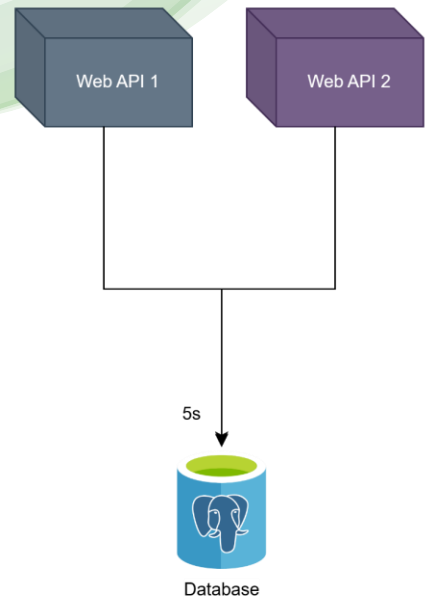




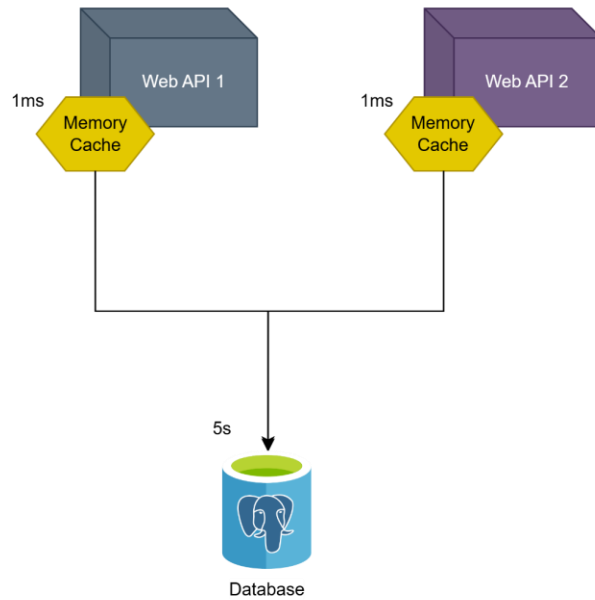
# Hybrid Cache

Deep dive into .NET Caching

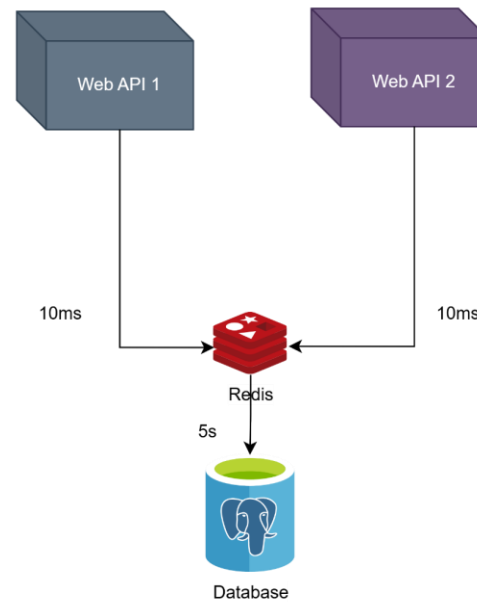
# From No-Cache to Hybrid-Cache



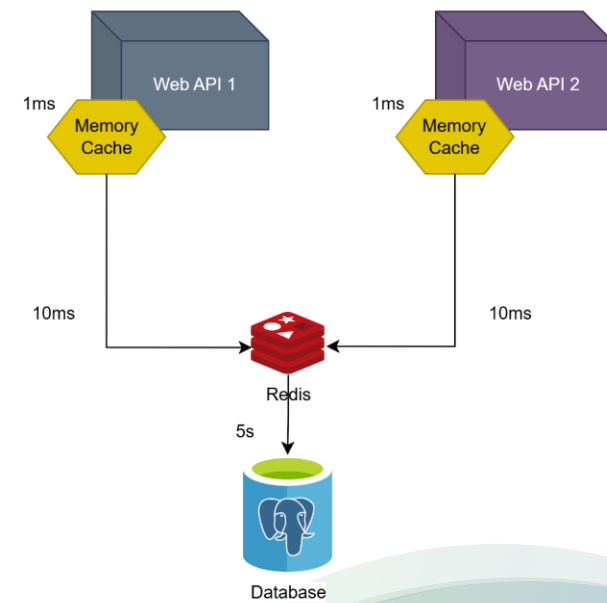
No Cache



Memory Cache (L1)



Distributed Cache (L2)



Hybrid Cache (L1 + L2)

# Process and Architecture

## Datenquelle

### Beschreibung:

Ursprüngliche Datenquelle, z.B. eine Datenbank oder ein Webservice.

### Technologie:

Entity Framework, Dapper, HTTP-Clients etc.

### Funktion:

Wird aufgerufen, wenn der Cache leer ist, um die benötigten Daten zu laden.

## L1 Cache (In-Memory Cache)

### Beschreibung:

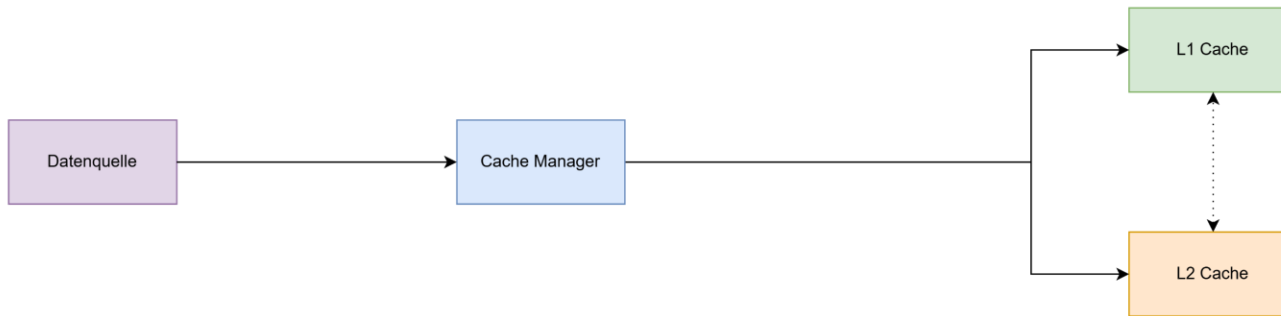
Schneller, speicherinterner Cache für häufig genutzte Daten.

### Technologie:

MemoryCache oder IMemoryCache in .NET.

### Vorteile:

Sehr schnelle Zugriffszeiten, ideal für kurzfristige Daten.



## Cache Manager (HybridCache)

### Beschreibung:

Verwaltung der Cache-Hierarchie und Entscheidung, welche Daten in L1 oder L2 gespeichert werden. Stamp

### Technologie:

Benutzerdefinierte Logik in .NET, die beide Caches koordiniert. Seit .NET 9 mit der Abstraktion "HybridCache" mit Microsoft oder mit FusionCache in .NET

### Zusätzliche Funktionen:

- Stamped Protection: Schutzmechanismus, der sicherstellt, dass nur einmalig, gültige Daten im Cache gespeichert werden.
- Tag-Based Validation: Ermöglicht die Invalidation von Cache-Einträgen basierend auf Tags, um gezielt Daten zu aktualisieren.

## L2 Cache (Distributed Cache)

### Beschreibung:

Verteilter Cache für Daten, die zwischen mehreren Instanzen geteilt werden.

### Technologie:

IDistributedCache in .NET, oft implementiert mit Redis oder SQL Server.

### Vorteile:

Skalierbar, Daten sind zwischen mehreren Servern konsistent.



Demo