TRAILHEAD
TECHNOLOGY PARTNERS

Cache Me If You Can

Supercharging .NET APIs with Caching

Jonathan “J.” Tower


TRAILHEAD
TECHNOLOGY PARTNERS


TRAILHEAD
TECHNOLOGY PARTNERS

Speed Up Your **.NET** Web API with **Redis** Caching

TRAILHEAD
TECHNOLOGY PARTNERS

Jonathan "J." Tower

Principal Consultant & Partner

TRAILHEAD
TECHNOLOGY PARTNERS

- Microsoft MVP in .NET
- jtower@trailheadtechnology.com
- trailheadtechnology.com/blog
- jtowermi
- Jonathan “J.” Tower
- Blue Blazes podcast

EXPERT
CONSULTATION

bit.ly/th-offer

https://github.com/trailheadtechnology/redis-aspnet-core

Overview

- Definitions
  - Caching
  - Redis
- Why Use Caching
- Getting Started in ASP.NET Core
  - StackExchange.Redis
  - ServiceStack.Redis
- The Redis Command Line Interface
- ASP.NET Core Redis Demo
  - ConnectionMultiplexer
  - IDistributedCache
- Common Mistakes

What is Caching?


TRAILHEAD
TECHNOLOGY PARTNERS


CODE
TRAILHEAD
TECHNOLOGY PARTNERS

What is Caching?

Temporarily storing **frequently accessed** data to **fast storage** that's located **close** to the application

TRAILHEAD
TECHNOLOGY PARTNERS

What is Caching?

Temporarily storing **frequently accessed** data to **fast storage** that's located **close** to the application

TRAILHEAD
TECHNOLOGY PARTNERS

What is Caching?

Temporarily storing **frequently accessed** data to **fast storage** that's located **close** to the application

What is Caching?

Temporarily storing **frequently accessed** data to **fast storage** that's located **close** to the application

What is Caching?

Temporarily storing **frequently accessed** data to **fast storage** that's located **close** to the application

What is Redis?

What is Redis?

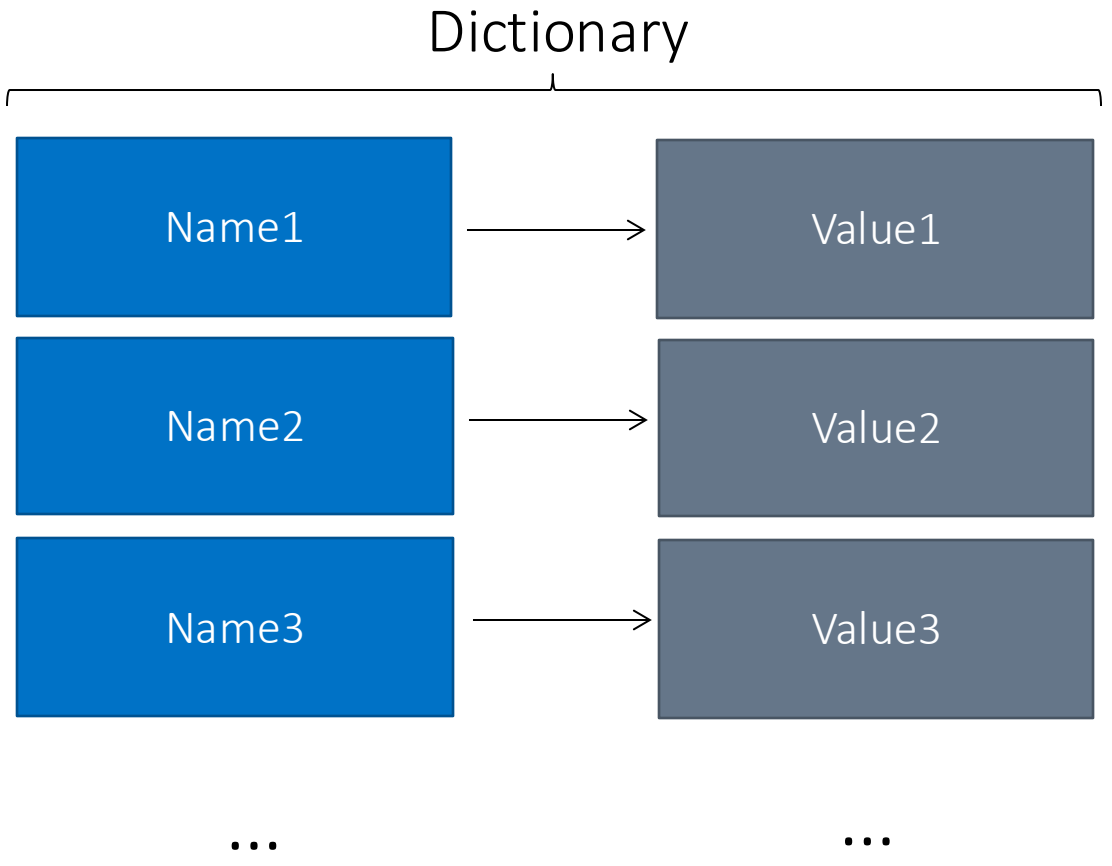
- The **open source**, **in-memory data store** used by millions of developers as a database, cache, streaming engine, and message broker.

