

Solution Cache: Local Stores

Elton Stoneman
elton@sixeyed.com

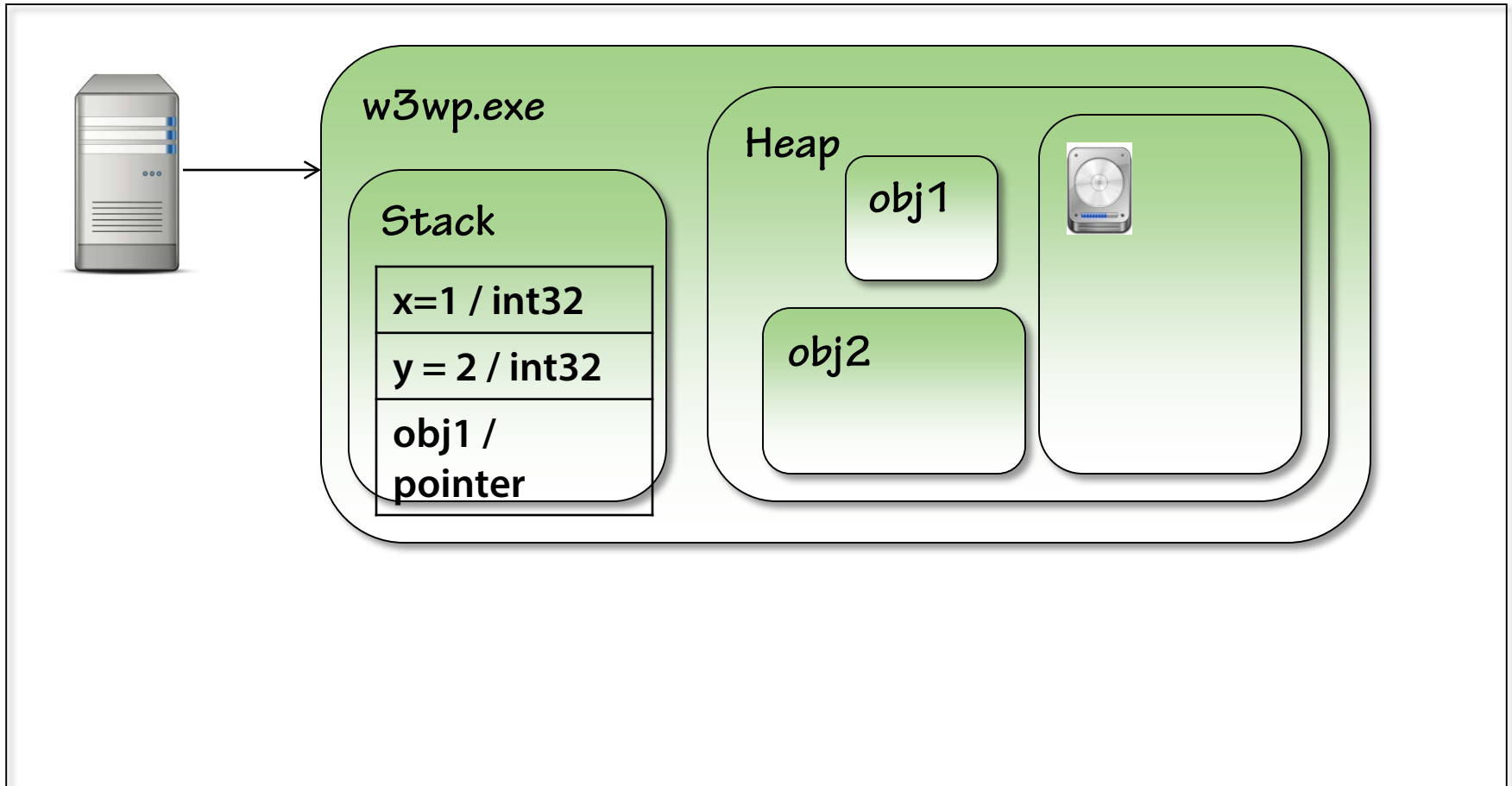


Outline

- **Types of local cache store**
 - In-process
 - Out-of-process
- **Abstracting the store from the solution**
 - ICache interface
- **Local caches**
 - Dependencies, usage and configuration
 - Management and extras
 - Matching against the decision matrix
- **.NET MemoryCache**
- **NCache Express**
- **Windows Server AppFabric Caching**
- **NullCache**

