Resilience Against Downstream Failures

Mario Fernandez

Wayfair

hceris.com/

What I Want to Cover

What I Want to Cover

Resilience Operators

What I Want to Cover

Resilience Operators

Fallback Cache

What I Want to Cover

Resilience Operators

Fallback Cache

Taking It to the Next Level

A Bit of Context

Partner Home @ Wayfair

wayfair PARTNER HOME

Your Wayfair Dashboard

Help

- Reporting
- Tickets
- Orders
- Stocking Orders
- CastleGate
- Inventory
- Products
- Premium Shelf
- Finance
- Developer
- Account Management
- Catalog
- Download Center
- Help & Support

Announcements | Previous Announcements

No Announcements! See Previous Announcements for important news you may have missed

Search PO
Enter PO Number
Search

Today's Tasks

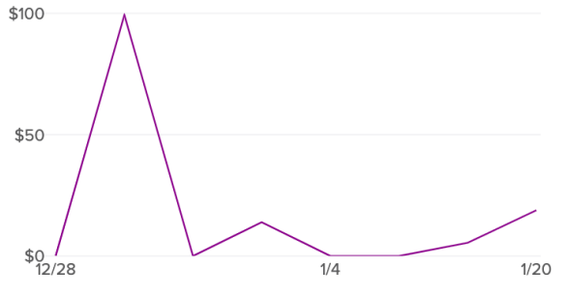
- 2 Order Alert(s)
- 13 to Ship Today
- 0 to Ship Next Business Day
- 3733 Ticket(s)

Performance(Last 30 Days)

View Sales Dashboard

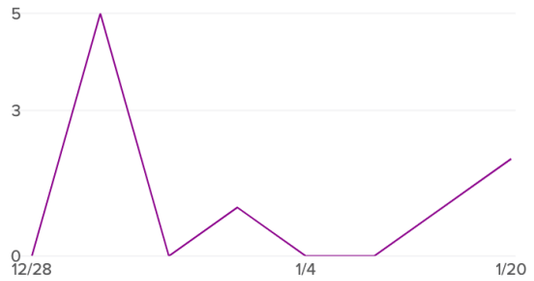
Revenue

$99.31



Units Sold

5



Quick Links

Reporting

Sales Dashboard
Forecast

Order & Operations

Order Management
Warehouse Calendar

Product Management

Add New Products
Manage Pricing
Manage Product Details
Manage Product Images
Manage Shipping Dimensions
Manage Tried & True