- **RE**mote **DI**ctionary **S**erver

TRAILHEAD
TECHNOLOGY PARTNERS

Redis: An In-Memory Dictionary

- Dictionary:
  An **unordered collection** of data represented by a set of **key-value pairs**



TRAILHEAD TECHNOLOGY PARTNERS

Redis is Not a Relational DB

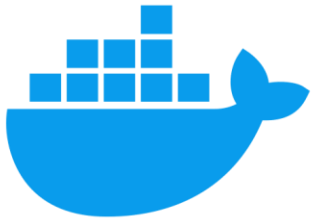
- Key/Value pairs
- Size limit
- In-memory
- Invalidation/expiration

TRAILHEAD
TECHNOLOGY PARTNERS

Redis Data Eviction Policy

Redis Hosting Options

Self-Host



WSL2

Host in Managed Cloud

Google Cloud

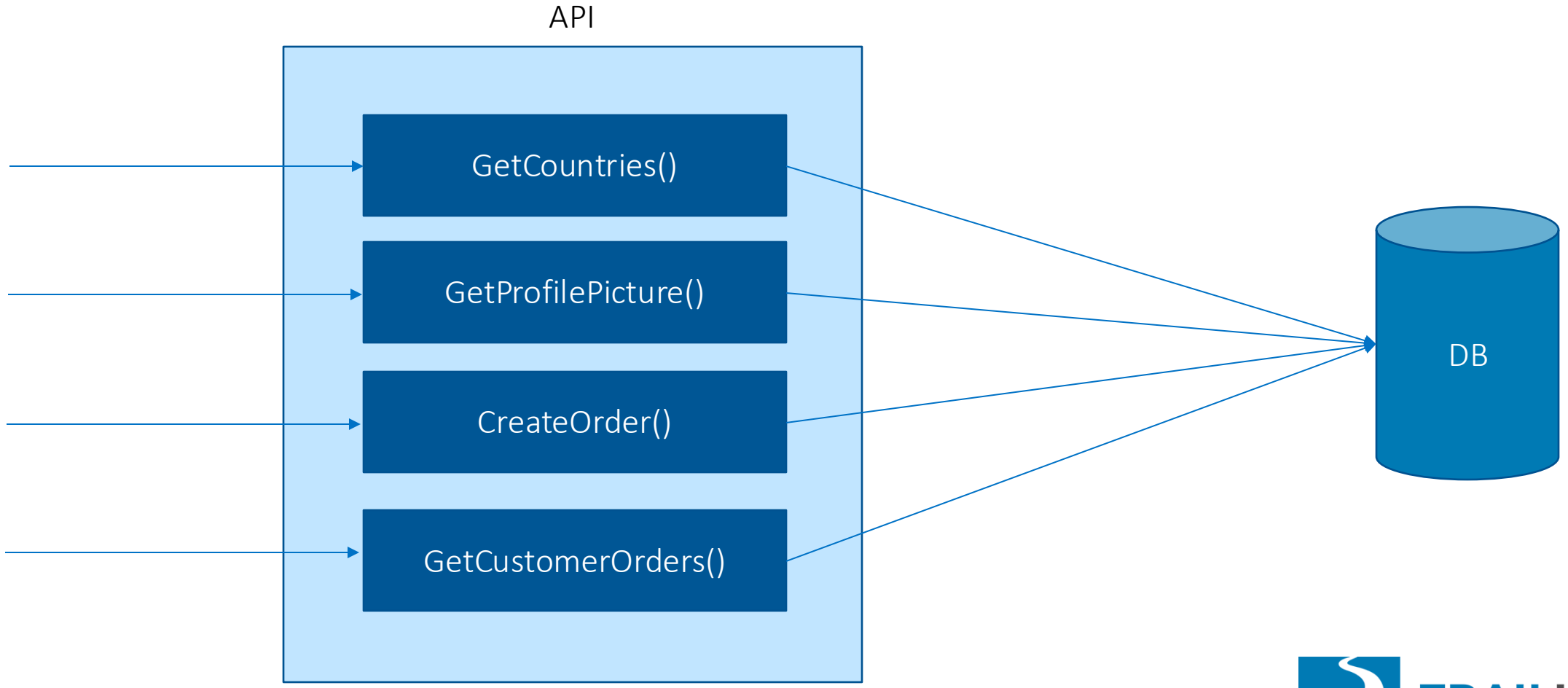
TRAILHEAD
TECHNOLOGY PARTNERS

Redis SKUs

		Redis Open Source	Redis Enterprise
Build	Boost app performance with Redis cache	●	●
	Support enterprise caching (read replica, write-behind, write-through)		●
	Build any real-time app on Redis with extended data model and processing engines support (JSON, search, time series, graph)		●
	Rapid development with out-of-the-box object mapping libraries for Spring, ASP.NET Core, FastAPI, and Express		●
	Redis GUI with predefined developer guides and tools	●	●
Deploy	Fully supported deployment on-premise or hybrid cloud		●
	Automated deployment on any cloud or multicloud		●
	Ingest data from external data sources with RedisConnect		●
Run	Deliver consistent real-time customer experience globally with geo-distributed Redis		●
	Automated database and cluster management (scaling, re-sharding, rebalancing)		●
	Built-in high-availability and disaster recovery management		●
	Enterprise-grade customer support from the creators of Redis		●

