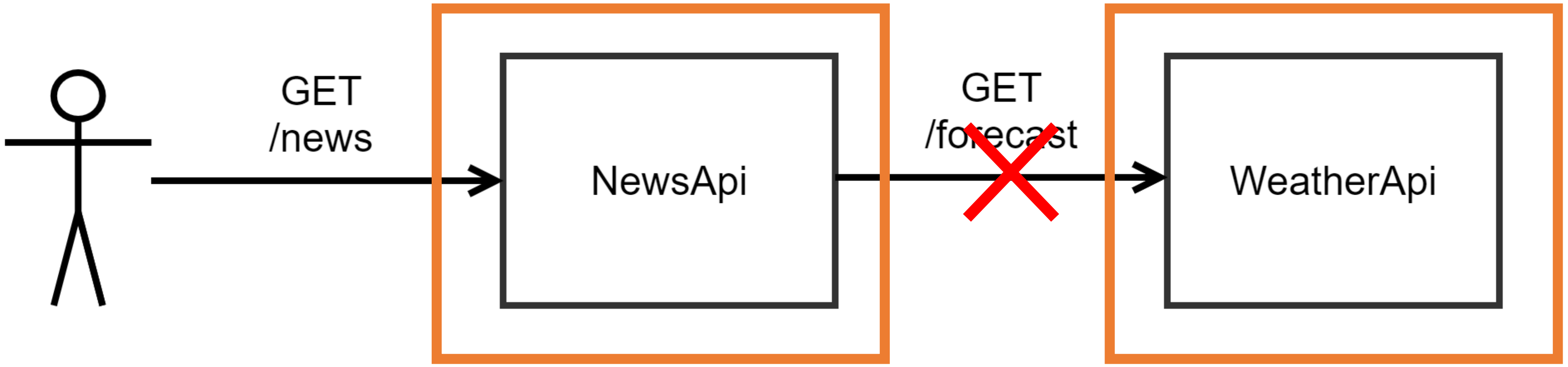
# How to integrate with a crappy Api

Artem Mikulich,
Tech Lead
at ISsoft Solutions


GET
/news
NewsApi
GET
/forecast
WeatherApi

# Possible root cause

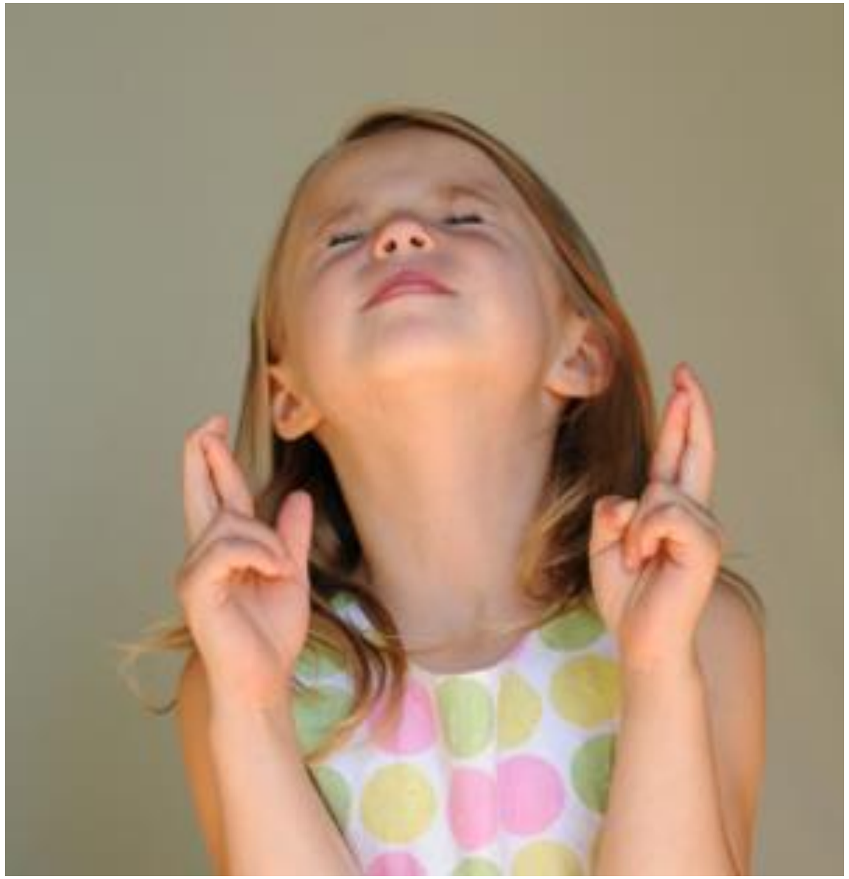
- Network lag
- 429 - too many requests
- Dead lock
- Server reboot
- Other

