Building Fault Tolerant Micro Services

Kristoffer Erlandsson

kristoffer.erlandsson@avanza.se

@kerlandsson

Building Fault Tolerant Micro Services

Why?

Stability patterns

Failure modes

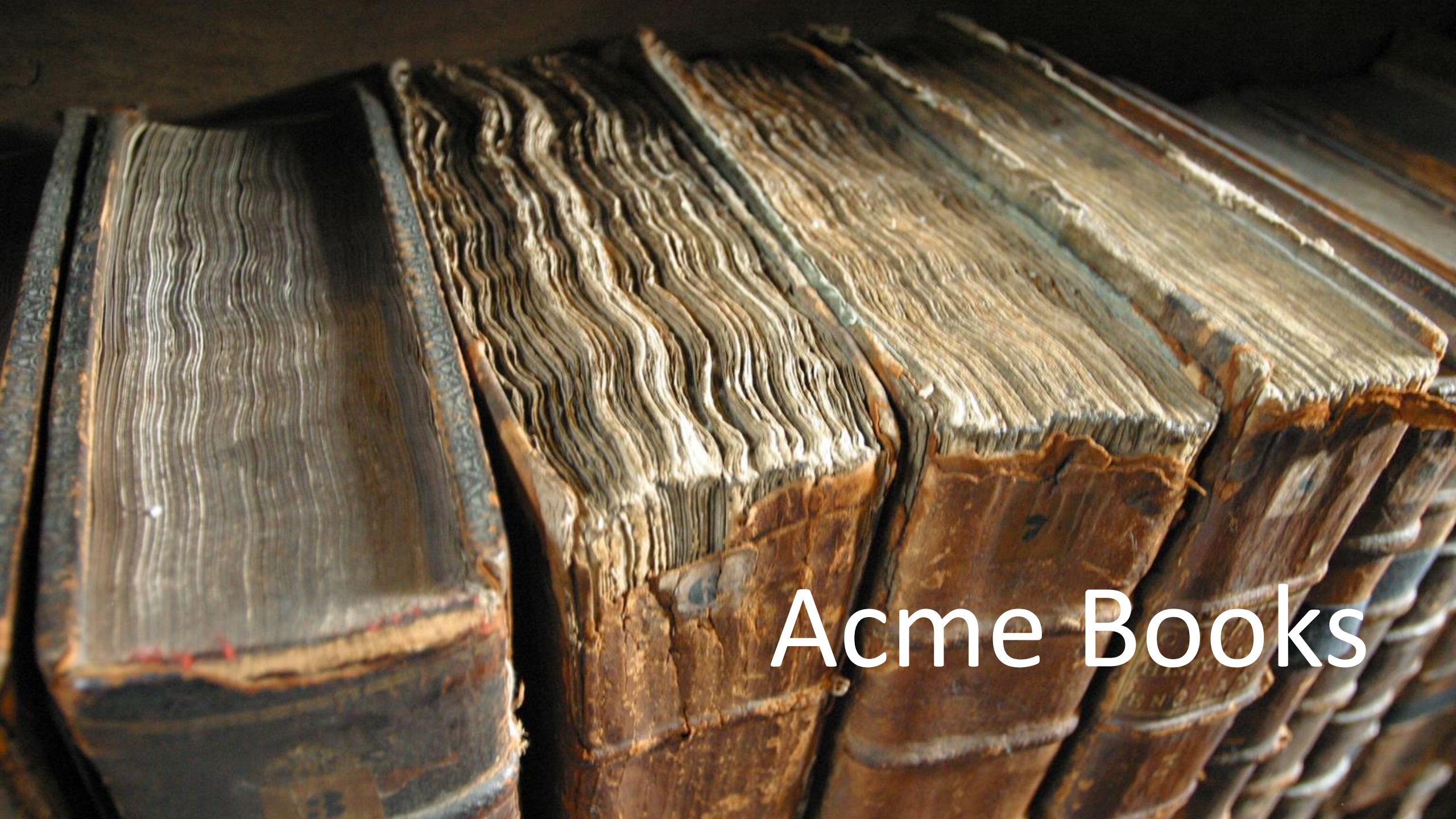
Monitoring guidelines

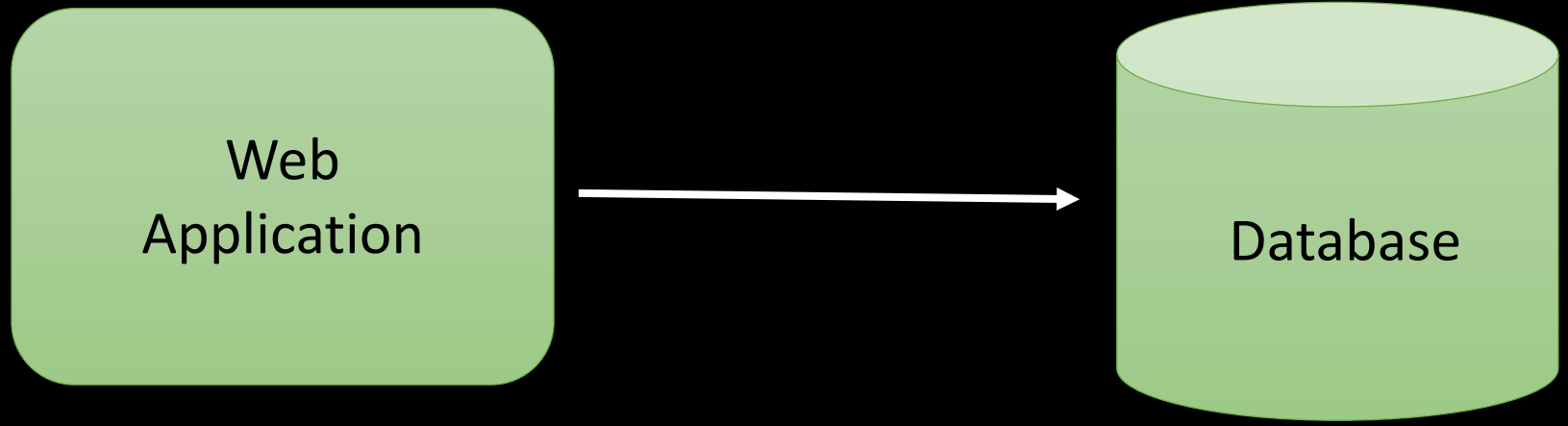
Kristoffer Erlandsson

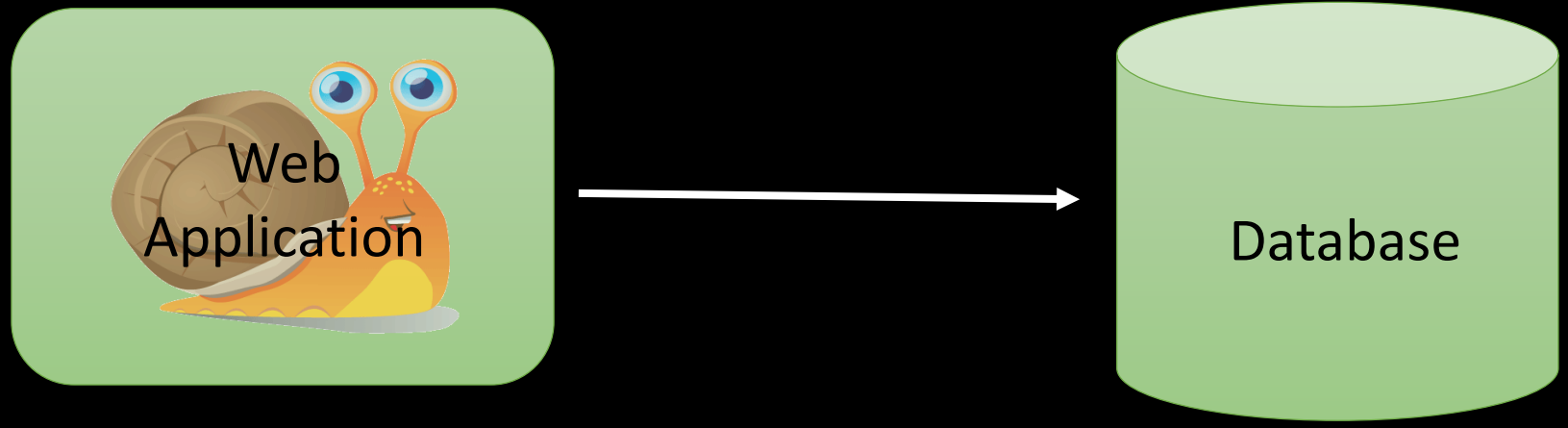
250+ services

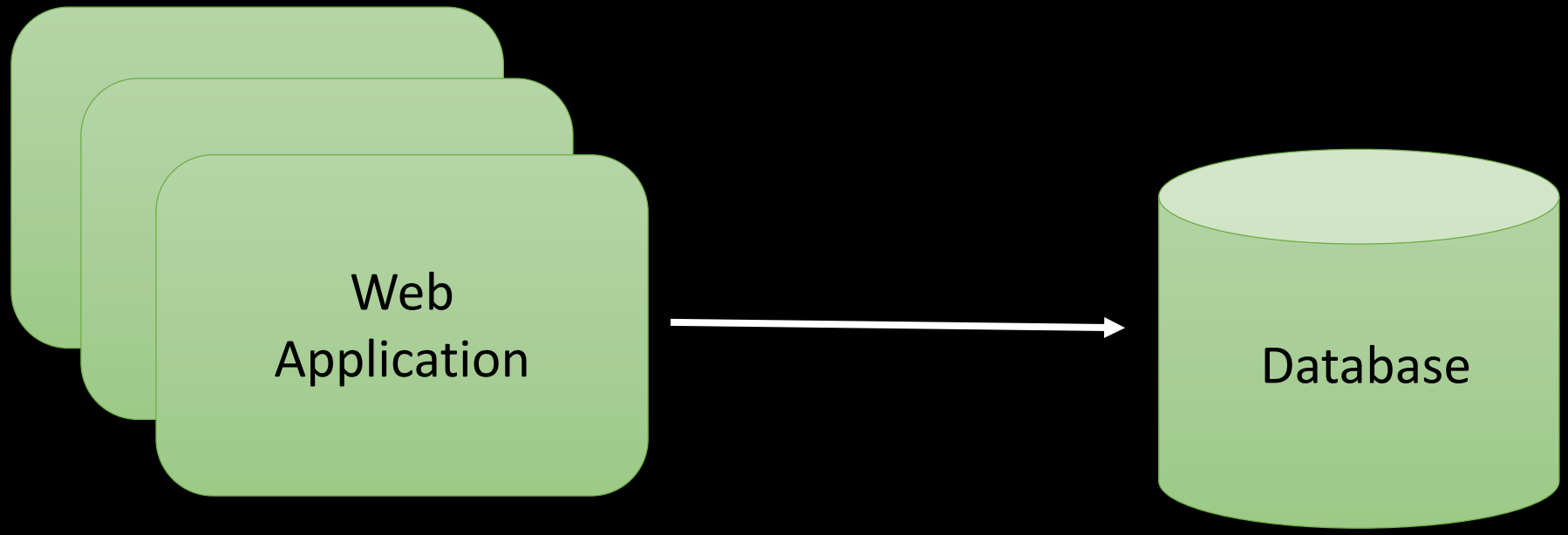
1000+ service instances (JVMs)

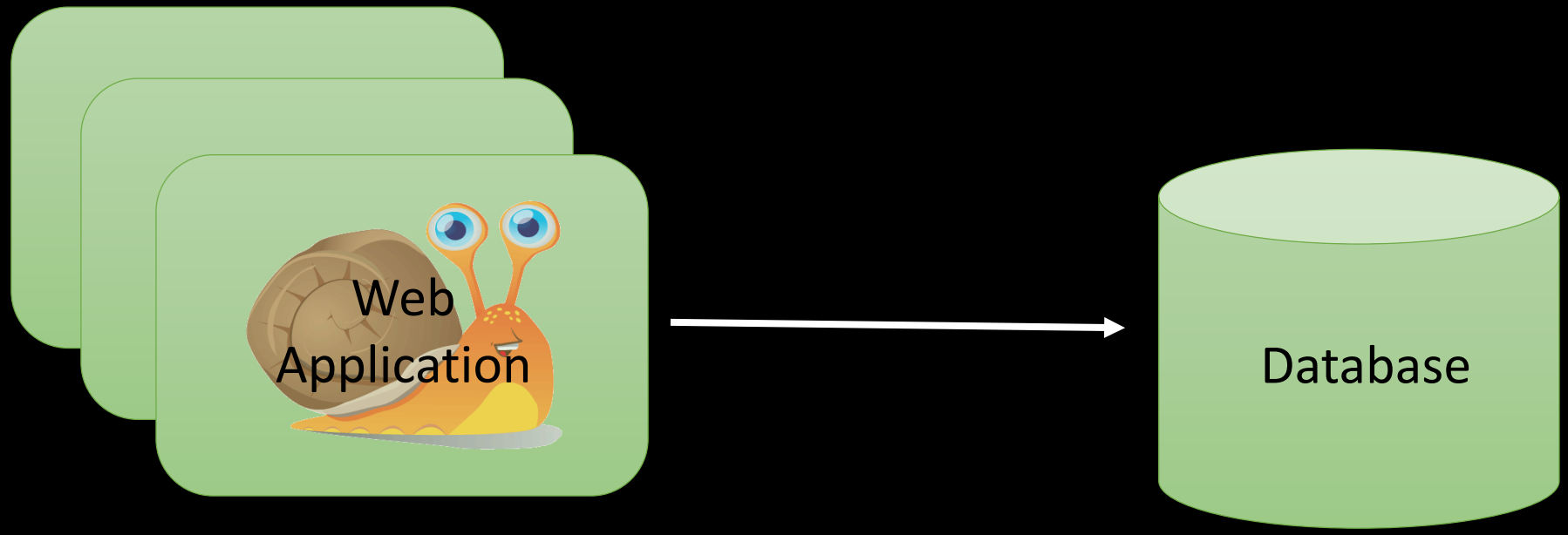
Largest actor on the Stockholm Stock Exchange