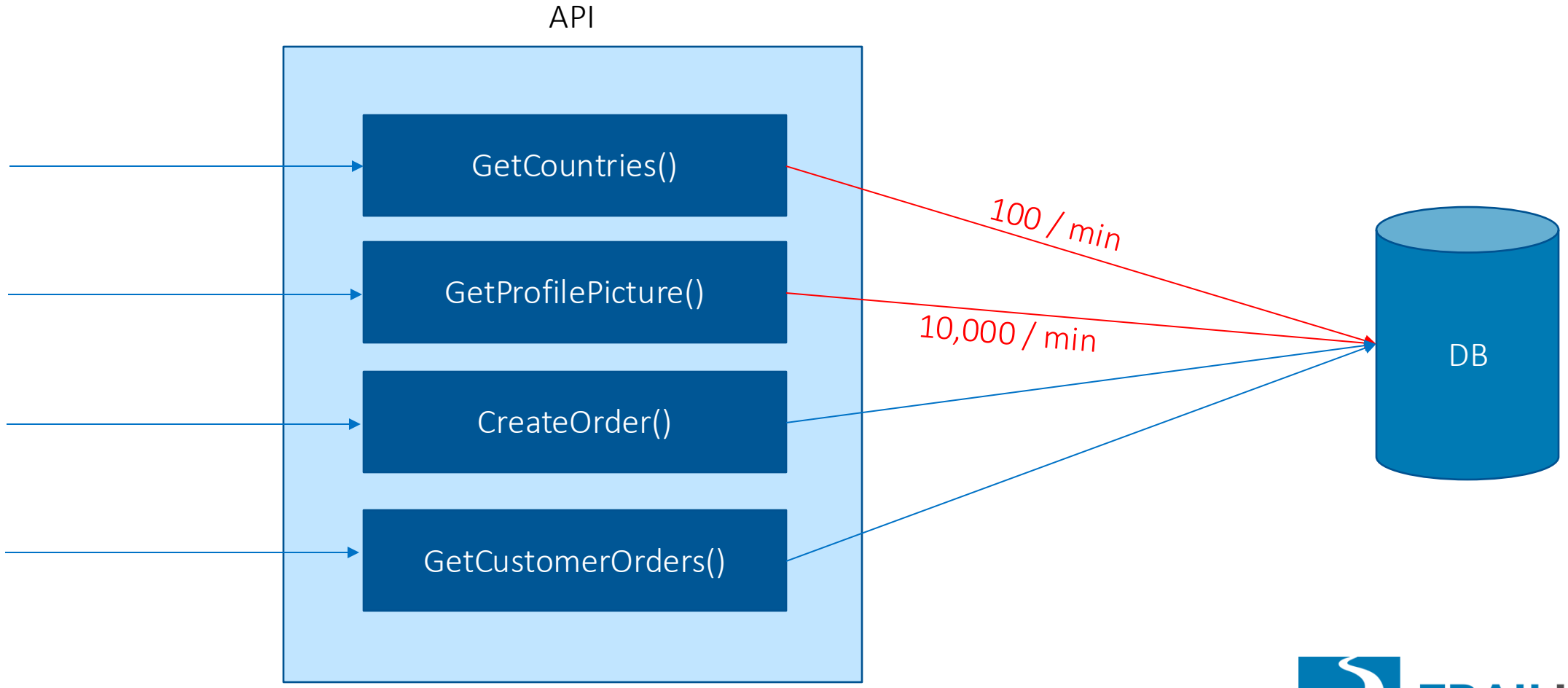
Why Use Caching?

