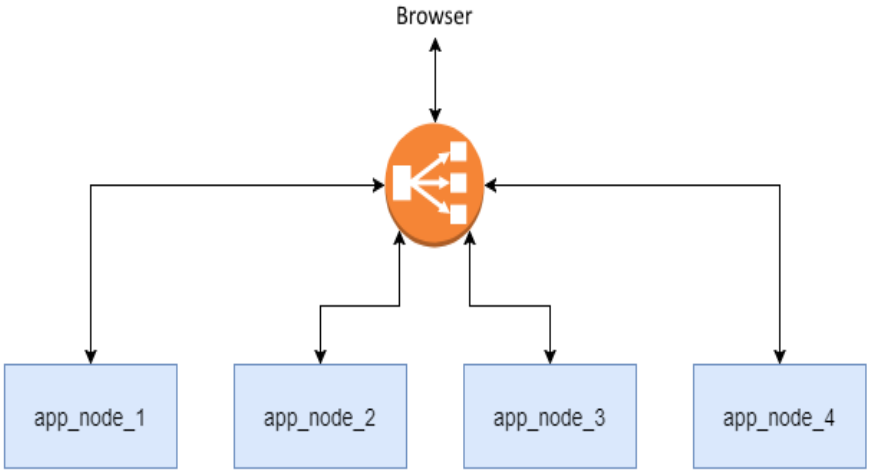
Chapter 4: Microservice Ecosystem

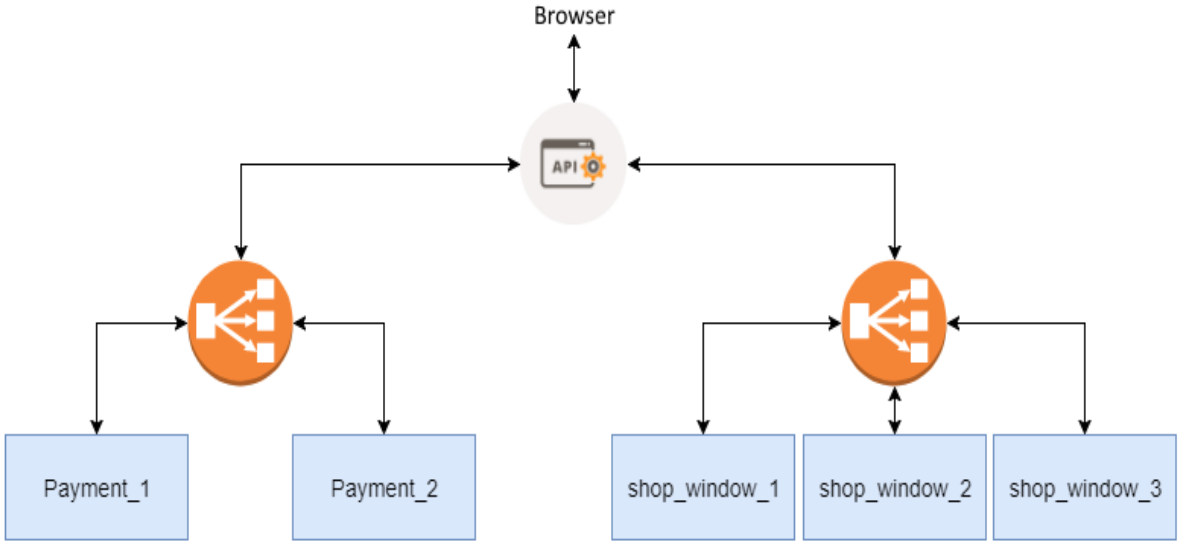


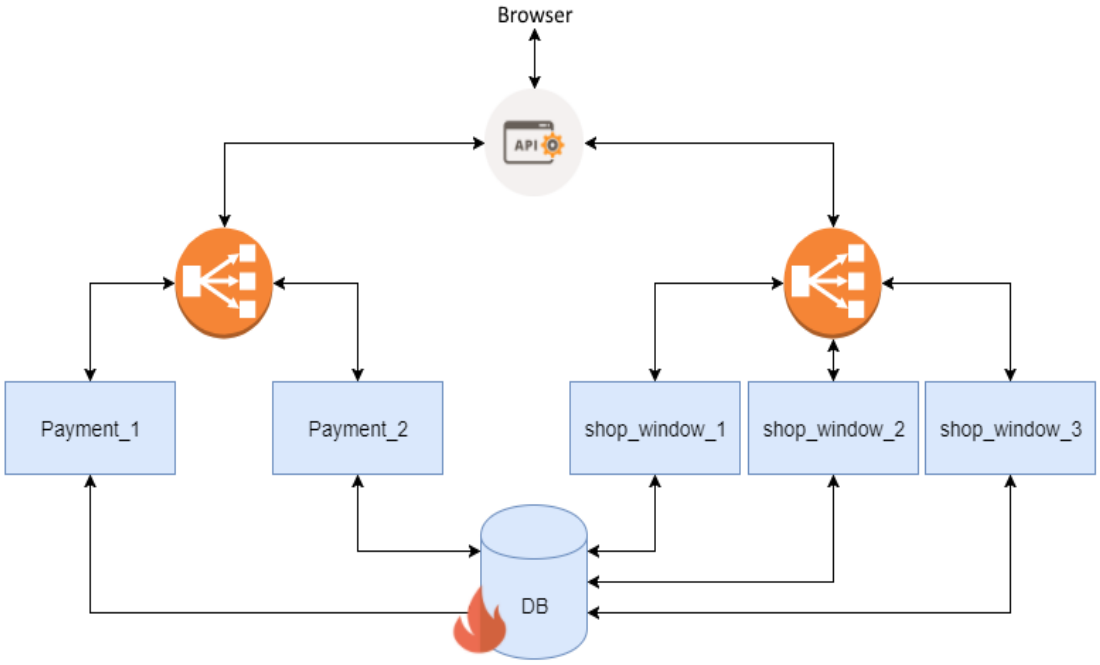


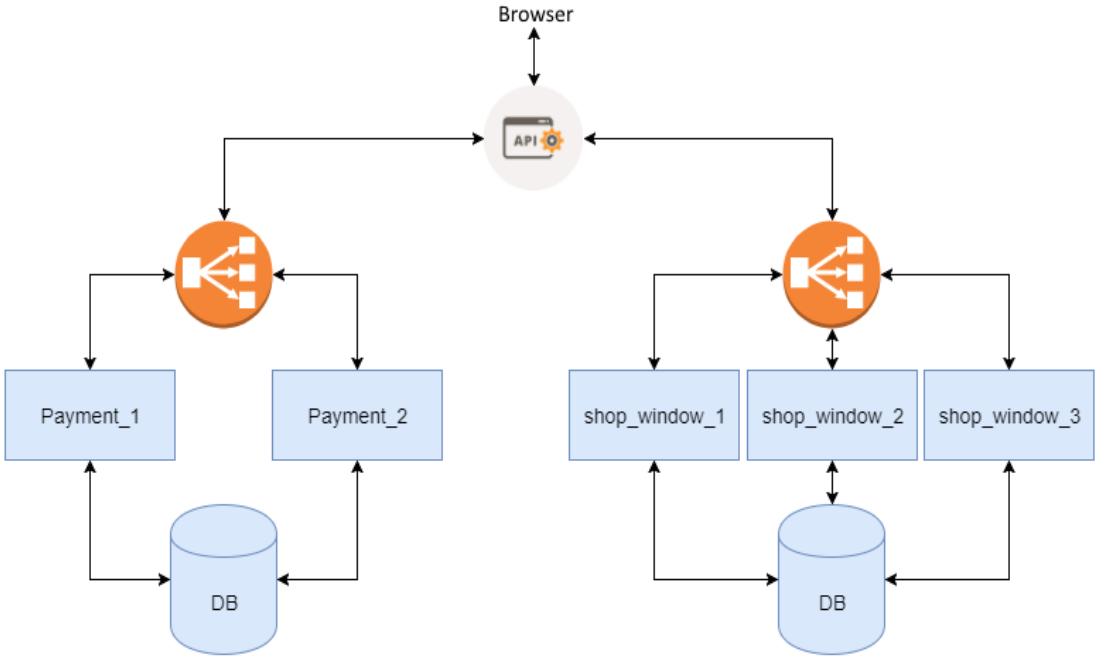

Browser
API
Orchestrator
(container)
DataAccess
(Container)
Business Logic
(Container)

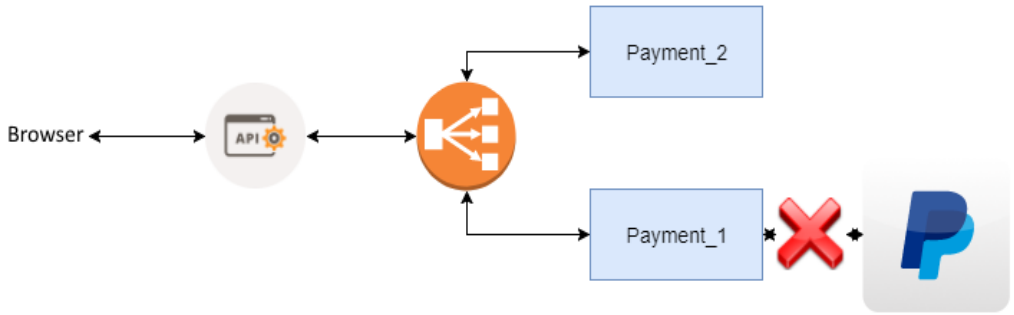

Browser
UsersService (Container)
API
Data
Access
Business Logic
Async
Workers
Container
Container
PostgreSQL