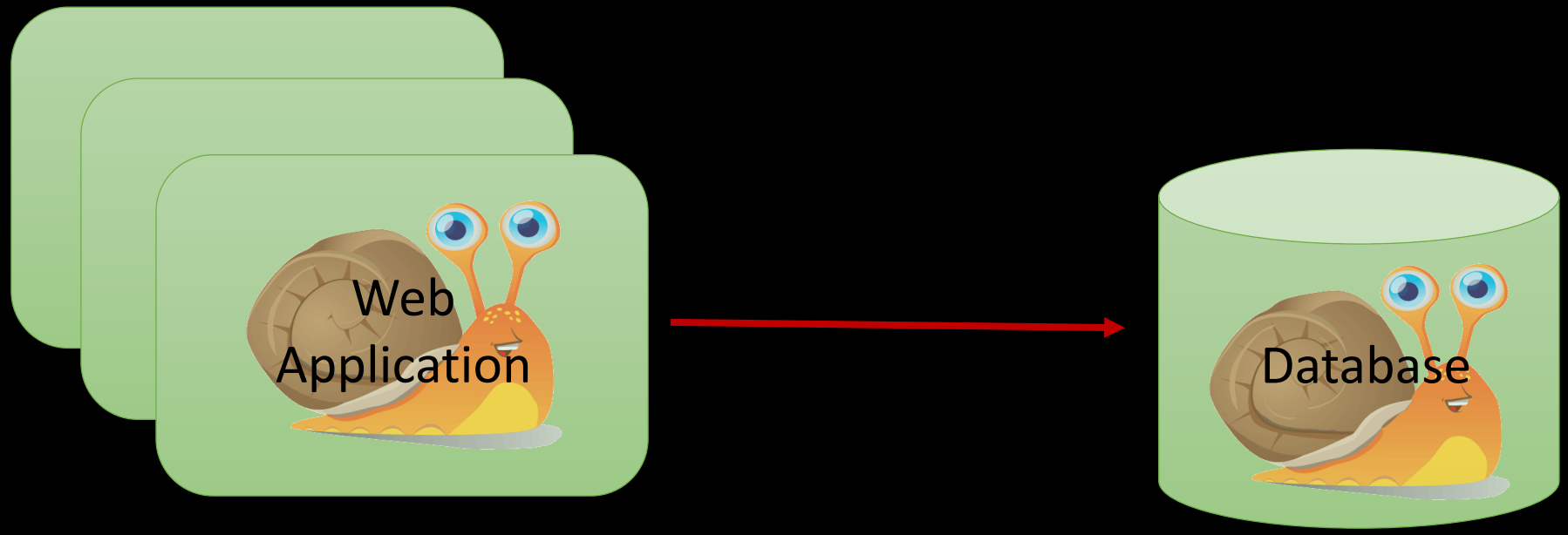

Acme Books

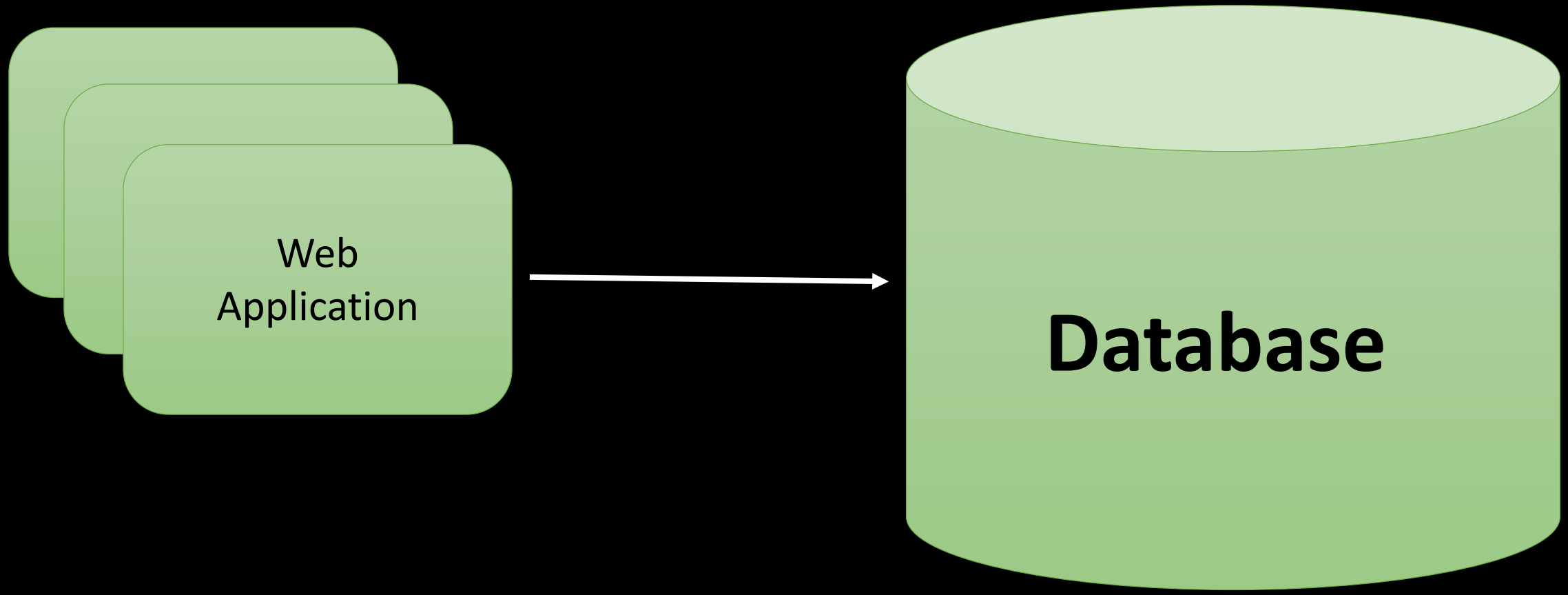

Web
Application
Database


Web
Application
Database

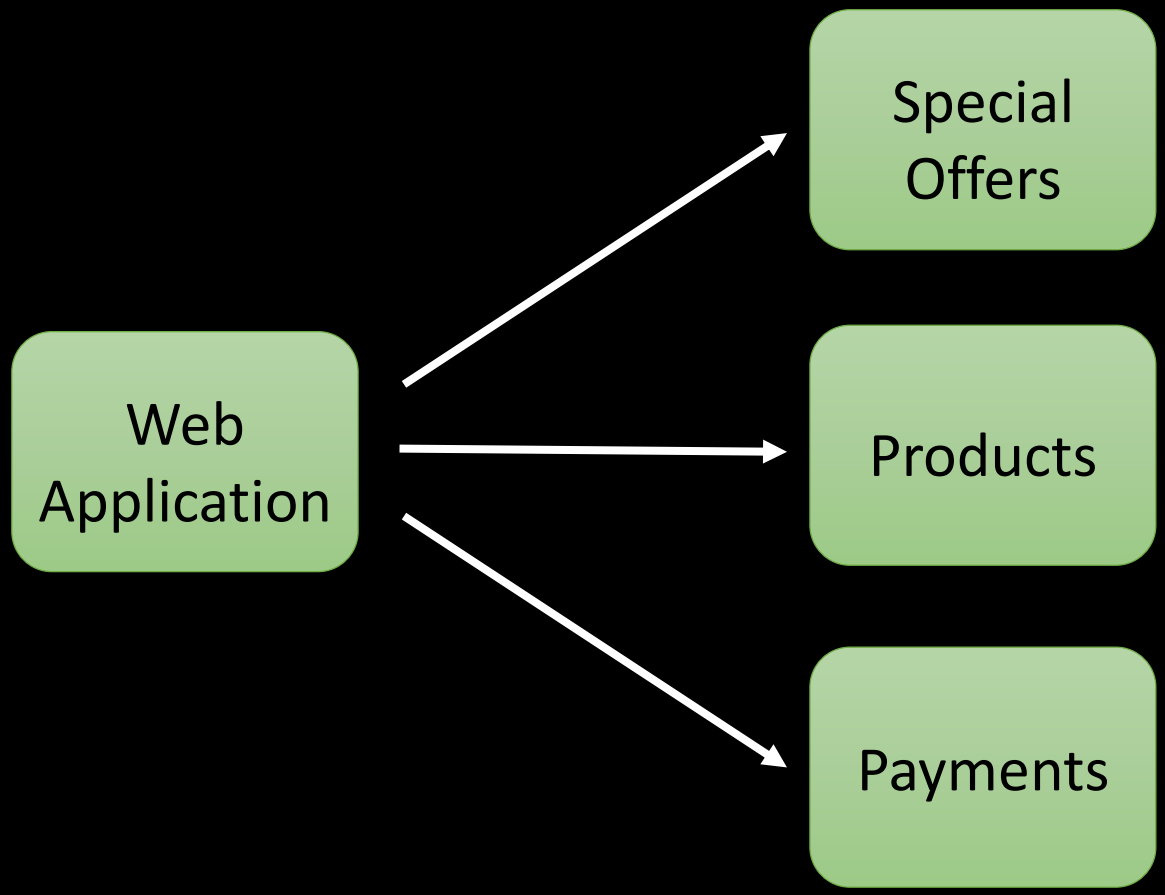

Web
Application
Database

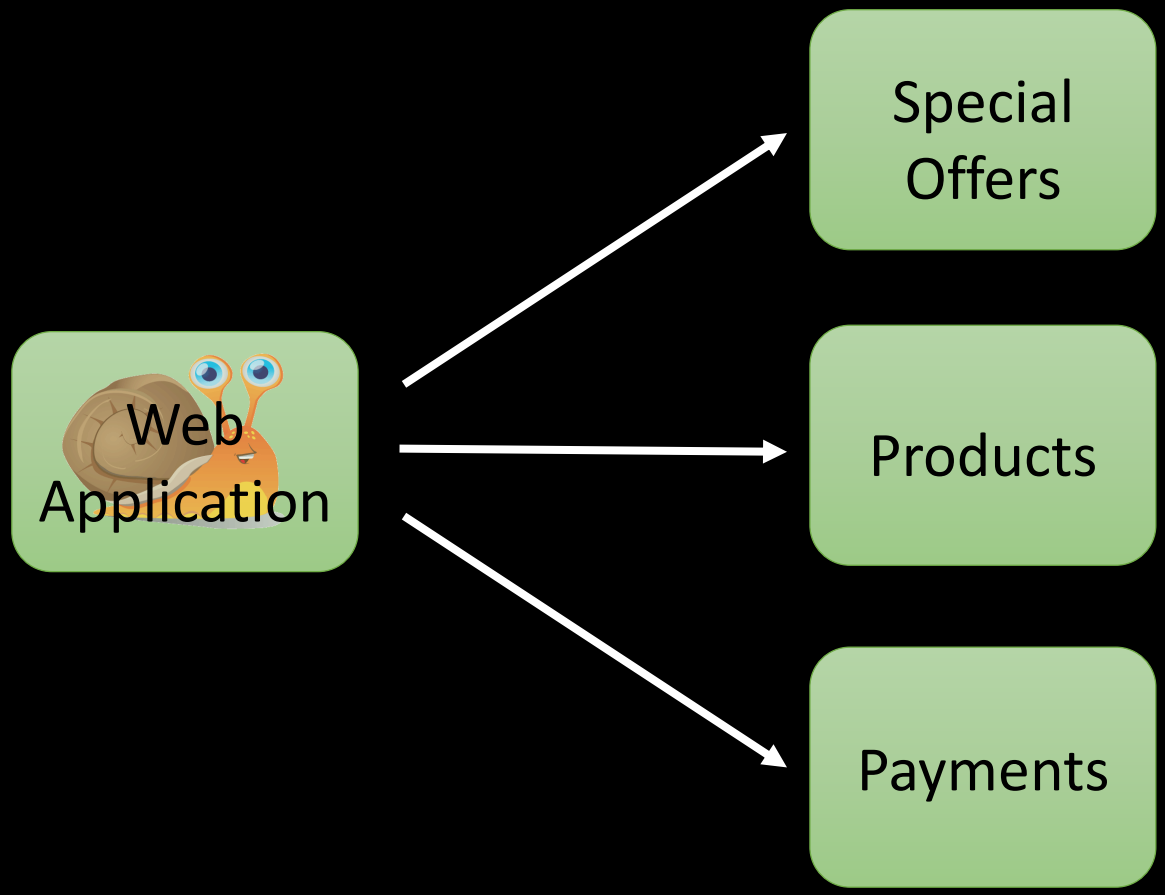

Web
Application
Database

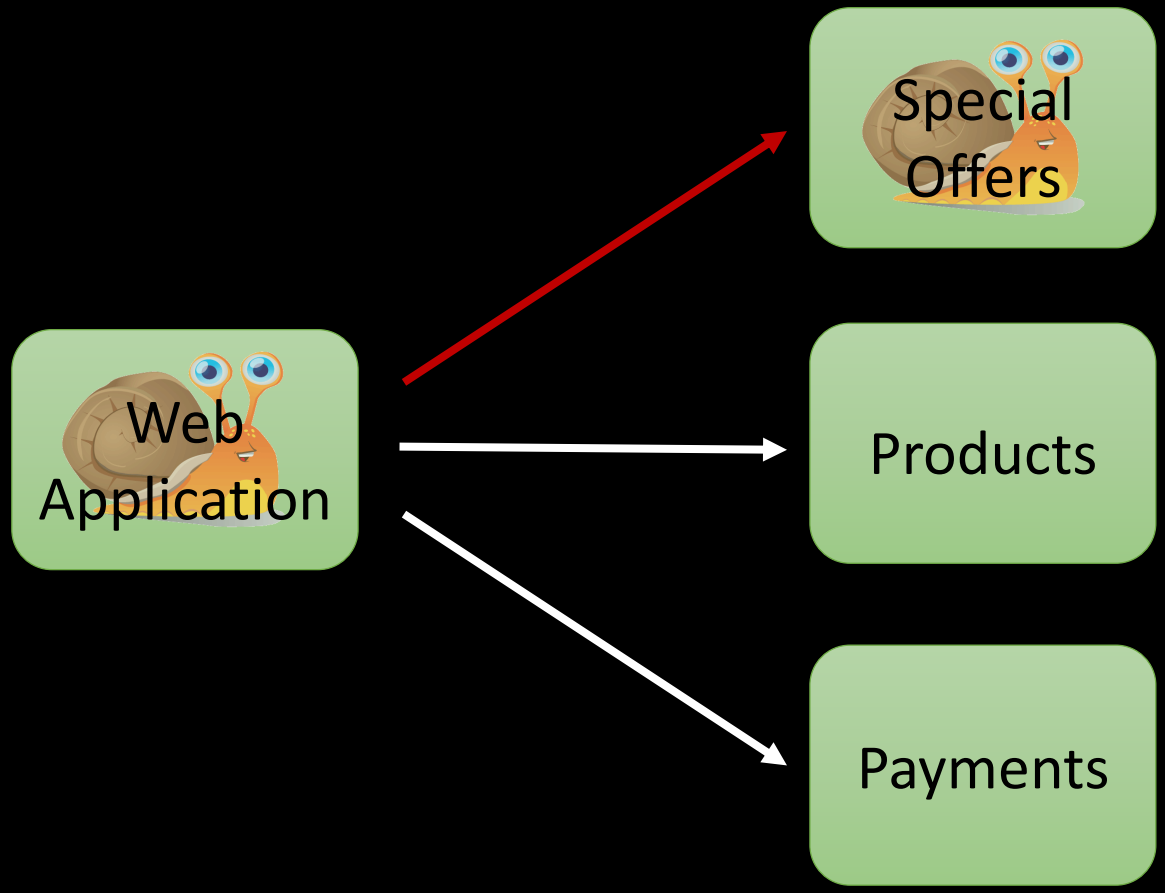

Web
Application
Database

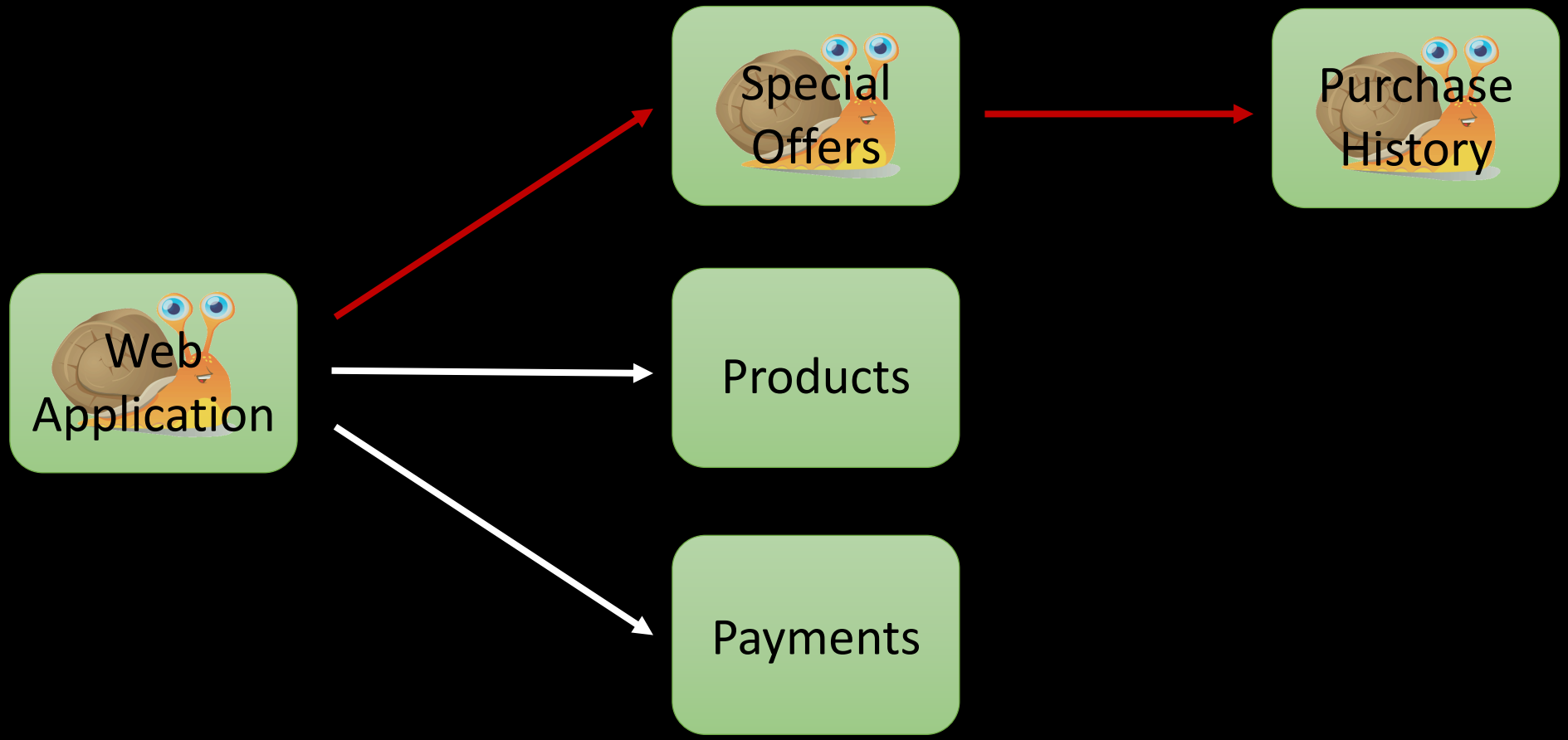

Web
Application
Database

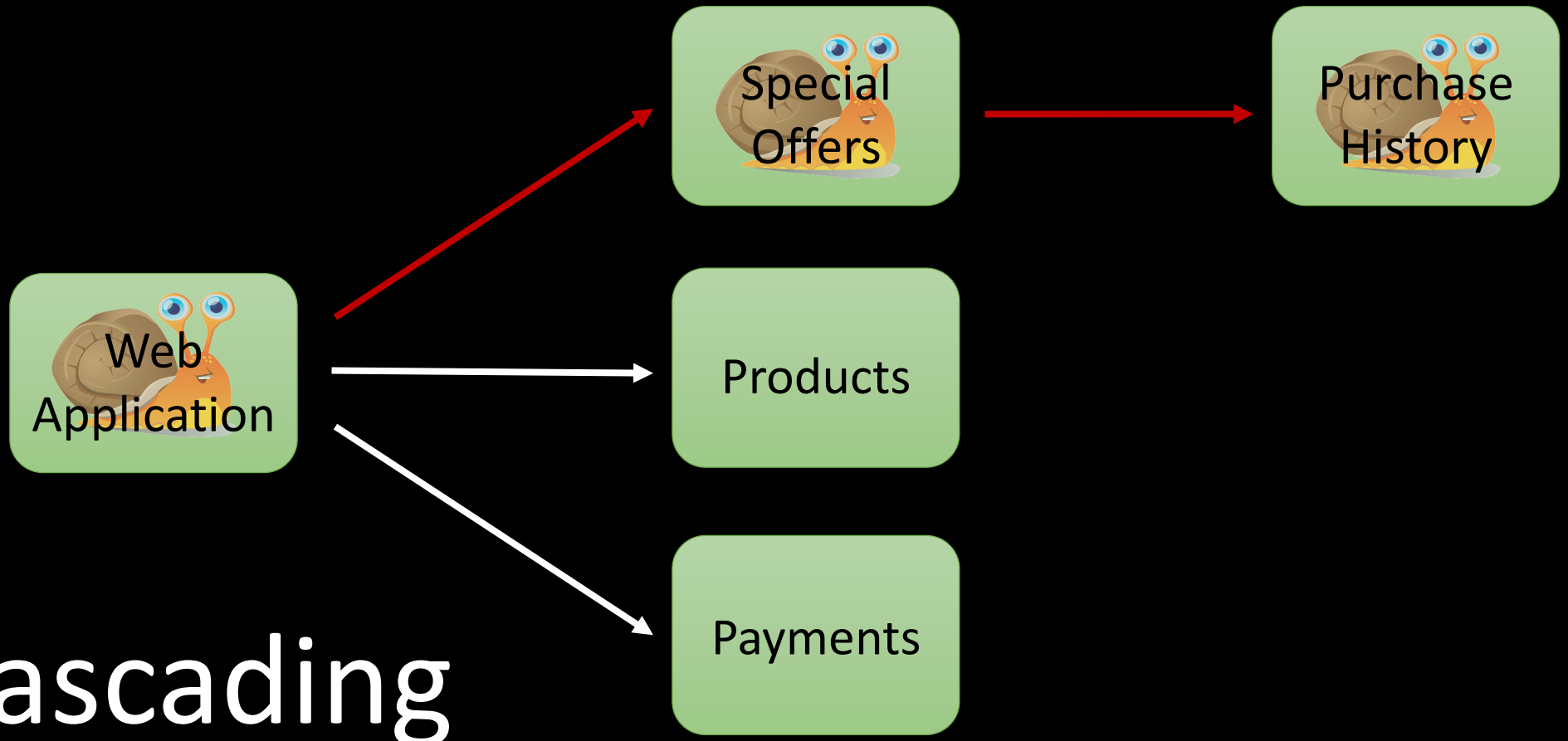

DOIS
$100
DOIS
$100


Web Application
Special Offers
Products
Payments


Web Application
Special Offers
Products
Payments


Special Offers
Web Application
Products
Payments


Special Offers
Purchase History
Web Application
Products
Payments



Cascading Failure

How about increasing availability?

0.99999

5 minutes per year

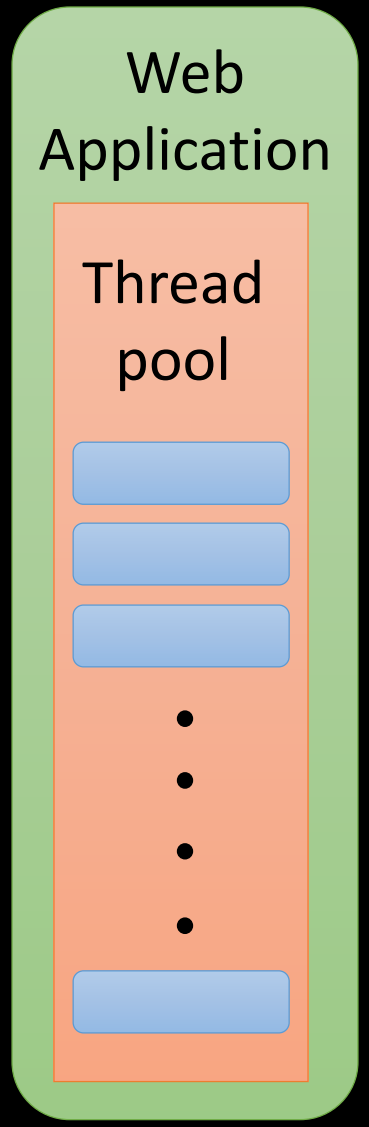
1000
service
instances?