Caching Sample



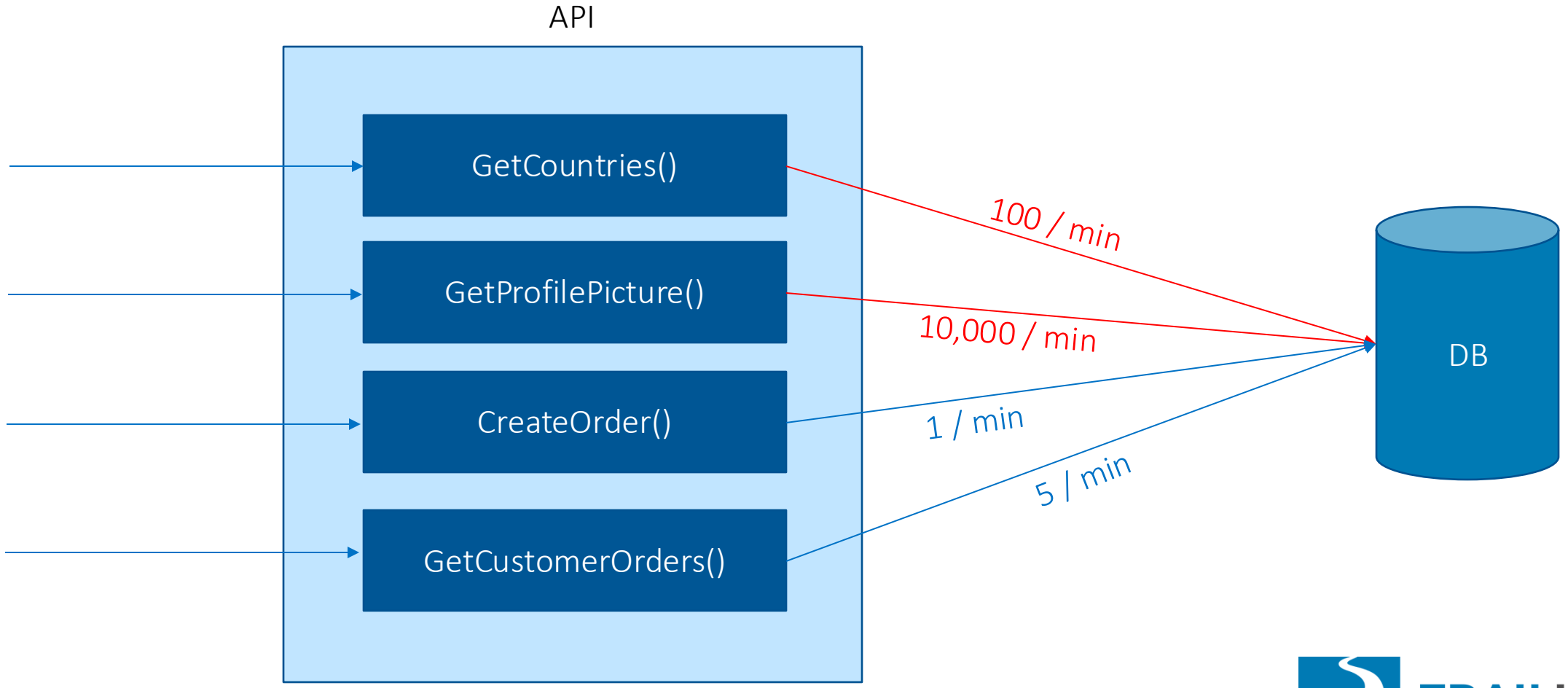
TRAILHEAD
TECHNOLOGY PARTNERS

Caching Sample



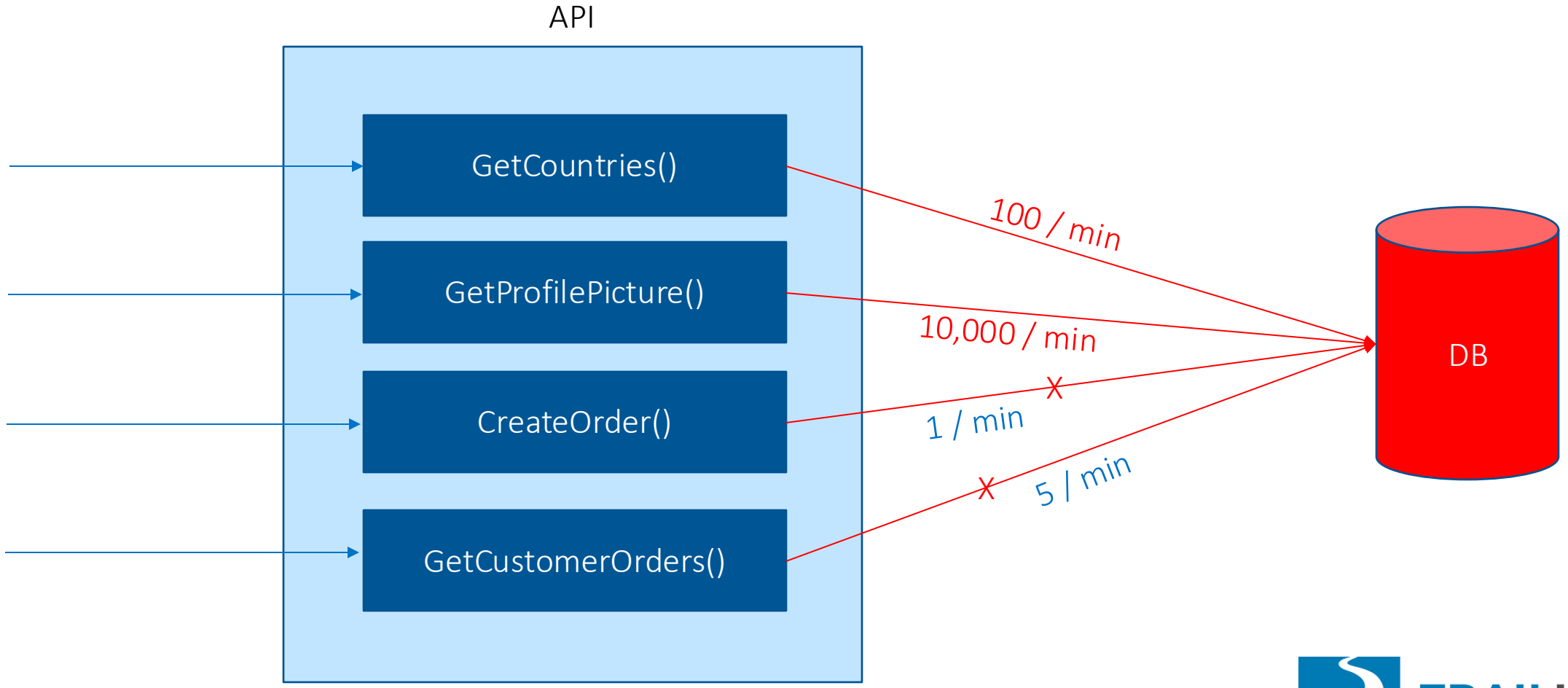
TRAILHEAD
TECHNOLOGY PARTNERS

Caching Sample