# Solution?

Do NOT make http requests

# Solution?

Cross fingers

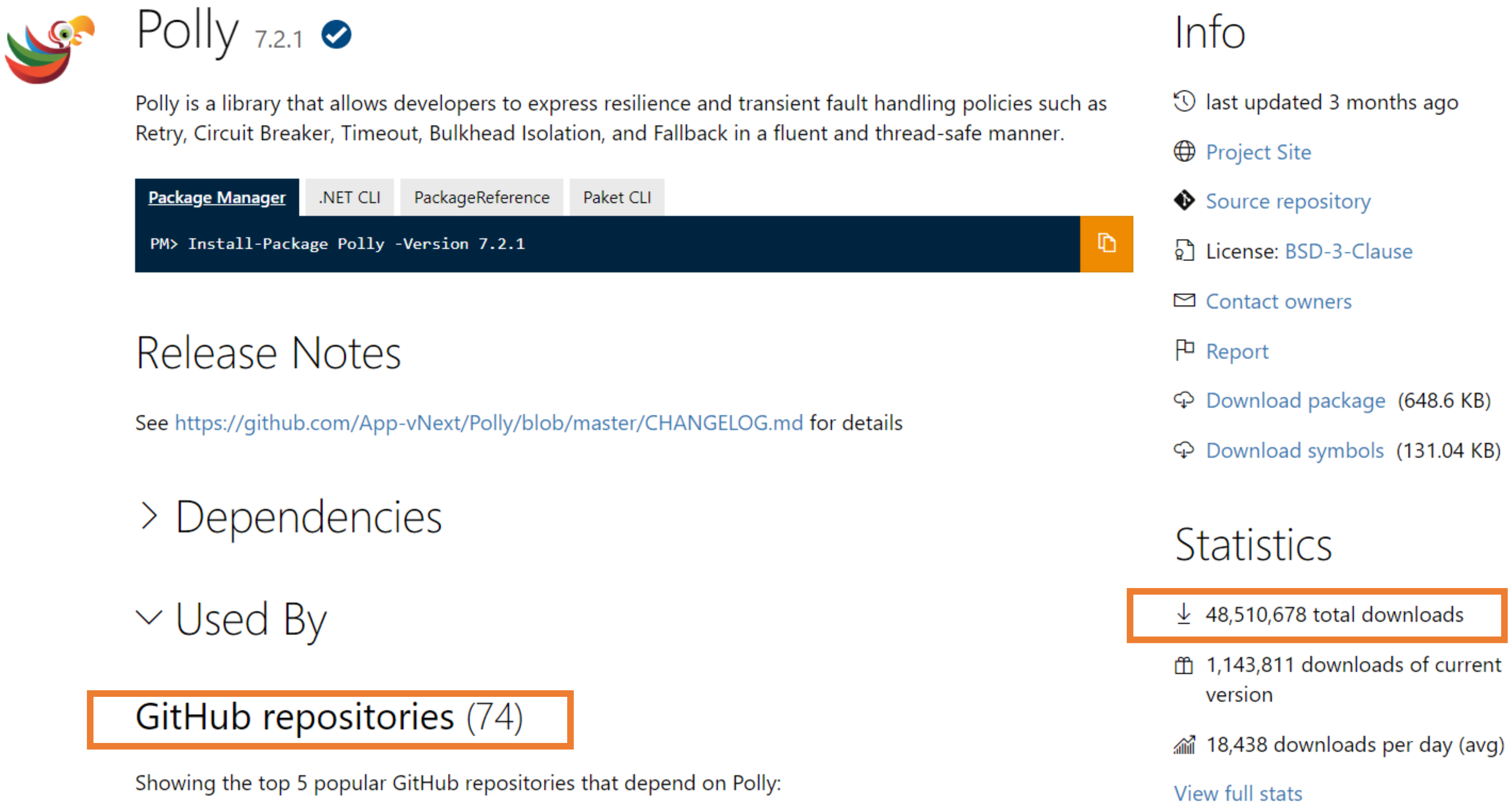


# Solution?

Re-invent the wheel

# Solution?

Find at nuget



# Polly 7.2.1

Polly is a library that allows developers to express resilience and transient fault handling policies such as Retry, Circuit Breaker, Timeout, Bulkhead Isolation, and Fallback in a fluent and thread-safe manner.

```
PM> Install-Package Polly -Version 7.2.1
```

## Release Notes

See https://github.com/App-vNext/Polly/blob/master/CHANGELOG.md for details