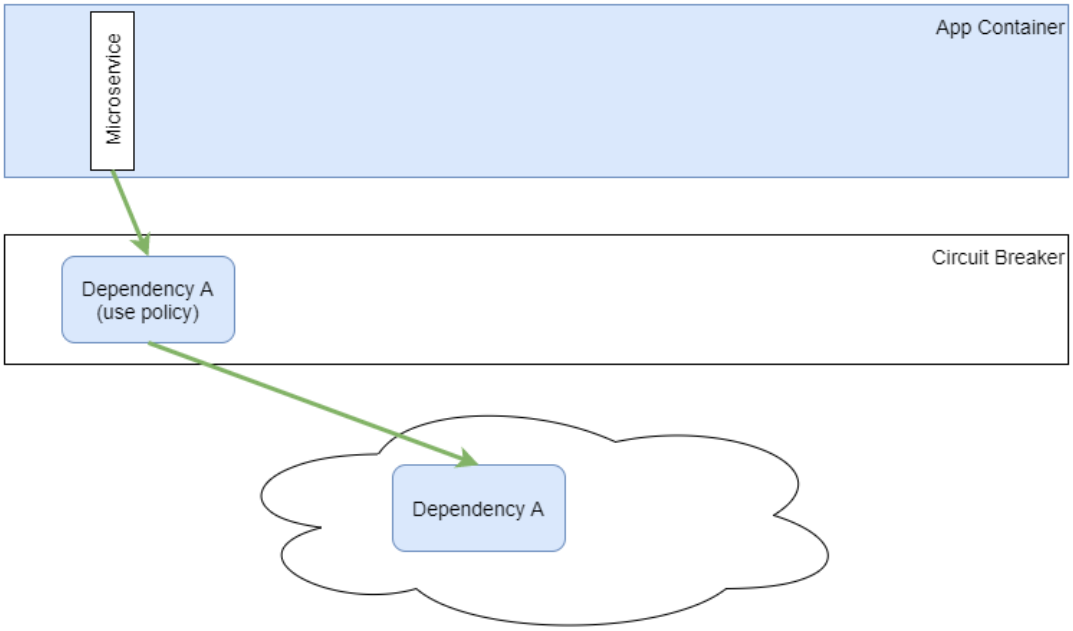

Browser
app_node_1
app_node_2
app_node_3
app_node_4


Browser
API
Payment_1
Payment_2
shop_window_1
shop_window_2
shop_window_3

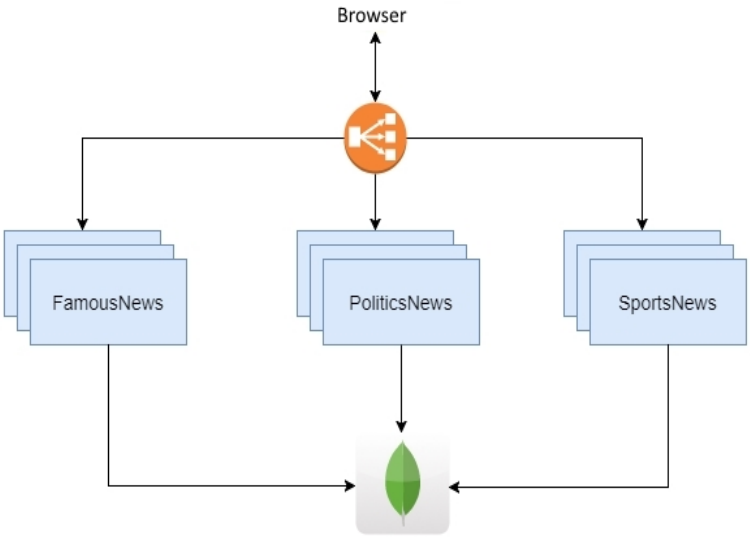

Browser
API
Payment_1
Payment_2
shop_window_1
shop_window_2
shop_window_3
DB


Browser
API
Payment_1
Payment_2
shop_window_1
shop_window_2
shop_window_3
DB
DB

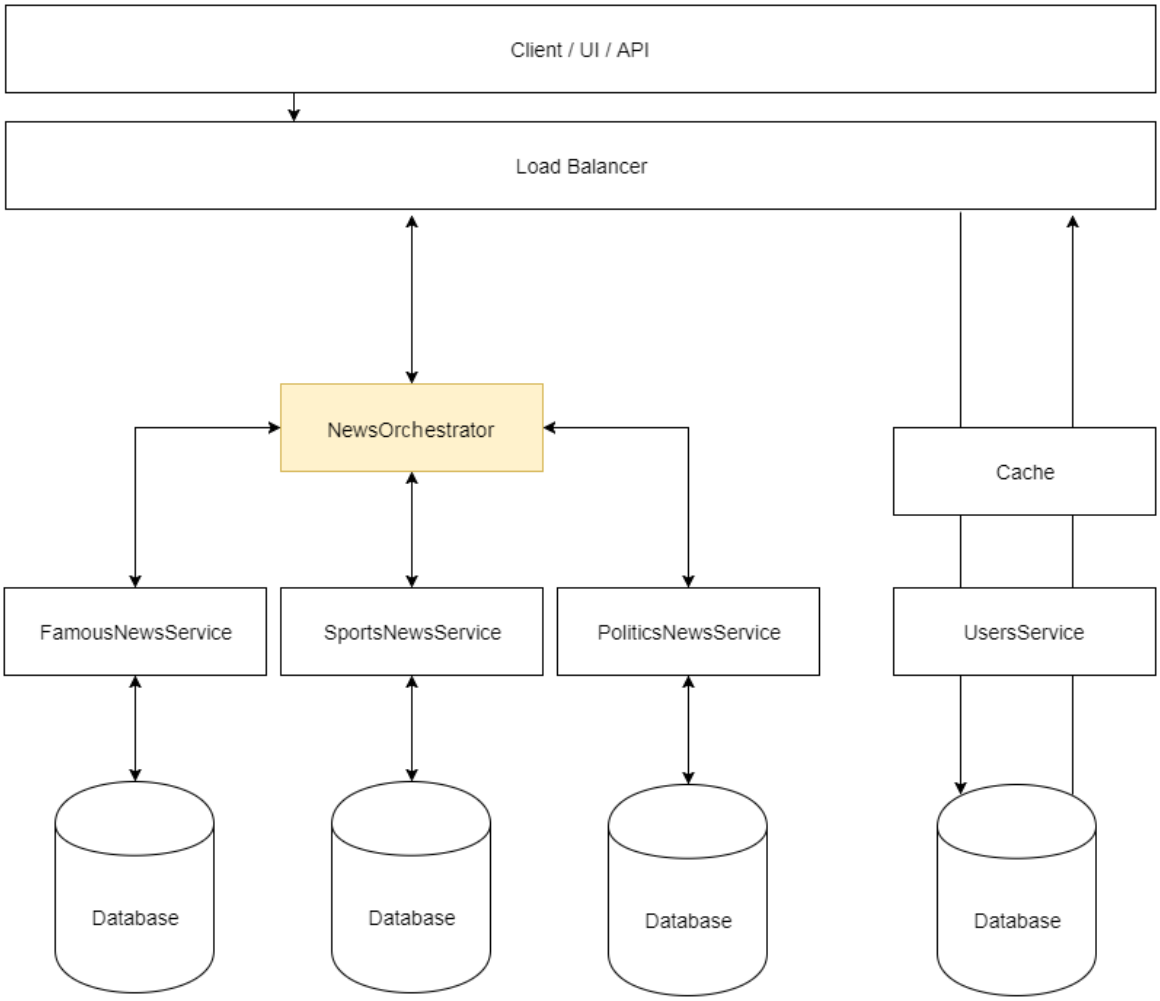

Browser
API
Payment_2
Payment_1

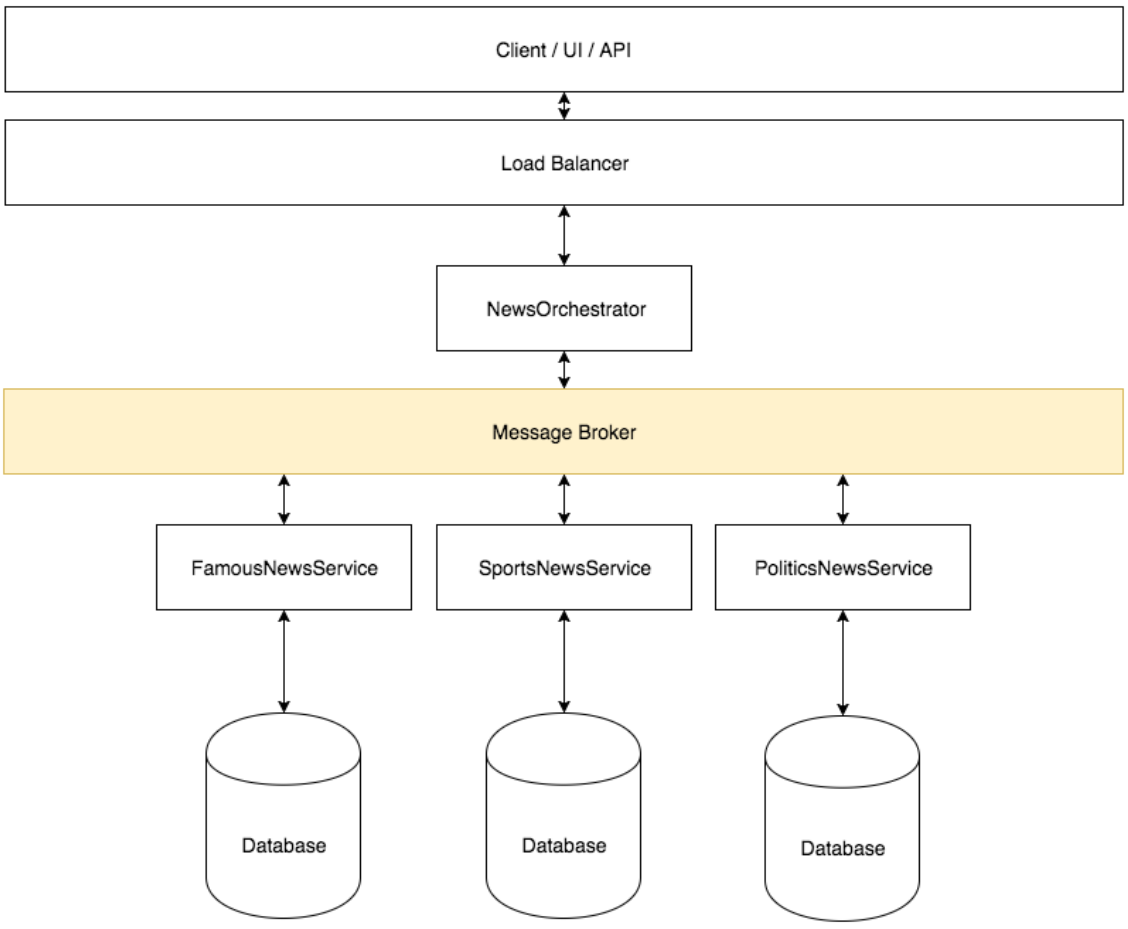

App Container
Microservice
Circuit Breaker
Dependency A
(use policy)
Dependency A

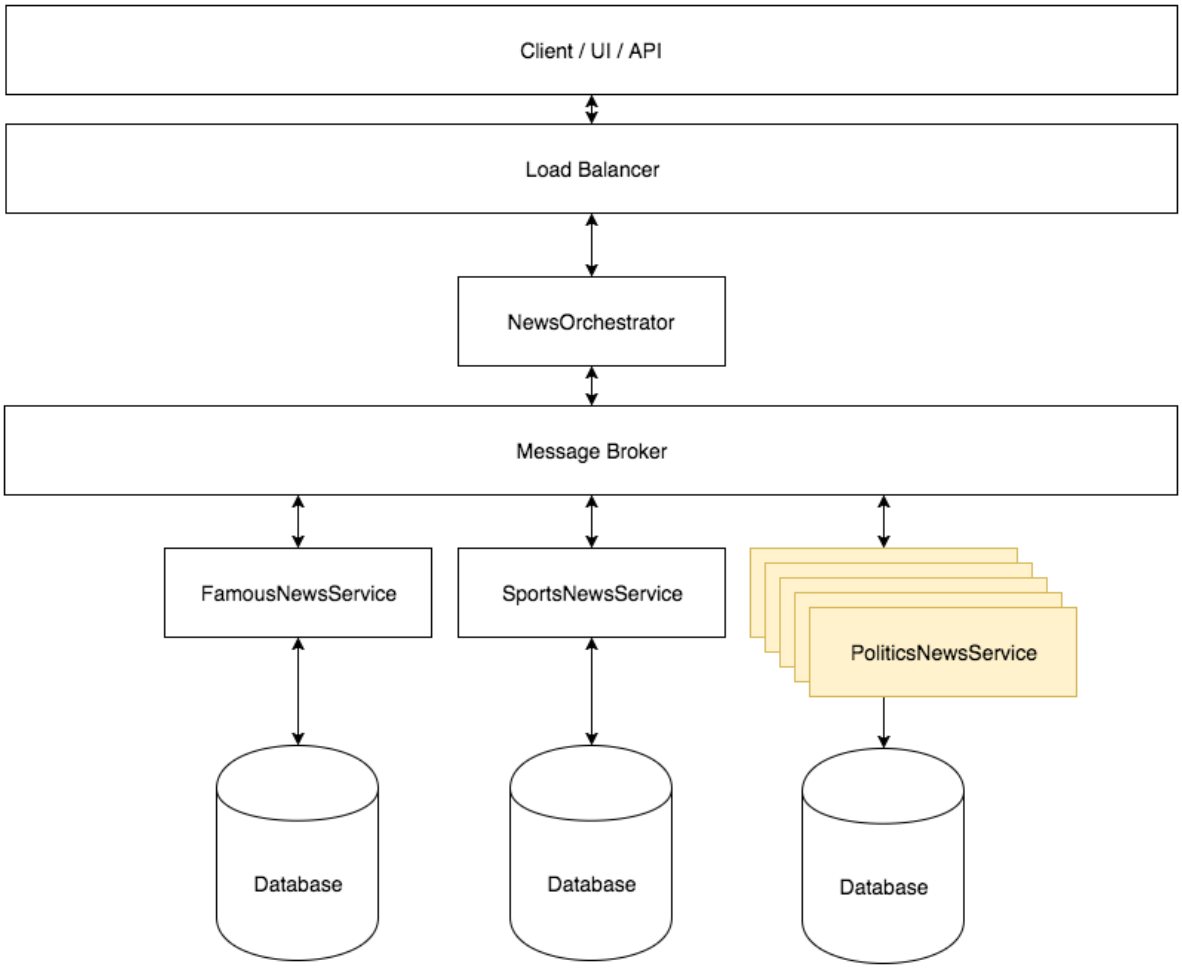
Chapter 5: Shared Data Microservice Design Pattern

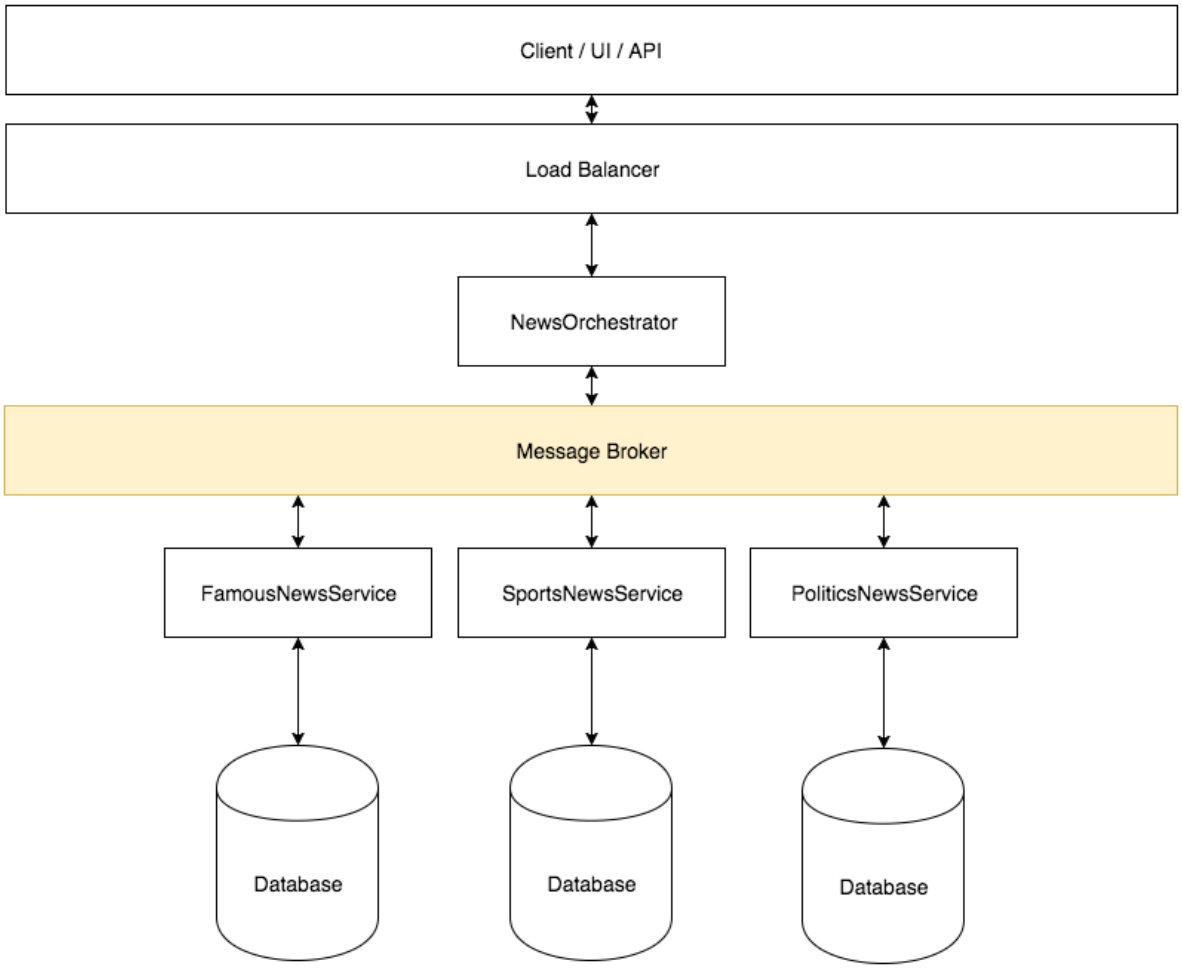


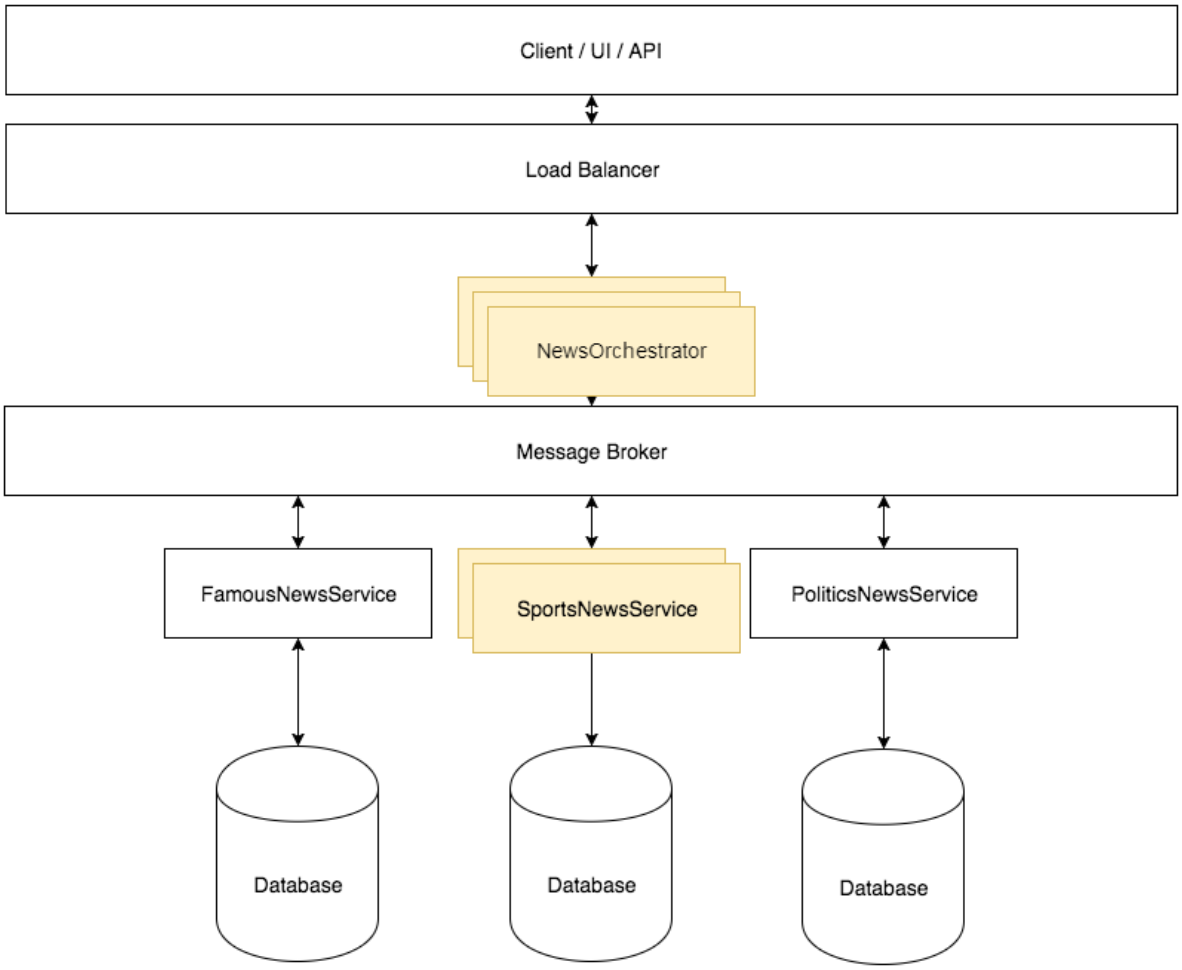
Chapter 6: Aggregator Microservice Design Pattern



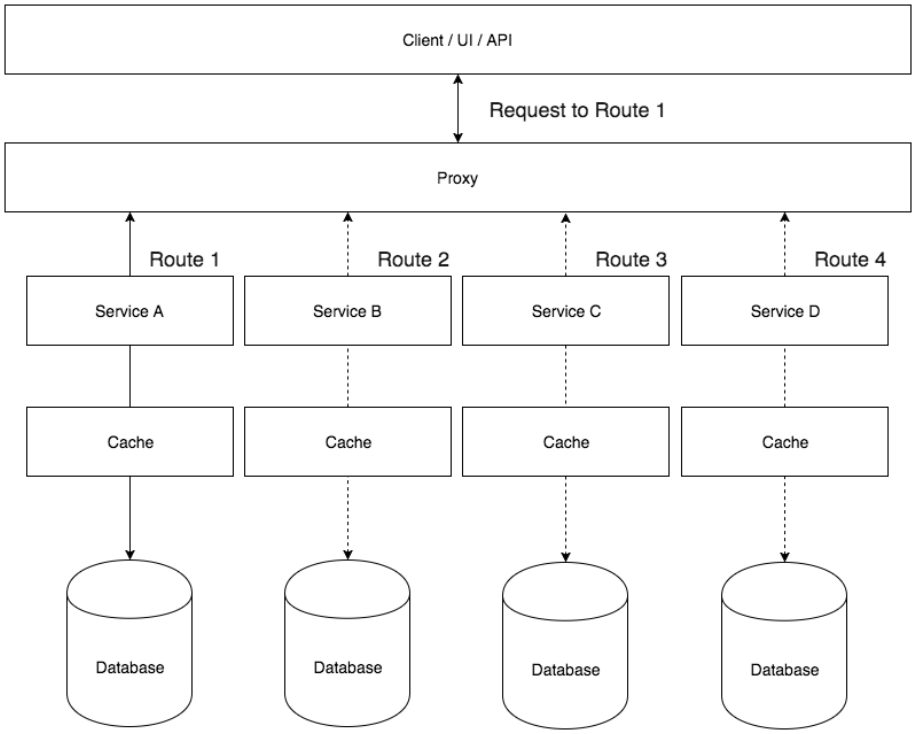

Client / UI / API
Load Balancer
NewsOrchestrator
Message Broker
FamousNewsService
SportsNewsService
PoliticsNewsService
Database
Database
Database