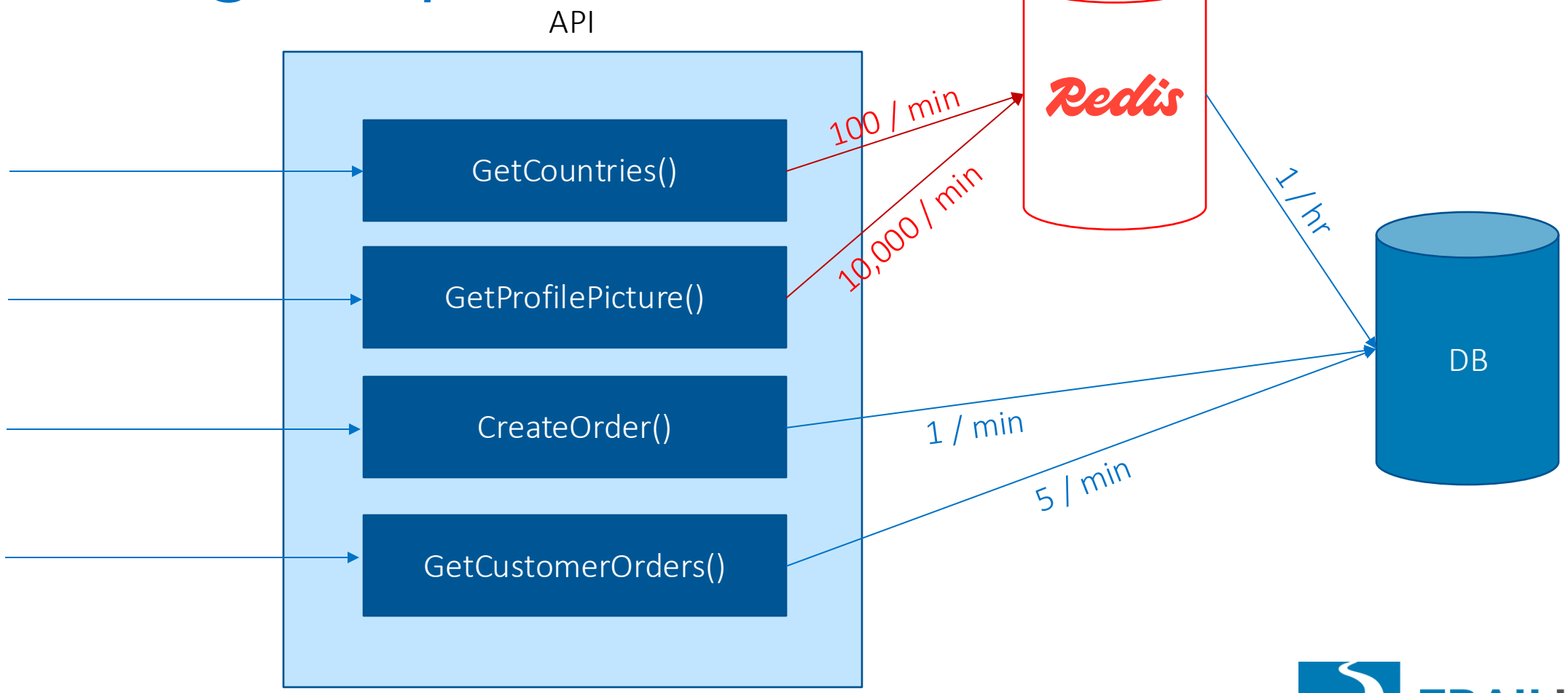
TRAILHEAD
TECHNOLOGY PARTNERS

Caching Sample



TRAILHEAD
TECHNOLOGY PARTNERS

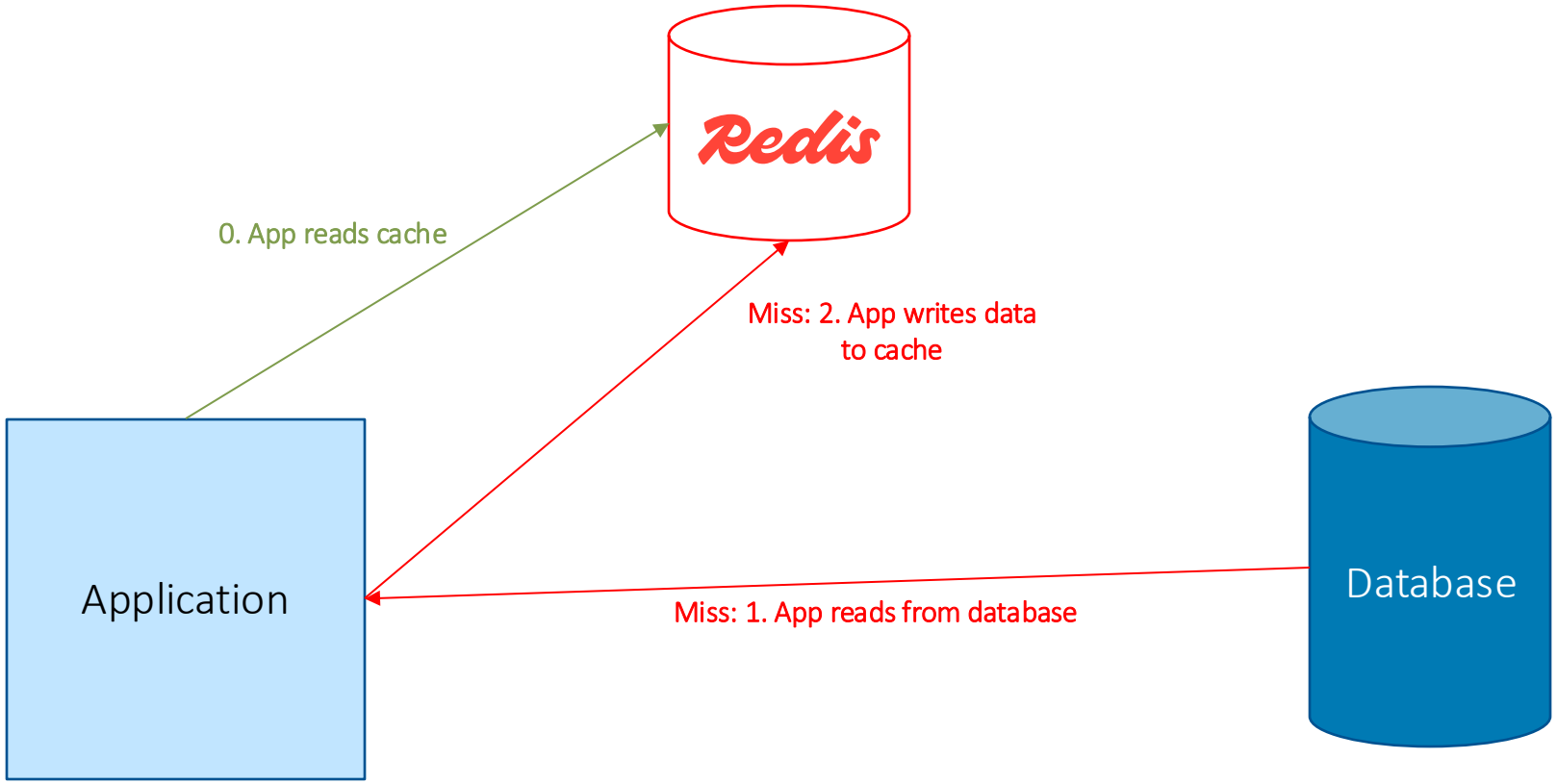
Caching Sample



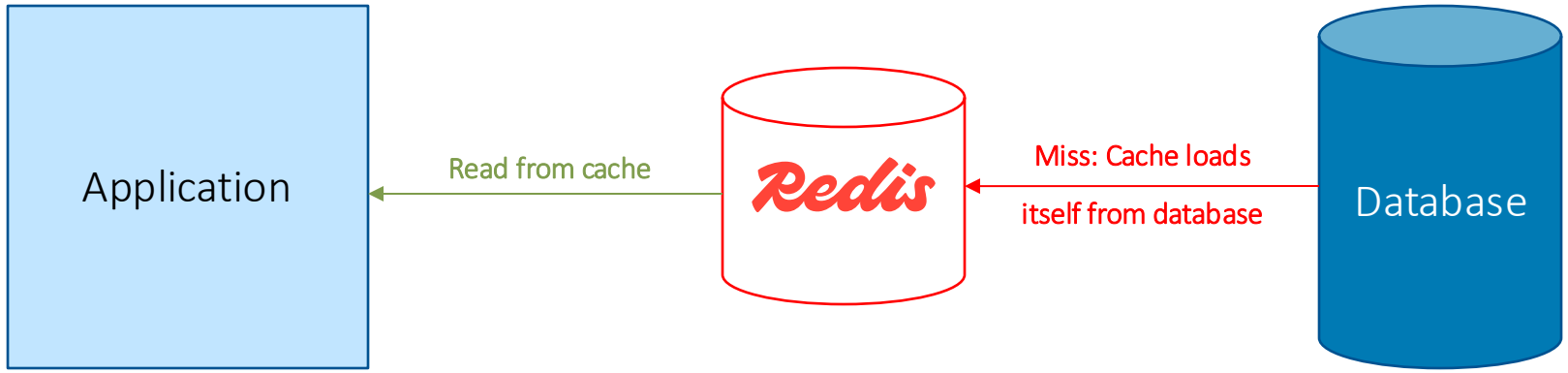
TRAILHEAD