## > Dependencies

## ⌄ Used By

### GitHub repositories (74)

Showing the top 5 popular GitHub repositories that depend on Polly:

## Info

- last updated 3 months ago
- Project Site
- Source repository
- License: BSD-3-Clause
- Contact owners
- Report
- Download package (648.6 KB)
- Download symbols (131.04 KB)

## Statistics

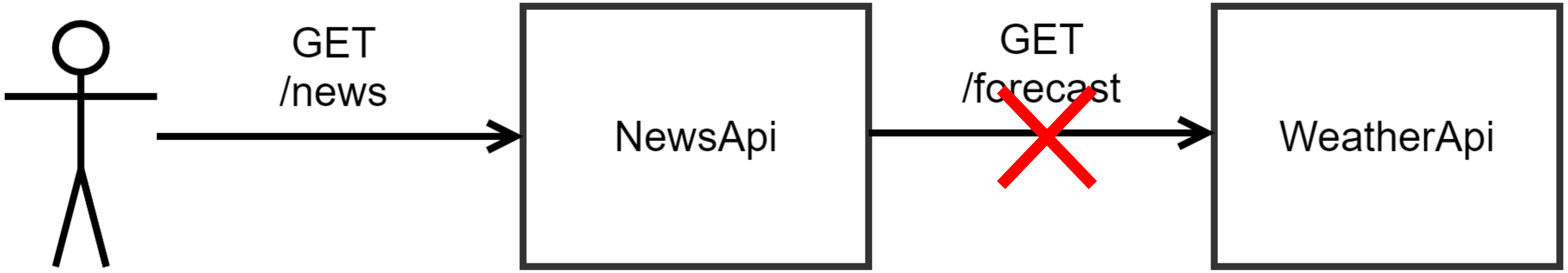
- 48,510,678 total downloads
- 1,143,811 downloads of current version
- 18,438 downloads per day (avg)

View full stats

# Agenda

- Case #1: sporadic errors
- Case #2: timeout
- Case #3: dynamic policies
- Case #4: caching
- Advanced cases