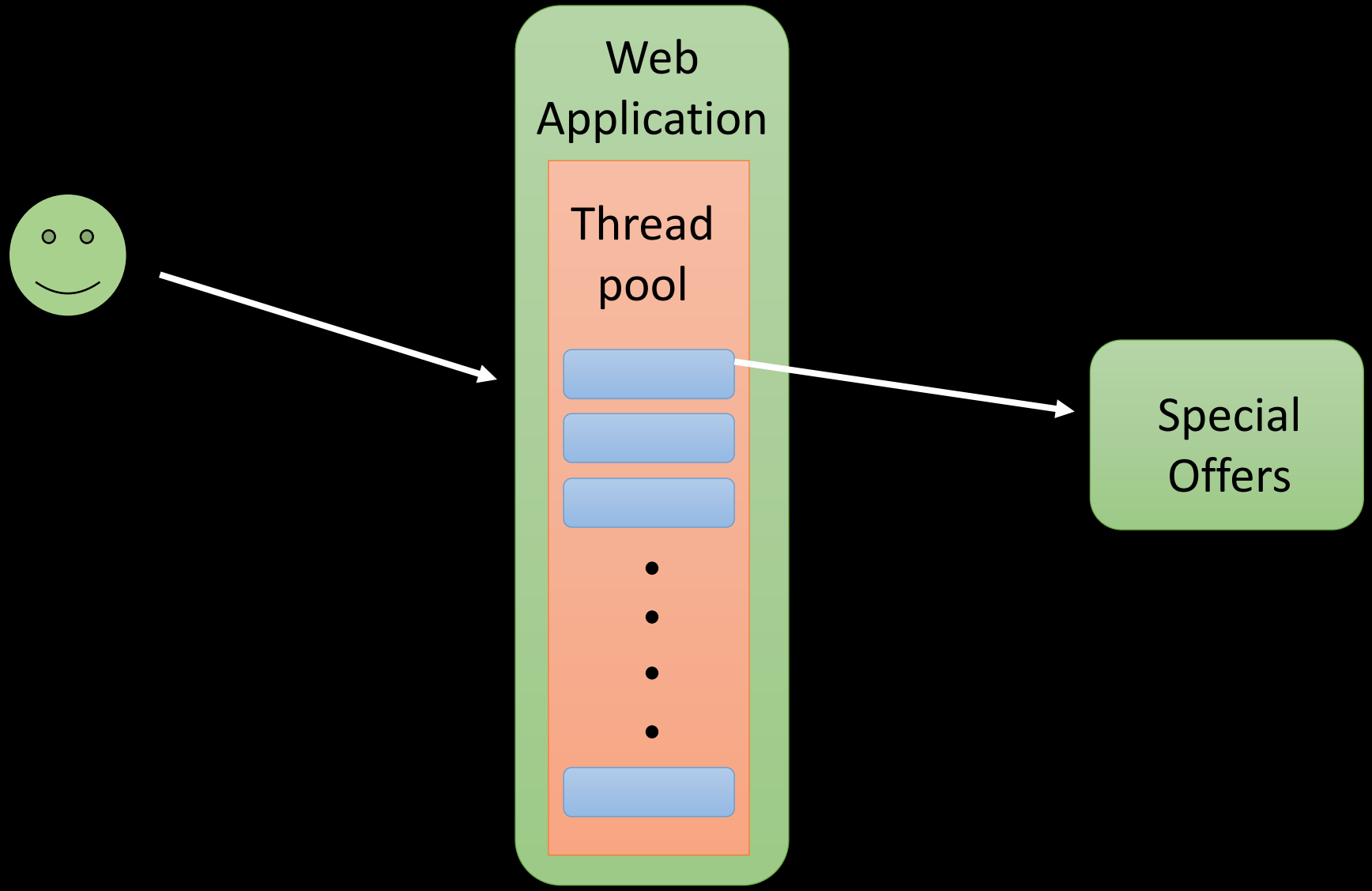
$$0.99999^{1000}$$

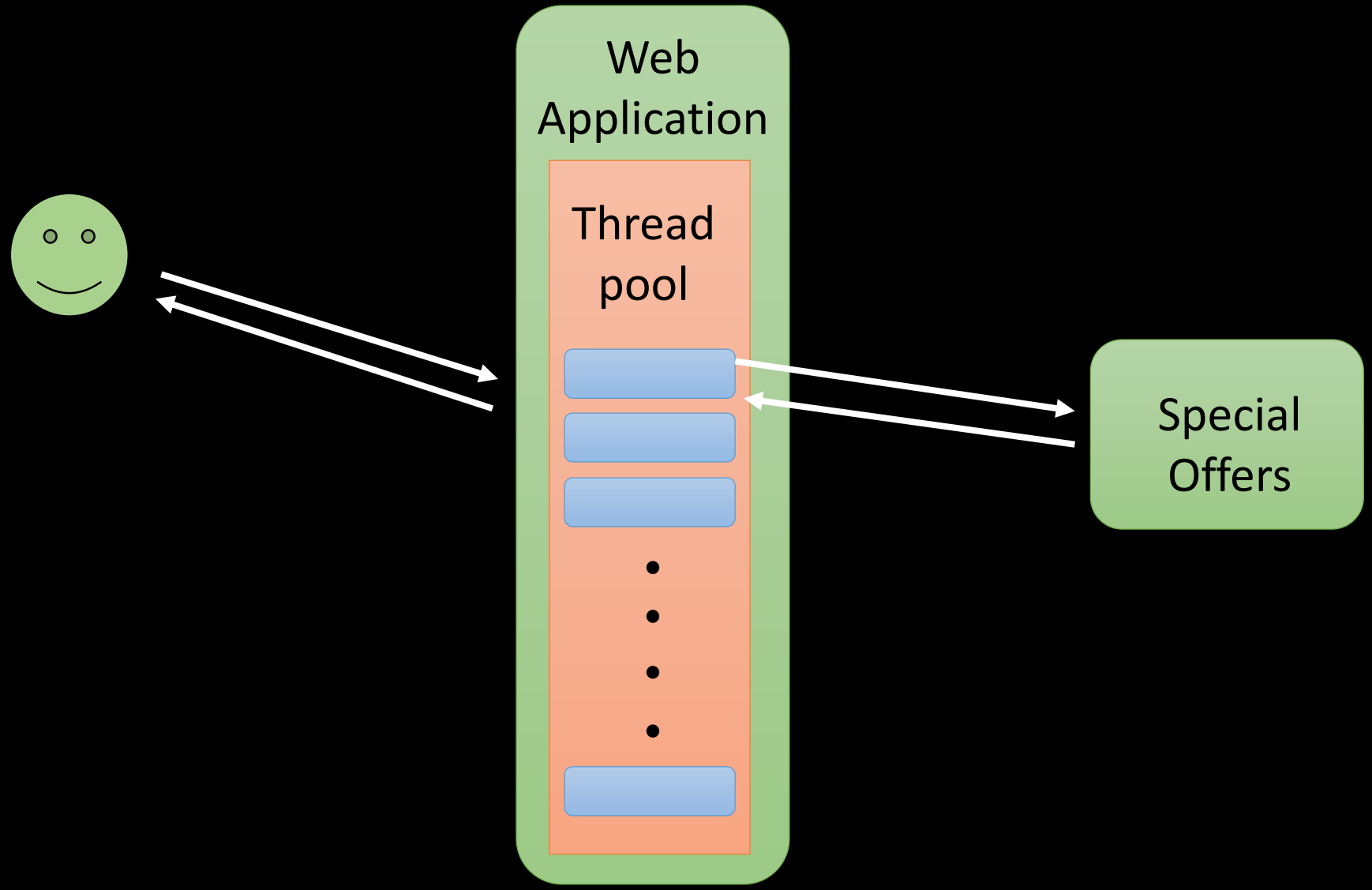
$$0.99999^{1000} \approx 0.99$$

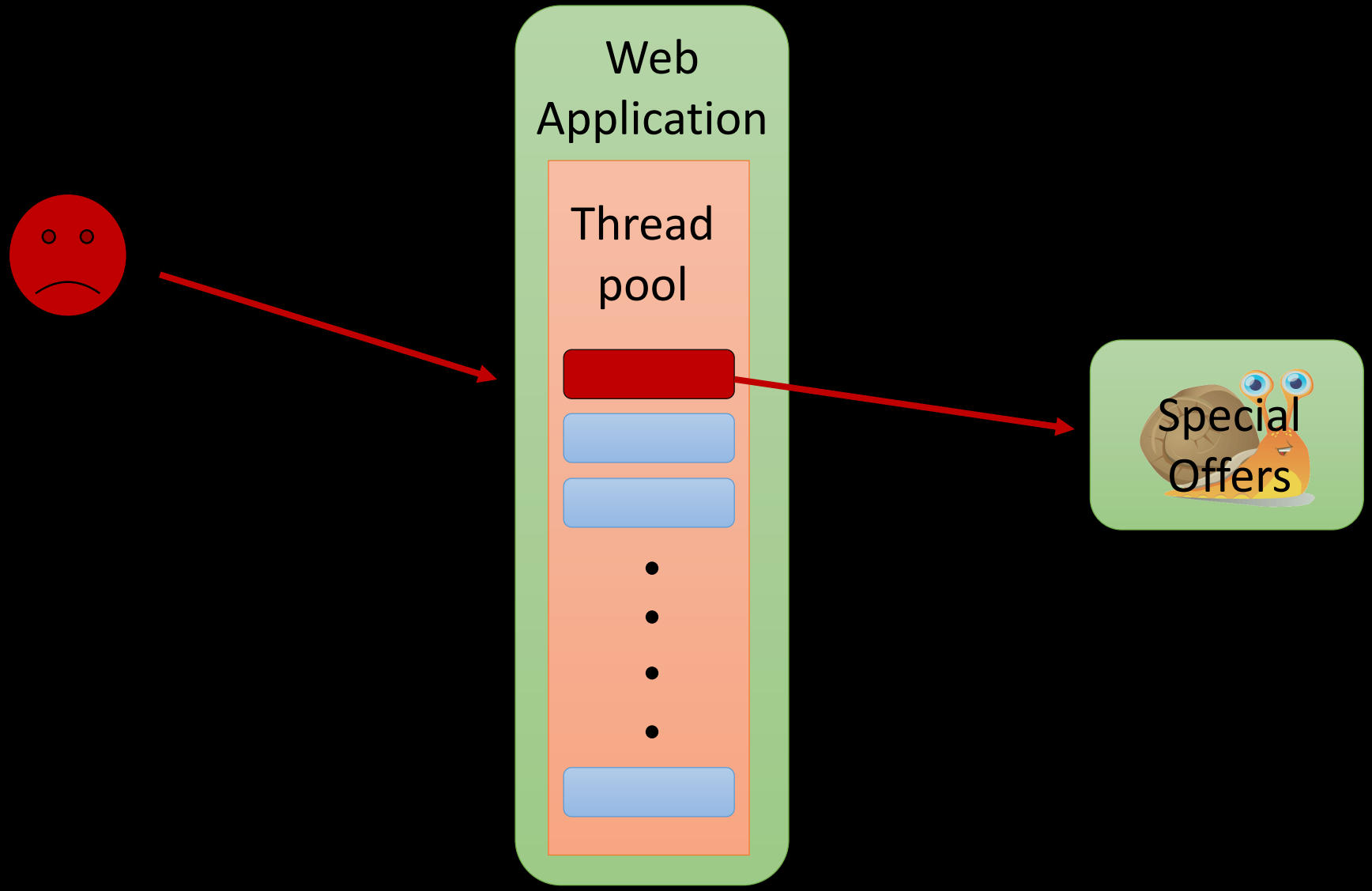
87 hours per year

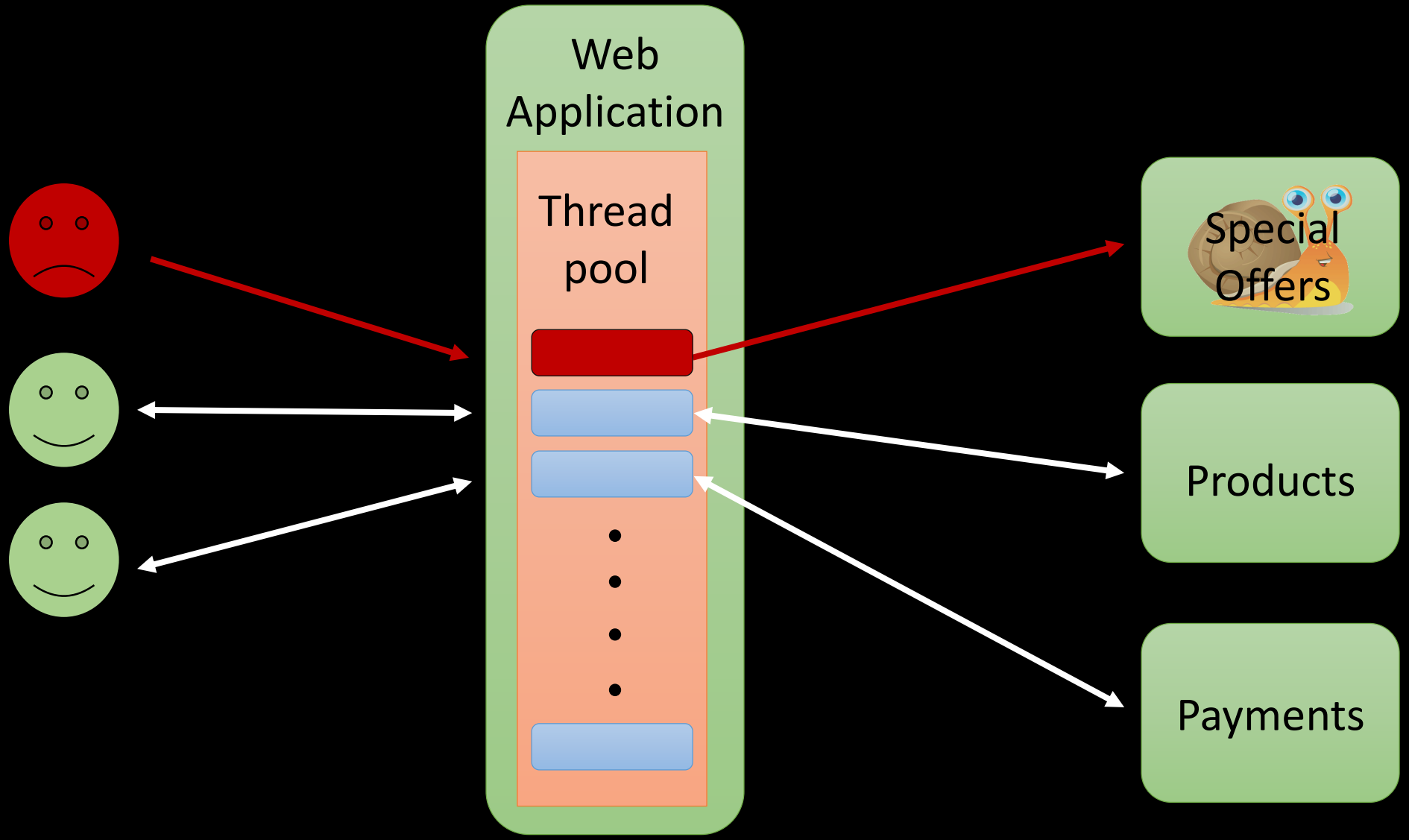
Design for Failure

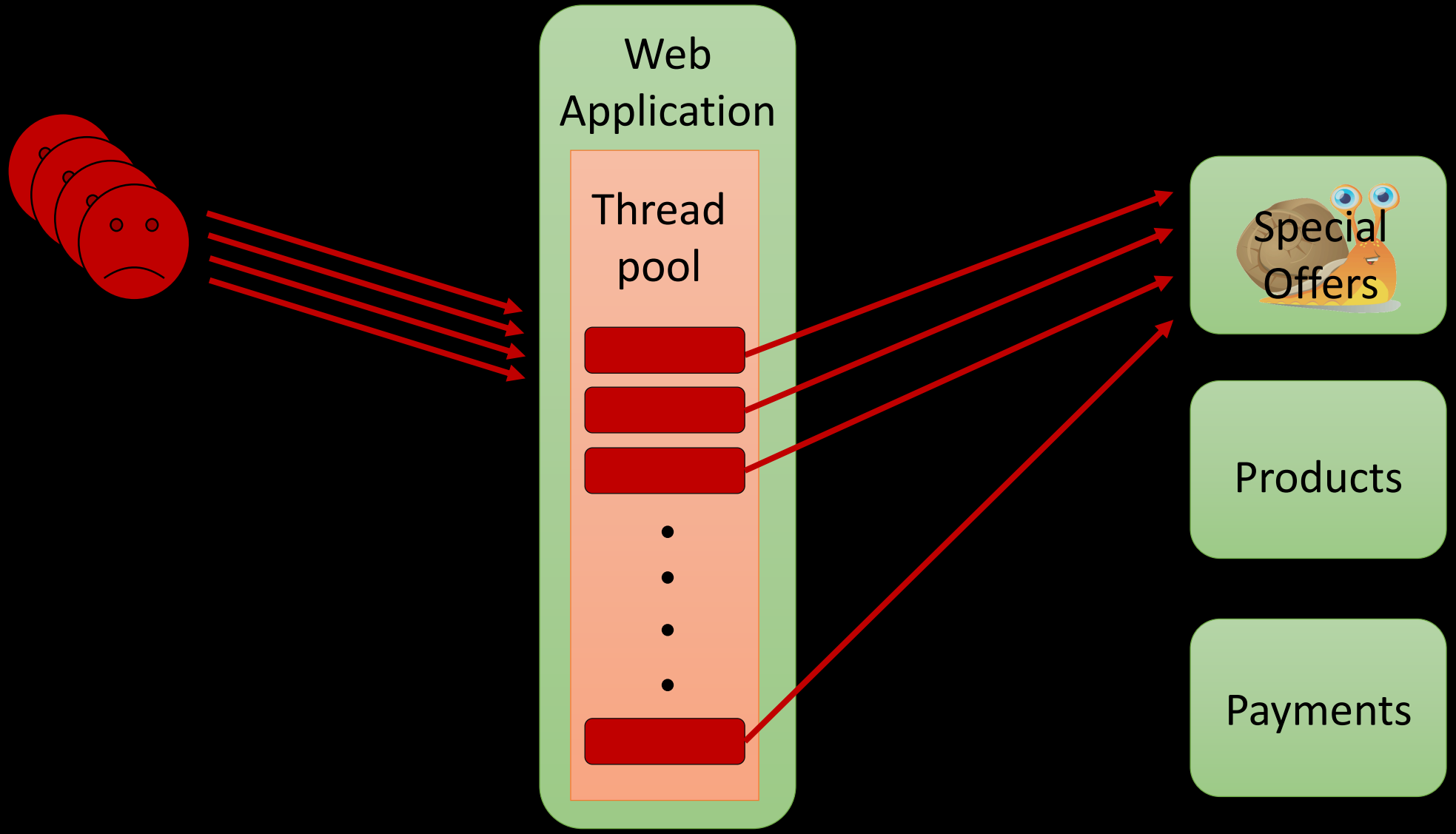

Web
Application
Thread
pool

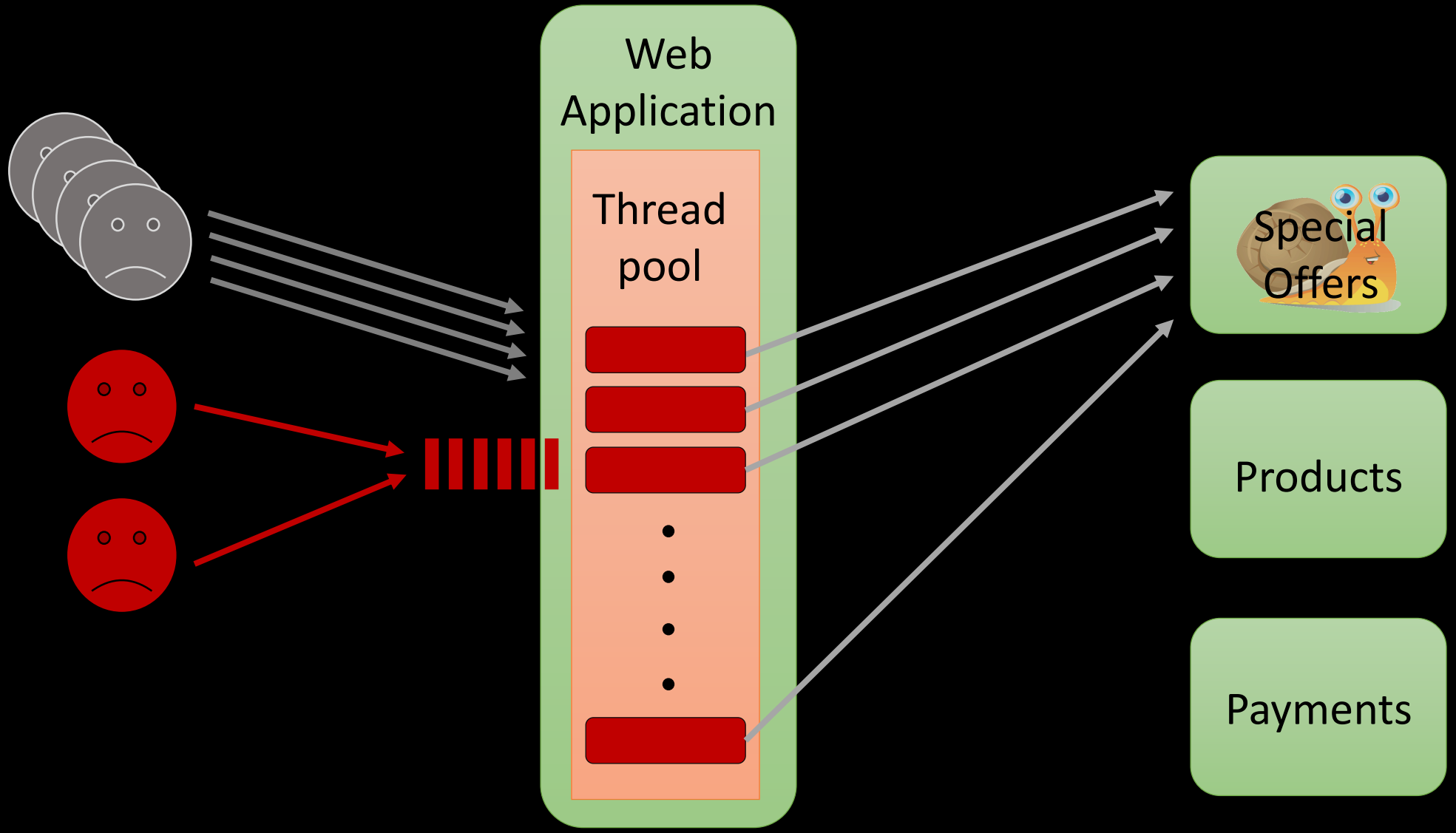

Web
Application
Thread
pool
Special
Offers


Web
Application
Thread
pool
Special
Offers


Web
Application
Thread
pool
Special
Offers


Web Application
Thread pool
Special Offers
Products
Payments


Web
Application
Thread
pool
Special
Offers
Products
Payments


Web Application
Thread pool
Special Offers
Products
Payments

```
URL url = new URL("http://acme-books.com/special-offers");
URLConnection connection = url.openConnection();
connection.connect();
InputStream inputStream = connection.getInputStream();
// Read response from stream
```

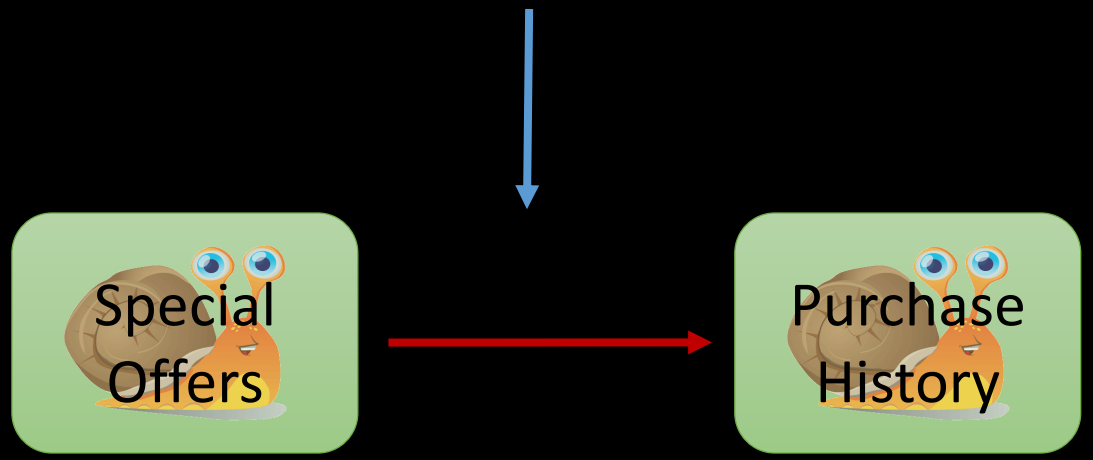
USE TIMEOUTS

Prevents blocked threads

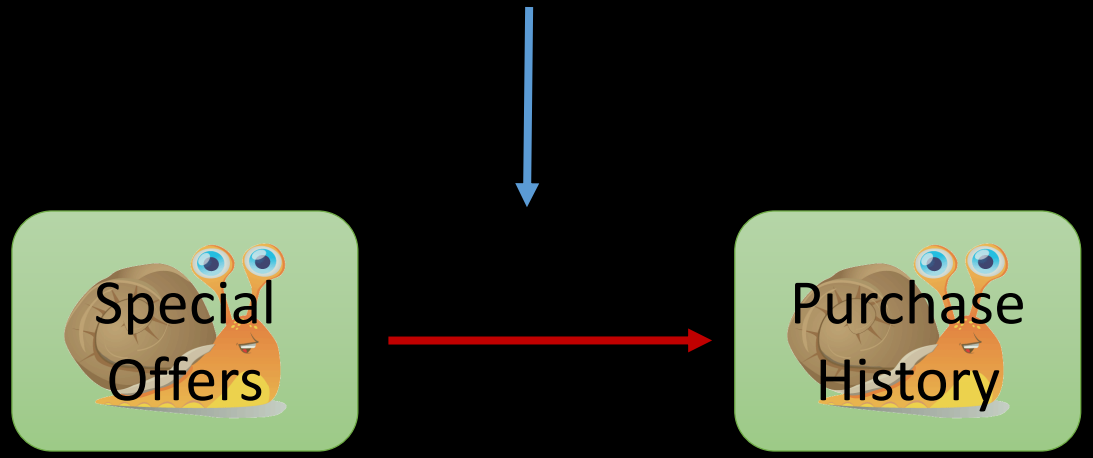
```
URL url = new URL("http://acme-books.com/special-offers");
URLConnection connection = url.openConnection();
connection.setConnectTimeout(100);
connection.setReadTimeout(500);
connection.connect();
InputStream inputStream = connection.getInputStream();
// Read response from stream
```

Set Aggresive Timeouts

Timeouts here?

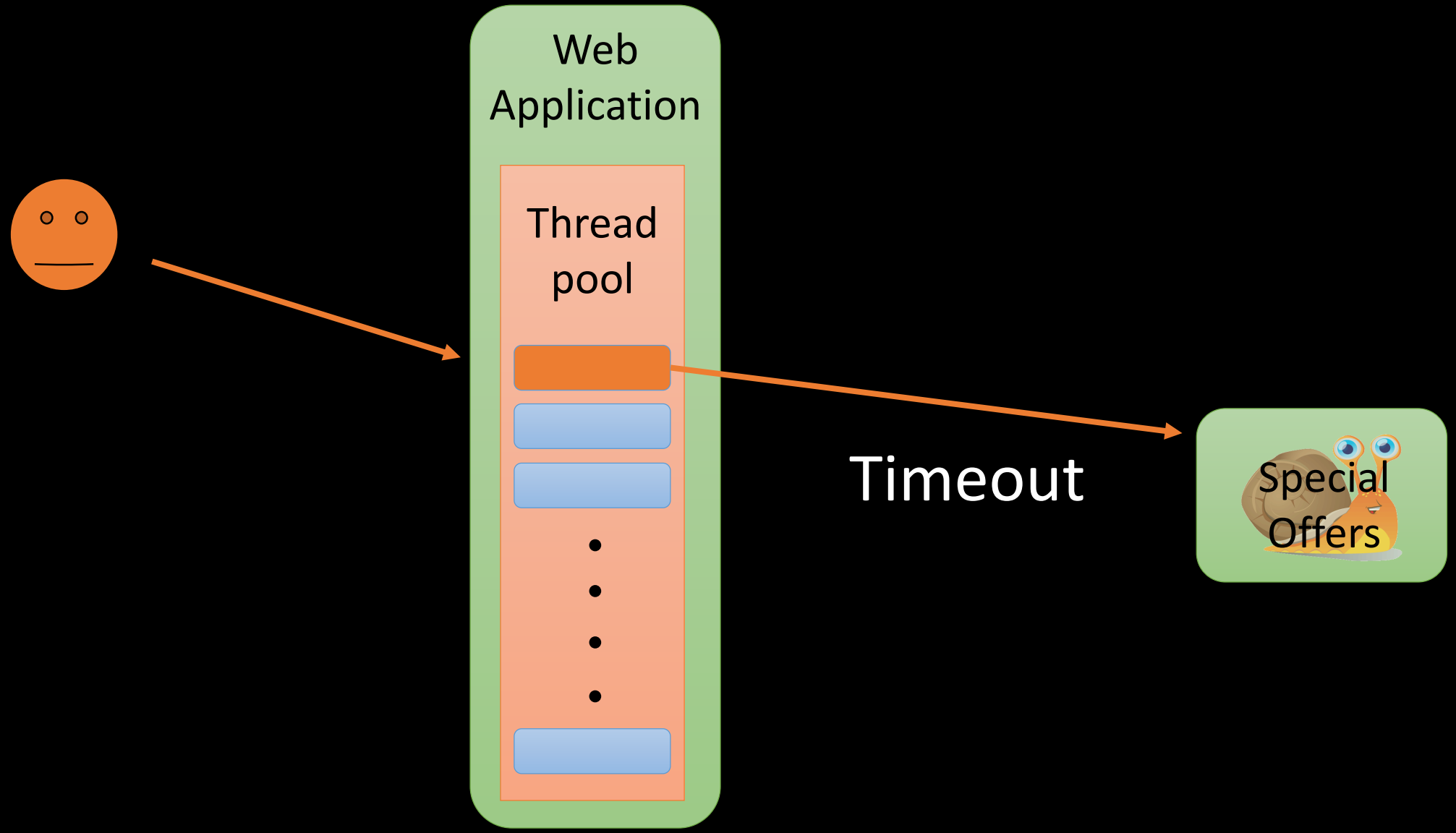


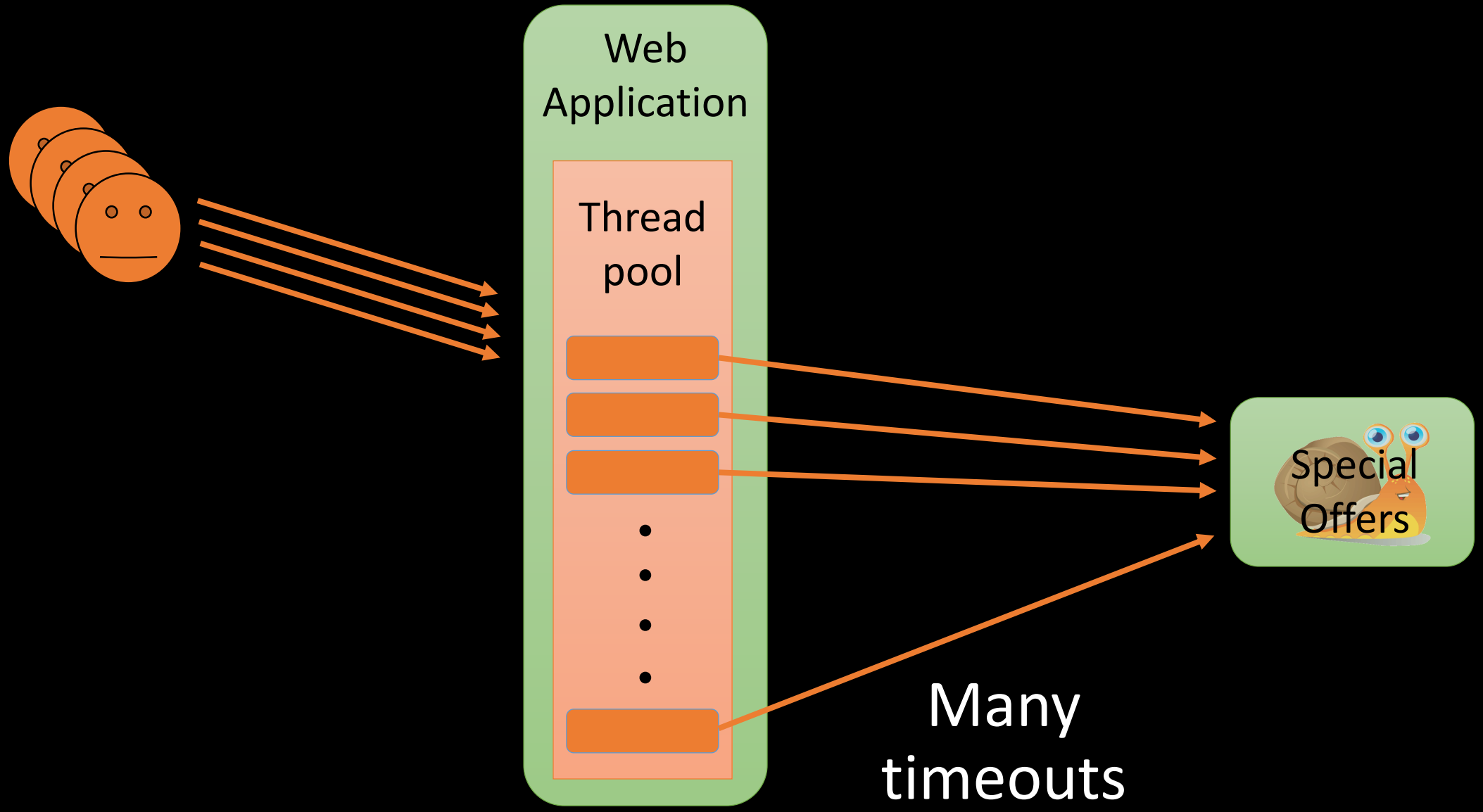
Here too!