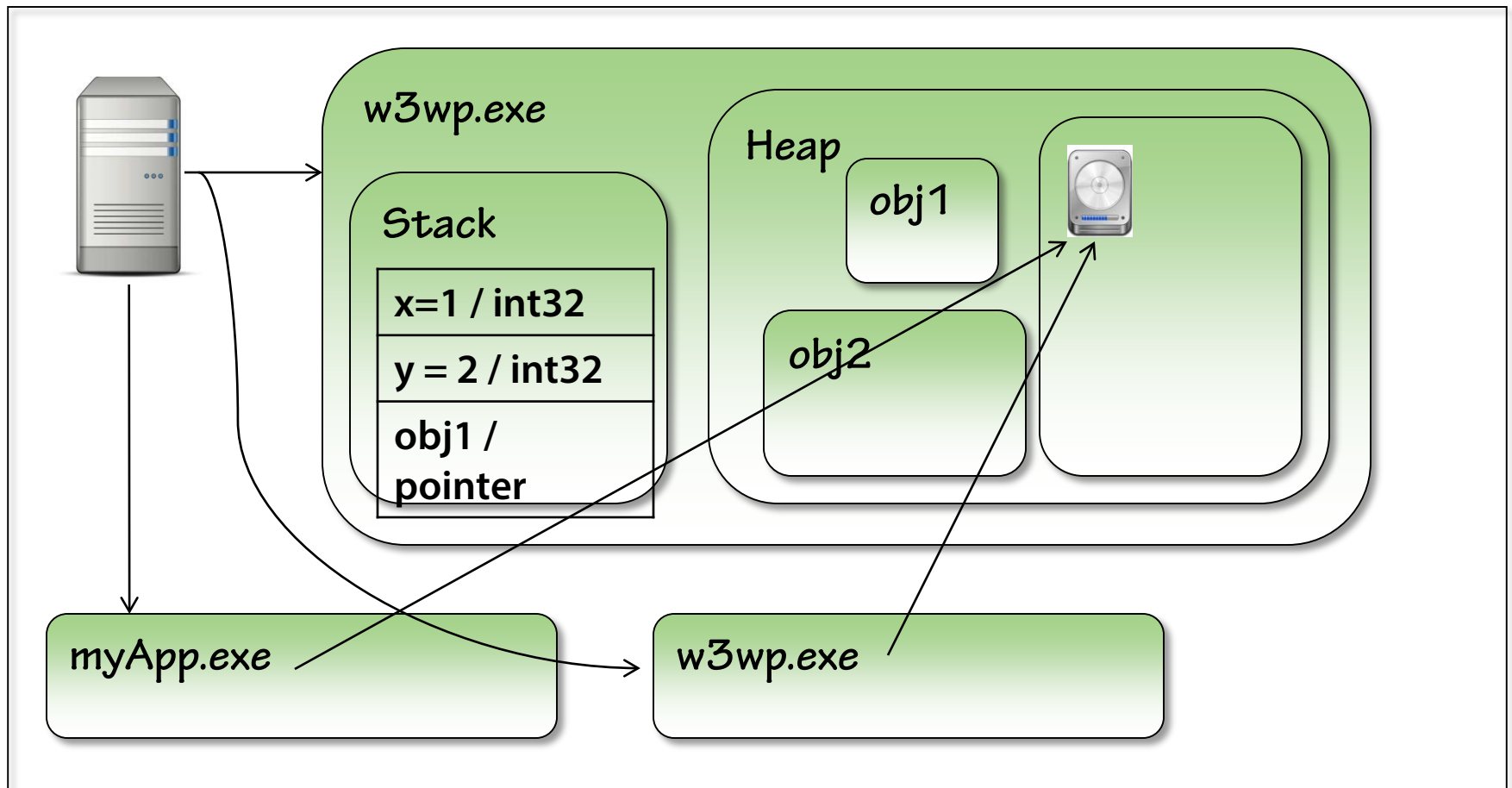
Types of Local Cache Store

- In-process