TECHNOLOGY PARTNERS

Caching Strategies

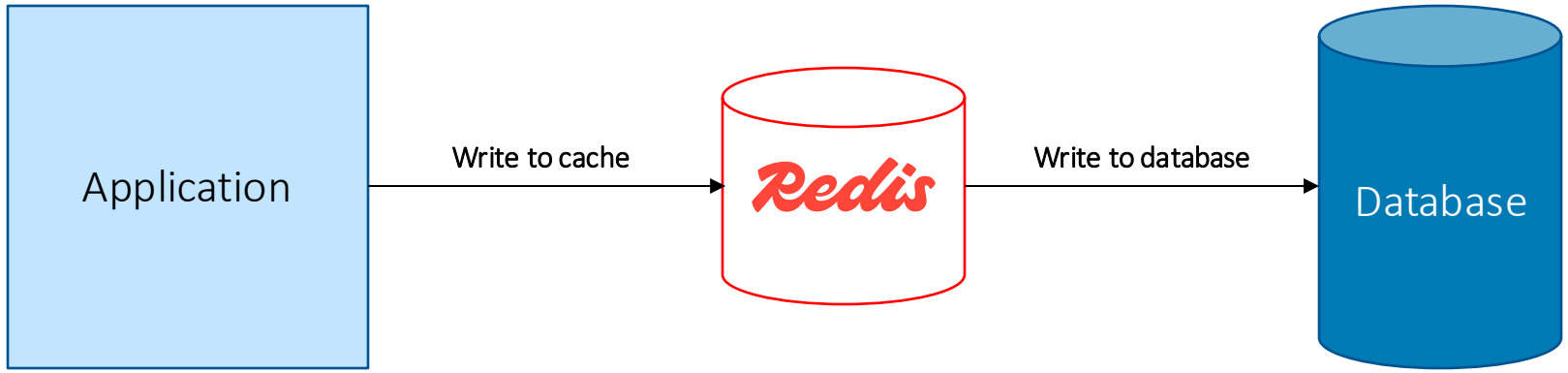
Cache-Aside


Redis
0. App reads cache
Miss: 2. App writes data
to cache
Application
Database
Miss: 1. App reads from database