# Case #1: Sporadic errors



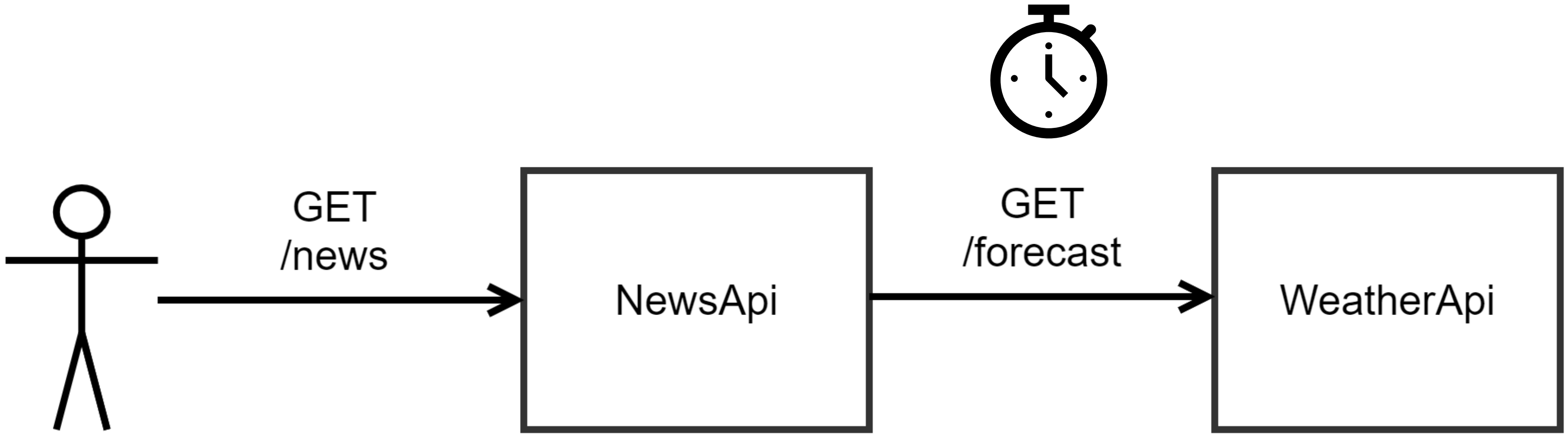
# Demo

# Retry

```
public void ConfigureServices(IServiceCollection services)
 {
     //...
     IAsyncPolicy<HttpResponseMessage> retry = Policy
         .HandleResult<HttpResponseMessage>(r => !r.IsSuccessStatusCode)
         .RetryAsync(3);

     services.AddHttpClient<CrapyWeatherApiClient>((client) => {/* ... */})
         .AddPolicyHandler(retry);
 }
```

# Demo: Retry

# Case #2: Timeout



# Possible root cause

- Network lag
- Database lock
- Database stats reset
- Server warms up
- Other

STILL
WAITING

# Handle Timeout

```
var retry = Policy
        .HandleResult<HttpResponseMessage>(r => !r.IsSuccessStatusCode)
        .Or<TimeoutRejectedException>()
        .RetryAsync(3);

var timeoutPolicy = Policy
        .TimeoutAsync(TimeSpan.FromSeconds(30), (ctx, t, task, e) => {/* ... */});


services.AddHttpClient<CrapyWeatherApiClient>((client) => {/* ... */})
                .AddPolicyHandler(retry.WrapAsync(timeoutPolicy));
```