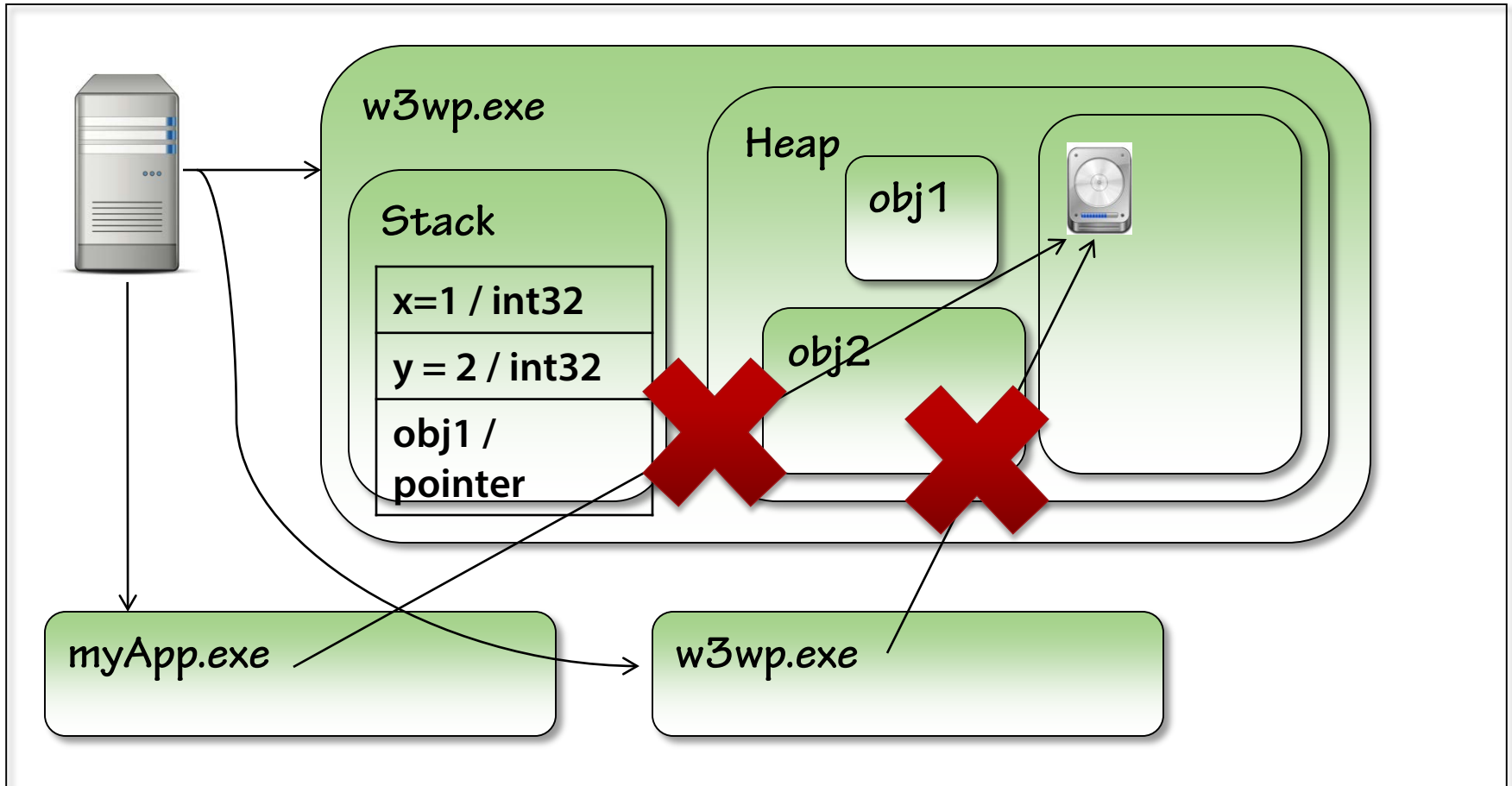
Types of Local Cache Store

- In-process



Types of Local Cache Store