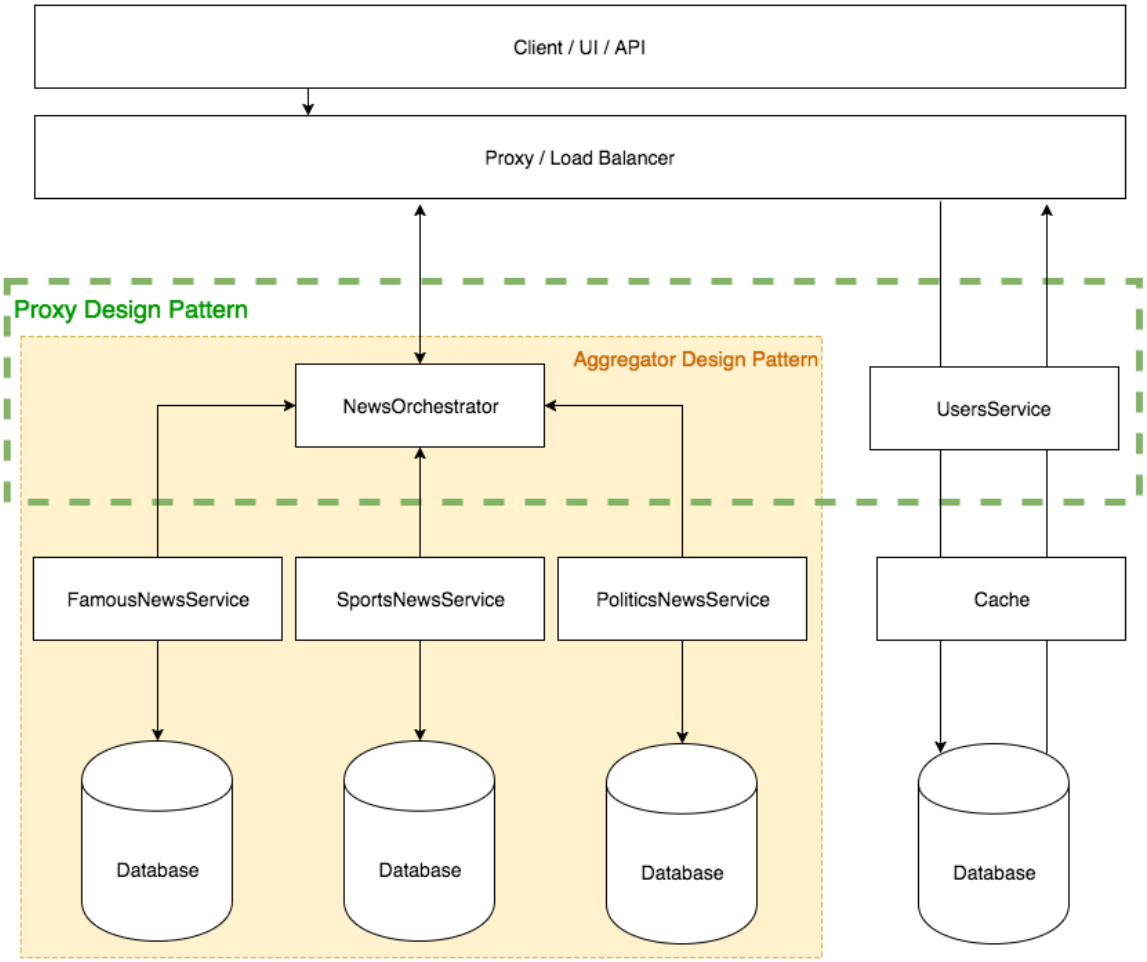

Client / UI / API
Load Balancer
NewsOrchestrator
Message Broker
FamousNewsService
SportsNewsService
PoliticsNewsService
Database
Database
Database

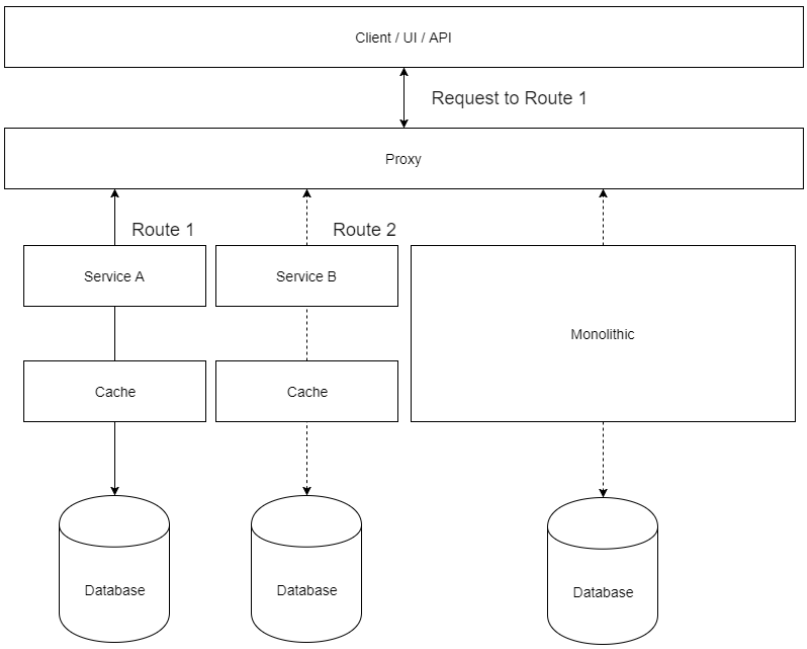

Client / UI / API
Load Balancer
NewsOrchestrator
Message Broker
FamousNewsService
SportsNewsService
PoliticsNewsService
Database
Database
Database

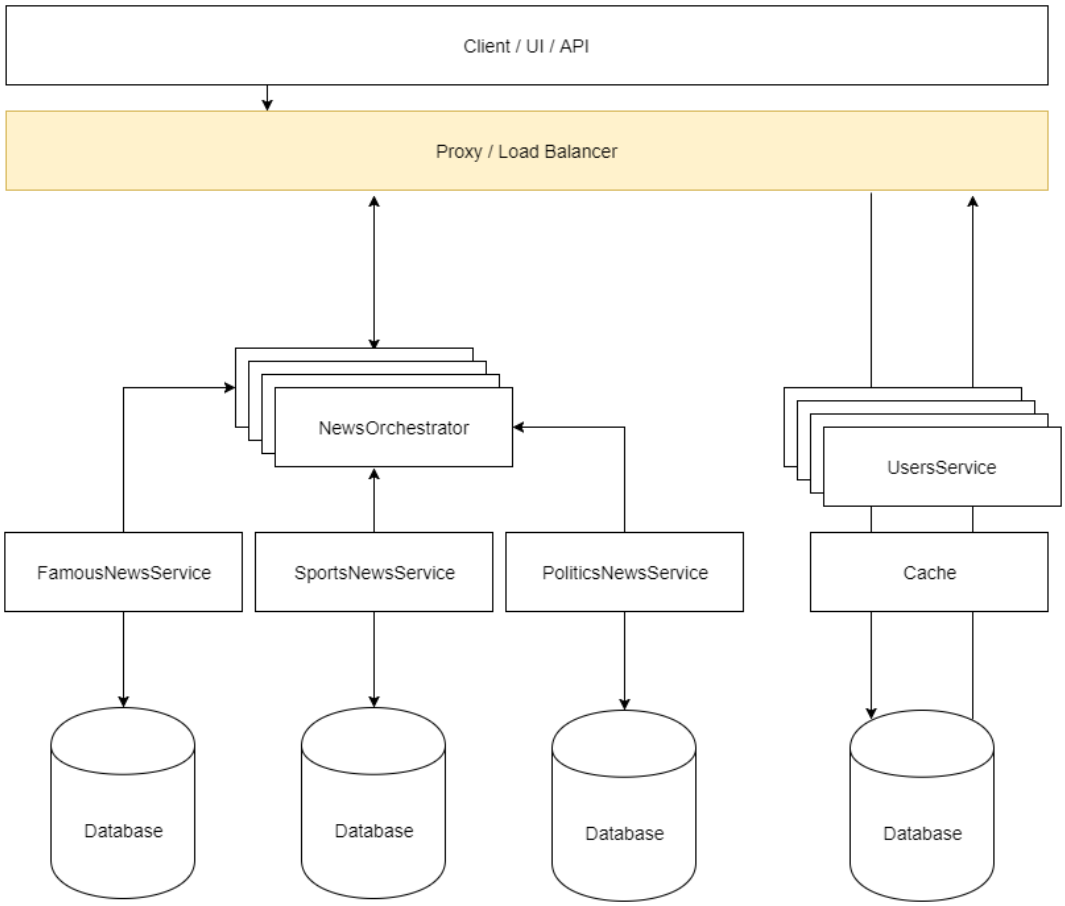

Client / UI / API
Load Balancer
NewsOrchestrator
Message Broker
FamousNewsService
SportsNewsService
PoliticsNewsService
Database
Database
Database

Chapter 7: Proxy Microservice Design Pattern

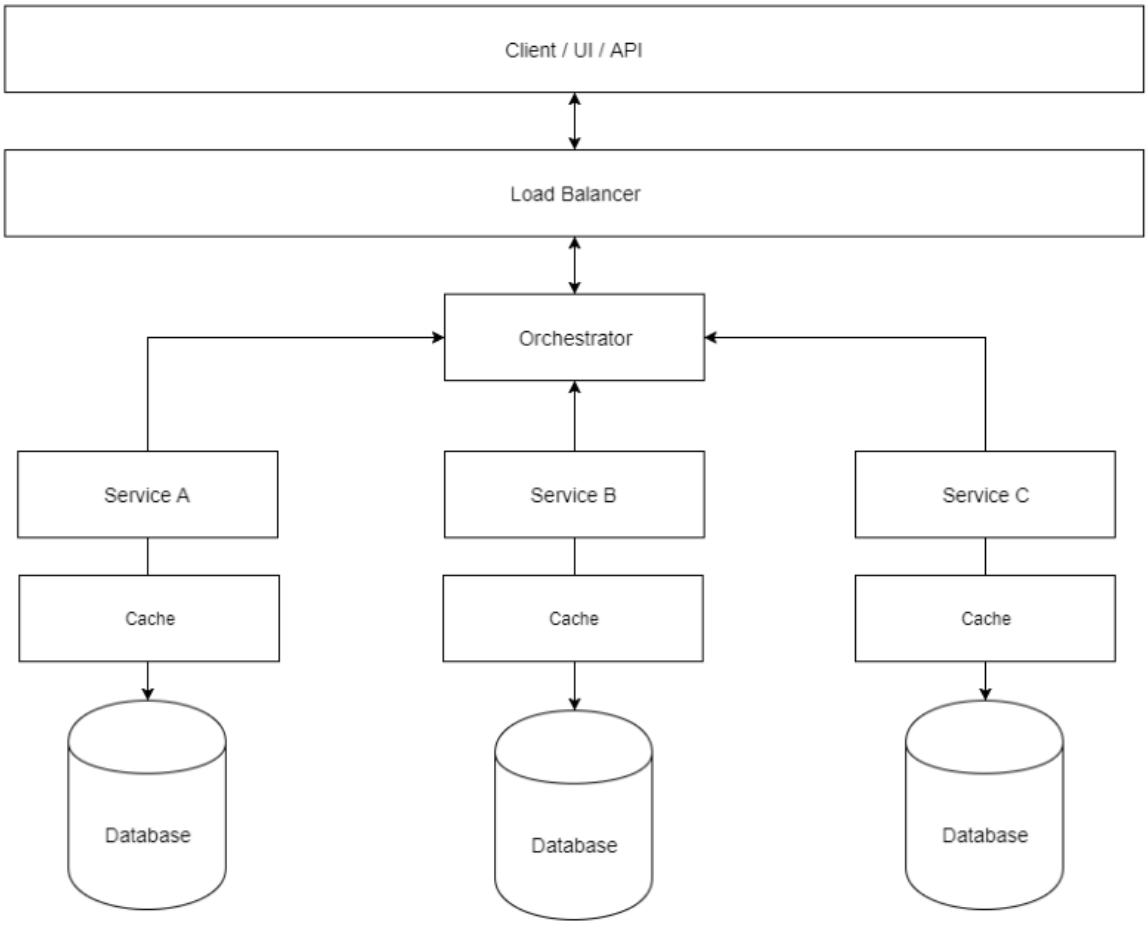


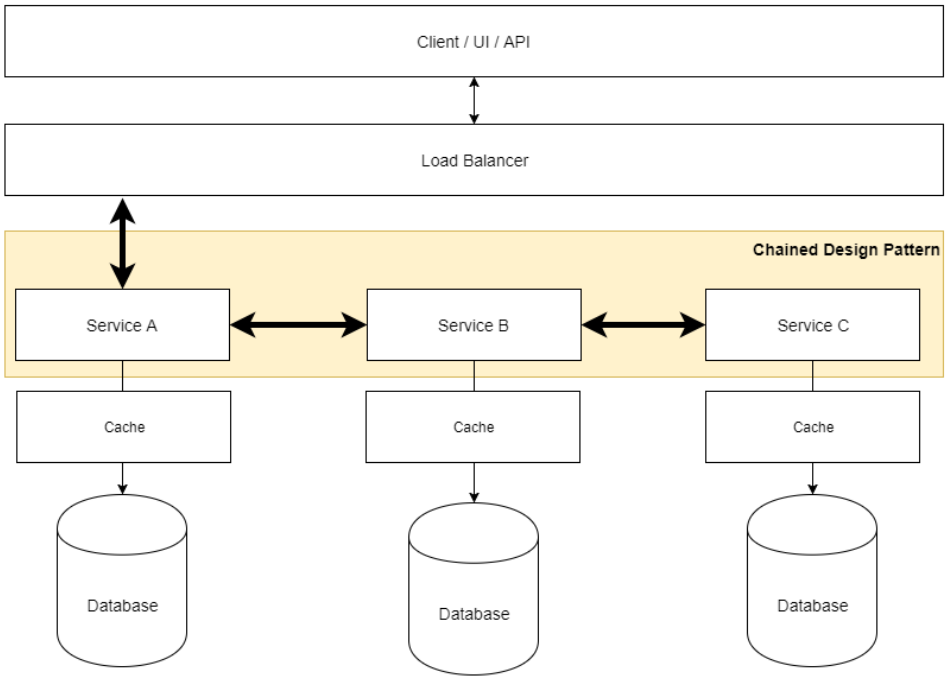

Client / UI / API
Proxy / Load Balancer
Proxy Design Pattern
Aggregator Design Pattern
NewsOrchestrator
UsersService
FamousNewsService
SportsNewsService
PoliticsNewsService
Cache
Database
Database
Database
Database

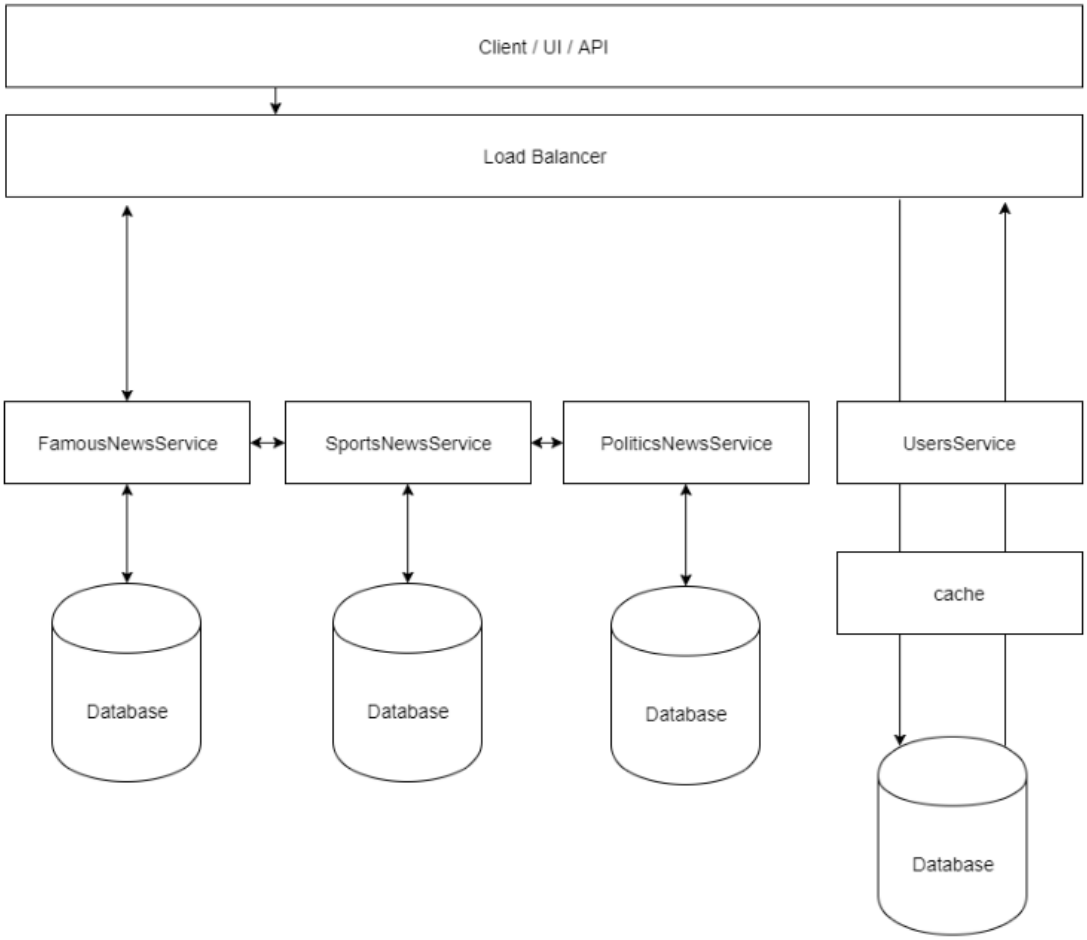

Client / UI / API
Request to Route 1
Proxy
Route 1
Route 2
Service A
Service B
Monolithic
Cache
Cache
Database
Database
Database