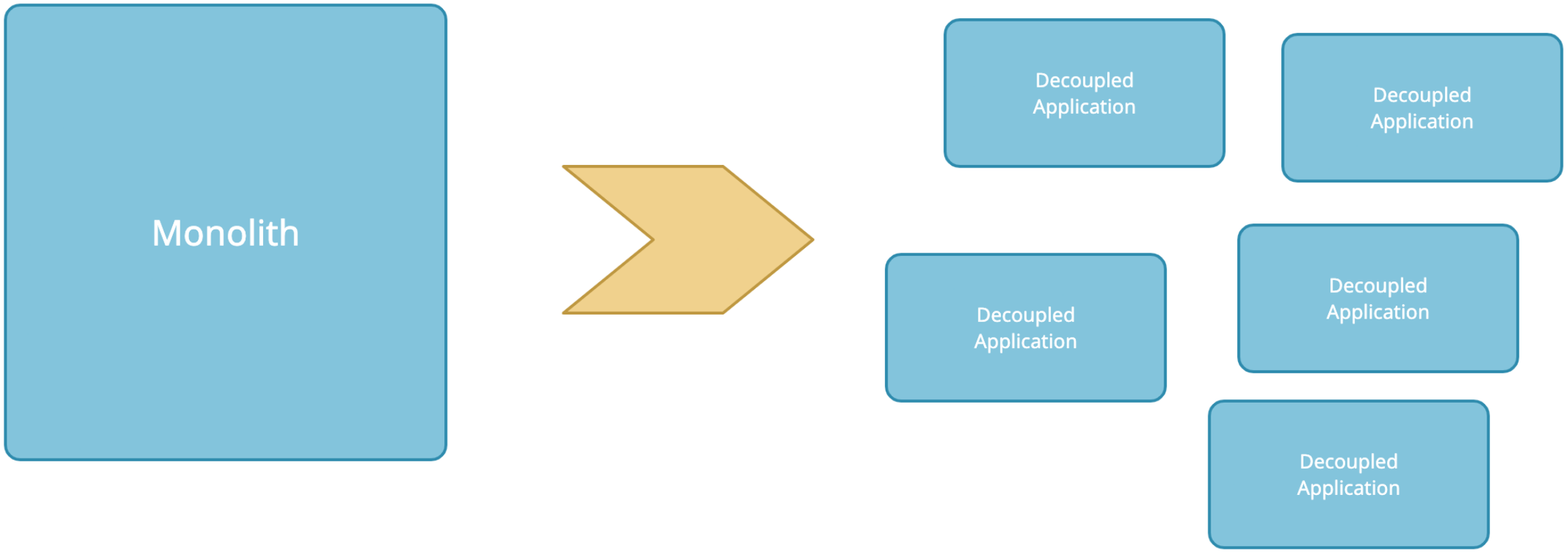
Accelerant Hub

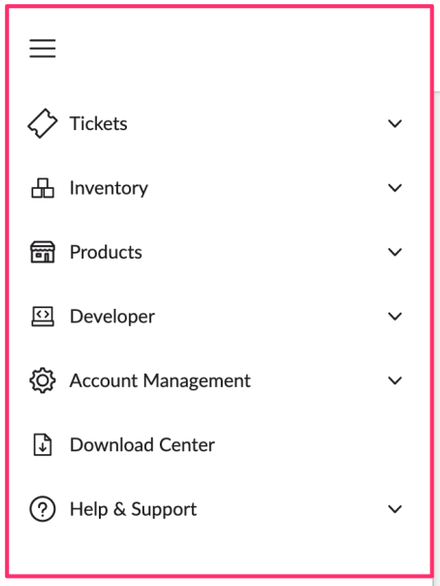
Accelerant Hub Home
Sponsored Products
Brand Page Management

Rough numbers

~100 flows

~30 experience teams


Monolith
Decoupled Application
Decoupled Application
Decoupled Application
Decoupled Application
Decoupled Application



wayfair PARTNER HOME

Powell

Your Wayfair Dashboard

Help

Announcements | Previous Announcements

12/10/2020

COVID-19 Guidance For Impacted Warehouses [North America]

If you are closing, extending a closure or reopening, **please update your Warehouse Closure Calendar, open orders & submit our COVID-19 impact form.** This will mitigate negative downstream effects on our mutual customers & you, our Supplier Partner.

Please reach out to your Performance and Relationship Managers with any questions.

06/25/2021

Current Order Management Tool Available Until September

The **current version of** Order Management will remain available until September, not July, as we take some extra time to polish the next version.

Until then, try out the new experience, including features we've added based on your feedback.

Search PO
Enter PO Number
Search

Quick Links

Product Management

Add New Products
Manage Pricing
Manage Product Details
Manage Product Images
Manage Shipping Dimensions
Manage Tried & True