# Demo: Handle Timeout

# Case #3: dynamic policies

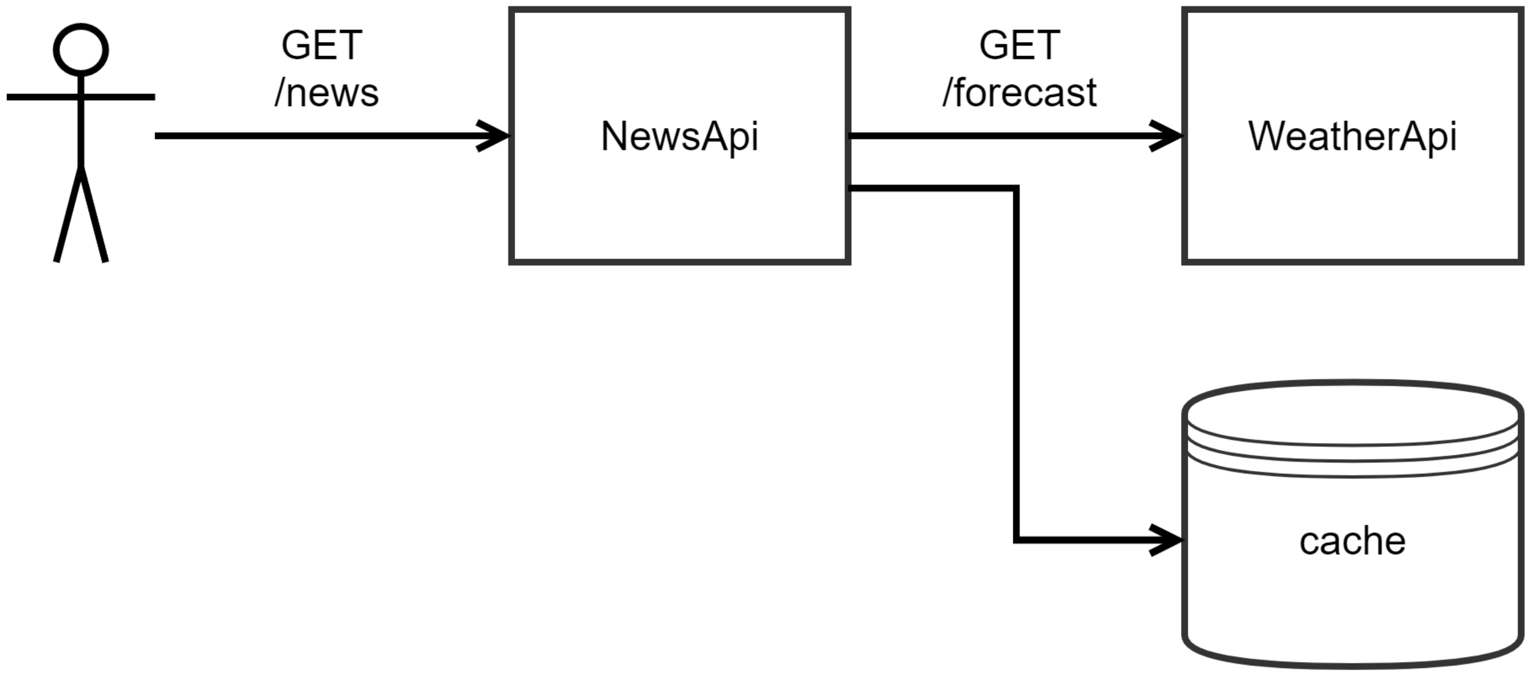
- Http method (GET/POST/PUT)
- Specific url (e.g.  /reports)
- Metrics (# of requests, date/time)
- Other

# Policy Registry

```
var registry = services.AddPolicyRegistry();

registry.Add("RetryPolicy", policy);

services.AddHttpClient<CrappyWeatherApiClient>((client) => {/* ... */})
   .AddPolicyHandlerFromRegistry((policyRegistry, request) =>
     {
       if (request.Method == HttpMethod.Get)
       {
           return registry.Get<IAsyncPolicy<HttpResponseMessage>>("RetryPolicy");
       }

       return Policy.NoOpAsync<HttpResponseMessage>();
     });
```