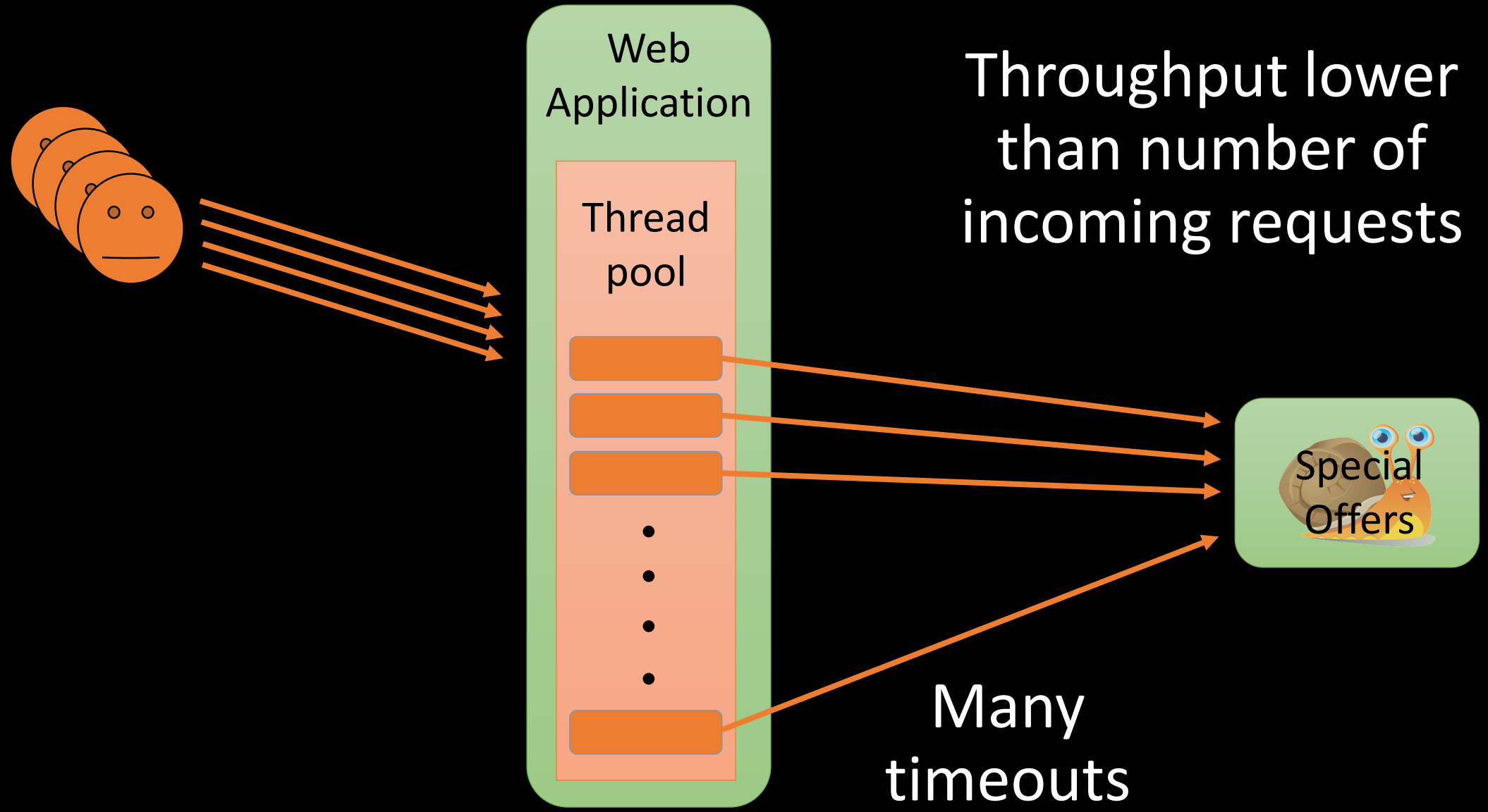


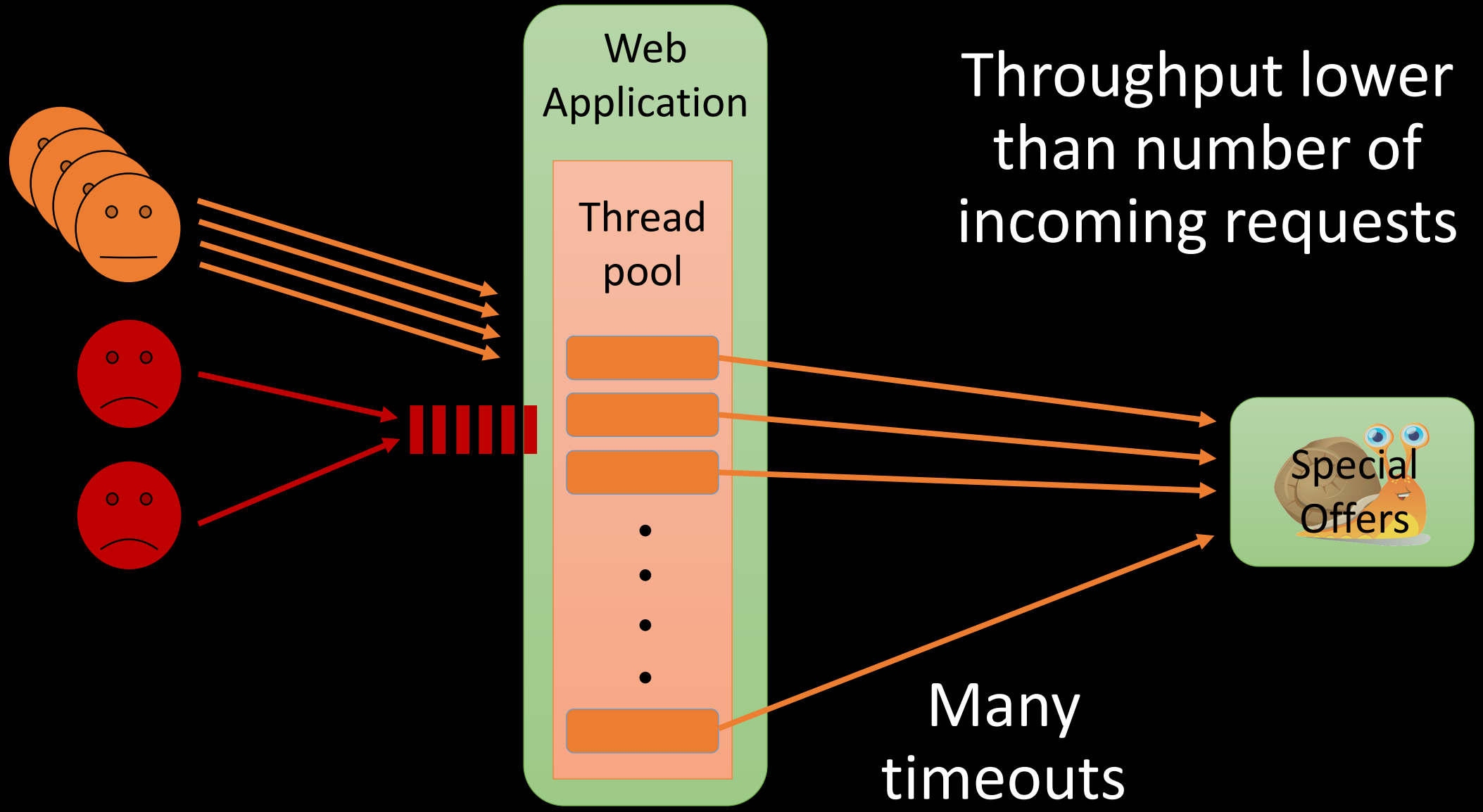
Terrible response times

Awful throughput


Web
Application
Thread
pool
Timeout
Special
Offers

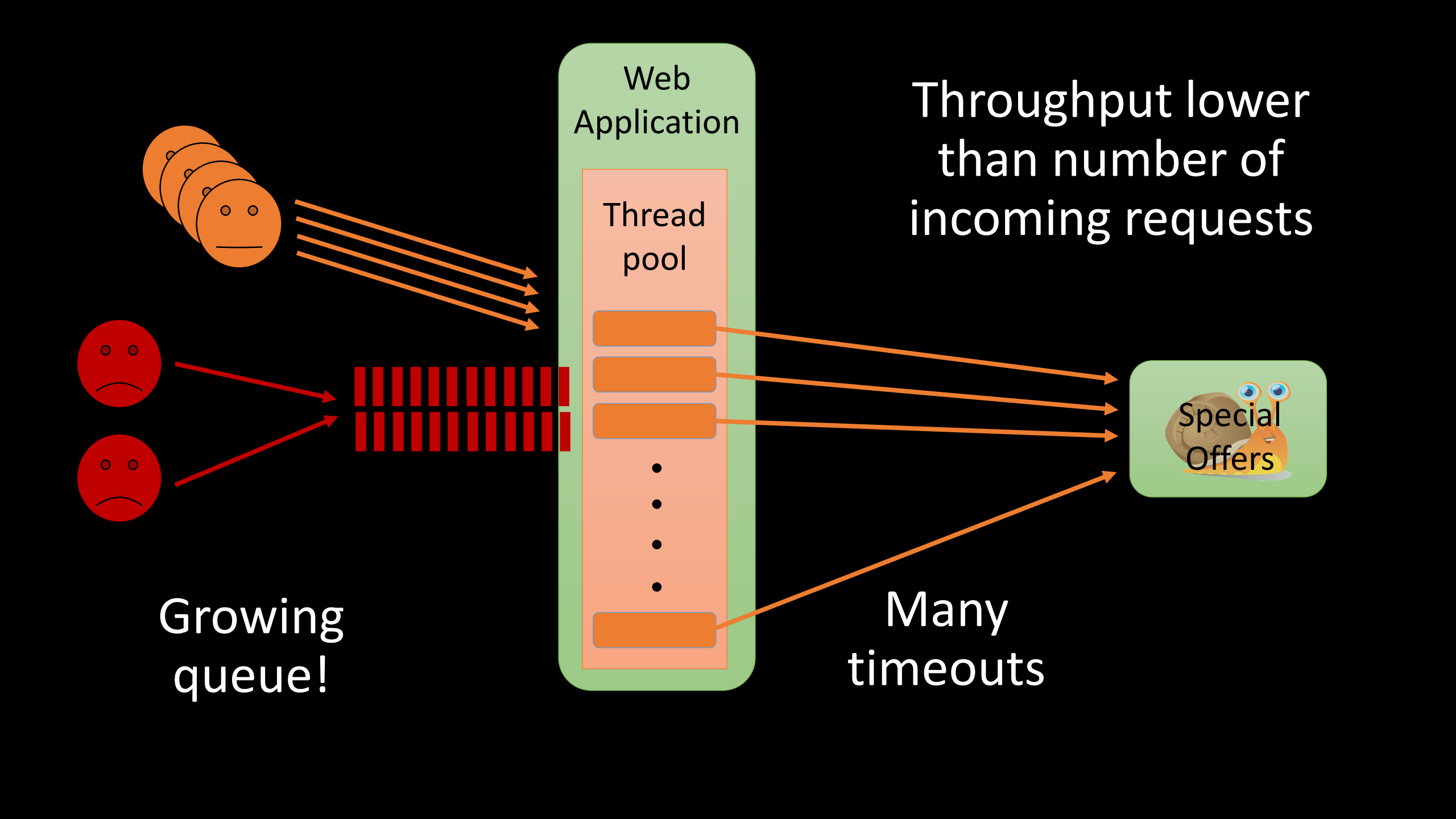

Web
Application
Thread
pool
Special
Offers
Many
timeouts


Web Application
Thread pool
Special Offers
Throughput lower than number of incoming requests
Many timeouts



Throughput lower than number of incoming requests

Many timeouts


Web Application
Thread pool
Throughput lower than number of incoming requests
Special Offers
Growing queue!
Many timeouts

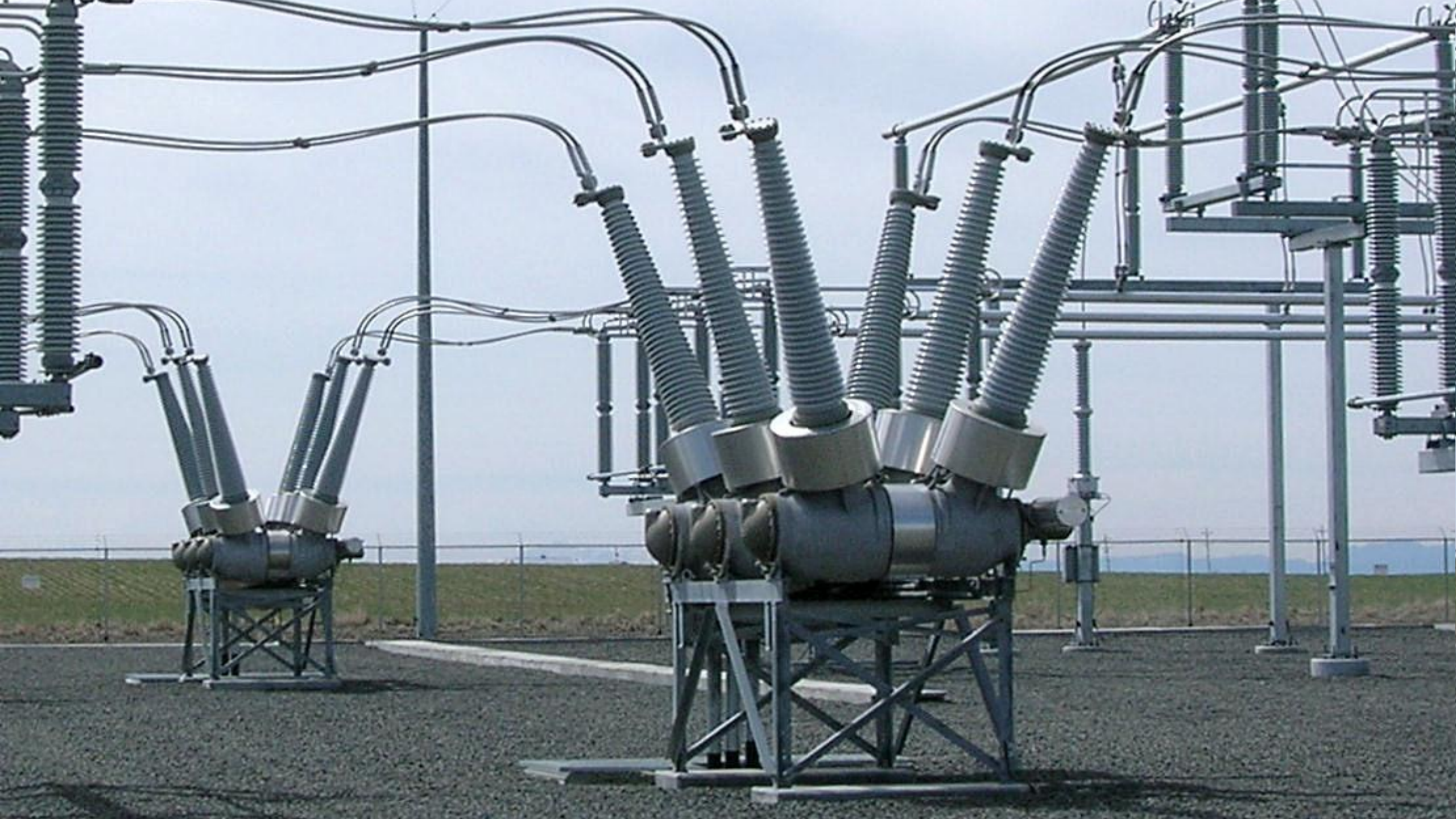
Frequently called service

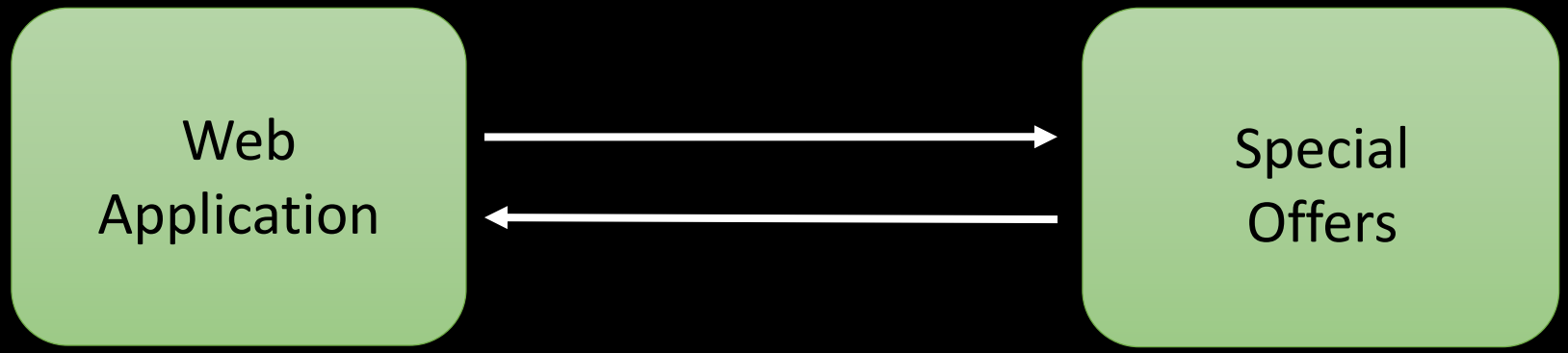
Timeouts are not enough

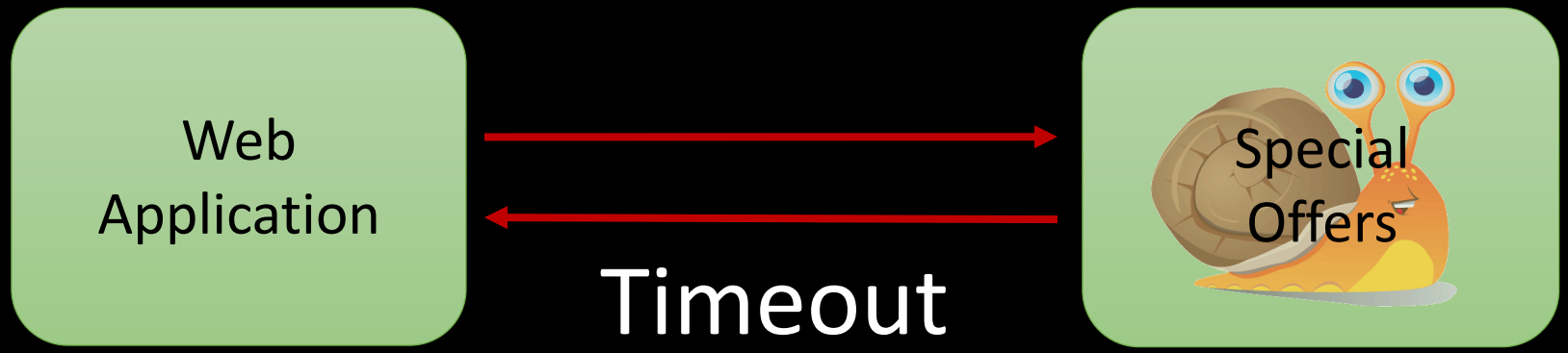
CIRCUIT BREAKERS

Calls to broken services fail fast

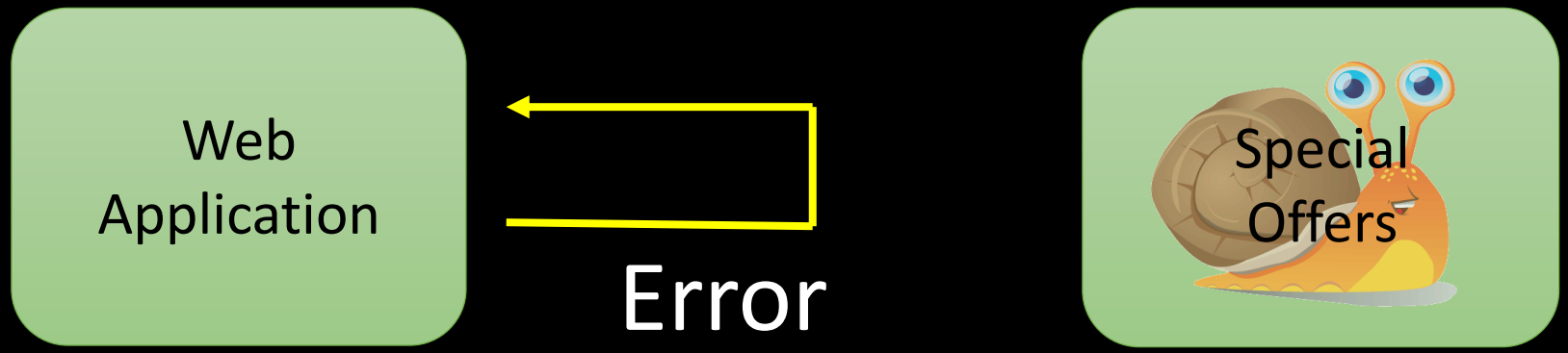
Offloads broken services



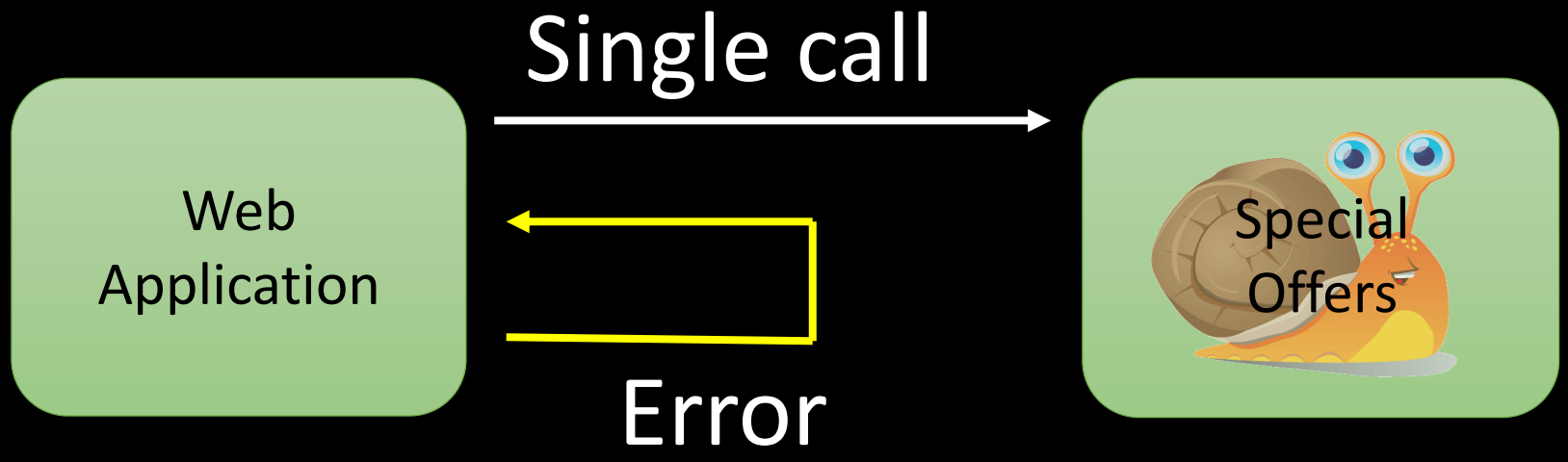

Web
Application
Special
Offers


Web
Application
Special
Offers
Timeout

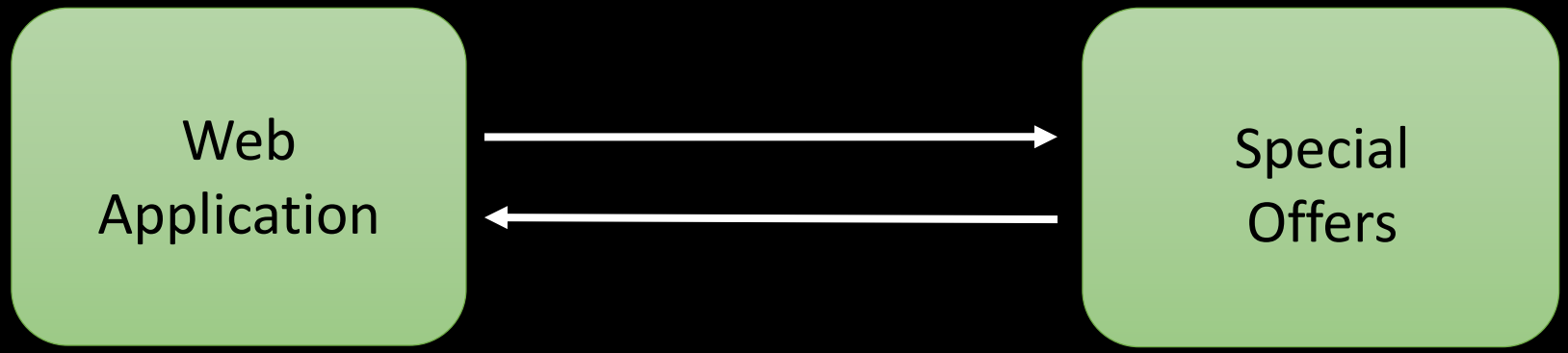
Open state



Half open state



Closed state



Timeouts over threshold

Unhandled errors over threshold

Known irrecoverable error occurs

HTTP ERROR: 500

Problem accessing /. Reason:

Server Error

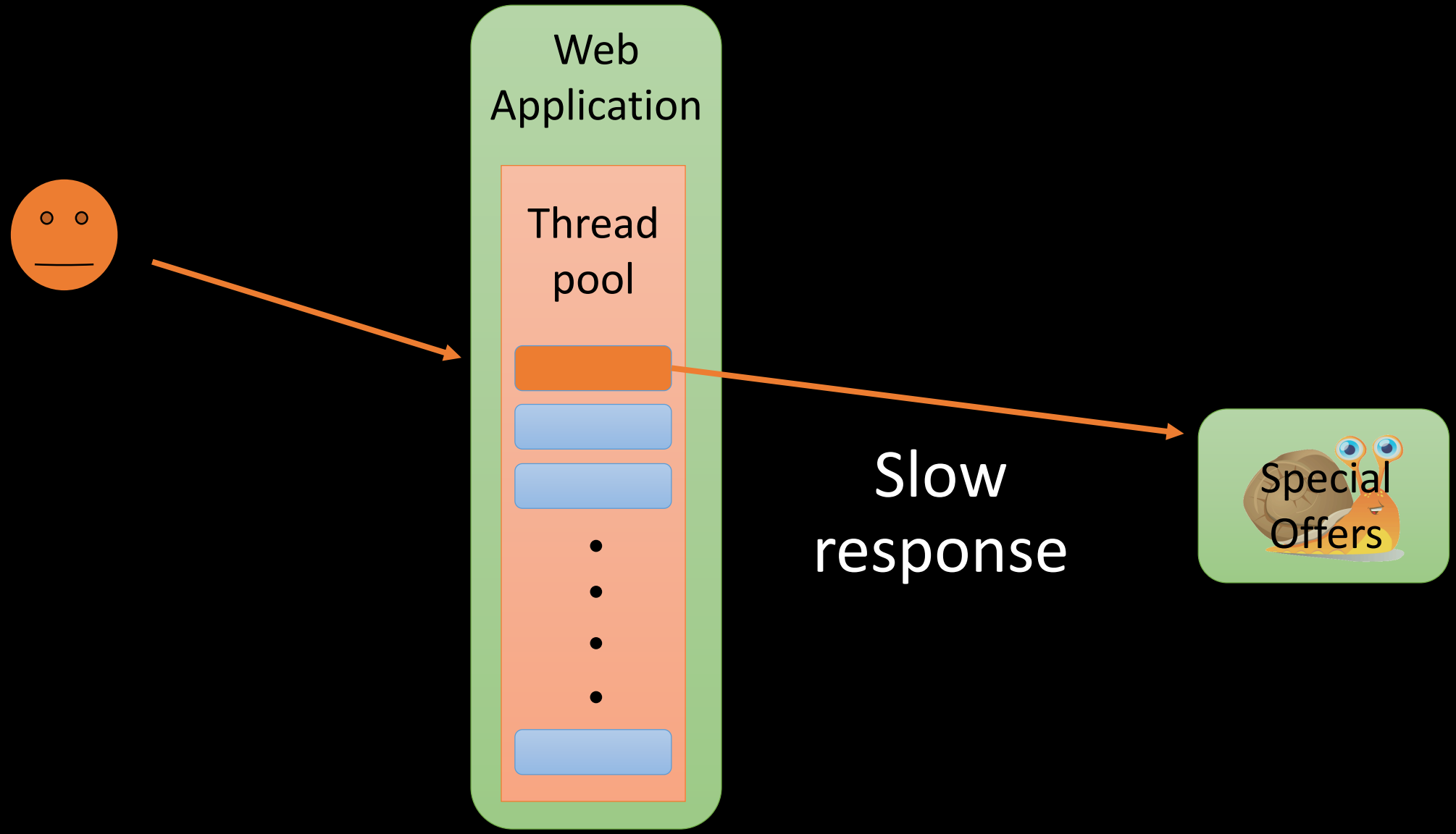
Handle service call errors

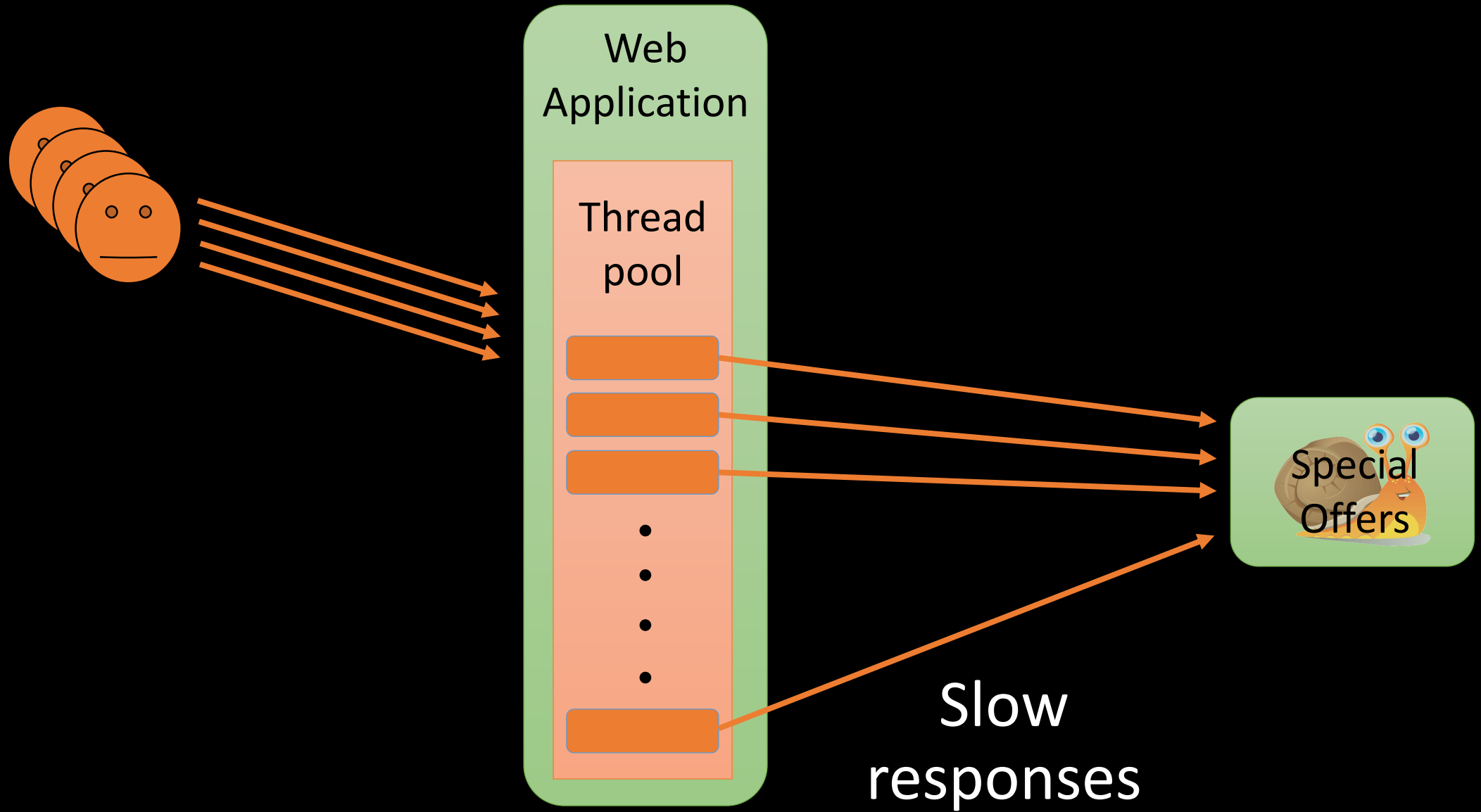
```
try {
    return specialOffers.getOffers();
} catch (Exception e) {
    return Offers.emptyOffers();
}
```

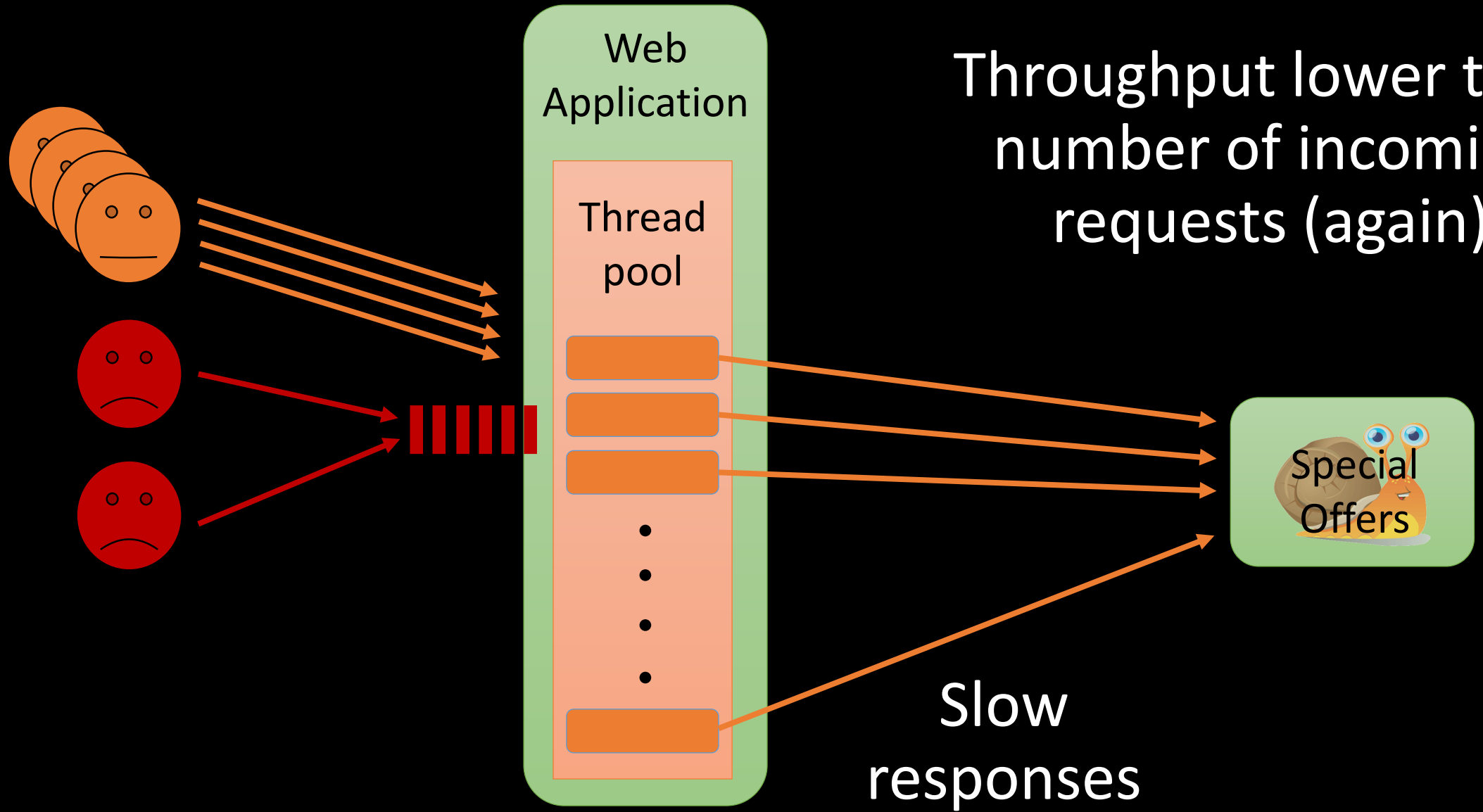
Terrible response times

Awful throughput

Again?!?


Web
Application
Thread
pool
Slow
response
Special
Offers


Web
Application
Thread
pool
Special
Offers
Slow
responses


Web Application
Thread pool
Special Offers
Throughput lower than number of incoming requests (again)
Slow responses