Today's Tasks

50 Order Alert(s)

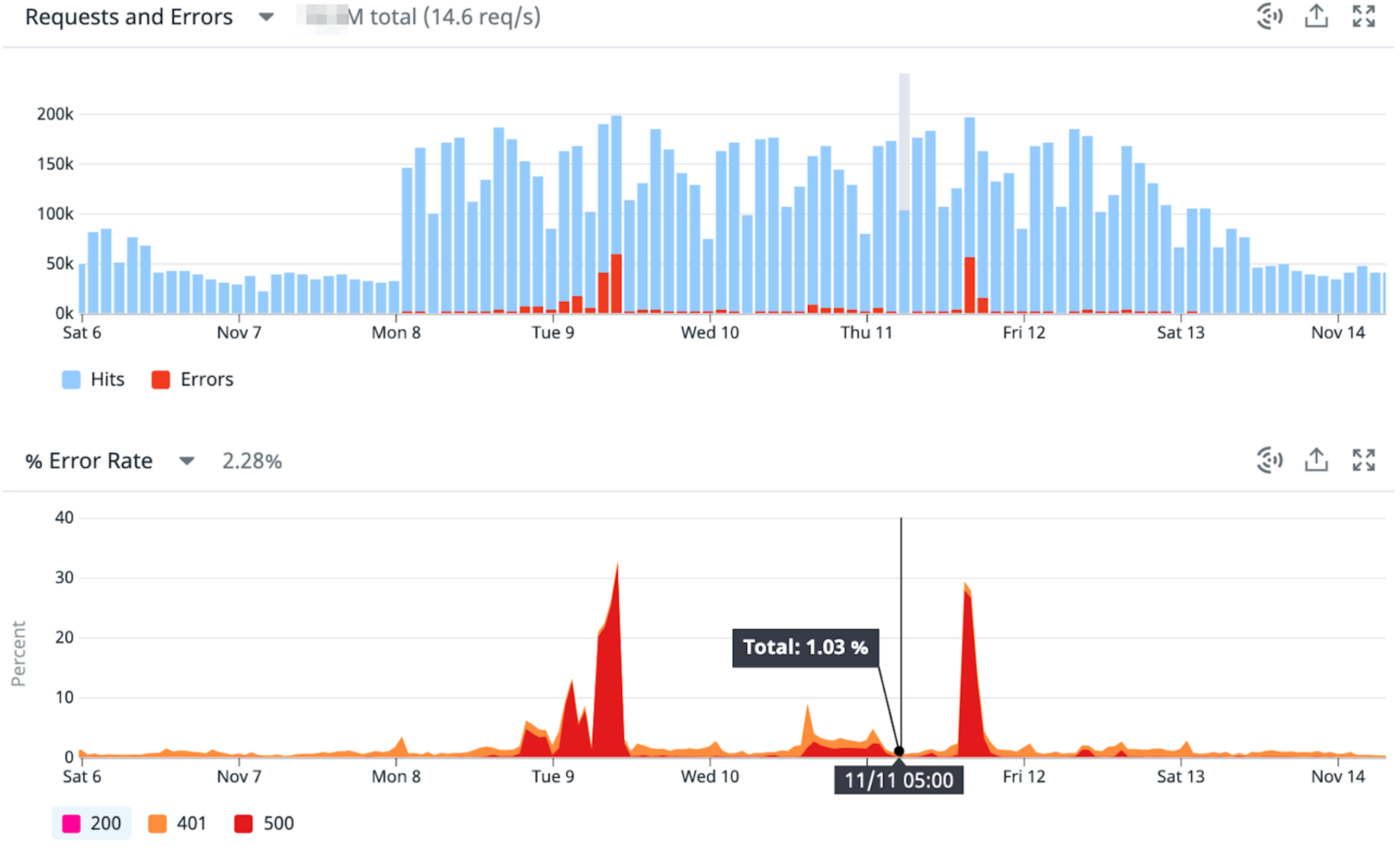
86 to Ship Today

36 to Ship Next Business Day

465 Ticket(s)

Issues 💣💥🔥

Reliant on legacy monolith


Requests and Errors
M total (14.6 req/s)
200k
150k
100k
50k
0k
Sat 6
Nov 7
Mon 8
Tue 9
Wed 10
Thu 11
Fri 12
Sat 13
Nov 14
Hits
Errors
% Error Rate
2.28%
40
30
20
10
0
Percent
Total: 1.03 %
11/11 05:00
200
401
500

User	Time on Call	High-Urgency Incidents	Acknowledged	Timeout Escalations	Manual Escalations	Reassignments	MTTA
[redacted]	282h 0m	109	1% (1)	12% (13)		5% (5)	2m
Mario Fernandez	1237h 38m	93	46% (43)	14% (13)		4% (4)	3m
[redacted]	1091h 30m	51	78% (40)	12% (6)		2% (1)	2m
[redacted]	1420h 41m	37	97% (36)				1m
[redacted]	1707h 38m	22	64% (14)	23% (5)		9% (2)	12m
[redacted]	2070h 56m	21	100% (21)			5% (1)	2m
[redacted]	1756h 0m	19	89% (17)	5% (1)			1m
[redacted]	1639h 38m	19	32% (6)	63% (12)	16% (3)	5% (1)	8m
[redacted]	1512h 11m	15	67% (10)				10m
[redacted]	930h 40m	4	25% (1)	50% (2)			9m
[redacted]	336h 0m	4	50% (2)	25% (1)		25% (1)	3m
[redacted]	1059h 38m	3	100% (3)	33% (1)			8m
[redacted]	1044h 58m	3	100% (3)				4m
[redacted]	840h 0m	3	33% (1)				5m
[redacted]	1252h 0m	2		50% (1)			
[redacted]	168h 4m						
[redacted]	36h 30m						