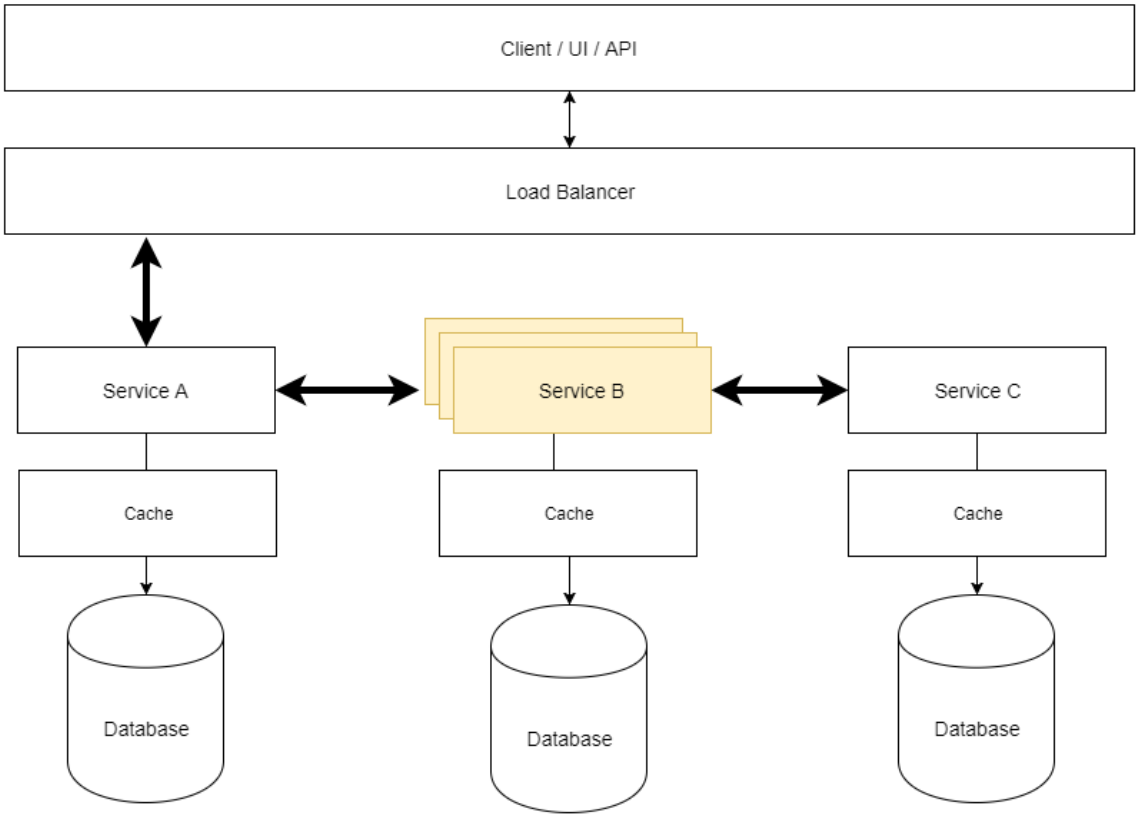

Client / UI / API
Proxy / Load Balancer
NewsOrchestrator
UsersService
FamousNewsService
SportsNewsService
PoliticsNewsService
Cache
Database
Database
Database
Database

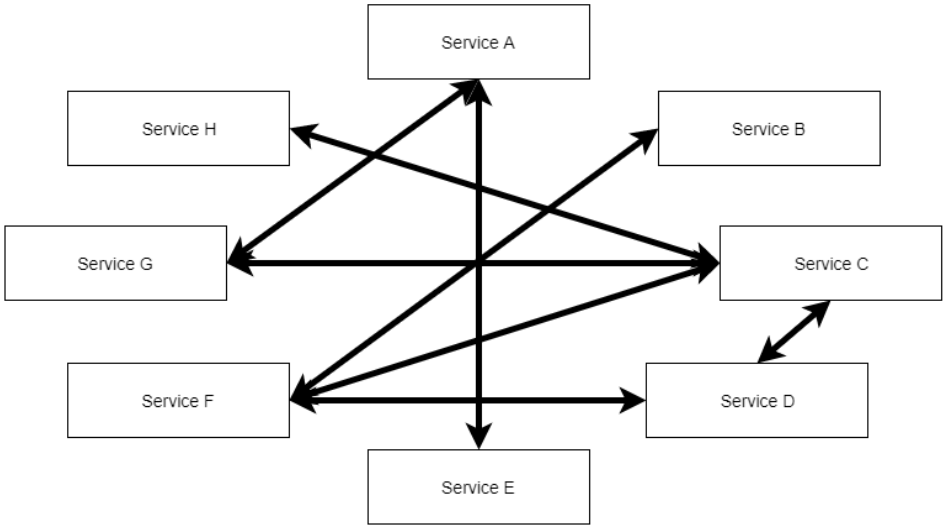
Chapter 8: Chained Microservice Design Pattern



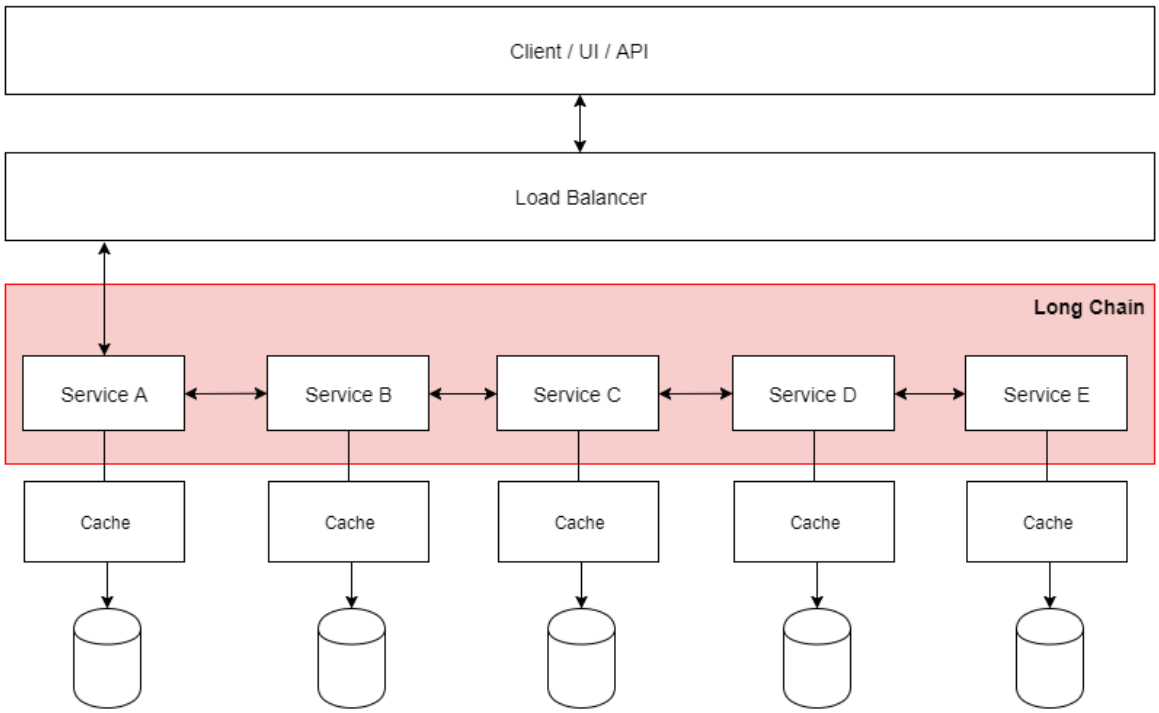

Client / UI / API
Load Balancer
Chained Design Pattern
Service A
Service B
Service C
Cache
Cache
Cache
Database
Database
Database

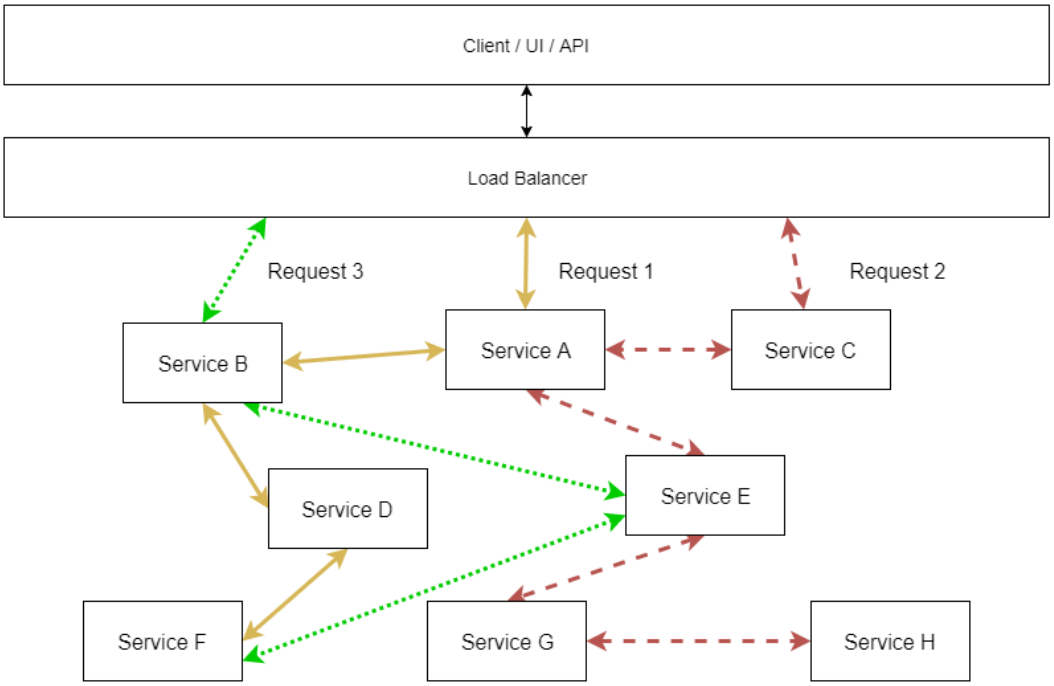

Client / UI / API
Load Balancer
FamousNewsService
SportsNewsService
PoliticsNewsService
UsersService
cache
Database
Database
Database
Database

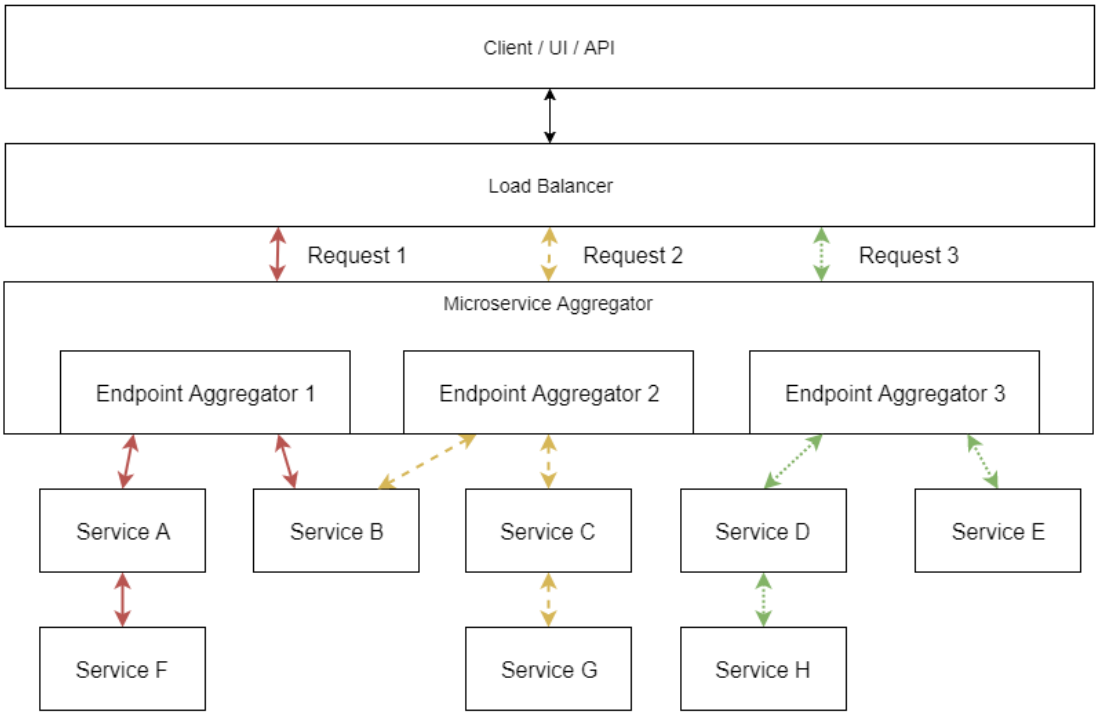

Client / UI / API
Load Balancer
Service A
Service B
Service C
Cache
Cache
Cache
Database
Database
Database

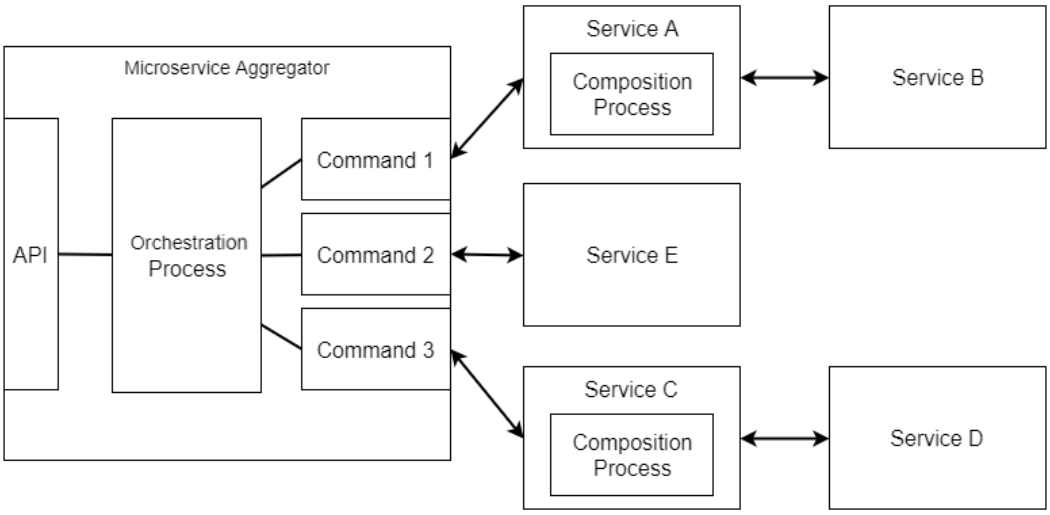

Service A
Service H
Service B
Service G
Service C
Service F
Service D
Service E

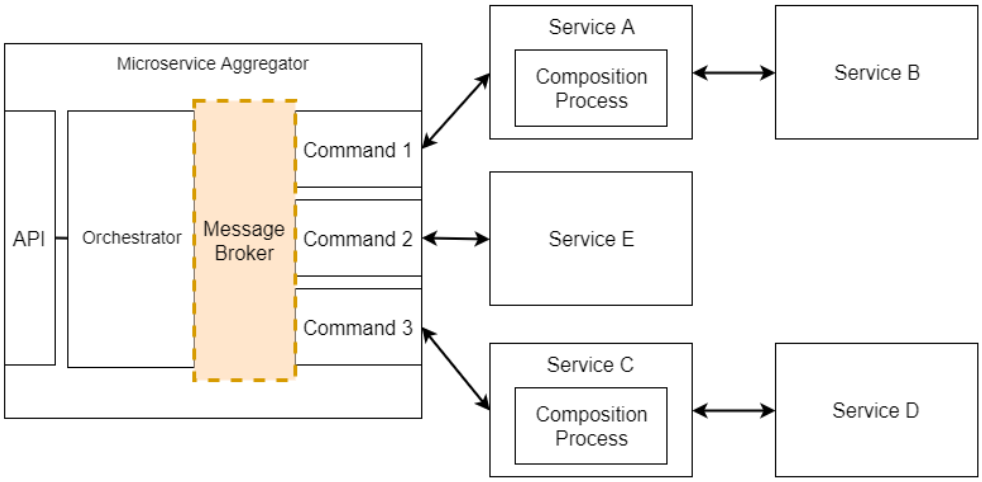
Chapter 9: Branch Microservice Design Pattern