Response time < timeout

Timeouts and circuit breakers are not enough

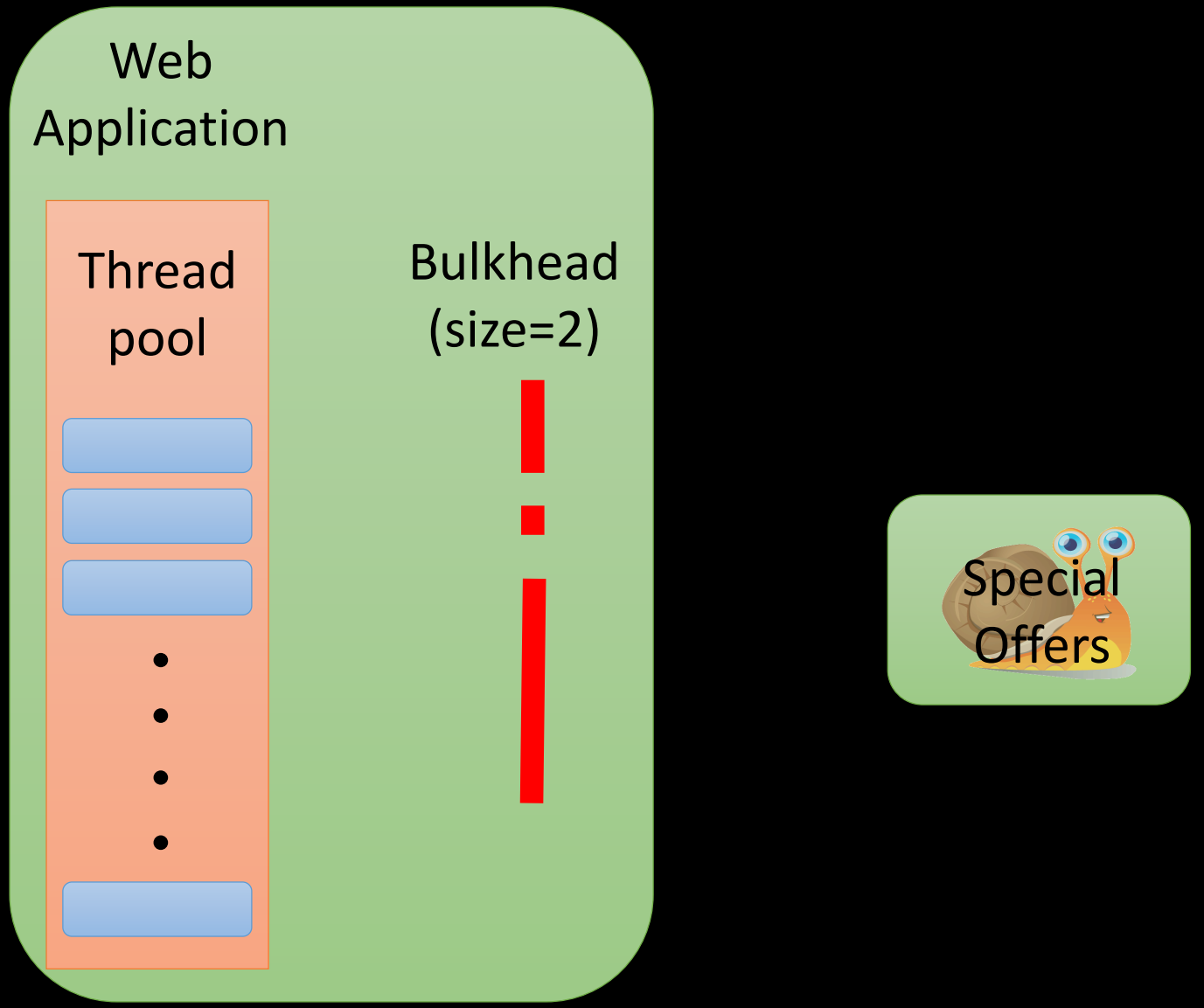
BULKHEADS

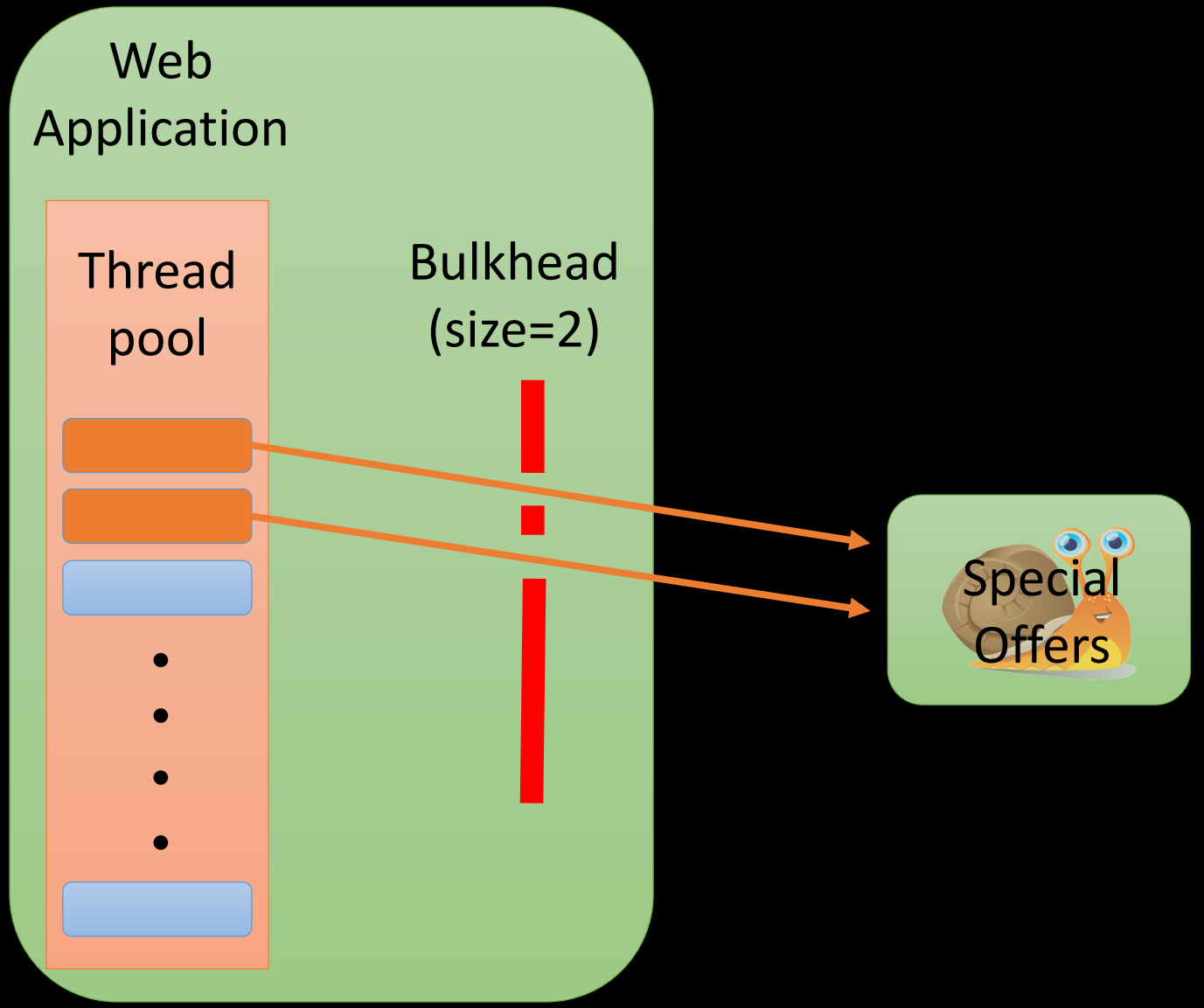
Isolates components

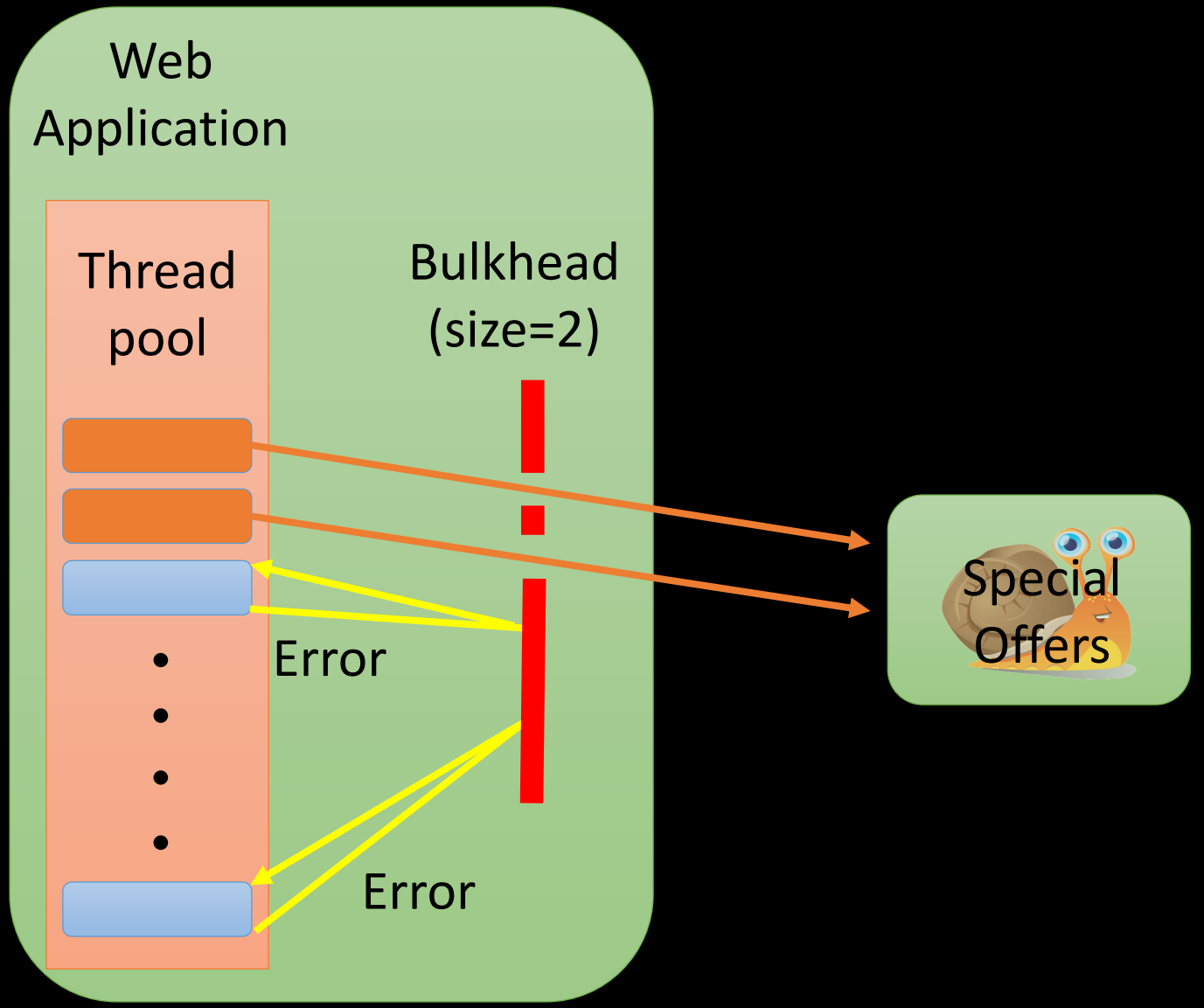
Prevents cascading

Limit number of concurrent calls

Upper bound on number of waiting threads

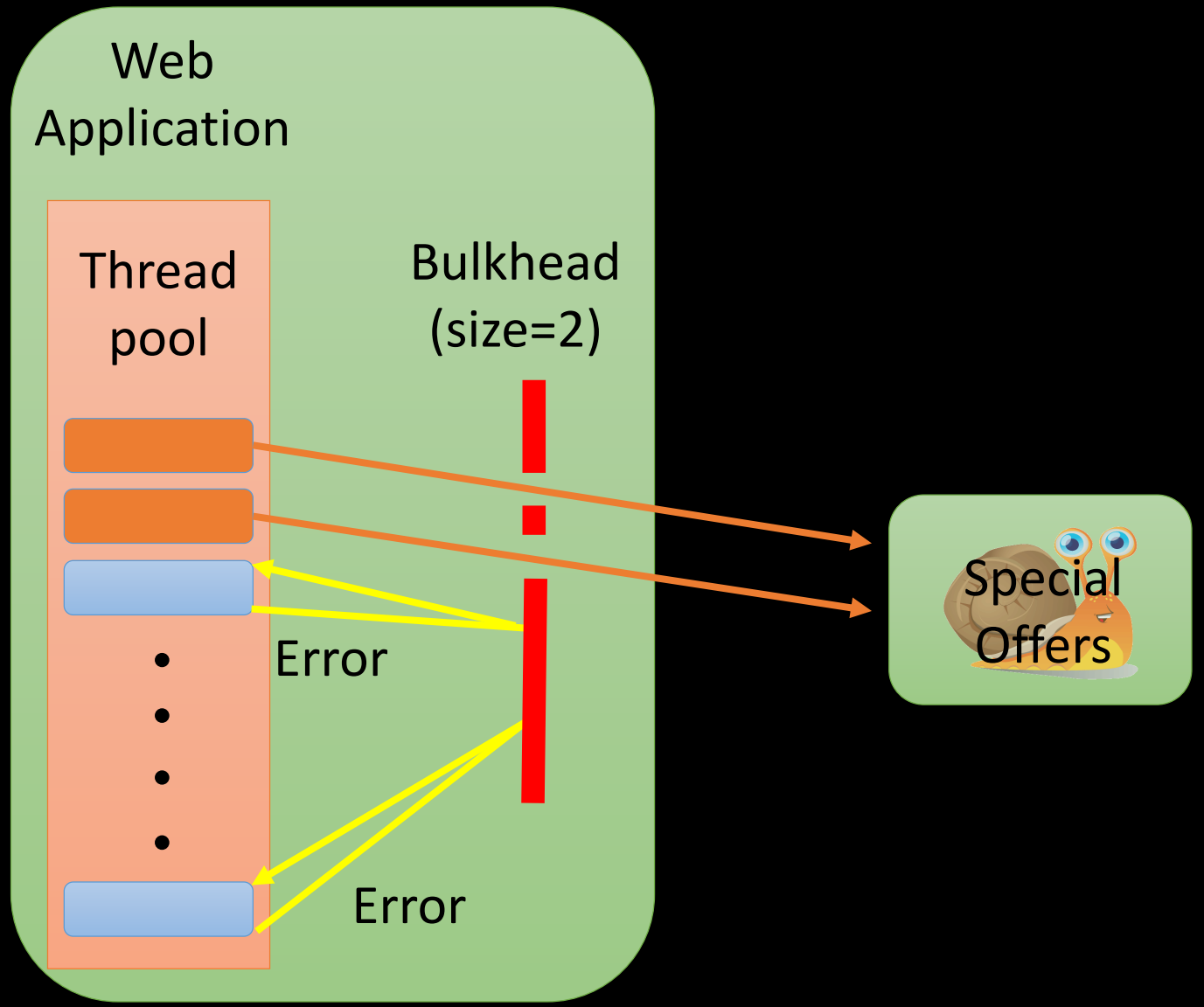

Web
Application
Thread
pool
Bulkhead
(size=2)
Special
Offers


Web
Application
Thread
pool
Bulkhead
(size=2)
Special
Offers

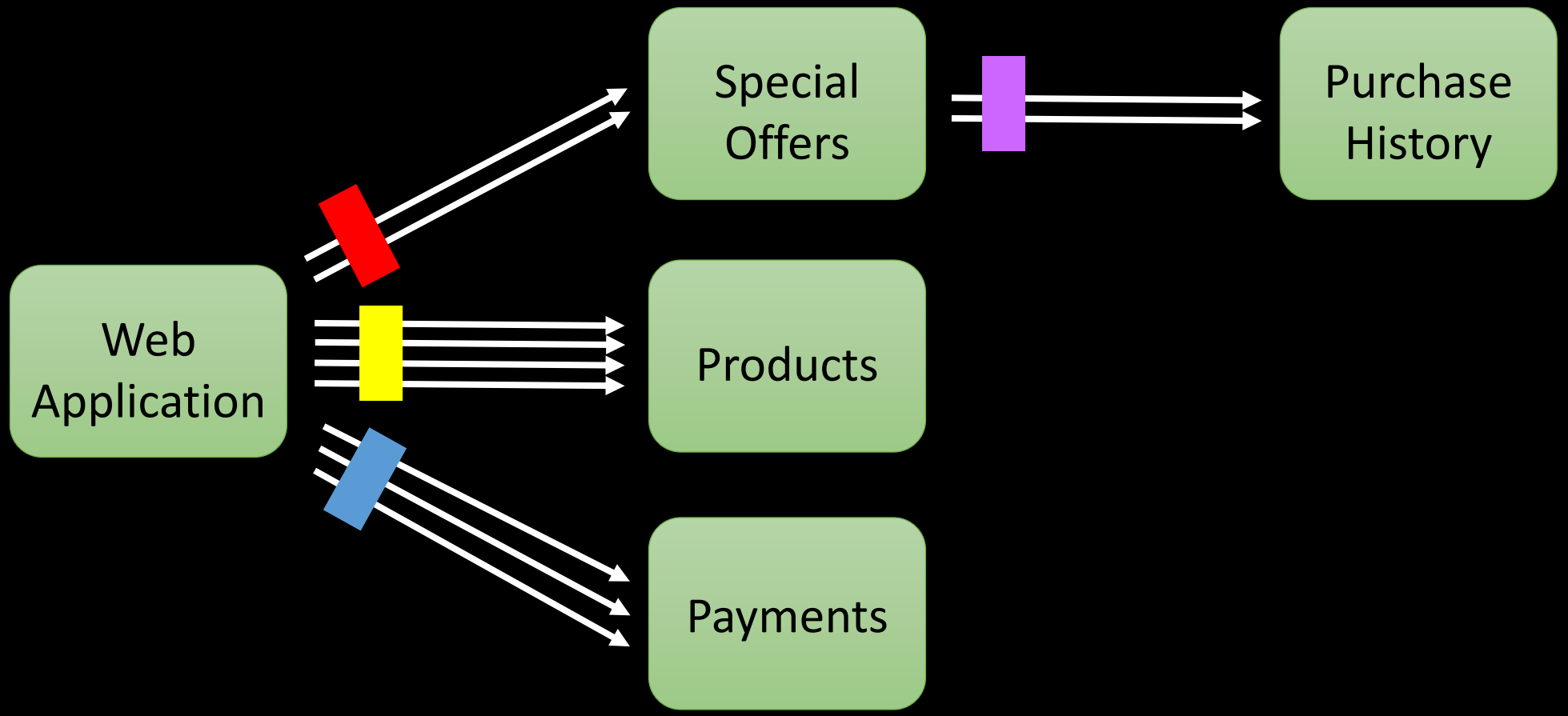

Web
Application
Thread
pool
Bulkhead
(size=2)
Error
Error
Special
Offers

- Slow page load
- Including special offers

- Fast page load
- No special offers



One bulkhead per service



Upper bound on number of waiting threads

Protects very well against cascading failure …

… if bulkhead sizes are …

… significantly smaller than request pool size

Peak load when healthy

40 requests per second (rps)

0.1 seconds response time

Suitable bulkhead size

40 rps x 0.1 seconds = 4

+ breathing room = 7

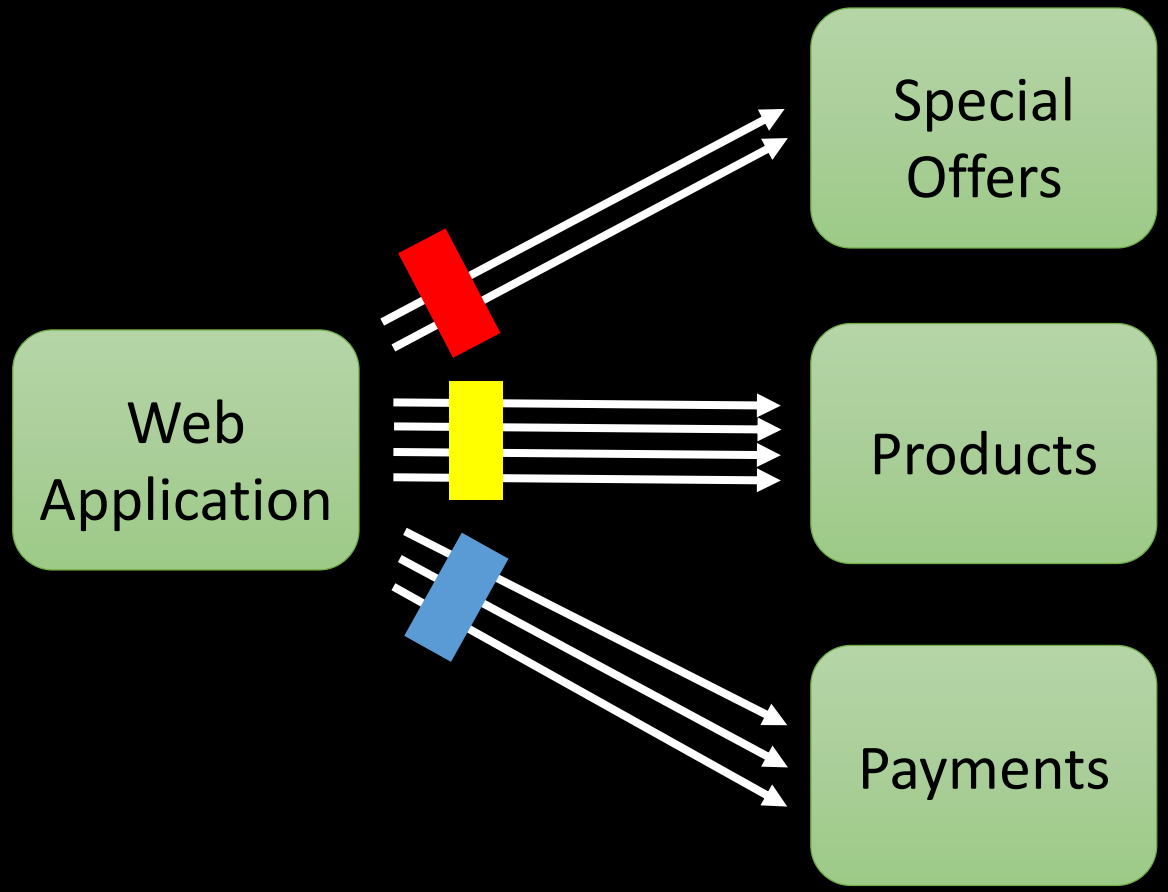
Bonus: protects services from overload

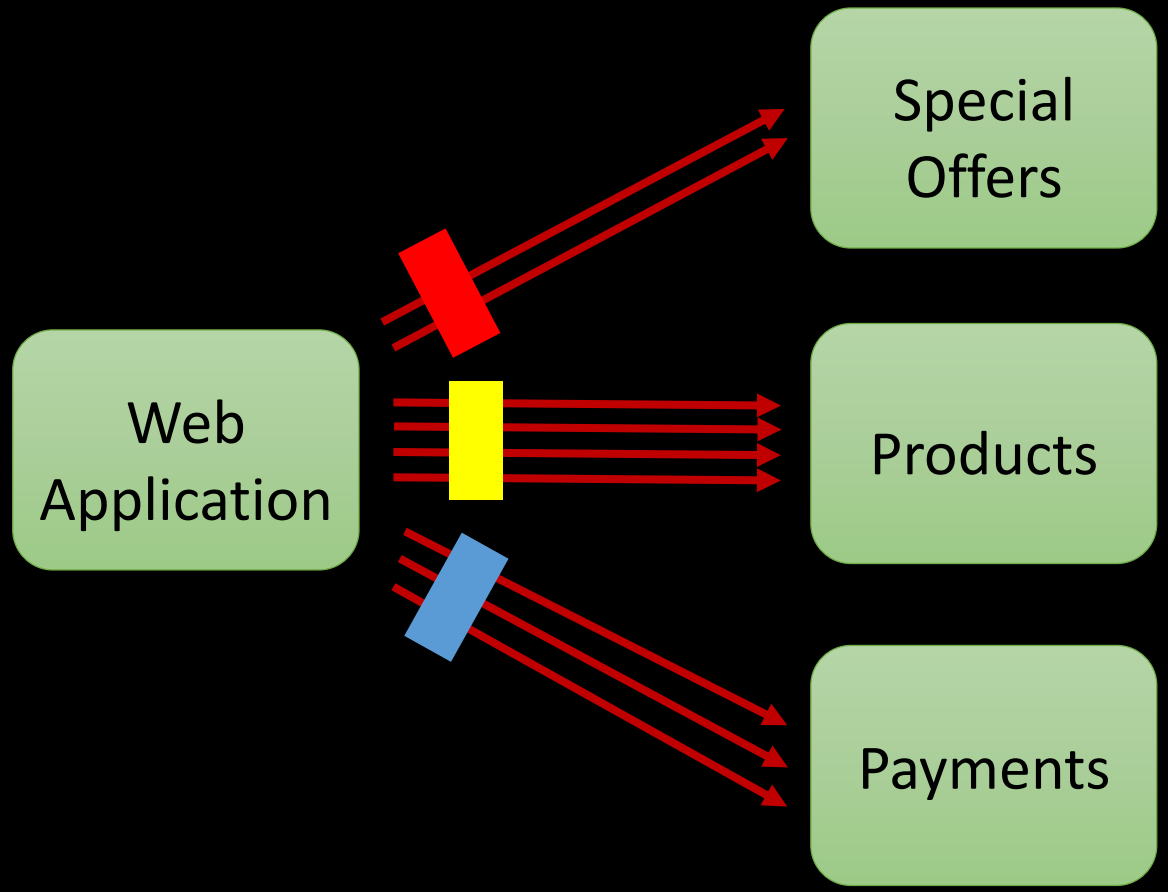
```
Semaphore bulkhead = new Semaphore(2);

Offers protectedGetOffers() {
    if (bulkhead.tryAcquire(0, TimeUnit.SECONDS)) {
        try {
            return specialOffers.getOffers();
        } finally {
            bulkhead.release();
        }
    } else {
        throw new RejectedByBulkheadException();
    }
}
```

Many threads are waiting

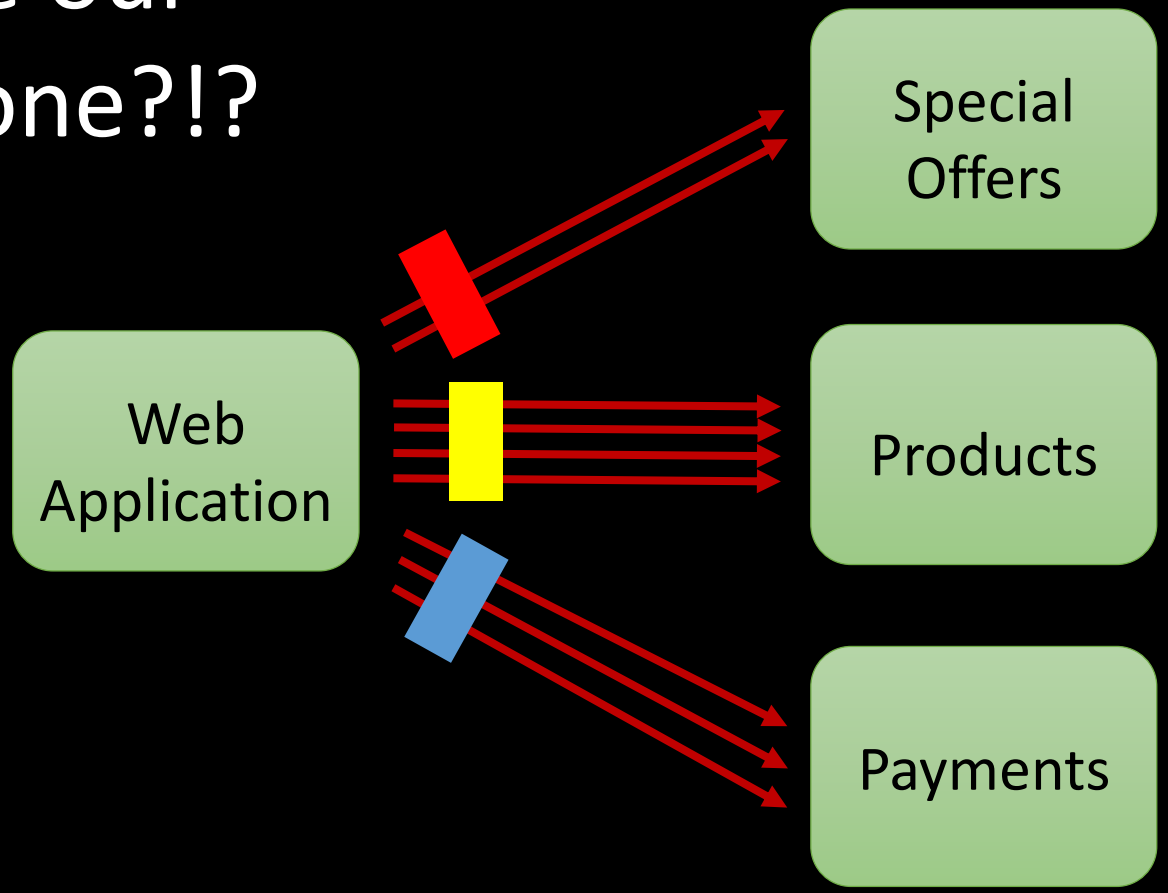
Few available threads - low throughput

All service calls are rejected!


Special Offers
Web Application
Products
Payments


Special Offers
Web Application
Products
Payments

Where have our timeouts gone?!?