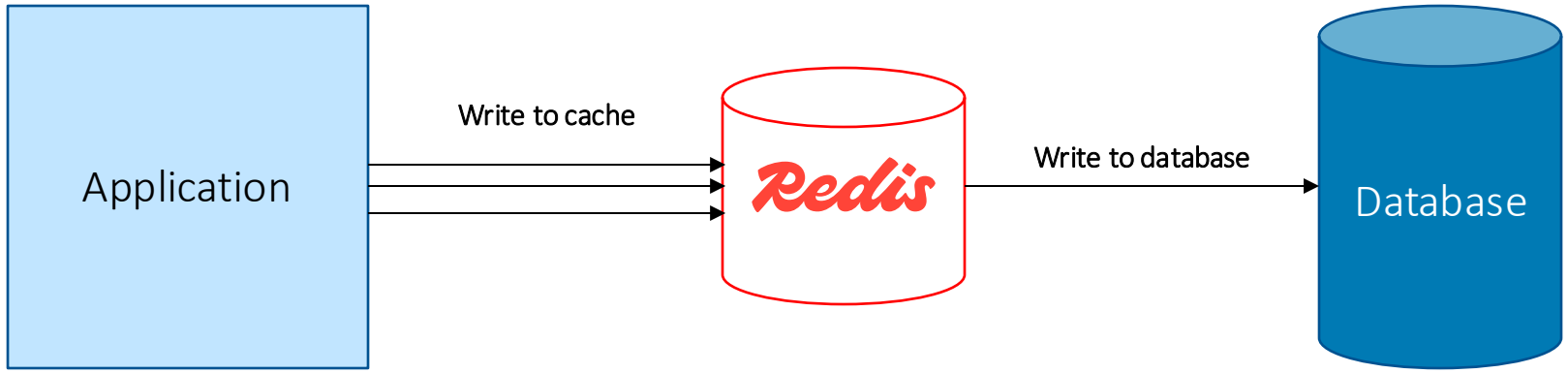
Read-Through



Write-Through



Write-Behind (or Write-Back)



Getting Started

Adding Redis to Your ASP.NET Core Project

Dev Env Install

Windows *Enable/install WSL2 (pre-req)*

```
sudo apt-get install redis
```

Mac

```
brew install redis
```

Linux

```
sudo apt-get install redis
```

TRAILHEAD
TECHNOLOGY PARTNERS

Deploying Redis

- On-prem or VM (No additional cost)
- Dockerized Redis (No additional cost)
- Managed Redis ($5+/ mo)
  - Redis Cloud
  - AWS ElastiCache for Redis
  - Azure Cache for Redis
  - Google Memorystore

TRAILHEAD
TECHNOLOGY PARTNERS

Adding Redis to ASP.NET Core

- Add StackExchange.Redis nuget package

- Either:
  - Wrapper for SINGLE instance of ConnectionMultiplexer
  - IDistributedCache through builder.Services.AddStackExchangeRedisCache(…)

StackExchange.Redis vs ServiceStack.Redis

	StackExchange.Redis	ServiceStack.Redis
Cost	Free	$300 - $1,000 / dev
Support	None	Commercial Support

TRAILHEAD
TECHNOLOGY PARTNERS

DEMO

Redis Command Line Interface (CLI)

DEMO

User Profile Images

Redis API vs IDistributedCache

- Microsoft.Extensions.Caching.StackExchangeRedis nuget package

```
builder.Services.AddStackExchangeRedisCache(options =>
{
    options.Configuration = "127.0.0.1:6379,allowAdmin=true";
    options.InstanceName = "dev-redis:";
});
```

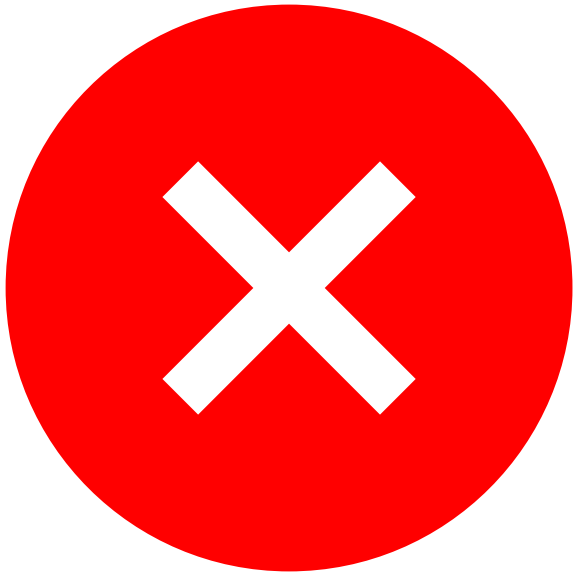
TRAILHEAD
TECHNOLOGY PARTNERS

Common Mistakes

With Redis Caching

Too Many ConnectionMultiplexers

```
var app = builder.Build();
app.MapGet("/", (ICacheService cache) => {
    using (var redis = ConnectionMultiplexer.Connect("localhost"))
    {
        var db = redis.GetDatabase();
        return db.StringGet(key);
    }
});
app.Run();
```



Too Many ConnectionMultiplexers

```
builder.Services.AddScoped<ICacheService, CacheService>();

var app = builder.Build();
app.MapGet("/", (ICacheService cache) => cache.GetCachedValue("foo"));
app.Run();
```