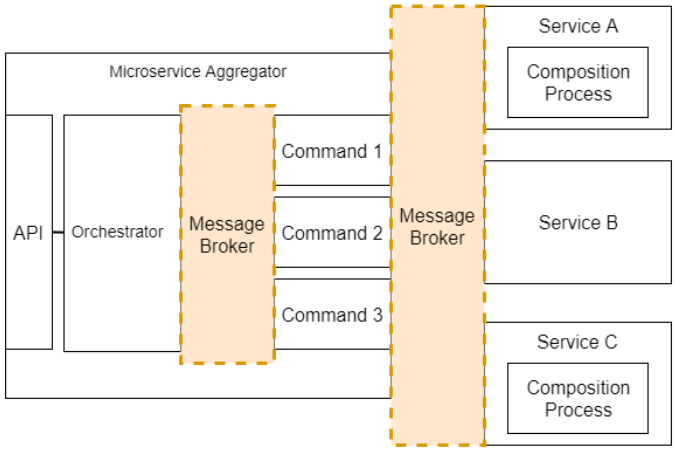




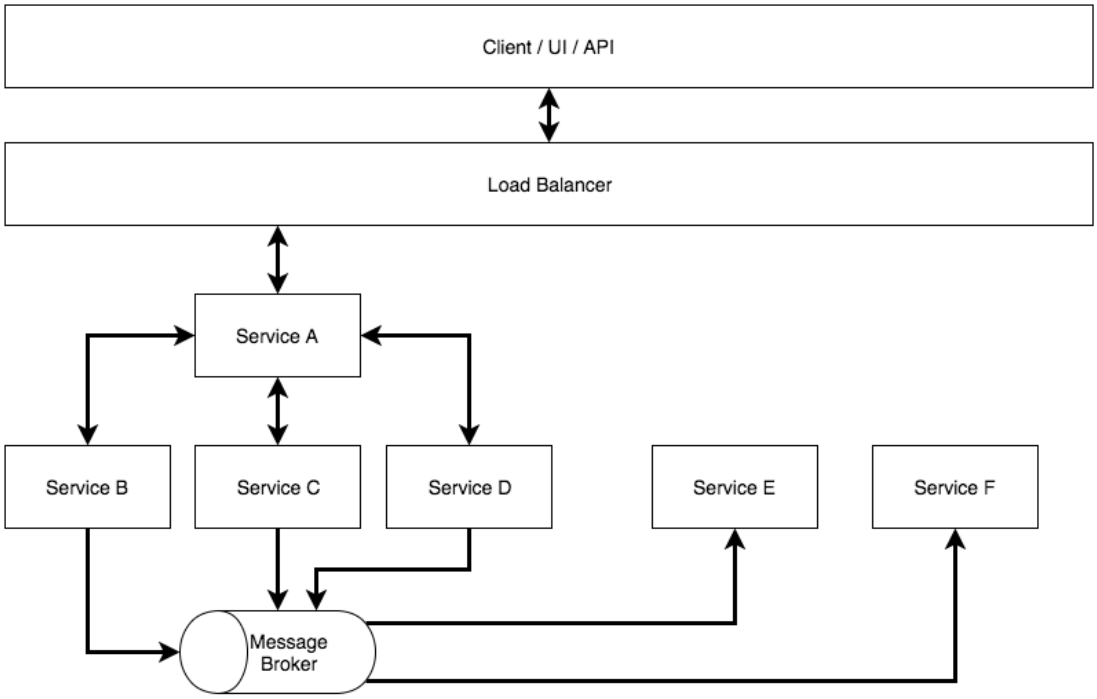

Client / UI / API
Load Balancer
Request 1
Request 2
Request 3
Microservice Aggregator
Endpoint Aggregator 1
Endpoint Aggregator 2
Endpoint Aggregator 3
Service A
Service B
Service C
Service D
Service E
Service F
Service G
Service H

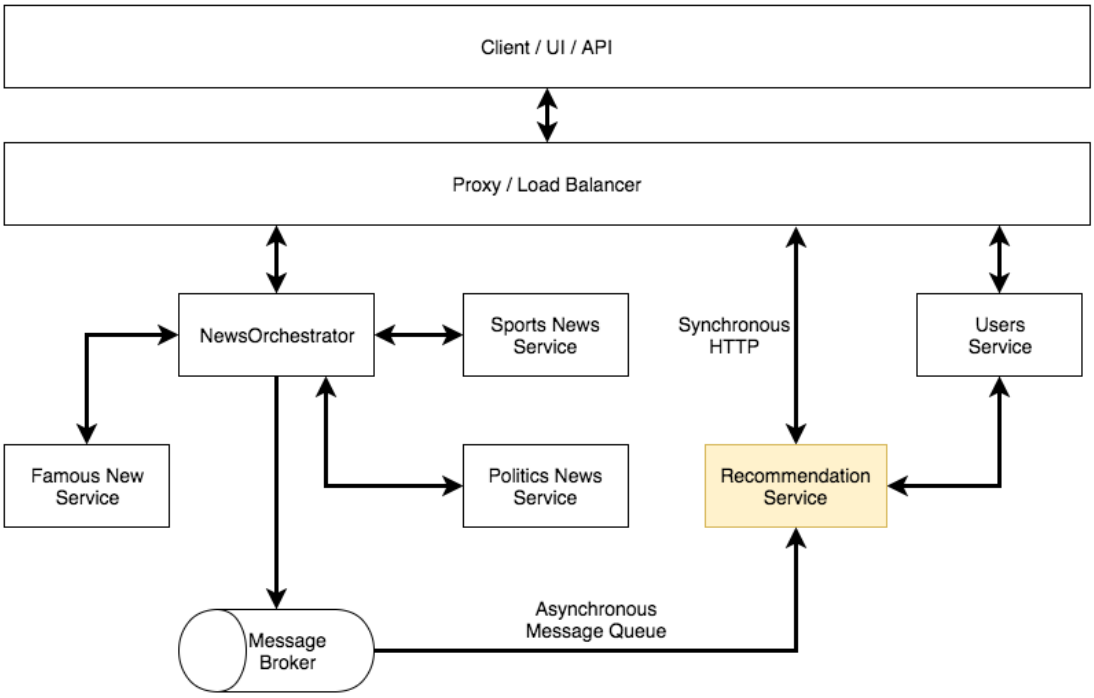

Microservice Aggregator
API
Orchestration Process
Command 1
Command 2
Command 3
Service A
Composition Process
Service B
Service E
Service C
Composition Process
Service D

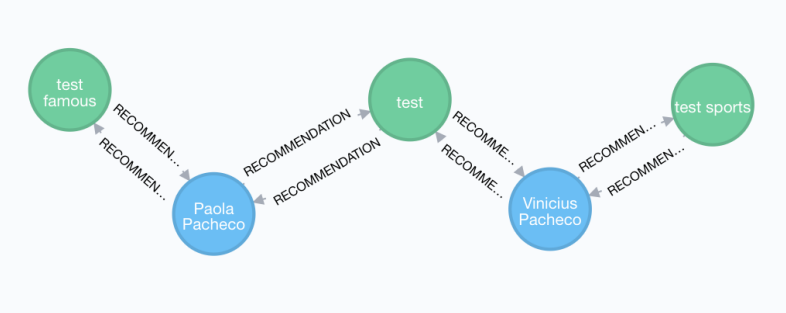

Microservice Aggregator
API
Orchestrator
Message Broker
Command 1
Command 2
Command 3
Service A
Composition Process
Service B
Service E
Service C
Composition Process
Service D

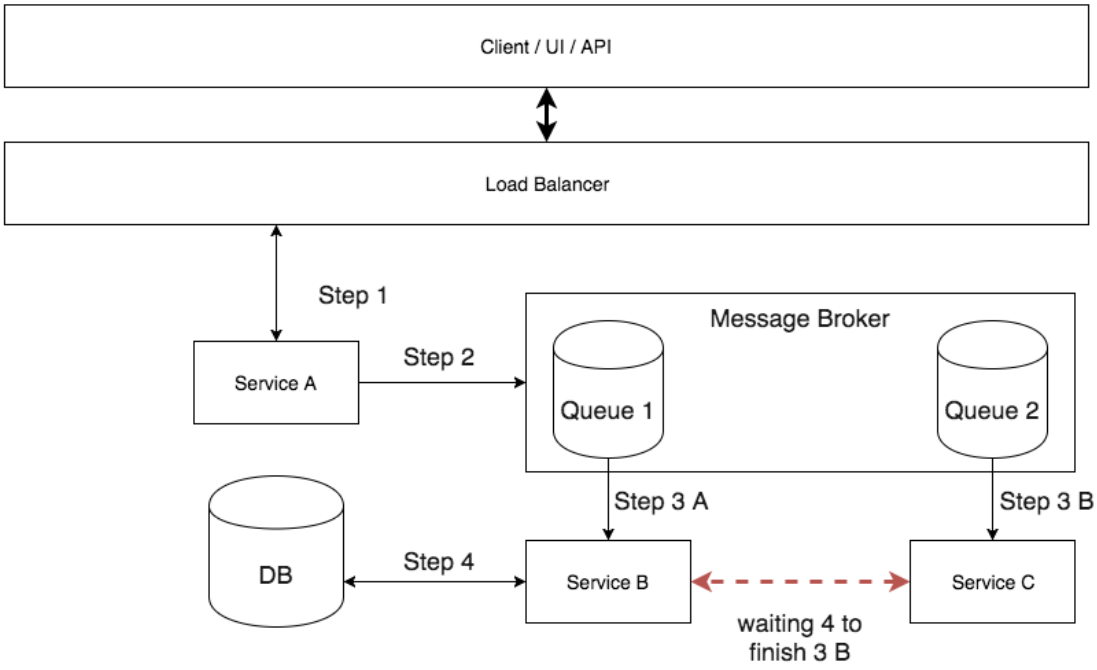

Microservice Aggregator
API
Orchestrator
Message Broker
Command 1
Command 2
Command 3
Message Broker
Service A
Composition Process
Service B
Service C
Composition Process

Chapter 10: Asynchronous Messaging Microservice

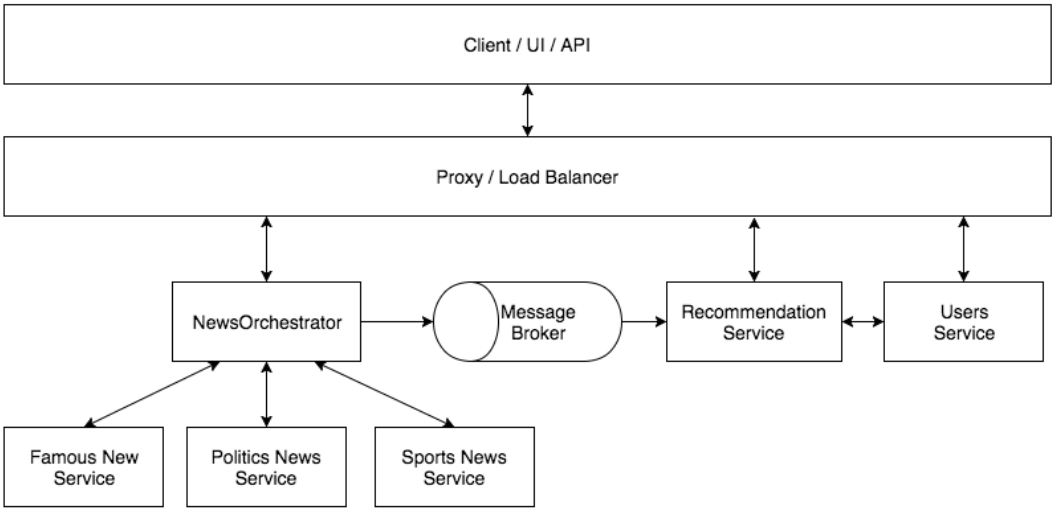


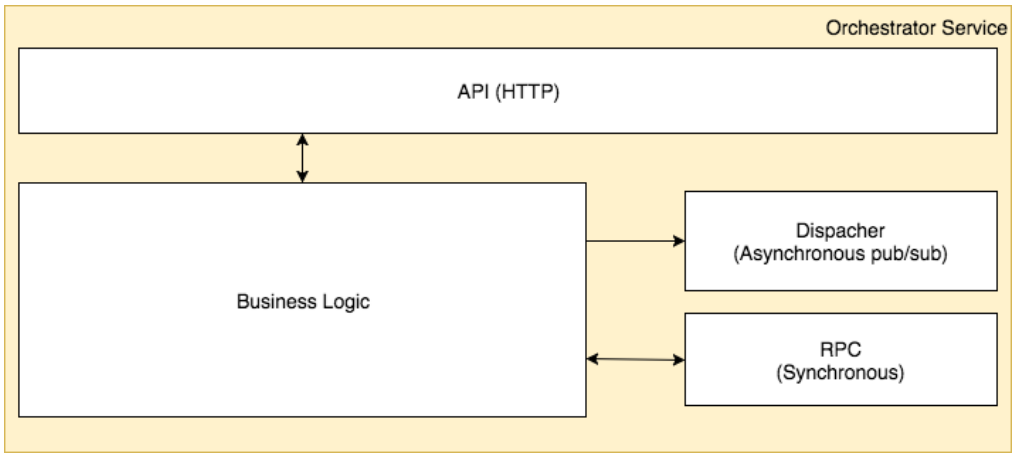

Client / UI / API
Proxy / Load Balancer
NewsOrchestrator
Sports News Service
Synchronous HTTP
Users Service
Famous New Service
Politics News Service
Recommendation Service
Message Broker
Asynchronous Message Queue

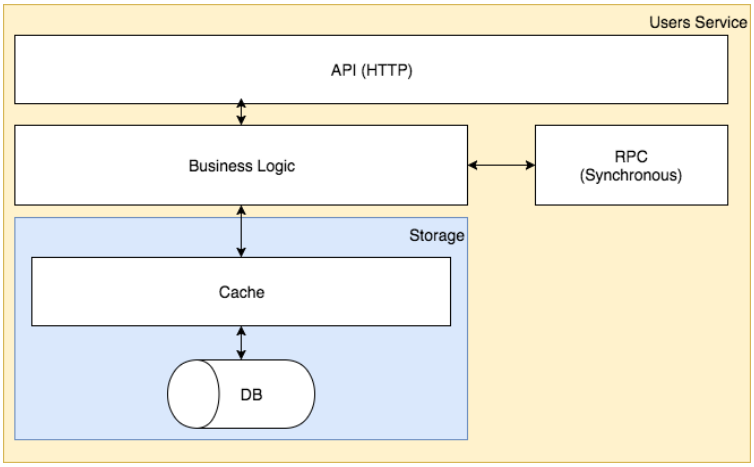

test famous
RECOMMEN...
RECOMMEN...
Paola Pacheco
RECOMMENDATION
RECOMMENDATION
test
RECOMME...
RECOMME...
Vinicius Pacheco
RECOMMEN...
RECOMMEN...
test sports

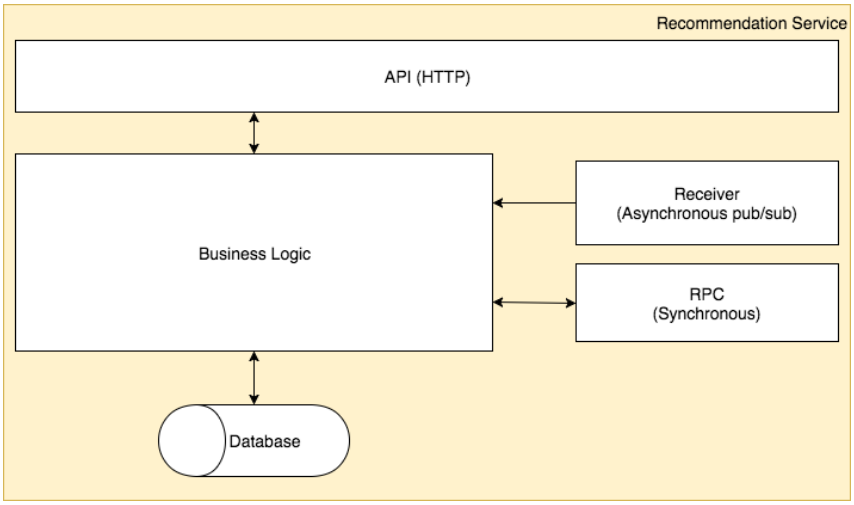

Client / UI / API
Load Balancer
Step 1
Service A
Step 2
Message Broker
Queue 1
Queue 2
Step 3 A
Step 3 B
DB
Step 4
Service B
Service C
waiting 4 to
finish 3 B