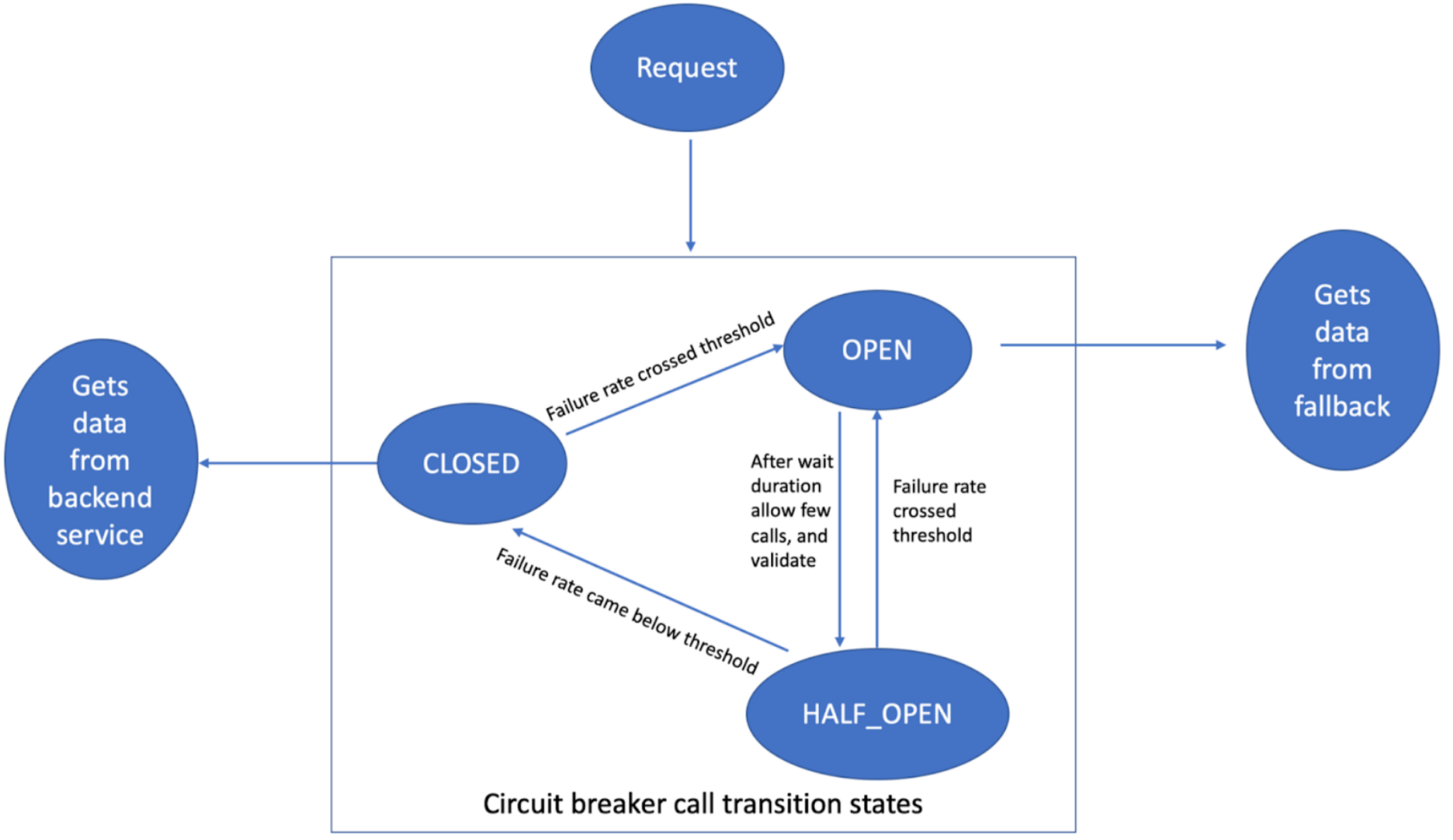
Humane On-Call

DevoxxUK 2021

www.youtube.com/watch?v=X59Sfaey5N4

Resilience Operators

Circuit breaker


Request
Gets data from backend service
CLOSED
Failure rate crossed threshold
OPEN
Gets data from fallback
After wait duration allow few calls, and validate
Failure rate crossed threshold
Failure rate came below threshold
HALF_OPEN
Circuit breaker call transition states

Retry


user
homepage
service
Request
Error
Retry Request
Success
user
recommendation
service

No need to build from scratch

Resilience4j

resilience4j.readme.io/

```java
@CircuitBreaker(name = BACKEND, fallbackMethod = "fallback")
@RateLimiter(name = BACKEND)
@Bulkhead(name = BACKEND)
@Retry(name = BACKEND, fallbackMethod = "fallback")
@TimeLimiter(name = BACKEND)
public Mono<String> method(String param1) {
    return Mono.error(new NumberFormatException());
}

private Mono<String> fallback(String param1, IllegalArgumentException e) {
    return Mono.just("test");
}

private Mono<String> fallback(String param1, RuntimeException e) {
    return Mono.just("test");
}
```

Alternative to annotations

Reusable abstraction ➡️ *SyncFallbackCache*

Creating an operator

```
1  Retry getRetry(String name, MeterRegistry meterRegistry) {
2    RetryRegistry retryRegistry = RetryRegistry.ofDefaults();
3    TaggedRetryMetrics.ofRetryRegistry(retryRegistry).bindTo(meterRegistry);
4    return retryRegistry.retry(name);
5  }
6
7  CircuitBreaker getCircuitBreaker(String name, MeterRegistry meterRegistry) {
8    CircuitBreakerRegistry circuitBreakerRegistry = CircuitBreakerRegistry
9        .ofDefaults();
10   TaggedCircuitBreakerMetrics
11       .ofCircuitBreakerRegistry(circuitBreakerRegistry).bindTo(meterRegistry);
12   return circuitBreakerRegistry.circuitBreaker(name);
13 }
```

Creating an operator

```
1  Retry getRetry(String name, MeterRegistry meterRegistry) {
2    RetryRegistry retryRegistry = RetryRegistry.ofDefaults();
3    TaggedRetryMetrics.ofRetryRegistry(retryRegistry).bindTo(meterRegistry);
4    return retryRegistry.retry(name);
5  }
6
7  CircuitBreaker getCircuitBreaker(String name, MeterRegistry meterRegistry) {
8    CircuitBreakerRegistry circuitBreakerRegistry = CircuitBreakerRegistry
9        .ofDefaults();
10   TaggedCircuitBreakerMetrics
11       .ofCircuitBreakerRegistry(circuitBreakerRegistry).bindTo(meterRegistry);
12   return circuitBreakerRegistry.circuitBreaker(name);
13 }
```