# Demo: dynamic policies

# Case #4: Caching

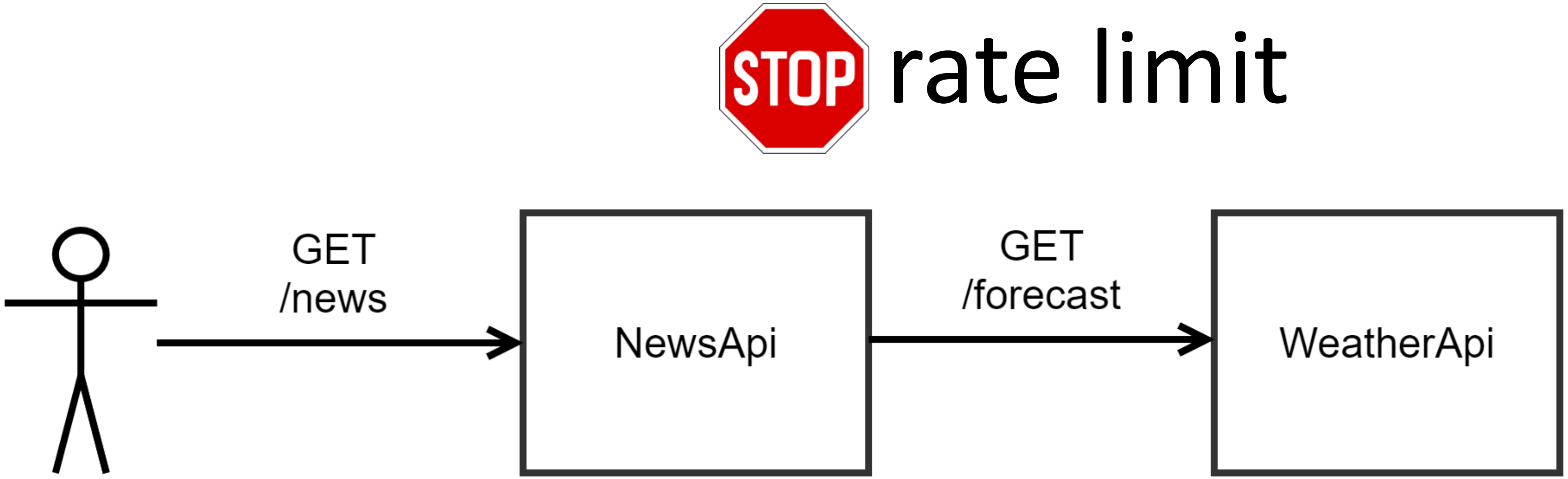


# Policy Registry extensions

```
policyRegistry.GetCachePolicyFor<SampleApiClient>();


policyRegistry.AddCachePolicyFor<SampleApiClient>(
        cacheProvider,
        TimeSpan.FromSeconds(60));
```

# Demo: Caching

# Advanced policies: Bulkhead isolation



STOP rate limit

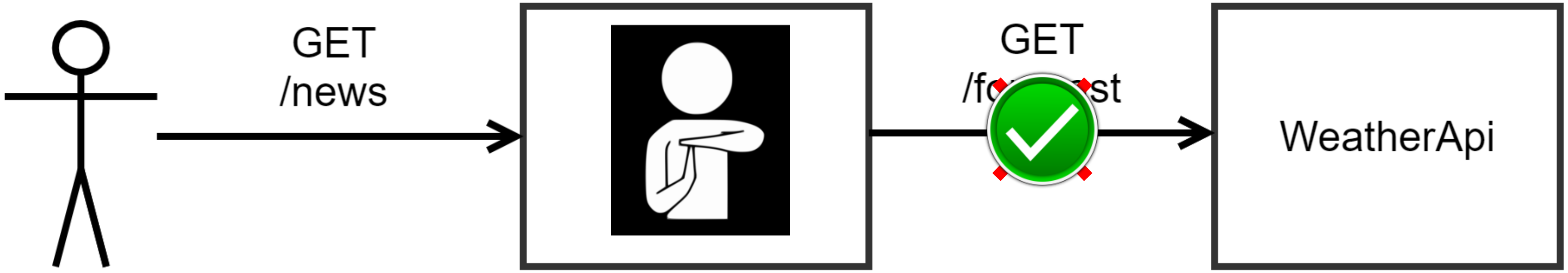
GET /news

NewsApi

GET /forecast

WeatherApi

# Advanced policies:
# Circuit Breaker



# Thanks for listening!

# Links

https://github.com/amikulich/polly-best-practices
https://github.com/amikulich/easyextensions.polly

https://www.nuget.org/packages/EasyExtensions.Polly/

https://www.facebook.com/devinjacket
https://www.youtube.com/channel/UCUwH3coyAnGL7zYCELMhjKA