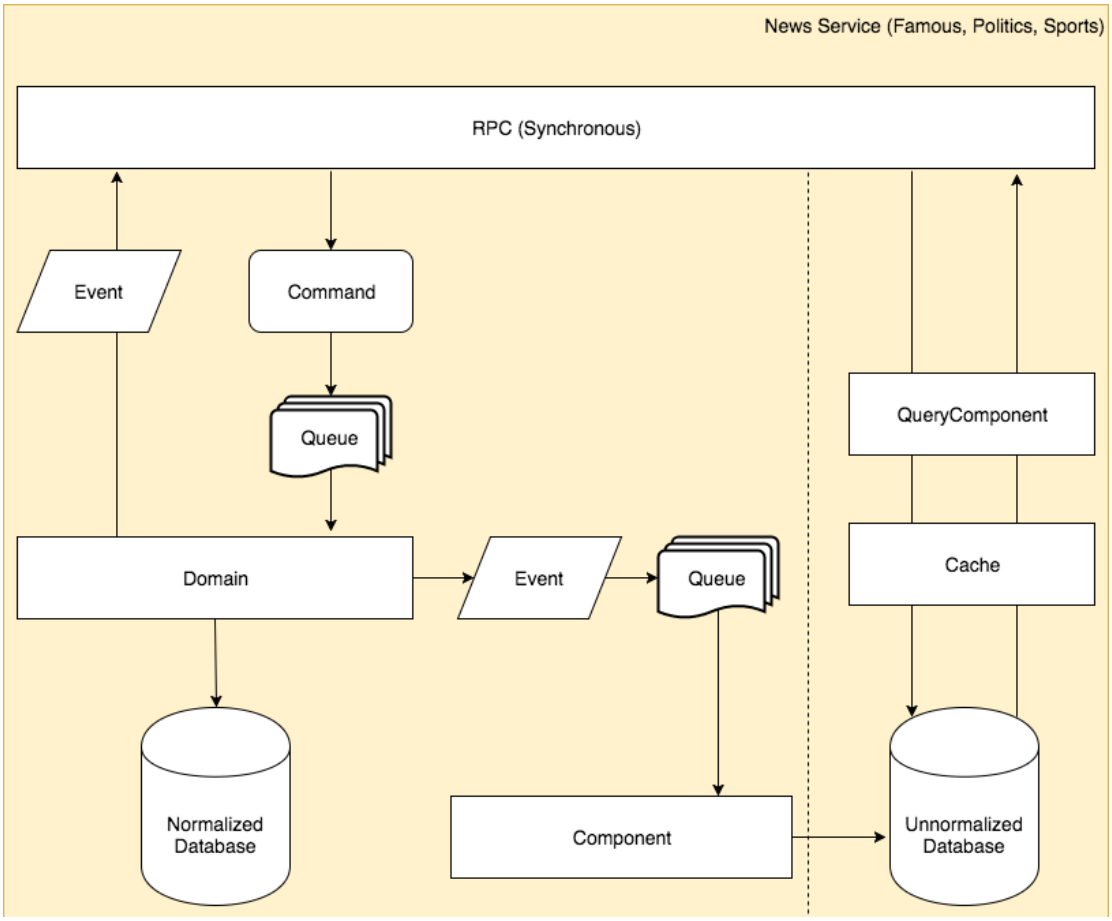
Chapter 11: Microservices Working Together

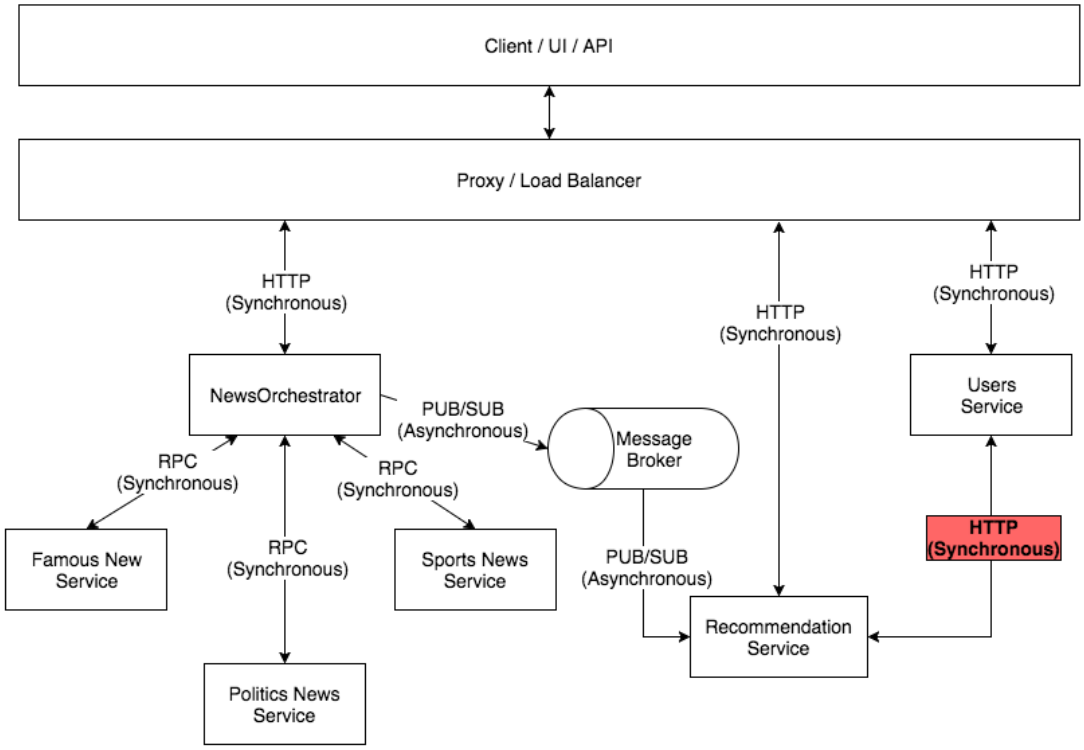


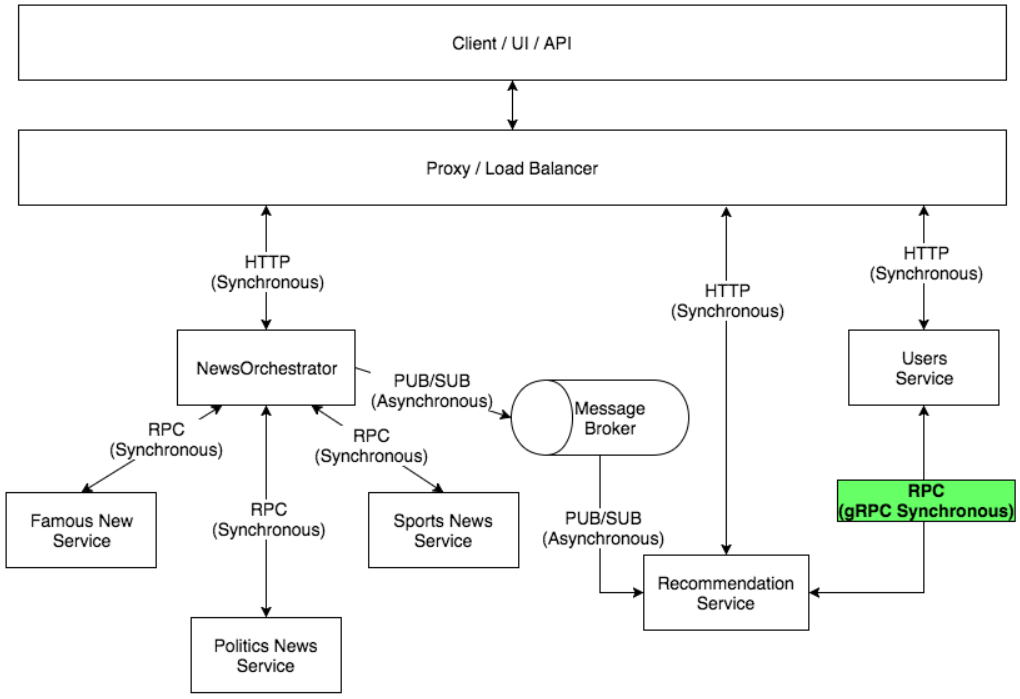


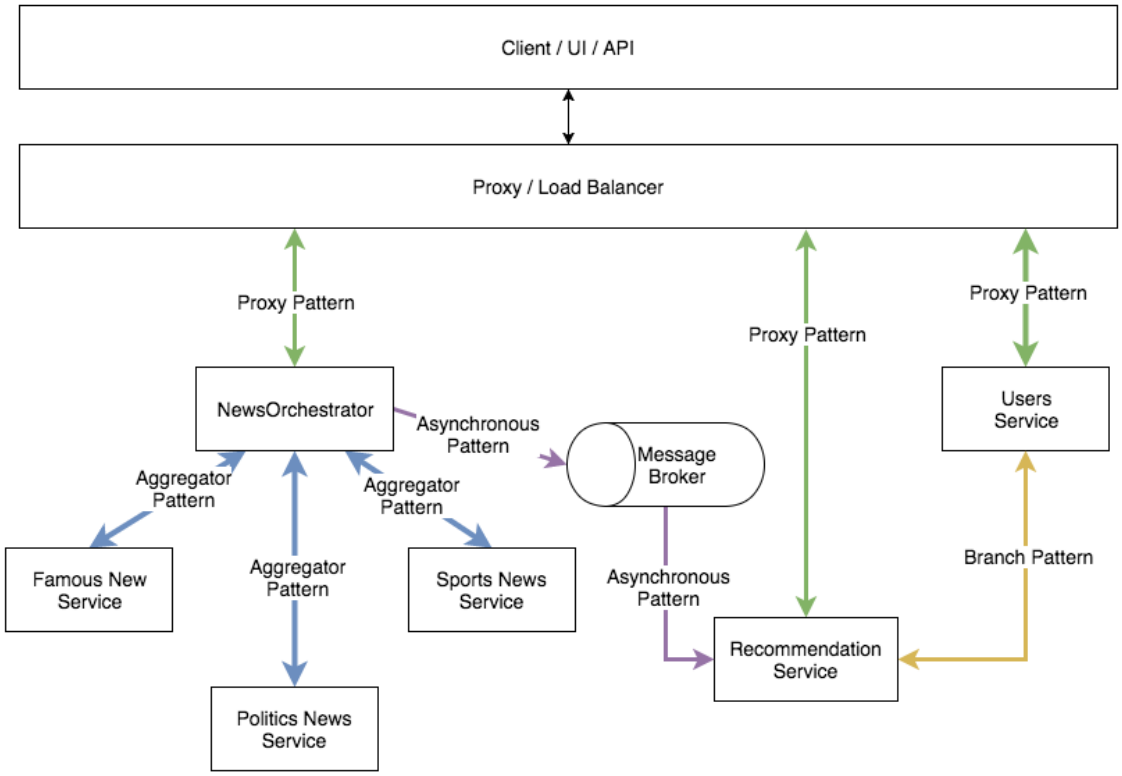

Users Service
API (HTTP)
Business Logic
RPC
(Synchronous)
Storage
Cache
DB


Recommendation Service
API (HTTP)
Business Logic
Receiver
(Asynchronous pub/sub)
RPC
(Synchronous)
Database

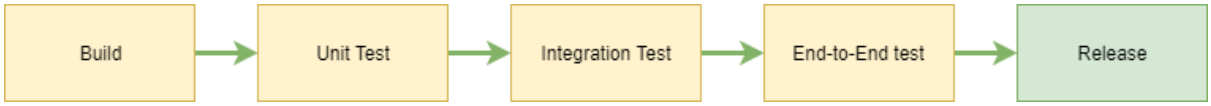

News Service (Famous, Politics, Sports)
RPC (Synchronous)
Event
Command
Queue
QueryComponent
Domain
Event
Queue
Cache
Normalized Database
Component
Unnormalized Database

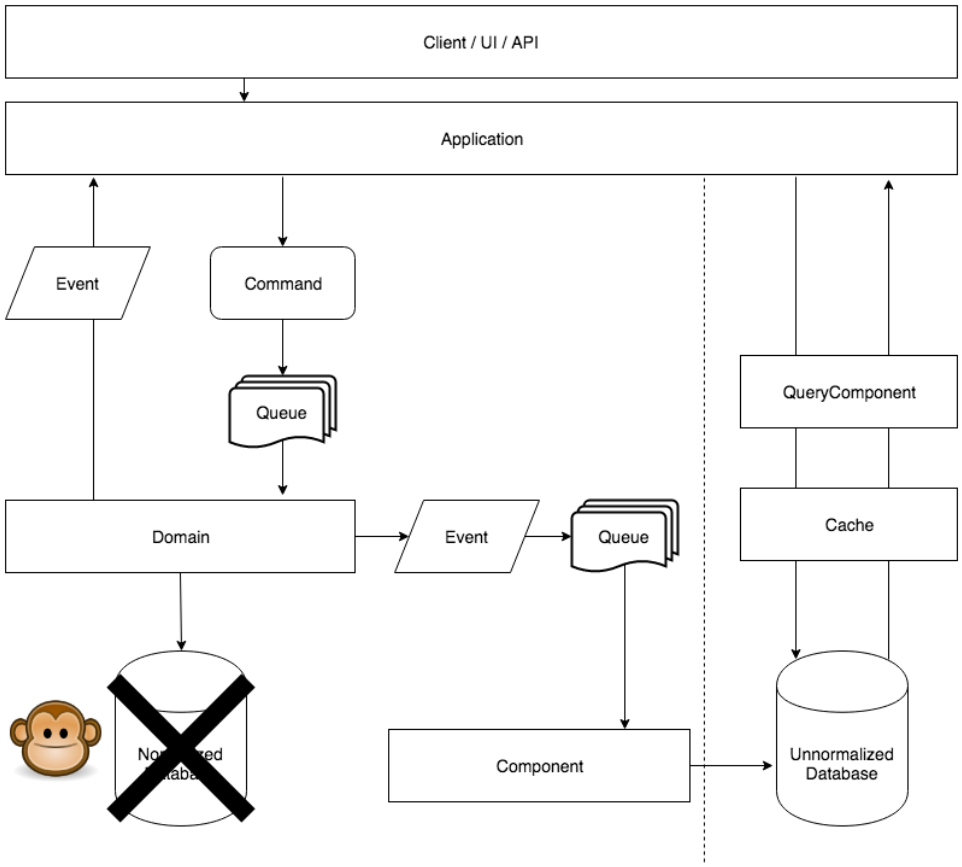

Client / UI / API
Proxy / Load Balancer
HTTP
(Synchronous)
HTTP
(Synchronous)
HTTP
(Synchronous)
NewsOrchestrator
PUB/SUB
(Asynchronous)
Message
Broker
Users
Service
RPC
(Synchronous)
RPC
(Synchronous)
Famous New
Service
RPC
(Synchronous)
Sports News
Service
PUB/SUB
(Asynchronous)
HTTP
(Synchronous)
Recommendation
Service
Politics News
Service


Client / UI / API
Proxy / Load Balancer
HTTP
(Synchronous)
HTTP
(Synchronous)
HTTP
(Synchronous)
NewsOrchestrator
PUB/SUB
(Asynchronous)
Message
Broker
Users
Service
RPC
(Synchronous)
RPC
(Synchronous)
Famous New
Service
RPC
(Synchronous)
Sports News
Service
PUB/SUB
(Asynchronous)
RPC
(gRPC Synchronous)
Recommendation
Service
Politics News
Service

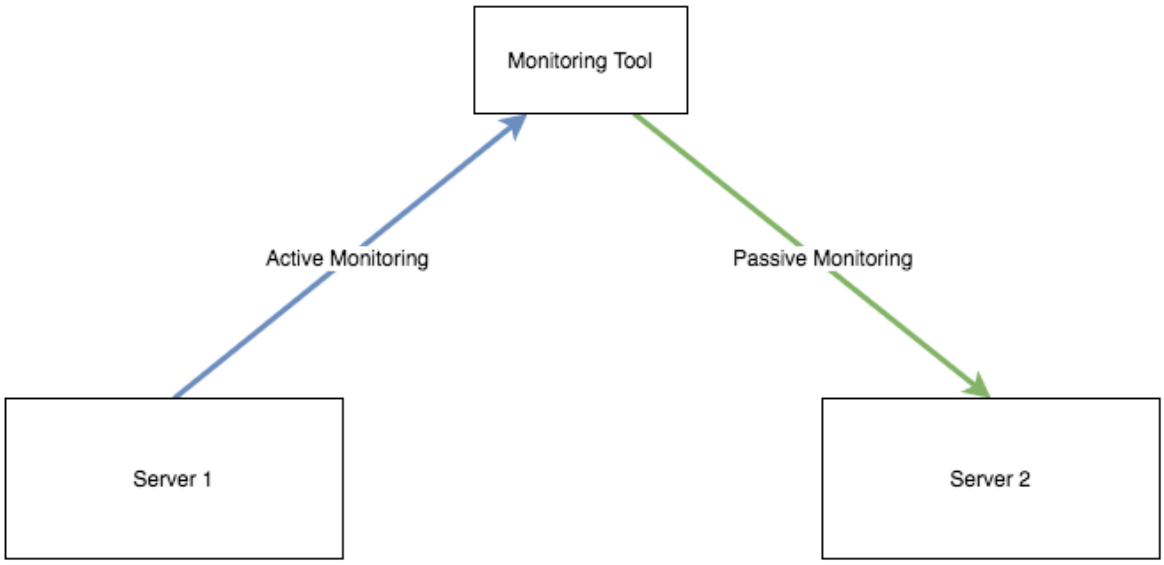

Client / UI / API
Proxy / Load Balancer
Proxy Pattern
Proxy Pattern
Proxy Pattern
NewsOrchestrator
Asynchronous Pattern
Message Broker
Users Service
Aggregator Pattern
Aggregator Pattern
Aggregator Pattern
Famous New Service
Sports News Service
Politics News Service
Asynchronous Pattern
Recommendation Service
Branch Pattern