Creating an operator

```
1  Retry getRetry(String name, MeterRegistry meterRegistry) {
2    RetryRegistry retryRegistry = RetryRegistry.ofDefaults();
3    TaggedRetryMetrics.ofRetryRegistry(retryRegistry).bindTo(meterRegistry);
4    return retryRegistry.retry(name);
5  }
6
7  CircuitBreaker getCircuitBreaker(String name, MeterRegistry meterRegistry) {
8    CircuitBreakerRegistry circuitBreakerRegistry = CircuitBreakerRegistry
9        .ofDefaults();
10   TaggedCircuitBreakerMetrics
11       .ofCircuitBreakerRegistry(circuitBreakerRegistry).bindTo(meterRegistry);
12   return circuitBreakerRegistry.circuitBreaker(name);
13 }
```

```java
/**
 * Fetches data synchronously
 *
 * @param key           the key that identifies the object
 * @param valueSupplier a supplier to perform the call
 * @return the data returned from the service, or from the cache
 * @throws io.github.resilience4j.circuitbreaker.CallNotPermittedException
 * @throws RuntimeException
 */
public T get(String key, Supplier<R> valueSupplier) {
  Supplier<T> decoratedSupplier = Decorators
      .ofSupplier(() -> getAndCache(key, valueSupplier))
      .withRetry(retry)
      .withCircuitBreaker(cb).decorate();

  return Try.ofSupplier(decoratedSupplier)
      .recover(RuntimeException.class,
          (exception) -> getFromCacheOrThrow(key, exception))
      .get();
}
```

```java
/**
 * Fetches data synchronously
 *
 * @param key           the key that identifies the object
 * @param valueSupplier a supplier to perform the call
 * @return the data returned from the service, or from the cache
 * @throws io.github.resilience4j.circuitbreaker.CallNotPermittedException
 * @throws RuntimeException
 */
public T get(String key, Supplier<R> valueSupplier) {
  Supplier<T> decoratedSupplier = Decorators
      .ofSupplier(() -> getAndCache(key, valueSupplier))
      .withRetry(retry)
      .withCircuitBreaker(cb).decorate();

  return Try.ofSupplier(decoratedSupplier)
      .recover(RuntimeException.class,
          (exception) -> getFromCacheOrThrow(key, exception))
      .get();
}
```

```java
/**
 * Fetches data synchronously
 *
 * @param key           the key that identifies the object
 * @param valueSupplier a supplier to perform the call
 * @return the data returned from the service, or from the cache
 * @throws io.github.resilience4j.circuitbreaker.CallNotPermittedException
 * @throws RuntimeException
 */
public T get(String key, Supplier<R> valueSupplier) {
  Supplier<T> decoratedSupplier = Decorators
      .ofSupplier(() -> getAndCache(key, valueSupplier))
      .withRetry(retry)
      .withCircuitBreaker(cb).decorate();

  return Try.ofSupplier(decoratedSupplier)
      .recover(RuntimeException.class,
          (exception) -> getFromCacheOrThrow(key, exception))
      .get();
}
```

```java
/**
 * Fetches data synchronously
 *
 * @param key           the key that identifies the object
 * @param valueSupplier a supplier to perform the call
 * @return the data returned from the service, or from the cache
 * @throws io.github.resilience4j.circuitbreaker.CallNotPermittedException
 * @throws RuntimeException
 */
public T get(String key, Supplier<R> valueSupplier) {
  Supplier<T> decoratedSupplier = Decorators
      .ofSupplier(() -> getAndCache(key, valueSupplier))
      .withRetry(retry)
      .withCircuitBreaker(cb).decorate();

  return Try.ofSupplier(decoratedSupplier)
      .recover(RuntimeException.class,
          (exception) -> getFromCacheOrThrow(key, exception))
      .get();
}
```

High-level usage

```
1  private SyncFallbackCache<Collection<SupplierUserPermission>> permissions;
2
3  private Collection<SupplierUserPermission> getPermissions(
4      String extranetUserId,
5      int supplierId) {
6    return permissions.get(
7        () -> String.format("exuId:%s;suId:%s", extranetUserId, supplierId),
8        () -> client.getUserPermissions(extranetUserId, supplierId)
9            .getData()
10           .getSupplierUserMyInformation()
11           .getPermissions()
12   );
13 }
```

High-level usage

```
1  private SyncFallbackCache<Collection<SupplierUserPermission>> permissions;
2
3  private Collection<SupplierUserPermission> getPermissions(
4      String extranetUserId,
5      int supplierId) {
6    return permissions.get(
7        () -> String.format("exuId:%s;suId:%s", extranetUserId, supplierId),
8        () -> client.getUserPermissions(extranetUserId, supplierId)
9            .getData()
10           .getSupplierUserMyInformation()
11           .getPermissions()
12   );
13 }
```

High-level usage

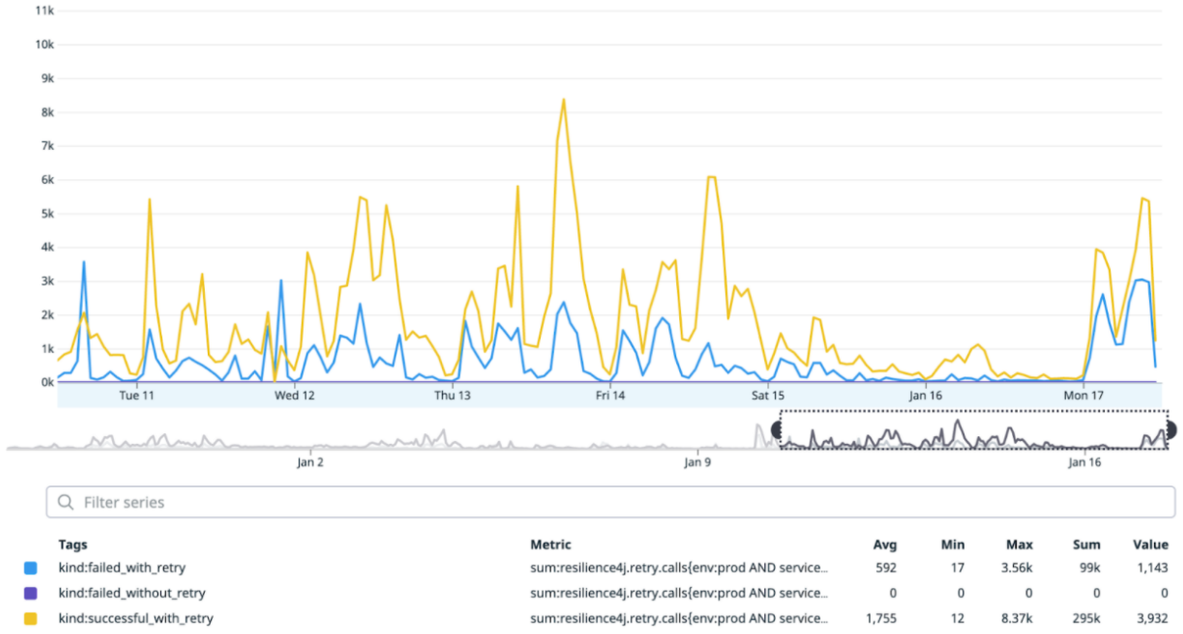
```
 1 private SyncFallbackCache<Collection<SupplierUserPermission>> permissions;
 2
 3 private Collection<SupplierUserPermission> getPermissions(
 4     String extranetUserId,
 5     int supplierId) {
 6   return permissions.get(
 7       () -> String.format("exuId:%s;suId:%s", extranetUserId, supplierId),
 8       () -> client.getUserPermissions(extranetUserId, supplierId)
 9           .getData()
10           .getSupplierUserMyInformation()
11           .getPermissions()
12   );
13 }
```

What about *Reactor*?