Not Using a Password

redis.conf

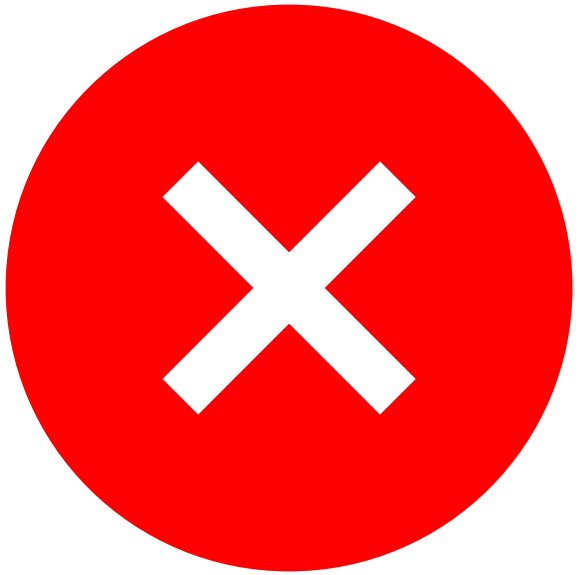
```
requirepass your_redis_password
```

code.cs

```
var redis = ConnectionMultiplexer.Connect(
    "servername:6379,password=your_redis_password");
```

Not Expecting Cache Misses

```
var cachedUser = db.StringGet($"user:{userId}");
return JsonConvert.DeserializeObject<User>(cachedUser);
```

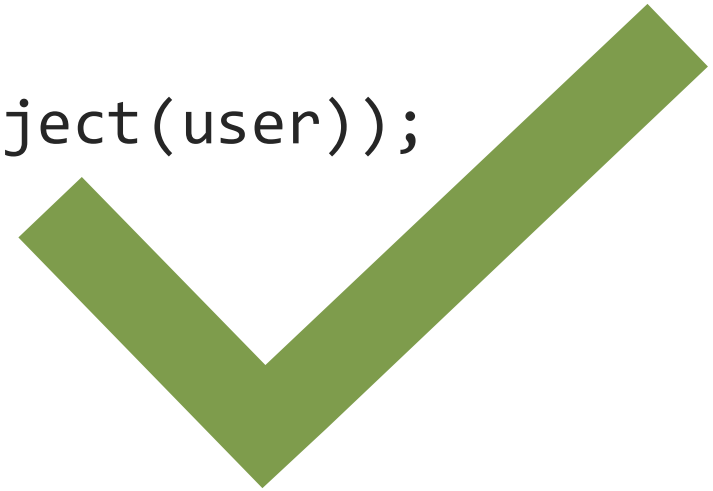


Not Expecting Cache Misses

```
var json = db.StringGet($"user:{userId}");
var user = !json.IsNullOrEmpty
    ? JsonConvert.DeserializeObject<User>(json)
    : _userRepository.GetUserById(userId);

if (json.IsNullOrEmpty && user != null)
    db.StringSet(key, JsonConvert.SerializeObject(user));

return user;
```

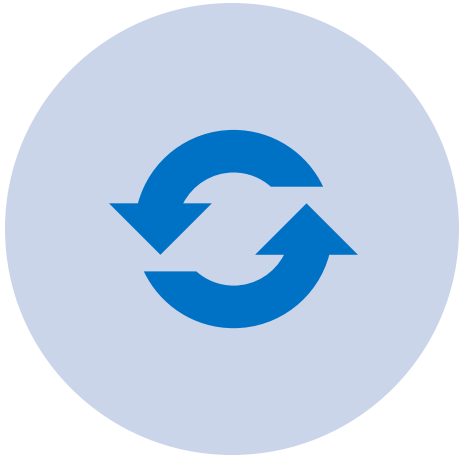


Caching Infrequently-Accessed Data

USES LIMITED STORAGE

LIMITED IMPACT ON PERFORMANCE

Caching Frequently-Changing Data



HIGH CHURN

NEGATIVE IMPACT ON PERFORMANCE

Not Having a Key Naming Convention

- Use colon (:) to separate items:
  - `namespace:object-type:id:field`
- Examples:
  - *user:1000:password*
  - *comment:1234:replyto*
- Optionally use # for ids
  - *logistics:building#23*

Not Setting Expiration (TTL)

- Memory bloat
- Stale or outdated data

- TTL & Programmatically Invalidate

```
db.StringSet("user:123:name", "...", TimeSpan.FromMinutes(30));

db.KeyDelete("user:123:name");
```

Over-Serialization

```
[MessagePackObject]
public class UserProfile
{
    [Key(0)]
    public int Id { get; set; }

    [Key(1)]
    public string Name { get; set; }
}

await redisDatabase.StringSetAsync("user:1", MessagePackSerializer.Serialize(
  new UserProfile { ... }));

var profile = MessagePackSerializer.Deserialize<UserProfile>(
  await redisDatabase.StringGetAsync("user:1"));
```

Summing Up

1. Using Redis correctly can speed up your application and take load off your database
2. Host Redis wherever you want (on-prem, cloud, etc)
3. Use StackExchange SDK for free or ServiceStack for commercial support
4. ConnectionMultiplexer or IDistributedCache

TRAILHEAD
TECHNOLOGY PARTNERS

Recap

- Definitions
  - Caching
  - Redis
- Why Use Caching
- Getting Started in ASP.NET Core
  - StackExchange.Redis
  - ServiceStack.Redis
- The Redis Command Line Interface
- ASP.NET Core Redis Demo
  - ConnectionMultiplexer
  - IDistributedCache
- Common Mistakes

Thank You! Questions?

Jonathan "J." Tower

- Microsoft MVP in .NET
- jtower@trailheadtechnology.com
- trailheadtechnology.com/blog
- jtowermi
- Jonathan “J.” Tower
- Blue Blazes podcast

EXPERT
CONSULTATION

bit.ly/th-offer

https://github.com/trailheadtechnology/redis-aspnet-core