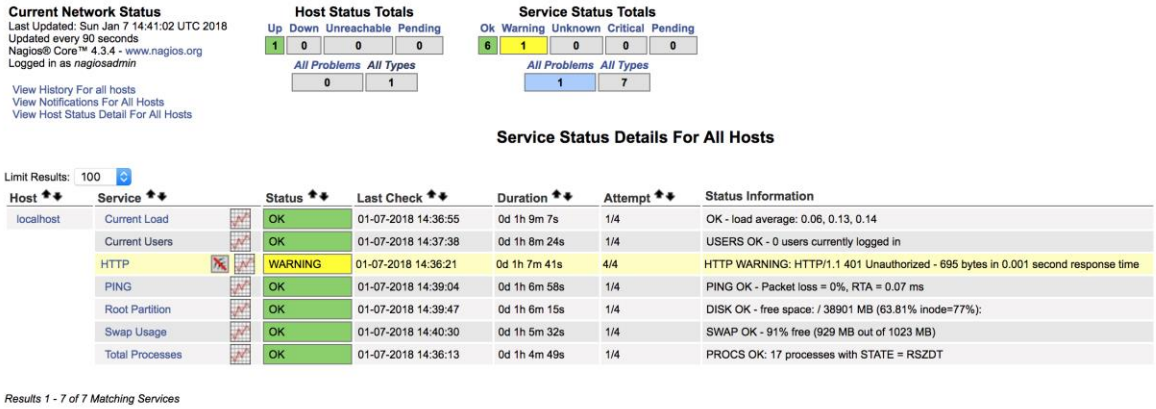
Chapter 12: Testing Microservices





Chapter 13: Monitoring Security and Deployment





Current Network Status
Last Updated: Sun Jan 7 14:41:02 UTC 2018
Updated every 90 seconds
Nagios® Core™ 4.3.4 - www.nagios.org
Logged in as *nagiosadmin*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

All Problems	All Types
0	1

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
6	1	0	0	0

All Problems	All Types
1	7

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	01-07-2018 14:36:55	0d 1h 9m 7s	1/4	OK - load average: 0.06, 0.13, 0.14
	Current Users	OK	01-07-2018 14:37:38	0d 1h 8m 24s	1/4	USERS OK - 0 users currently logged in
	HTTP	WARNING	01-07-2018 14:36:21	0d 1h 7m 41s	4/4	HTTP WARNING: HTTP/1.1 401 Unauthorized - 695 bytes in 0.001 second response time
	PING	OK	01-07-2018 14:39:04	0d 1h 6m 58s	1/4	PING OK - Packet loss = 0%, RTA = 0.07 ms
	Root Partition	OK	01-07-2018 14:39:47	0d 1h 6m 15s	1/4	DISK OK - free space: / 38901 MB (63.81% inode=77%):
	Swap Usage	OK	01-07-2018 14:40:30	0d 1h 5m 32s	1/4	SWAP OK - 91% free (929 MB out of 1023 MB)
	Total Processes	OK	01-07-2018 14:36:13	0d 1h 4m 49s	1/4	PROCS OK: 17 processes with STATE = RSZDT

Results 1 - 7 of 7 Matching Services

Service Information
Last Updated: Sun Jan 7 14:42:51 UTC 2018
Updated every 90 seconds
Nagios® Core™ 4.3.4 - www.nagios.org
Logged in as *nagiosadmin*

View Information For This Host
View Status Detail For This Host
View Alert History For This Service
View Trends For This Service
View Alert Histogram For This Service
View Availability Report For This Service
View Notifications For This Service

Service
PING

On Host
localhost
(localhost)

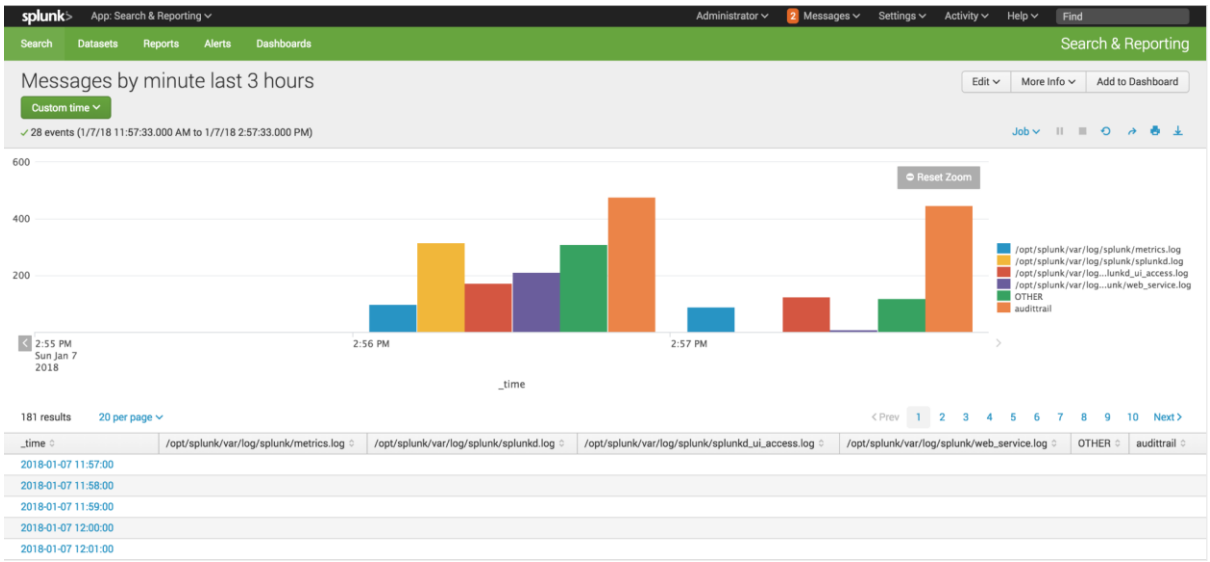
Member of
No servicegroups.

127.0.0.1

Service State Information

Current Status:	**OK** (for 0d 1h 8m 47s)
Status Information:	PING OK - Packet loss = 0%, RTA = 0.07 ms
Performance Data:	rta=0.066000ms;100.000000;500.000000;0.000000 pl=0%;20;60;0
Current Attempt:	1/4 (HARD state)
Last Check Time:	01-07-2018 14:39:04
Check Type:	ACTIVE
Check Latency / Duration:	0.001 / 4.158 seconds
Next Scheduled Check:	01-07-2018 14:44:04
Last State Change:	01-07-2018 13:34:04
Last Notification:	N/A (notification 0)
Is This Service Flapping?	**NO** (0.00% state change)
In Scheduled Downtime?	**NO**
Last Update:	01-07-2018 14:42:43 (0d 0h 0m 8s ago)

Active Checks:	**ENABLED**
Passive Checks:	**ENABLED**
Obsessing:	**ENABLED**
Notifications:	**ENABLED**
Event Handler:	**ENABLED**
Flap Detection:	**ENABLED**



Vinicius Feitosa Pacheco / Go

Issues Overview User Feedback Releases

Star Project | Project Settings

Unresolved Issues (2)

Sort by: Last Seen | is:unresolved

Resolve | Ignore | GRAPH: 24H 14D | EVENTS | USERS

Error ***os.PathError** app/vendor/github.com/getsentry/raven-go in CaptureErrorAndWait
open filename.ext: no such file or directory
GO-2 | 26 minutes ago — 26 minutes old | root

1 0

API Docs Contribute

---

*os.PathError app/vendor/github.com/getsentry/raven-go in CaptureErrorAnd...

open filename.ext: no such file or directory root

ISSUE # GO-2 | ASSIGNED | EVENTS 1 | USERS 0

Resolve | Ignore | Share | Link Issue Tracker

Details Comments 0 User Feedback 0 Tags Events Merged Similar Issues

Event c2d446a24a3a47a9a2ebb0b13b6c5369
Jan 7, 2018 4:59:09 PM UTC | JSON (10.8 KB)

Older | Newer

TAGS

level error | logger root | server_name 9079b78d8a10

MESSAGE

```
open filename.ext: no such file or directory
```

EXCEPTION (most recent call first) Full | Raw

***os.PathError**

```
open filename.ext: no such file or directory
```

(Default Environment)

LAST 24 HOURS

LAST 30 DAYS

FIRST SEEN

When: 29 minutes ago
Jan 7, 2018 4:59:09 PM UTC

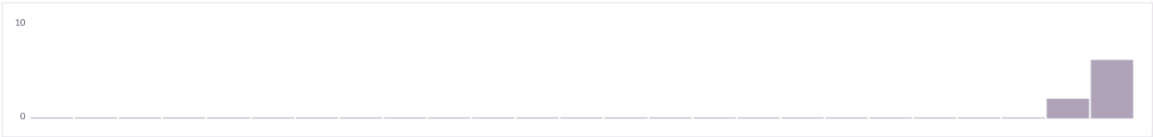
Release: *not configured*

LAST SEEN

When: 29 minutes ago
Jan 7, 2018 4:59:09 PM UTC

Overview

1 hour | 1 day | 1 week

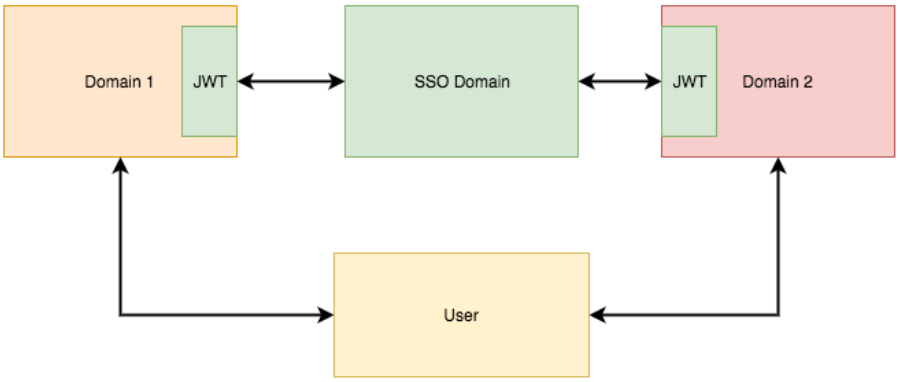


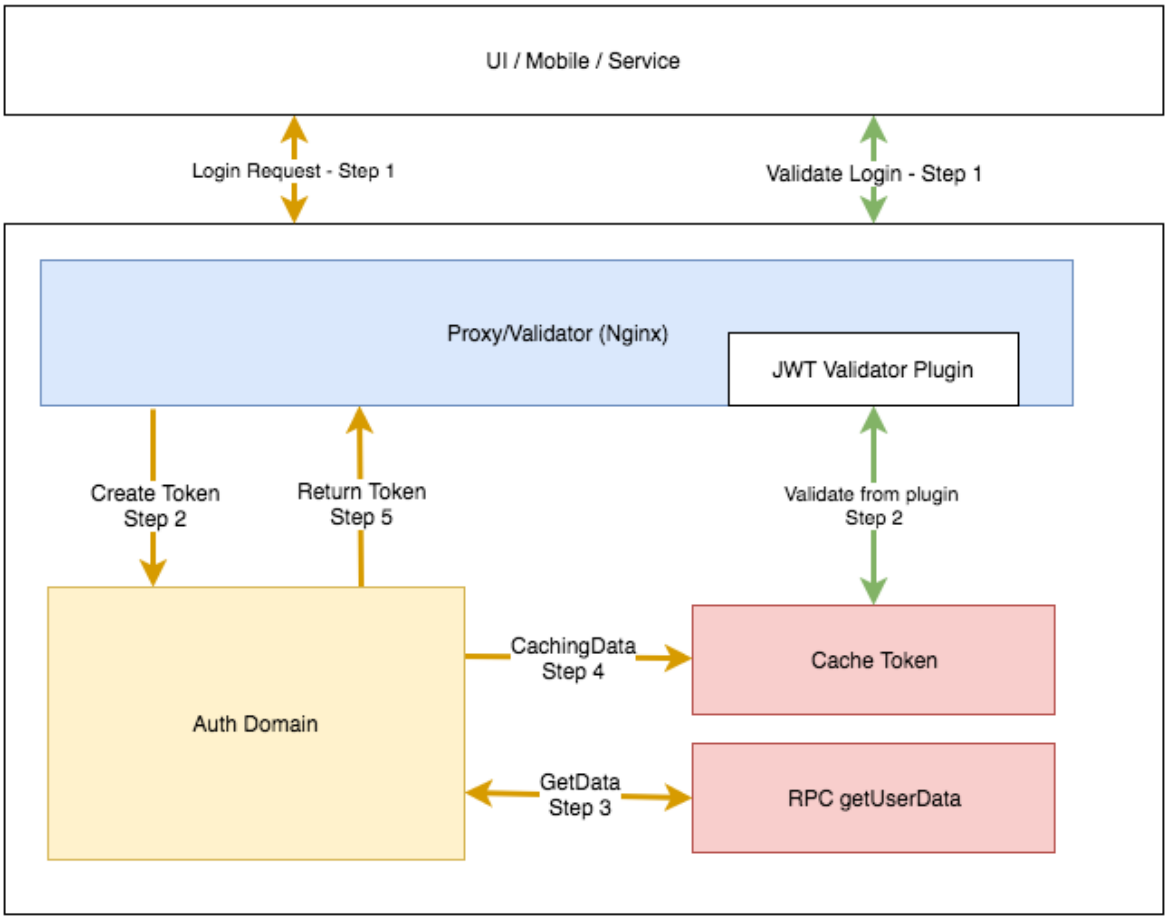
January 6, 2018

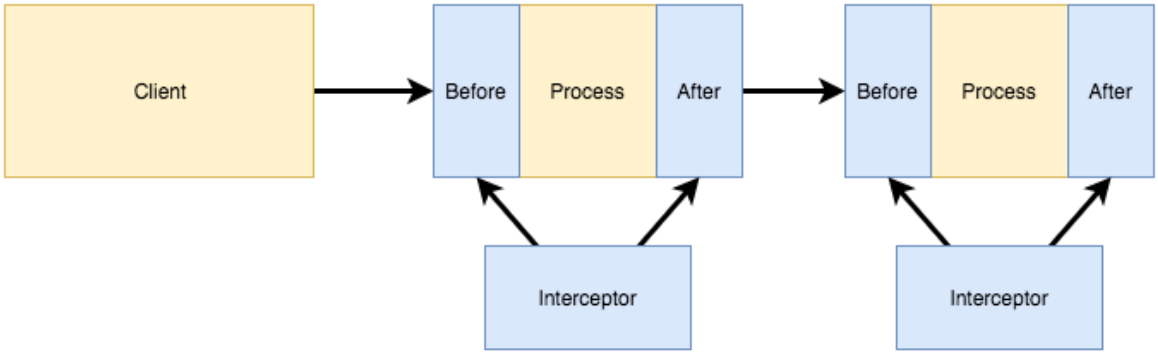
TRENDING ISSUES	EVENTS	USERS
***os.PathError** app/vendor/github.com/getsentry/raven-go ... open filename.ext: no such file or directory a minute ago — 31 minutes old	7	0

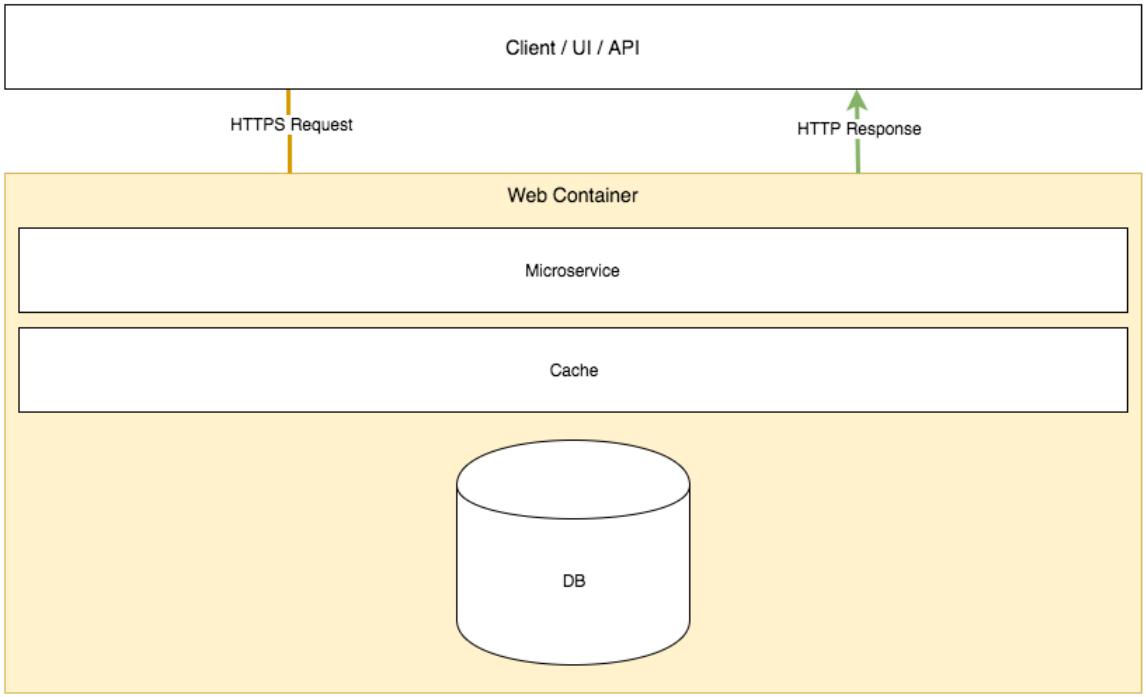
NEW ISSUES	EVENTS	USERS
***os.PathError** app/vendor/github.com/getsentry/raven-go ... open filename.ext: no such file or directory a minute ago — 31 minutes old	7	0