Support for *Mono* ➡️ *AsyncFallbackCache*

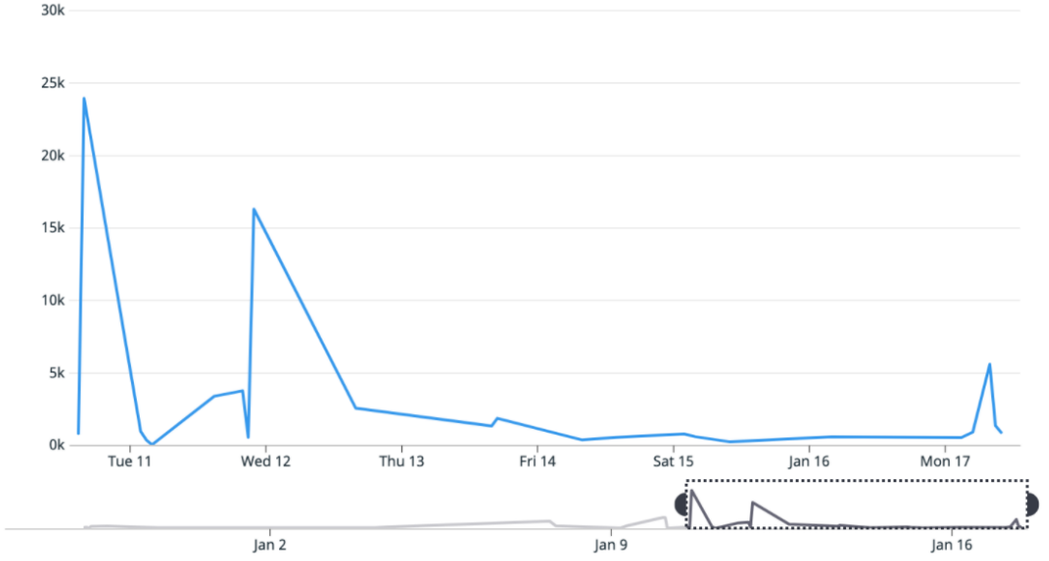
Let's not forget monitoring

Nav Service Retry



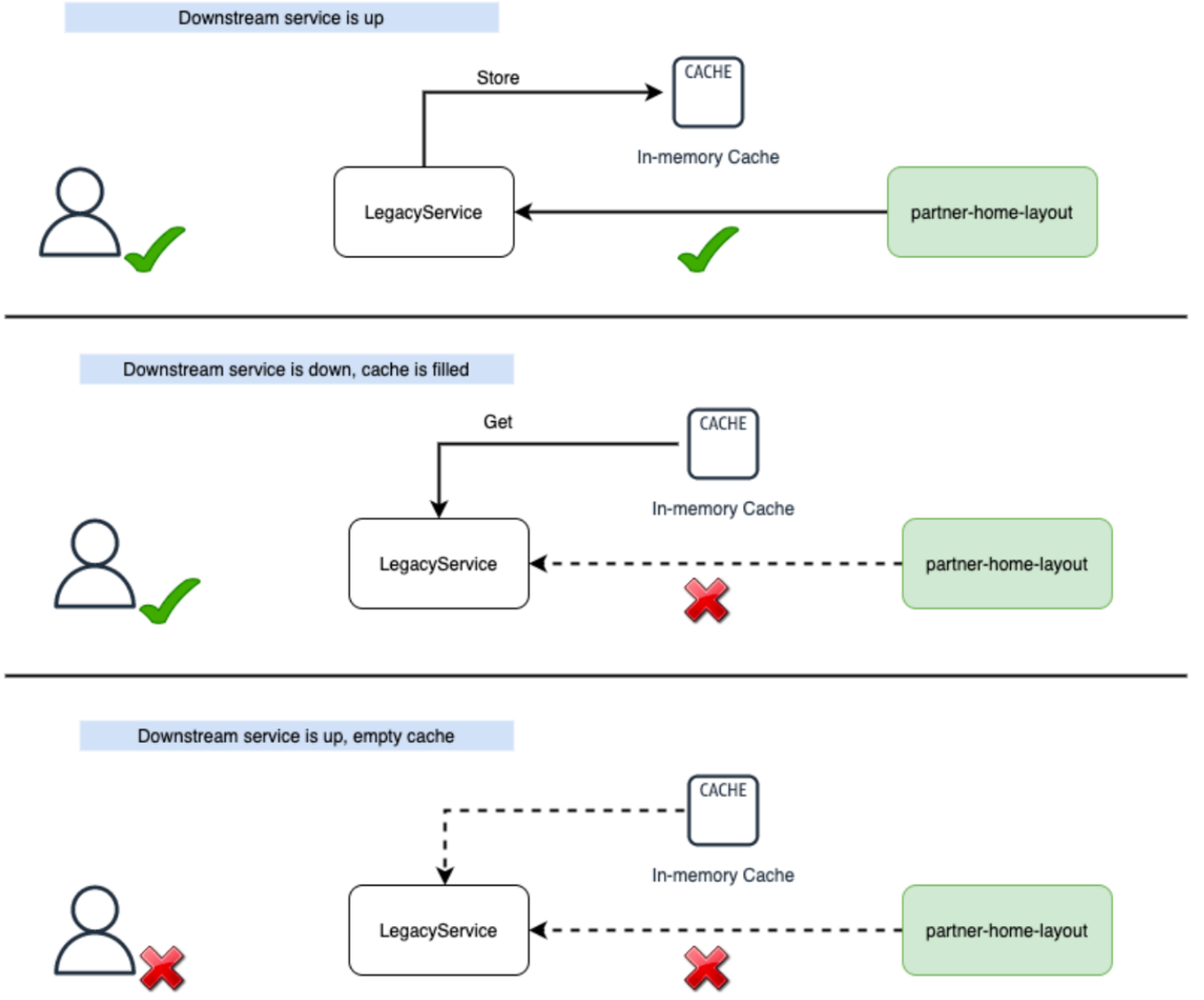
Tags	Metric	Avg	Min	Max	Sum	Value
kind:failed_with_retry	sum:resilience4j.retry.calls{env:prod AND service...	592	17	3.56k	99k	1,143
kind:failed_without_retry	sum:resilience4j.retry.calls{env:prod AND service...	0	0	0	0	0
kind:successful_with_retry	sum:resilience4j.retry.calls{env:prod AND service...	1,755	12	8.37k	295k	3,932