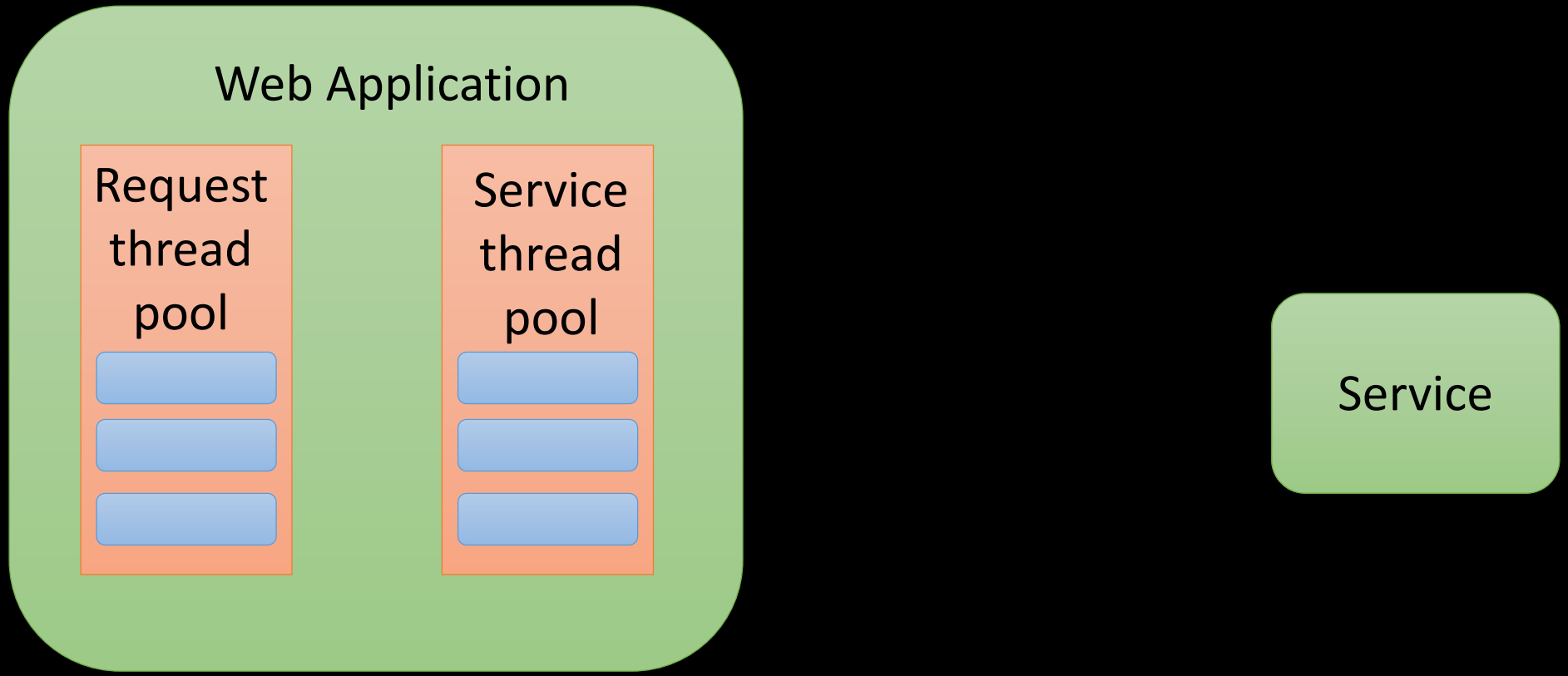
Broken Client Library

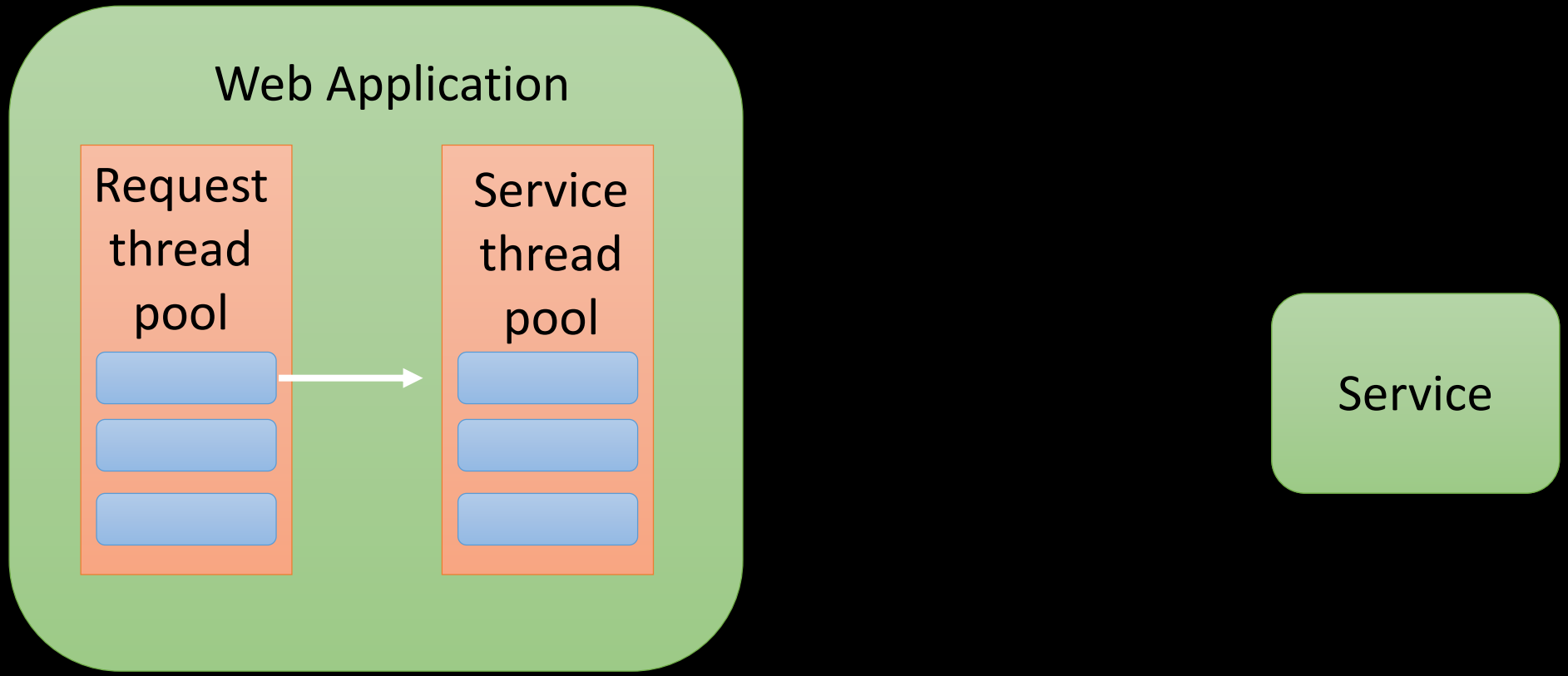
More protection required

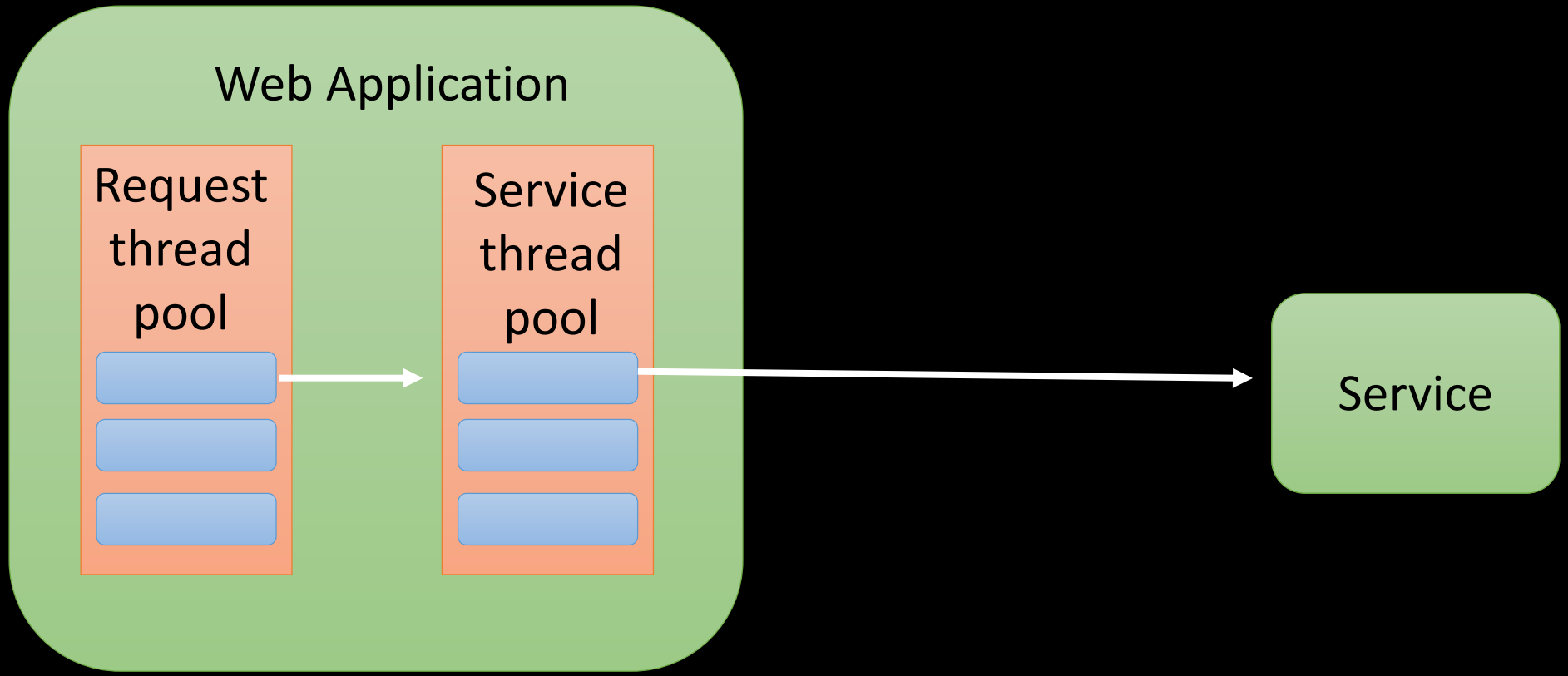
Thread Pool Handovers

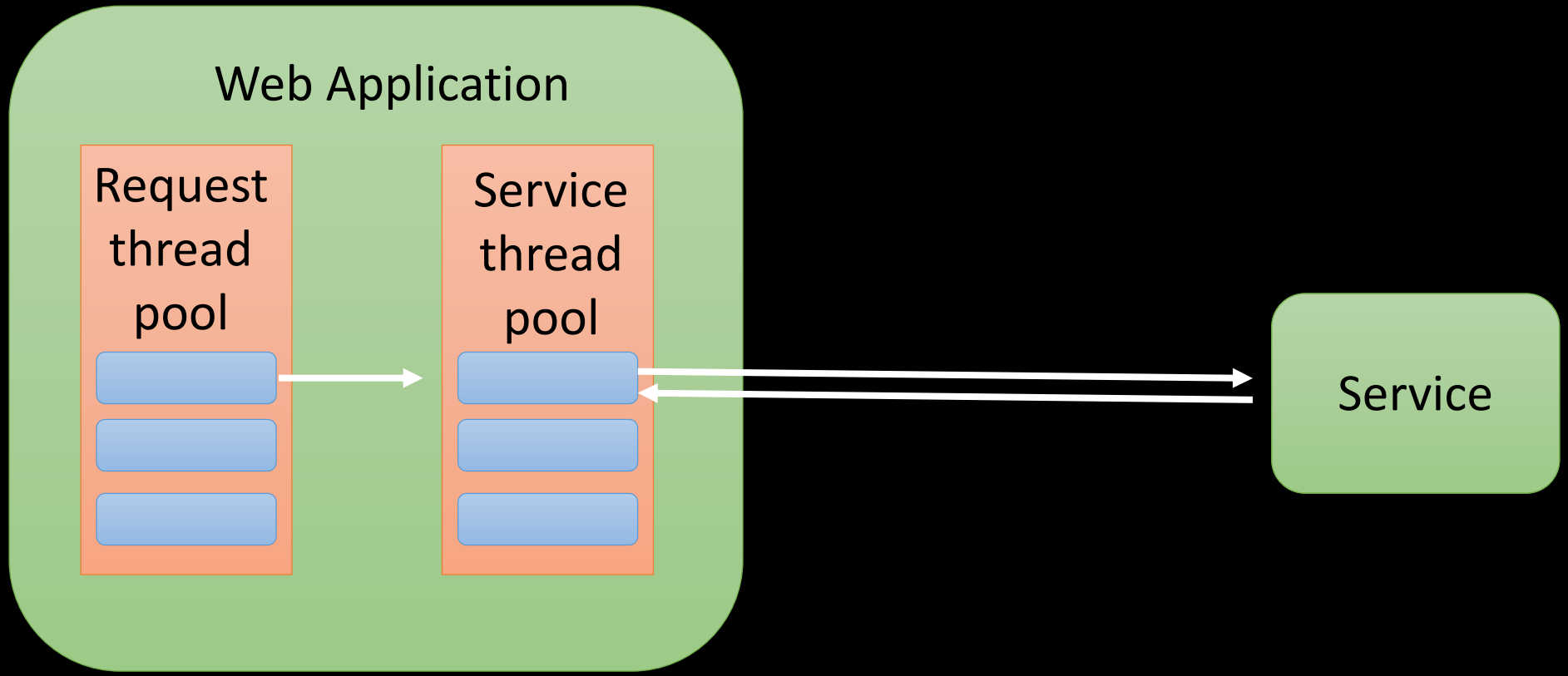
Calling threads can always walk away

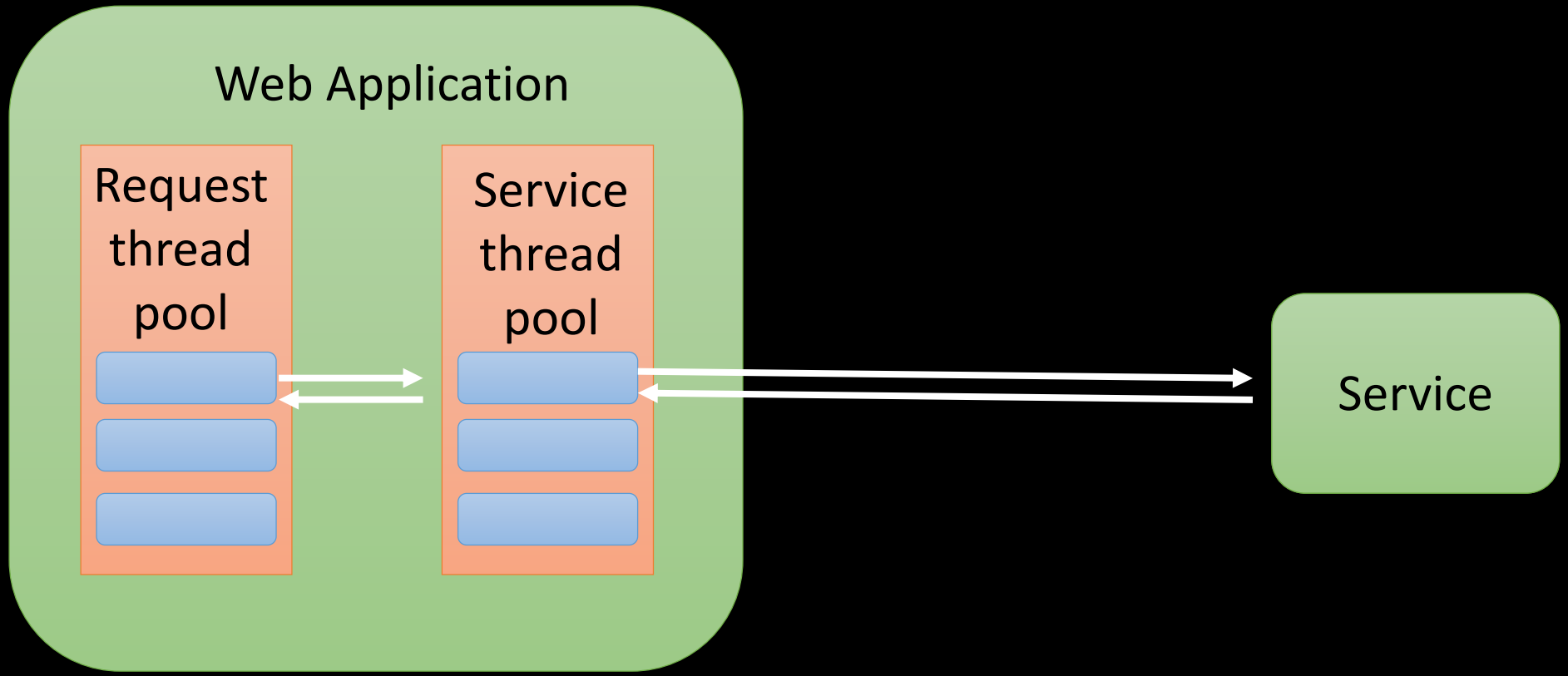
Generic timeouts

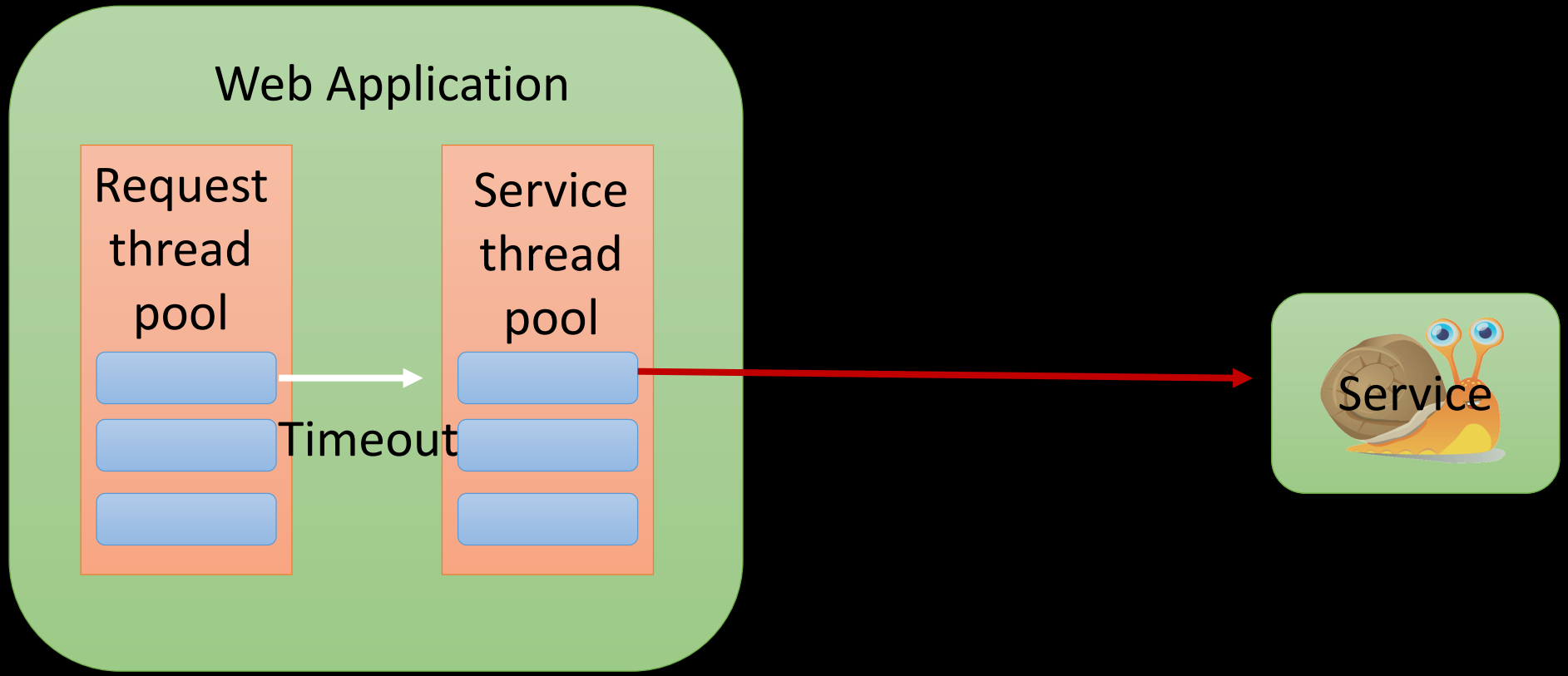

Web Application
Request thread pool
Service thread pool
Service

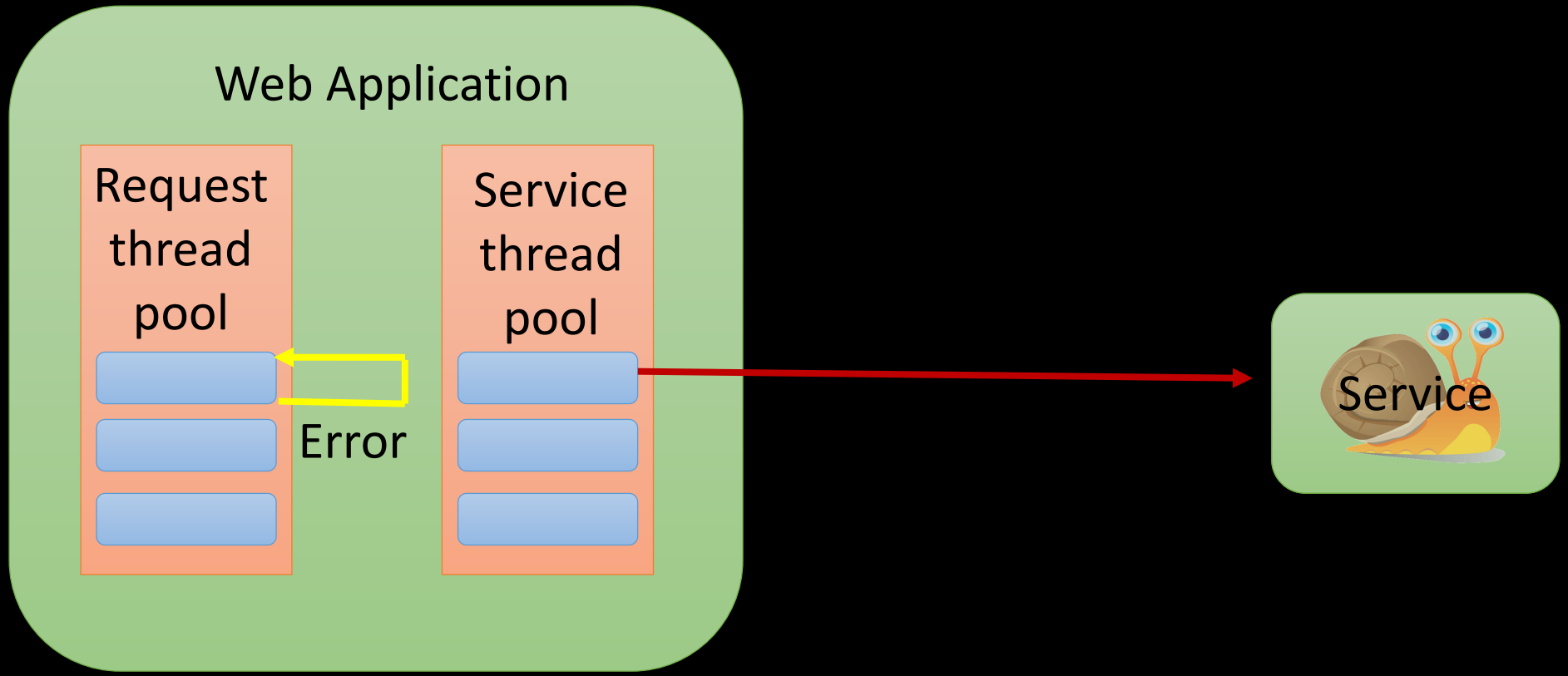

Web Application
Request thread pool
Service thread pool
Service


Web Application
Request thread pool
Service thread pool
Service

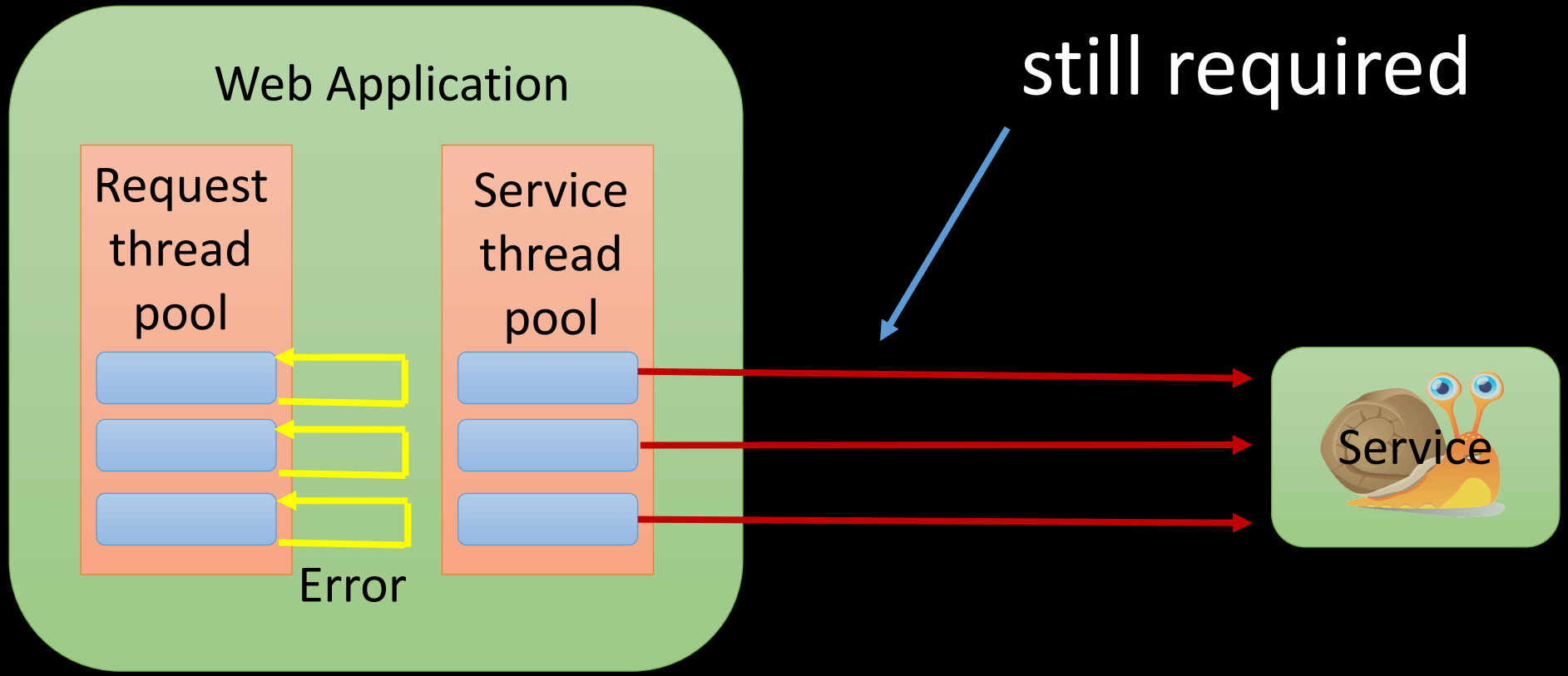

Web Application
Request thread pool
Service thread pool
Service


Web Application
Request thread pool
Service thread pool
Service

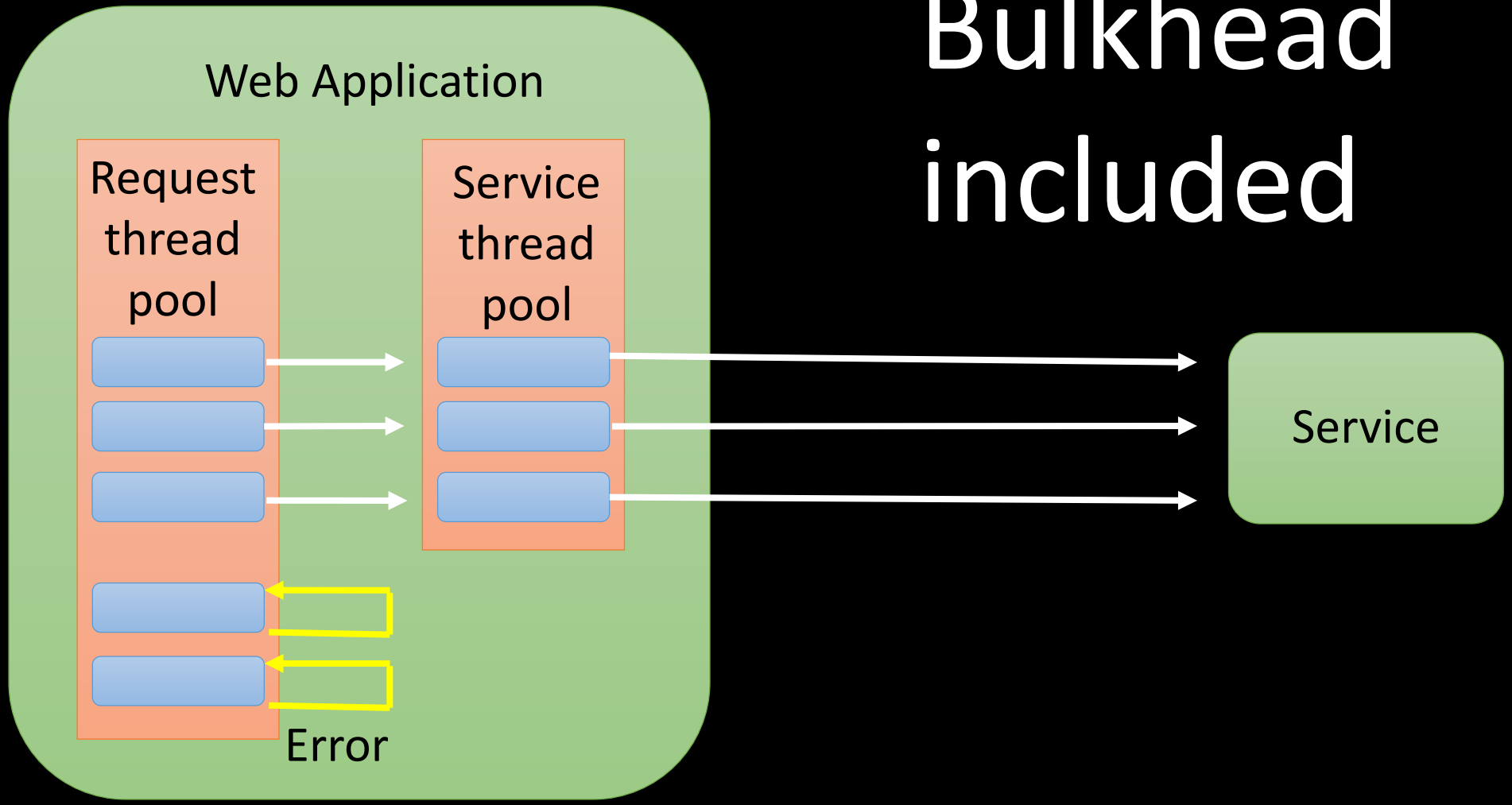

Web Application
Request thread pool
Service thread pool
Timeout
Service


Web Application
Request thread pool
Service thread pool
Error
Service

Service call timeouts still required



Bulkhead included



```
ExecutorService executor = new ThreadPoolExecutor(3, 3, 1,
    TimeUnit.MINUTES, new SynchronousQueue<>());

Offers protectedGetOffers() {
    try {
        Future<Offers> future =
            executor.submit(specialOffers::getOffers);
        return future.get(1, TimeUnit.SECONDS);
    } catch (RejectedExecutionException e) {
        throw new RejectedByBulkheadException();
    } catch (TimeoutException e) {
        throw new ServiceCallTimeoutException();
    }
}
```

Thread pool handovers are very powerful!

What about performance?