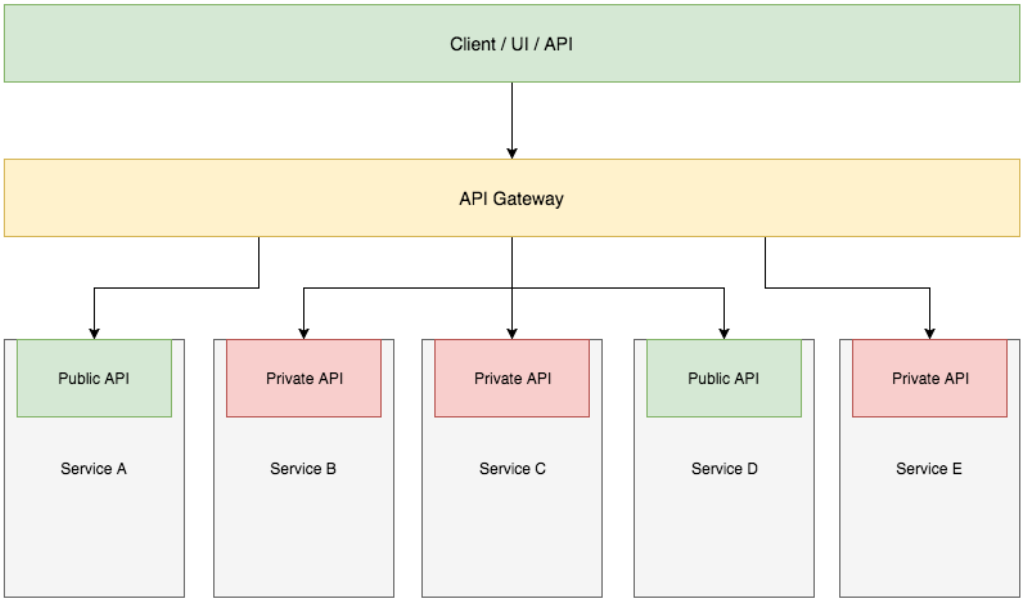
API Docs Contribute

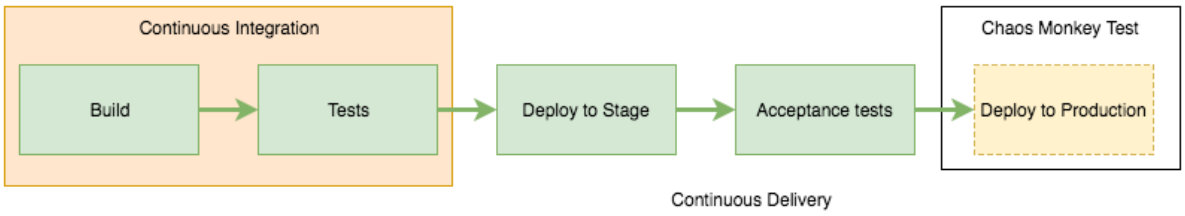

Domain 1
JWT
SSO Domain
JWT
Domain 2
User


UI / Mobile / Service
Login Request - Step 1
Validate Login - Step 1
Proxy/Validator (Nginx)
JWT Validator Plugin
Create Token
Step 2
Return Token
Step 5
Validate from plugin
Step 2
Auth Domain
CachingData
Step 4
Cache Token
GetData
Step 3
RPC getUserData

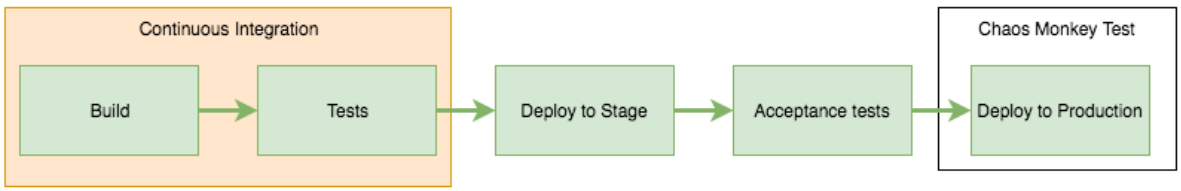

Client
Before
Process
After
Before
Process
After
Interceptor
Interceptor


Client / UI / API
HTTPS Request
HTTP Response
Web Container
Microservice
Cache
DB

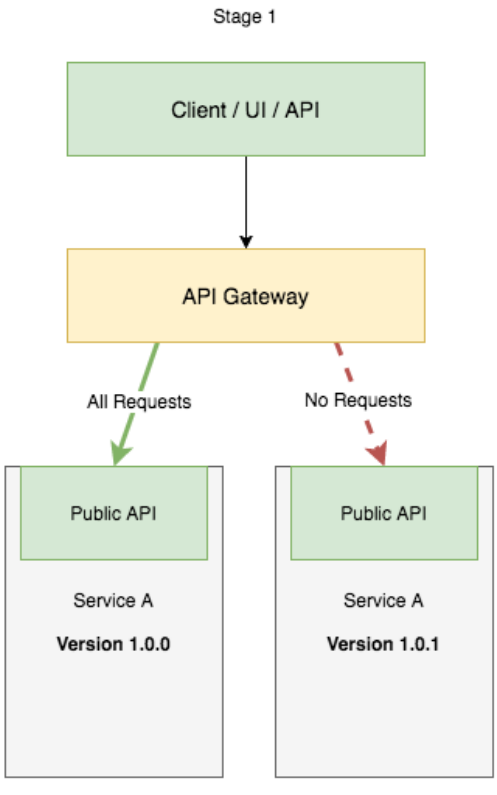


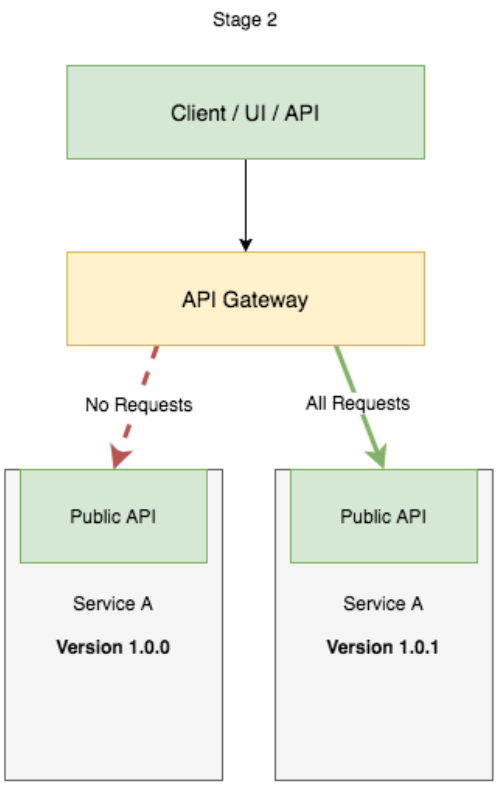


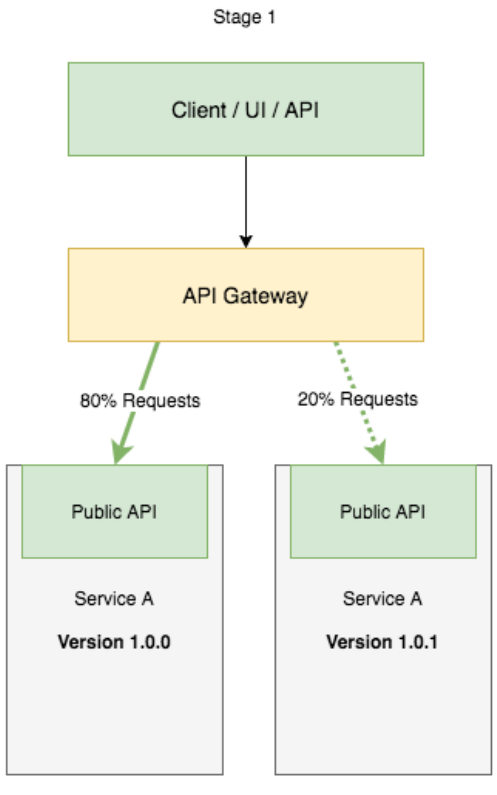
Continuous Delivery

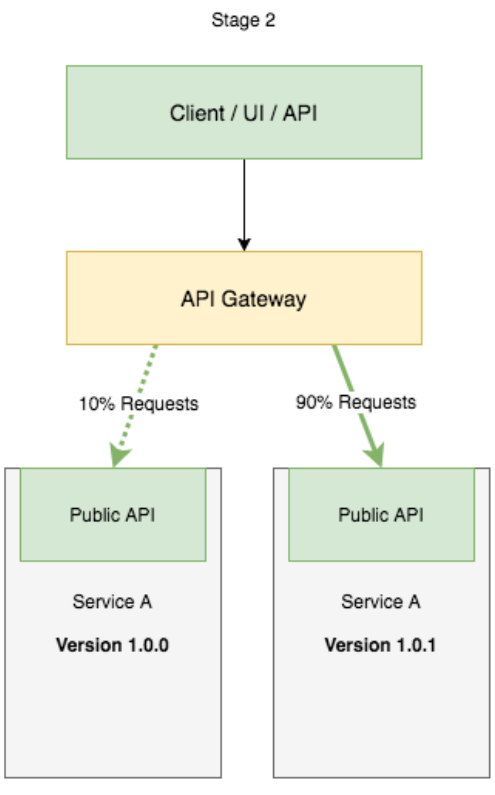


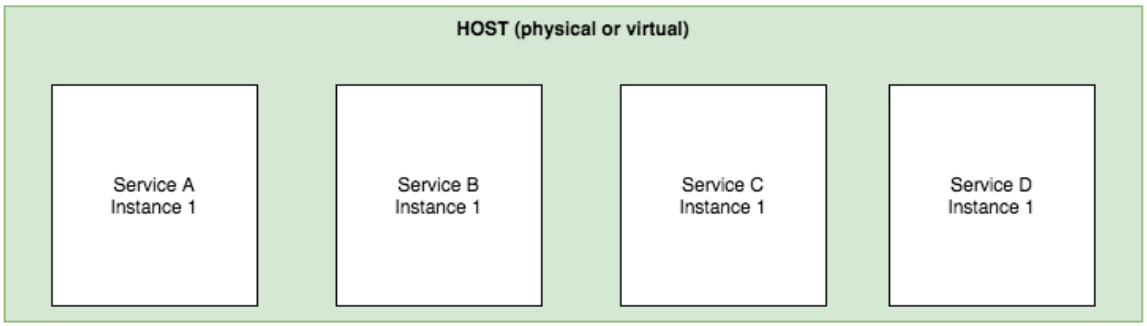
Continuous Deploy

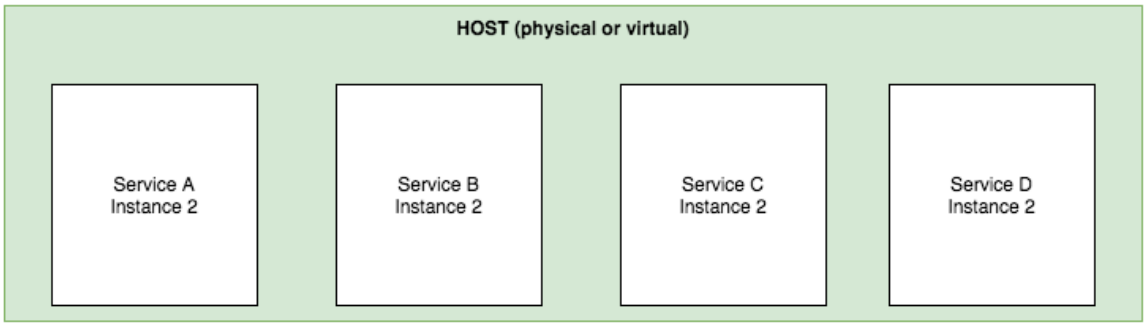

Stage 1
Client / UI / API
API Gateway
All Requests
No Requests
Public API
Service A
Version 1.0.0
Public API
Service A
Version 1.0.1

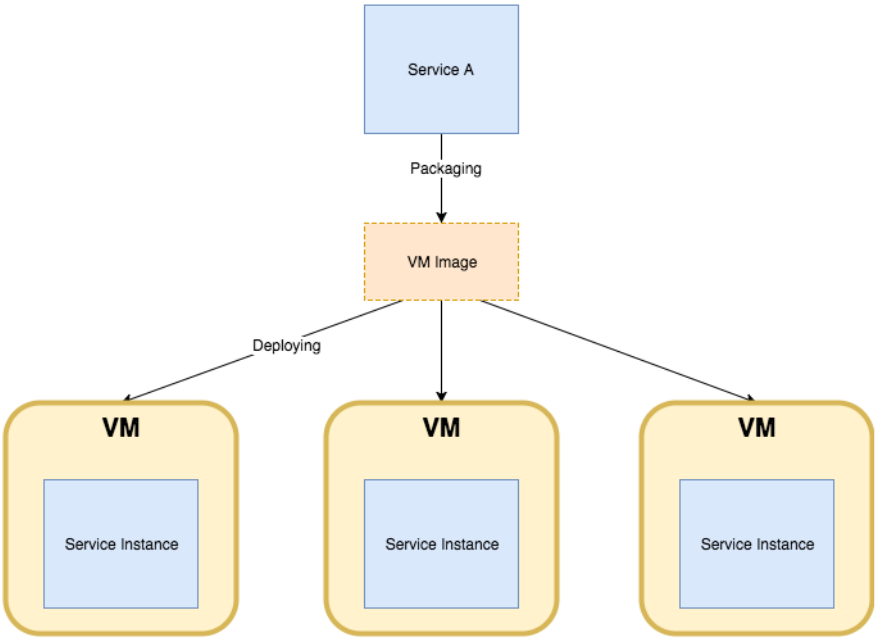

Stage 2
Client / UI / API
API Gateway
No Requests
All Requests
Public API
Service A
Version 1.0.0
Public API
Service A
Version 1.0.1

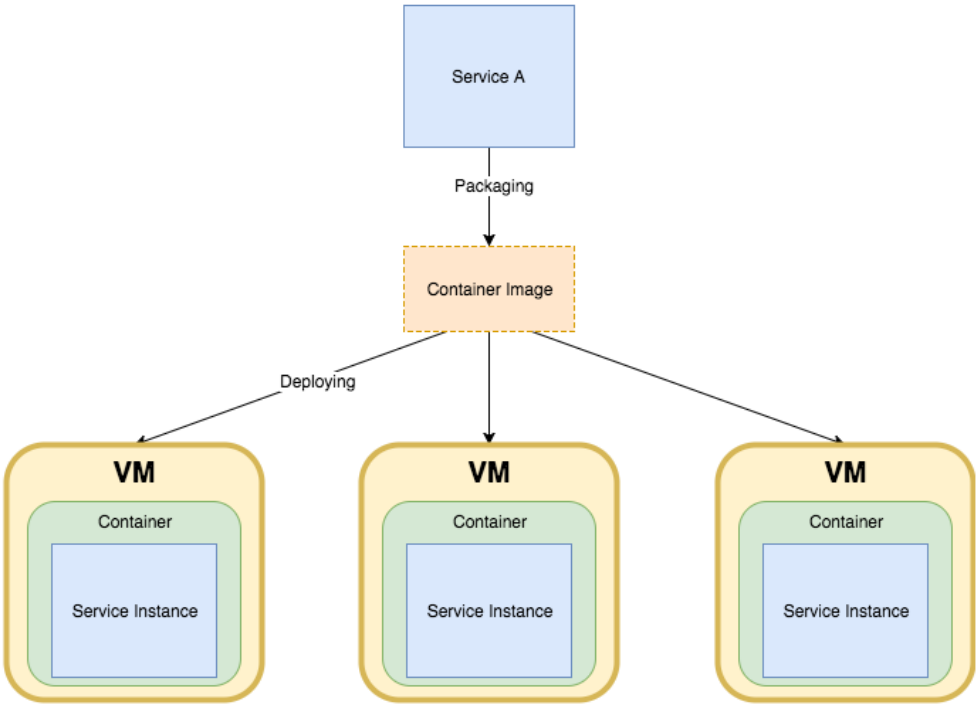

Stage 1
Client / UI / API
API Gateway
80% Requests
20% Requests
Public API
Service A
Version 1.0.0
Public API
Service A
Version 1.0.1


Stage 2
Client / UI / API
API Gateway
10% Requests
90% Requests
Public API
Service A
Version 1.0.0
Public API
Service A
Version 1.0.1


HOST (physical or virtual)
Service A
Instance 1
Service B
Instance 1
Service C
Instance 1
Service D
Instance 1


HOST (physical or virtual)
Service A
Instance 2
Service B
Instance 2
Service C
Instance 2
Service D
Instance 2


Service A
Packaging
VM Image
Deploying
VM
Service Instance
VM
Service Instance
VM
Service Instance


Service A
Packaging
Container Image
Deploying
VM
Container
Service Instance
VM
Container
Service Instance
VM
Container
Service Instance