Nav Service Circuit Breaker - CallNotPermitted



Fallback Cache

Graceful degradation


Downstream service is up
Store
CACHE
In-memory Cache
LegacyService
partner-home-layout
Downstream service is down, cache is filled
Get
CACHE
In-memory Cache
LegacyService
partner-home-layout
Downstream service is up, empty cache
CACHE
In-memory Cache
LegacyService
partner-home-layout

Caffeine

github.com/ben-manes/caffeine

```java
static <R> Cache<String, R> cache(String name, MeterRegistry meterRegistry) {
  Cache<String, R> cache = Caffeine.newBuilder()
      .recordStats()
      .expireAfterWrite(Duration.ofHours(24))
      .maximumSize(6000)
      .build();

  CaffeineCacheMetrics.monitor(meterRegistry, cache, name);
  return cache;
}
```

```java
static <R> Cache<String, R> cache(String name, MeterRegistry meterRegistry) {
  Cache<String, R> cache = Caffeine.newBuilder()
      .recordStats()
      .expireAfterWrite(Duration.ofHours(24))
      .maximumSize(6000)
      .build();

  CaffeineCacheMetrics.monitor(meterRegistry, cache, name);
  return cache;
}
```

```java
static <R> Cache<String, R> cache(String name, MeterRegistry meterRegistry) {
  Cache<String, R> cache = Caffeine.newBuilder()
      .recordStats()
      .expireAfterWrite(Duration.ofHours(24))
      .maximumSize(6000)
      .build();

  CaffeineCacheMetrics.monitor(meterRegistry, cache, name);
  return cache;
}
```

```
protected T getAndCache(Supplier<String> keySupplier, Supplier<T> valueSupplier) {
  var result = valueSupplier.get();
  if (result != null) {
    cache.put(keySupplier.get(), result);
  }
  return result;
}
```

```
protected T getFromCacheOrThrow(
    Supplier<String> keySupplier,
    RuntimeException exception) {
  var value = cache.getIfPresent(keySupplier.get());

  if (value == null) {
    log.error(String.format(
        "FallbackCache[%s] got an error without a cached value",
        cb.getName()),
        exception);
    throw exception;
  }

  log.info(String.format(
      "FallbackCache[%s] got cached value",
      cb.getName()),
      value);

  return value;
}
```

Does the caching work?

Hit Rate %

Caches

CACHE	SERVICE	HITS	TOTAL	↓ HITRATE%
togglescache	ph-core-navigation-service	717	740	96.89
alertscache	ph-core-navigation-service	302.39k	394.67k	76.62
permissionscache	ph-core-layout30-service	2.74k	3.62k	75.56
appcuescache	ph-core-navigation-service	48.45k	73.35k	66.05
analyticscache	ph-core-navigation-service	128.59k	207.42k	62
navmenucache	ph-core-navigation-service	138.22k	233.92k	59.09
minlayoutcache	ph-core-navigation-service	104.84k	202.8k	51.7

Taking It to the Next Level

Ephemeral caching won't cut it

Aerospike

aerospike.com/

Spring Data

```
public interface NavigationRepository extends
    AerospikeRepository<CachedNavigation, String> {
}
```