Monitor Service Calls

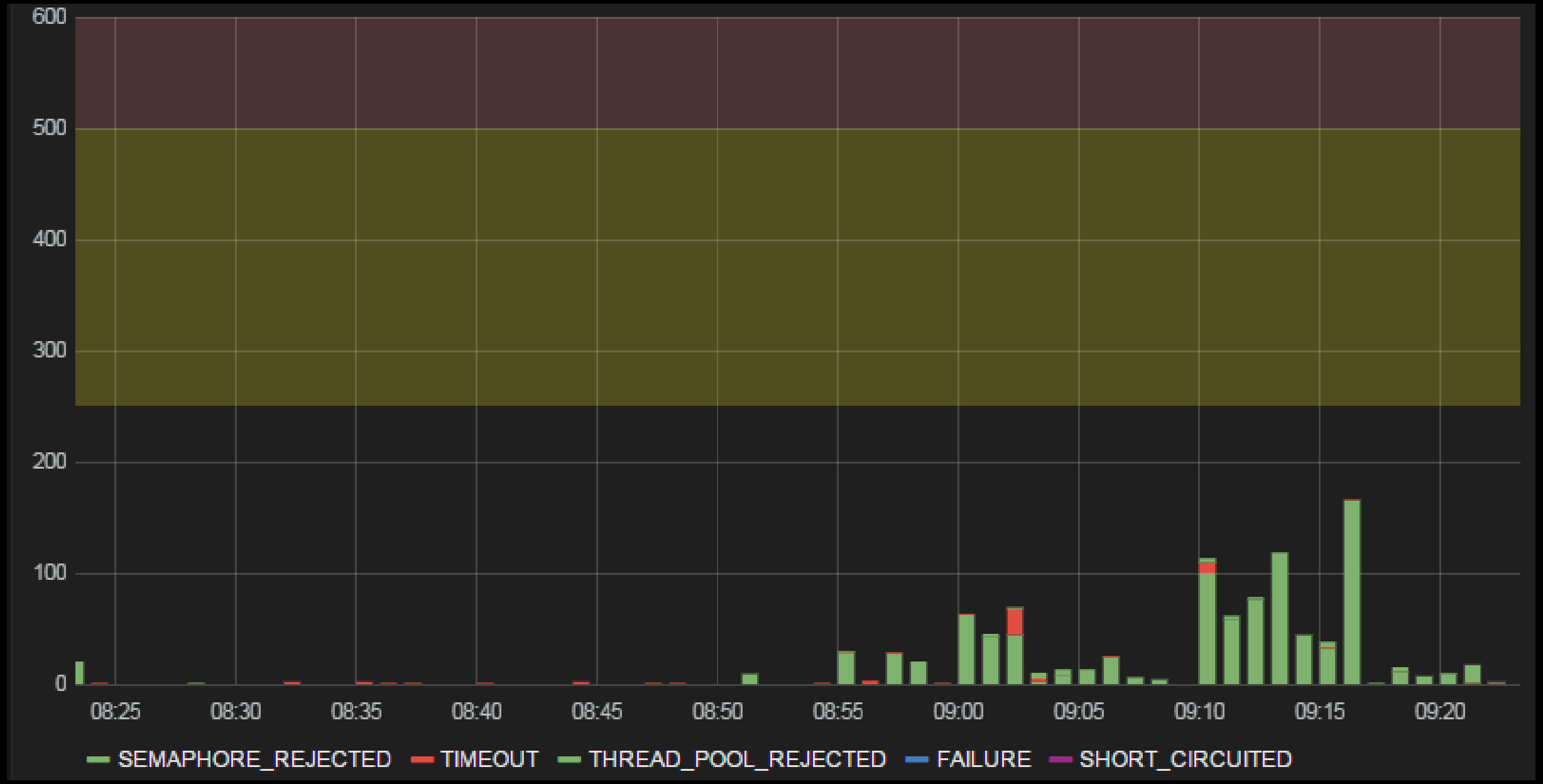
Timeout rate

Failure/success rate

Rejected call rate

Response times

Short circuit rate


600
500
400
300
200
100
0
08:25
08:30
08:35
08:40
08:45
08:50
08:55
09:00
09:05
09:10
09:15
09:20
SEMAPHORE_REJECTED
TIMEOUT
THREAD_POOL_REJECTED
FAILURE
SHORT_CIRCUITED

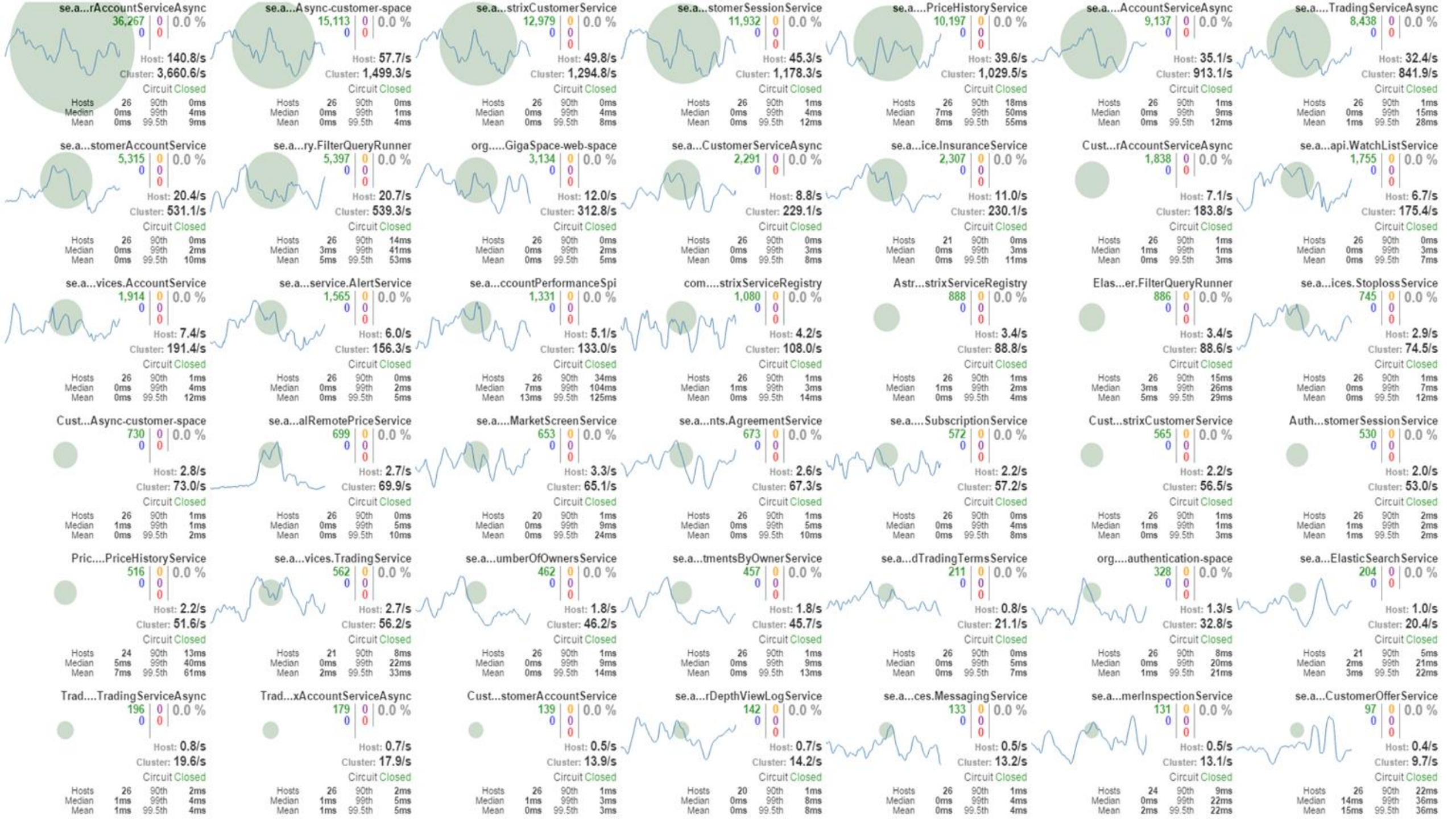
Understand problems before changing configuration

”All this seems like a lot of work!”

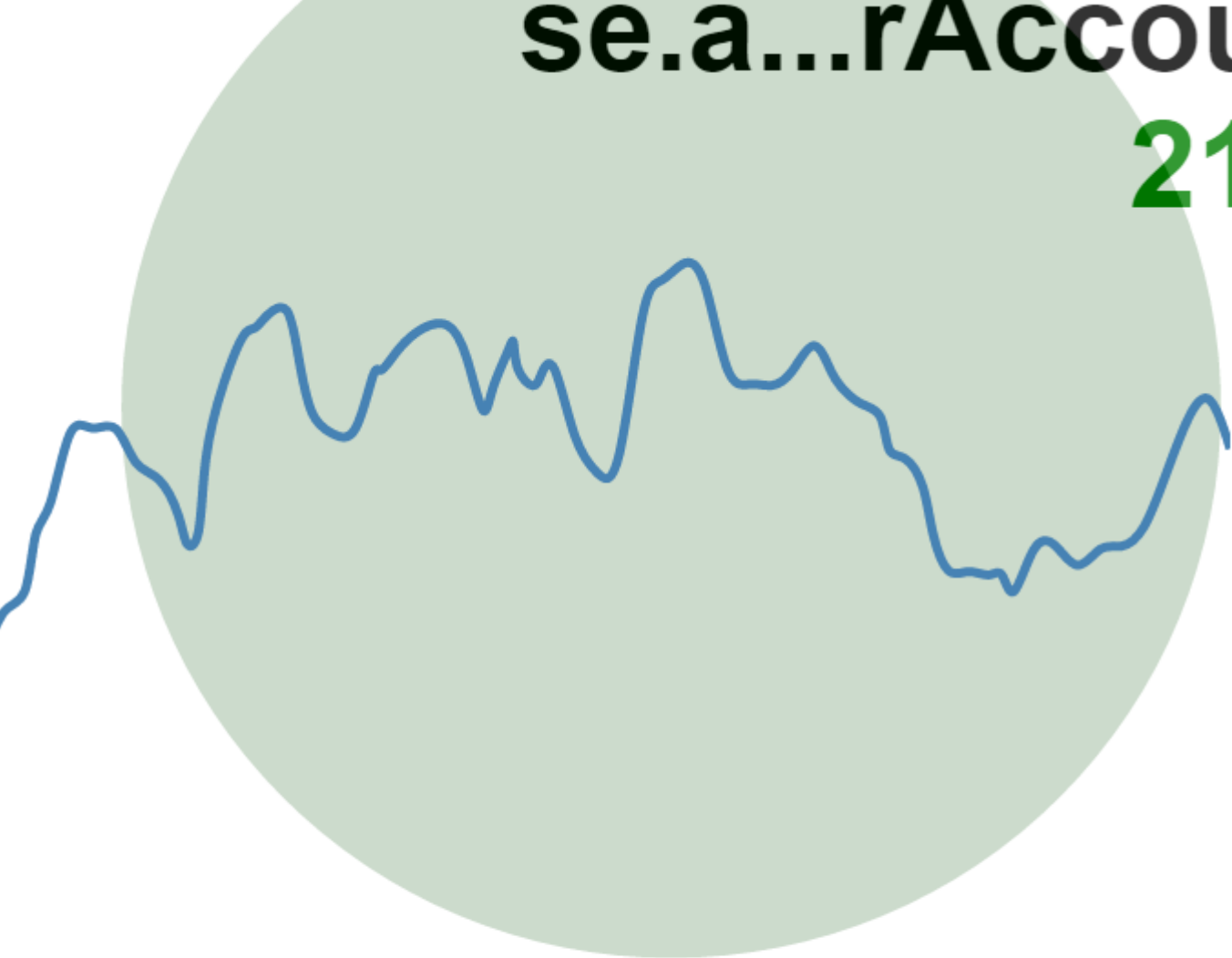

HYSTRIX
DEFEND YOUR APP

```
class GetOffersCommand extends HystrixCommand<Offers> {

    public GetOffersCommand() {
        super(HystrixCommandGroupKey
                .Factory.asKey("SpecialOffers"));
    }

    @Override
    protected Offers run() throws Exception {
        return specialOffers.getOffers();
    }
}

public Offers getOffers() {
    return new GetOffersCommand().execute();
}
```

```
class GetOffersCommand extends HystrixCommand<Offers> {

    // ...

    @Override
    protected Offers getFallback() {
        return Offers.emptyOffers();
    }
}
```


se.a...rAccountServiceAsync
36,267 0 0.0 %
0 0
Host: 140.8/s
Cluster: 3,660.6/s
Circuit Closed
Hosts 26 90th 0ms
Median 0ms 99th 4ms
Mean 0ms 99.5th 9ms
se.a...Async-customer-space
15,113 0 0.0 %
0 0
Host: 57.7/s
Cluster: 1,499.3/s
Circuit Closed
Hosts 26 90th 0ms
Median 0ms 99th 1ms
Mean 0ms 99.5th 4ms
se.a...strixCustomerService
12,979 0 0.0 %
0 0
Host: 49.8/s
Cluster: 1,294.8/s
Circuit Closed
Hosts 26 90th 0ms
Median 0ms 99th 4ms
Mean 0ms 99.5th 8ms
se.a...stomerSessionService
11,932 0 0.0 %
0 0
Host: 45.3/s
Cluster: 1,178.3/s
Circuit Closed
Hosts 26 90th 1ms
Median 0ms 99th 4ms
Mean 0ms 99.5th 12ms
se.a....PriceHistoryService
10,197 0 0.0 %
0 0
Host: 39.6/s
Cluster: 1,029.5/s
Circuit Closed
Hosts 26 90th 18ms
Median 7ms 99th 50ms
Mean 8ms 99.5th 55ms
se.a....AccountServiceAsync
9,137 0 0.0 %
0 0
Host: 35.1/s
Cluster: 913.1/s
Circuit Closed
Hosts 26 90th 1ms
Median 0ms 99th 9ms
Mean 0ms 99.5th 12ms
se.a....TradingServiceAsync
8,438 0 0.0 %
0 0
Host: 32.4/s
Cluster: 841.9/s
Circuit Closed
Hosts 26 90th 1ms
Median 0ms 99th 15ms
Mean 1ms 99.5th 28ms
se.a...stomerAccountService
5,315 0 0.0 %
0 0
Host: 20.4/s
Cluster: 531.1/s
Circuit Closed
Hosts 26 90th 0ms
Median 0ms 99th 2ms
Mean 0ms 99.5th 10ms
se.a...ry.FilterQueryRunner
5,397 0 0.0 %
0 0
Host: 20.7/s
Cluster: 539.3/s
Circuit Closed
Hosts 26 90th 14ms
Median 3ms 99th 41ms
Mean 5ms 99.5th 53ms
org.....GigaSpace-web-space
3,134 0 0.0 %
0 0
Host: 12.0/s
Cluster: 312.8/s
Circuit Closed
Hosts 26 90th 0ms
Median 0ms 99th 2ms
Mean 0ms 99.5th 5ms
se.a...CustomerServiceAsync
2,291 0 0.0 %
0 0
Host: 8.8/s
Cluster: 229.1/s
Circuit Closed
Hosts 26 90th 0ms
Median 0ms 99th 3ms
Mean 0ms 99.5th 8ms
se.a...ice.InsuranceService
2,307 0 0.0 %
0 0
Host: 11.0/s
Cluster: 230.1/s
Circuit Closed
Hosts 21 90th 0ms
Median 0ms 99th 3ms
Mean 0ms 99.5th 11ms
Cust...rAccountServiceAsync
1,838 0 0.0 %
0 0
Host: 7.1/s
Cluster: 183.8/s
Circuit Closed
Hosts 26 90th 1ms
Median 1ms 99th 1ms
Mean 0ms 99.5th 3ms
se.a...api.WatchListService
1,755 0 0.0 %
0 0
Host: 6.7/s
Cluster: 175.4/s
Circuit Closed
Hosts 26 90th 0ms
Median 0ms 99th 3ms
Mean 0ms 99.5th 7ms
se.a...vices.AccountService
1,914 0 0.0 %
0 0
Host: 7.4/s
Cluster: 191.4/s
Circuit Closed
Hosts 26 90th 1ms
Median 0ms 99th 4ms
Mean 0ms 99.5th 12ms
se.a...service.AlertService
1,565 0 0.0 %
0 0
Host: 6.0/s
Cluster: 156.3/s
Circuit Closed
Hosts 26 90th 0ms
Median 0ms 99th 2ms
Mean 0ms 99.5th 5ms
se.a...ccountPerformanceSpi
1,331 0 0.0 %
0 0
Host: 5.1/s
Cluster: 133.0/s
Circuit Closed
Hosts 26 90th 34ms
Median 7ms 99th 104ms
Mean 13ms 99.5th 125ms
com....strixServiceRegistry
1,080 0 0.0 %
0 0
Host: 4.2/s
Cluster: 108.0/s
Circuit Closed
Hosts 26 90th 1ms
Median 1ms 99th 3ms
Mean 0ms 99.5th 14ms
Astr...strixServiceRegistry
888 0 0.0 %
0 0
Host: 3.4/s
Cluster: 88.8/s
Circuit Closed
Hosts 26 90th 1ms
Median 1ms 99th 2ms
Mean 0ms 99.5th 4ms
Elas...er.FilterQueryRunner
886 0 0.0 %
0 0
Host: 3.4/s
Cluster: 88.6/s
Circuit Closed
Hosts 26 90th 15ms
Median 3ms 99th 26ms
Mean 5ms 99.5th 29ms
se.a...ices.StoplossService
745 0 0.0 %
0 0
Host: 2.9/s
Cluster: 74.5/s
Circuit Closed
Hosts 26 90th 1ms
Median 0ms 99th 7ms
Mean 0ms 99.5th 12ms
Cust...Async-customer-space
730 0 0.0 %
0 0
Host: 2.8/s
Cluster: 73.0/s
Circuit Closed
Hosts 26 90th 1ms
Median 1ms 99th 1ms
Mean 0ms 99.5th 2ms
se.a...alRemotePriceService
699 0 0.0 %
0 0
Host: 2.7/s
Cluster: 69.9/s
Circuit Closed
Hosts 26 90th 0ms
Median 0ms 99th 5ms
Mean 0ms 99.5th 10ms
se.a....MarketScreenService
653 0 0.0 %
0 0
Host: 3.3/s
Cluster: 65.1/s
Circuit Closed
Hosts 20 90th 0ms
Median 0ms 99th 9ms
Mean 0ms 99.5th 24ms
se.a...nts.AgreementService
673 0 0.0 %
0 0
Host: 2.6/s
Cluster: 67.3/s
Circuit Closed
Hosts 26 90th 1ms
Median 0ms 99th 5ms
Mean 0ms 99.5th 10ms
se.a....SubscriptionService
572 0 0.0 %
0 0
Host: 2.2/s
Cluster: 57.2/s
Circuit Closed
Hosts 26 90th 0ms
Median 0ms 99th 4ms
Mean 0ms 99.5th 8ms
Cust...strixCustomerService
565 0 0.0 %
0 0
Host: 2.2/s
Cluster: 56.5/s
Circuit Closed
Hosts 26 90th 1ms
Median 1ms 99th 1ms
Mean 0ms 99.5th 3ms
Auth...stomerSessionService
530 0 0.0 %
0 0
Host: 2.0/s
Cluster: 53.0/s
Circuit Closed
Hosts 26 90th 2ms
Median 1ms 99th 2ms
Mean 1ms 99.5th 2ms
Pric....PriceHistoryService
516 0 0.0 %
0 0
Host: 2.2/s
Cluster: 51.6/s
Circuit Closed
Hosts 24 90th 13ms
Median 5ms 99th 40ms
Mean 7ms 99.5th 61ms
se.a...vices.TradingService
562 0 0.0 %
0 0
Host: 2.7/s
Cluster: 56.2/s
Circuit Closed
Hosts 21 90th 8ms
Median 0ms 99th 22ms
Mean 2ms 99.5th 33ms
se.a...umberOfOwnersService
462 0 0.0 %
0 0
Host: 1.8/s
Cluster: 46.2/s
Circuit Closed
Hosts 26 90th 1ms
Median 0ms 99th 9ms
Mean 0ms 99.5th 14ms
se.a...tmentsByOwnerService
457 0 0.0 %
0 0
Host: 1.8/s
Cluster: 45.7/s
Circuit Closed
Hosts 26 90th 1ms
Median 0ms 99th 9ms
Mean 0ms 99.5th 13ms
se.a...dTradingTermsService
211 0 0.0 %
0 0
Host: 0.8/s
Cluster: 21.1/s
Circuit Closed
Hosts 26 90th 0ms
Median 0ms 99th 5ms
Mean 0ms 99.5th 7ms
org....authentication-space
328 0 0.0 %
0 0
Host: 1.3/s
Cluster: 32.8/s
Circuit Closed
Hosts 26 90th 8ms
Median 0ms 99th 20ms
Mean 1ms 99.5th 21ms
se.a...ElasticSearchService
204 0 0.0 %
0 0
Host: 1.0/s
Cluster: 20.4/s
Circuit Closed
Hosts 21 90th 5ms
Median 2ms 99th 21ms
Mean 3ms 99.5th 22ms
Trad....TradingServiceAsync
196 0 0.0 %
0 0
Host: 0.8/s
Cluster: 19.6/s
Circuit Closed
Hosts 26 90th 2ms
Median 1ms 99th 4ms
Mean 1ms 99.5th 4ms
Trad...xAccountServiceAsync
179 0 0.0 %
0 0
Host: 0.7/s
Cluster: 17.9/s
Circuit Closed
Hosts 26 90th 2ms
Median 1ms 99th 5ms
Mean 1ms 99.5th 5ms
Cust...stomerAccountService
139 0 0.0 %
0 0
Host: 0.5/s
Cluster: 13.9/s
Circuit Closed
Hosts 26 90th 1ms
Median 1ms 99th 3ms
Mean 0ms 99.5th 3ms
se.a...rDepthViewLogService
142 0 0.0 %
0 0
Host: 0.7/s
Cluster: 14.2/s
Circuit Closed
Hosts 20 90th 1ms
Median 0ms 99th 8ms
Mean 0ms 99.5th 8ms
se.a...ces.MessagingService
133 0 0.0 %
0 0
Host: 0.5/s
Cluster: 13.2/s
Circuit Closed
Hosts 26 90th 1ms
Median 0ms 99th 4ms
Mean 0ms 99.5th 4ms
se.a...merInspectionService
131 0 0.0 %
0 0
Host: 0.5/s
Cluster: 13.1/s
Circuit Closed
Hosts 24 90th 9ms
Median 0ms 99th 22ms
Mean 2ms 99.5th 22ms
se.a...CustomerOfferService
97 0 0.0 %
0 0
Host: 0.4/s
Cluster: 9.7/s
Circuit Closed
Hosts 26 90th 22ms
Median 14ms 99th 36ms
Mean 15ms 99.5th 36ms

se.a...rAccountServiceAsync



21,259	0	0.0 %
0	0	

Host: **84.9/s**

Cluster: **2,123.4/s**

Circuit Closed

Hosts	**25**	90th	**0ms**
Median	**0ms**	99th	**2ms**
Mean	**0ms**	99.5th	**6ms**

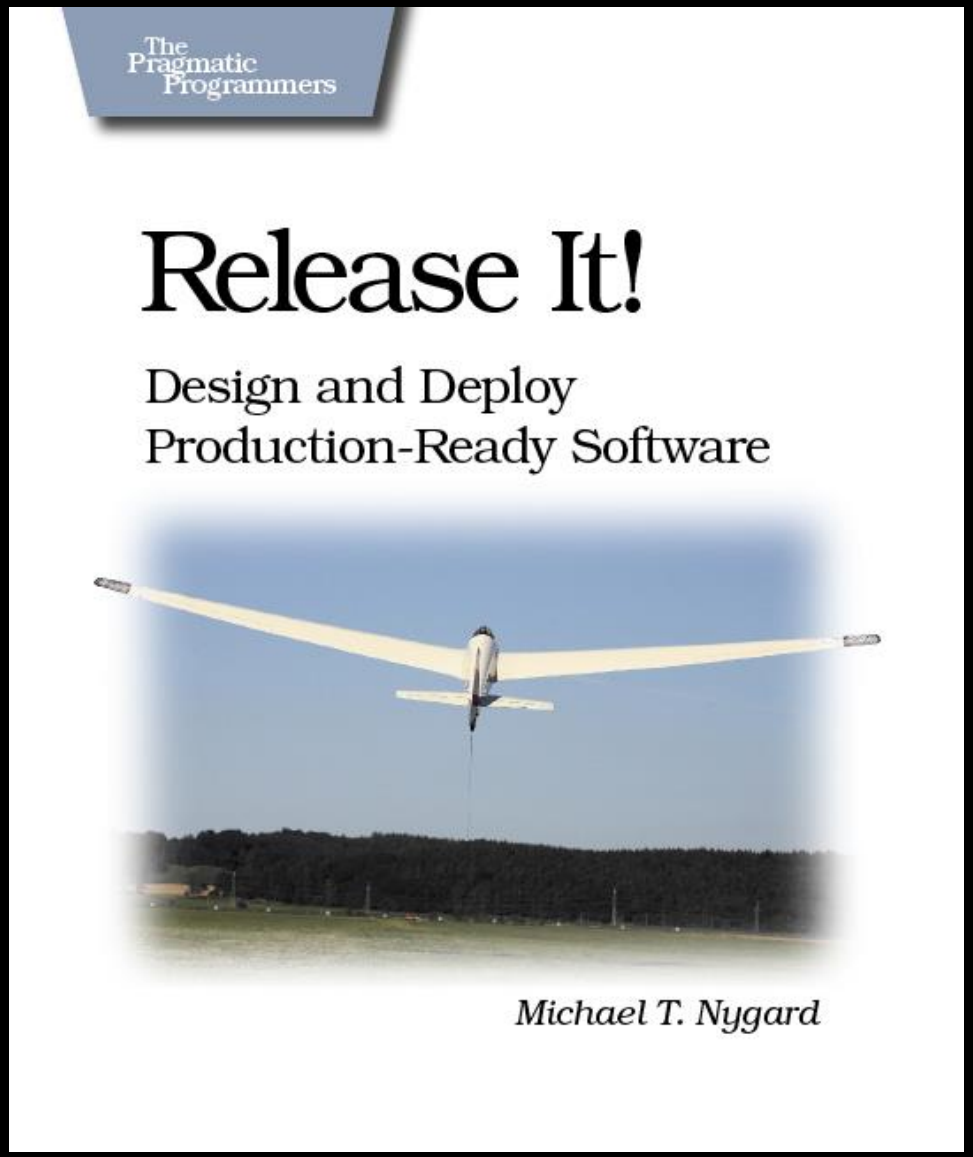
Design for failure

Use timeouts

Circuit Breakers

Bulkheads

Monitor service calls



https://github.com/Netflix/Hystrix

Image Attributions

- Polycelis felina" by Eduard Solà - Own work. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Polycelis_felina.jpg#/media/File:Polycelis_felina.jpg
- "Old book bindings" by Tom Murphy VII - Own work. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Old_book_bindings.jpg#/media/File:Old_book_bindings.jpg
- "Cute Snail" by gniyuhs - Own work. Licensed under CC BY-SA 3.0 via deviantart - http://gniyuhs.deviantart.com/art/Cute-Snail-278597934
- ”Cash” by 401(K) 2012 – Own work. Licensed under CC BY-SA 2.0 via Flickr - https://www.flickr.com/photos/68751915@N05/6355816649
- "Circuit breakers at substation near Denver International Airport, Colorado" by Greg Goebel from Loveland CO, USA - Yipws_2bUploaded by PDTillman. Licensed under CC BY-SA 2.0 via Wikimedia Commons - https://commons.wikimedia.org/wiki/File:Circuit_breakers_at_substation_near_Denver_International_Airport,_Colorado.jpg#/media/File:Circuit_breakers_at_substation_near_Denver_International_Airport,_Colorado.jpg
- ”The control room of the nuclear ship NS Savannah, Baltimore, Maryland, USA” - Own work. Licensed under CC BY-SA 3.0 via Commons - https://en.wikipedia.org/wiki/File:NS_Savannah_control_room_MD1.jpg#/media/File:NS_Savannah_control_room_MD1.jpg

Thank you! - Questions?

Kristoffer Erlandsson

kristoffer.erlandsson@avanza.se

@kerlandsson