Serializable models

```
@Value
@Document(collection = "navigation_CachedNavigationMenu")
public class CachedNavigation implements Cacheable<List<LegacyNavMenuItem>> {
  @Id
  String id;
  List<LegacyNavMenuItem> navigation;

  @Override
  public List<LegacyNavMenuItem> data() {
    return navigation;
  }
}
```

```
| PK                                             | appcues
                                                                                                 | @_class                                                        | layout
                                           | analytics
                                                                                                                      |
+-----------------------------------------------+------------------------------------------------------------------------------------------------------------------------------------------------------------------
+---------------------------------------------------------------------------------------------+---------------------------------------------------------------+------------------------------------------
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
+-------------------------------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------
+-----------------------------------------------------------------------------------------------------------------------+
| "empId:1054878;exuId:151887;suId:22770" | MAP('{"userTypeId":1, "isAdminUser":true, "environment":"prod", "@_class":"com.wayfair.ph.core.navigation.service.payload.serviceresponse.layout2
0.Appcues", "isViewingPageAsAdmin":false, "userLanguageId":"eng_US", "id":"emp1062003"}') | "com.wayfair.ph.core.navigation.service.domain.CachedAppcues"   |
                                          |
                                                                                                                      |
| "empId:1054878;exuId:151887;suId:22770" |
                                                                                                 | "com.wayfair.ph.core.navigation.service.domain.CachedMinLayout" | MAP('{"analyticsDefaultData":{"
userType":1, "userId":151887, "isInternal":true, "employeeId":"1023547", "languageId":"deu", "catalogId":6, "googleUid":"emId_1023547", "@_class":"com.wayfair.ph.core.navigation.service.pay
load.serviceresponse.layout20.Anal |
                                                                                                                      |
| "empId:1054878;exuId:151887;suId:22770" |
                                                                                                 | "com.wayfair.ph.core.navigation.service.domain.CachedAnalytics" |
                                          | MAP('{"userType":1, "userId":164529, "isInternal":true, "employeeId":"1062003", "languageId":"eng_US", "catalogId":1, "googleUid":"emId_1062003", "@_cla
ss":"com.wayfair.ph.core.navigation.service.payload.serviceresponse.layout20.Analytics", "subentityId" |
+-------------------------------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------
+---------------------------------------------------------------------------------------------+---------------------------------------------------------------+------------------------------------------
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
+-------------------------------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------
+-----------------------------------------------------------------------------------------------------------------------+
3 rows in set (0.265 secs)

OK
```

Side note

Testcontainers is a life saver!

www.testcontainers.org/

Did it work this time?

Hit Rate %

Caches

CACHE	SERVICE	HITS	TOTAL	↓ HITRATE%
alerts	ph-core-navigation-service	43.89k	44.23k	99.23
navmenu	ph-core-navigation-service	20.22k	20.49k	98.68
analytics	ph-core-navigation-service	57.15k	58.03k	98.47
appcues	ph-core-navigation-service	7.63k	7.82k	97.57
minlayout	ph-core-navigation-service	24.06k	24.97k	96.36
permissions	ph-core-layout30-service	2.87k	3.22k	88.99

Let's Finish With Some Numbers

Error Rate %

Error Rate %

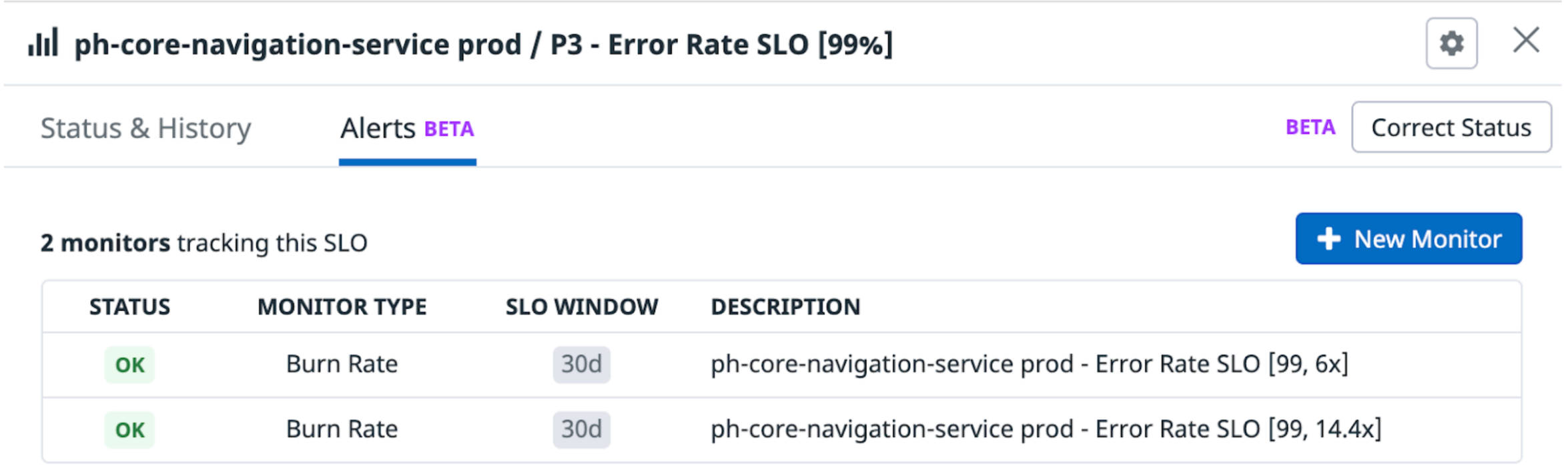
2.2%

Error Rate %

2.2% ➡️

Error Rate %

2.2% ➡️ 0.03%



hceris.com/multiwindow-multi-burn-rate-alerts-in-datadog/