- In-process



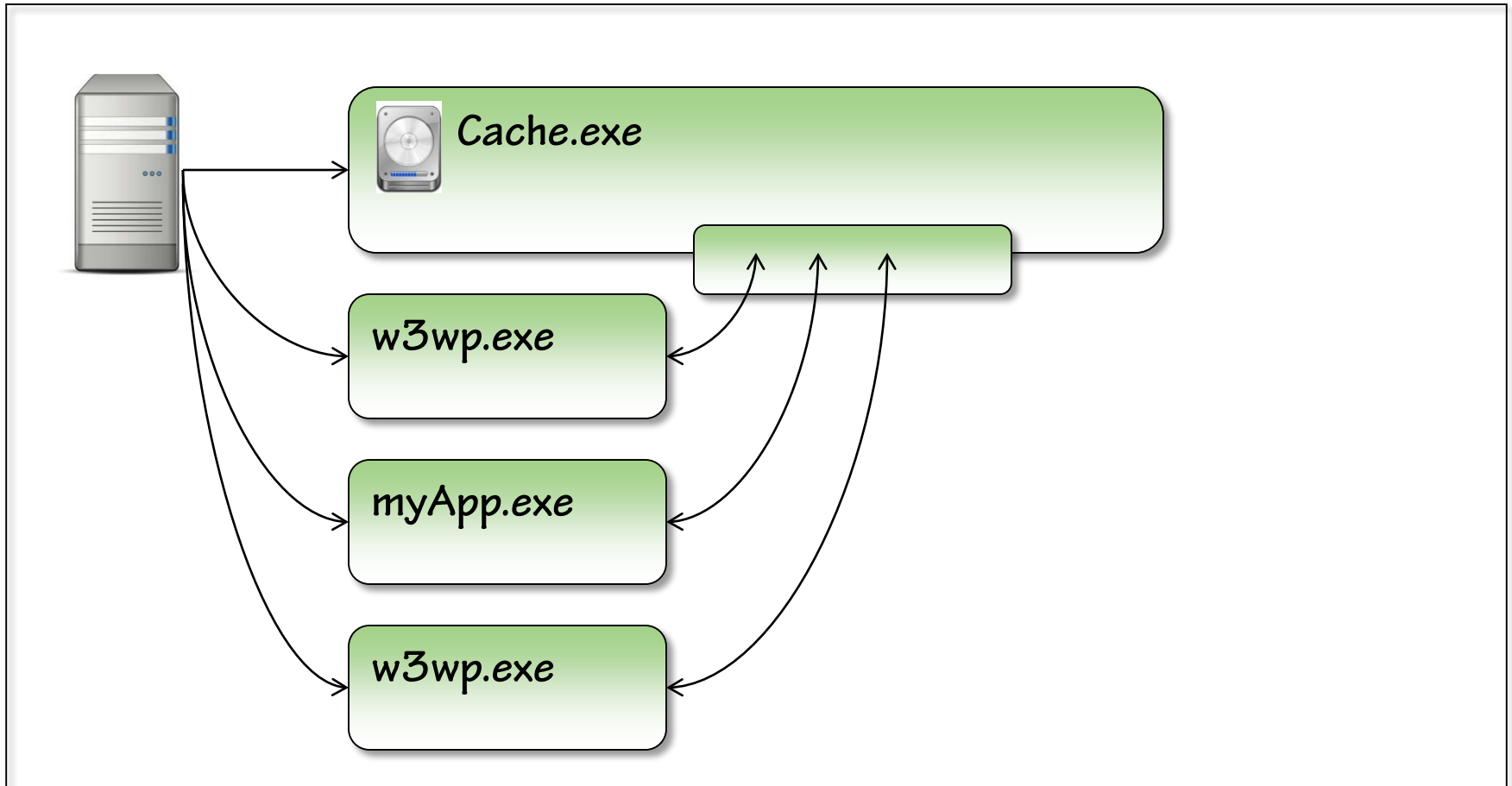
Types of Local Cache Store

- Out-of-process



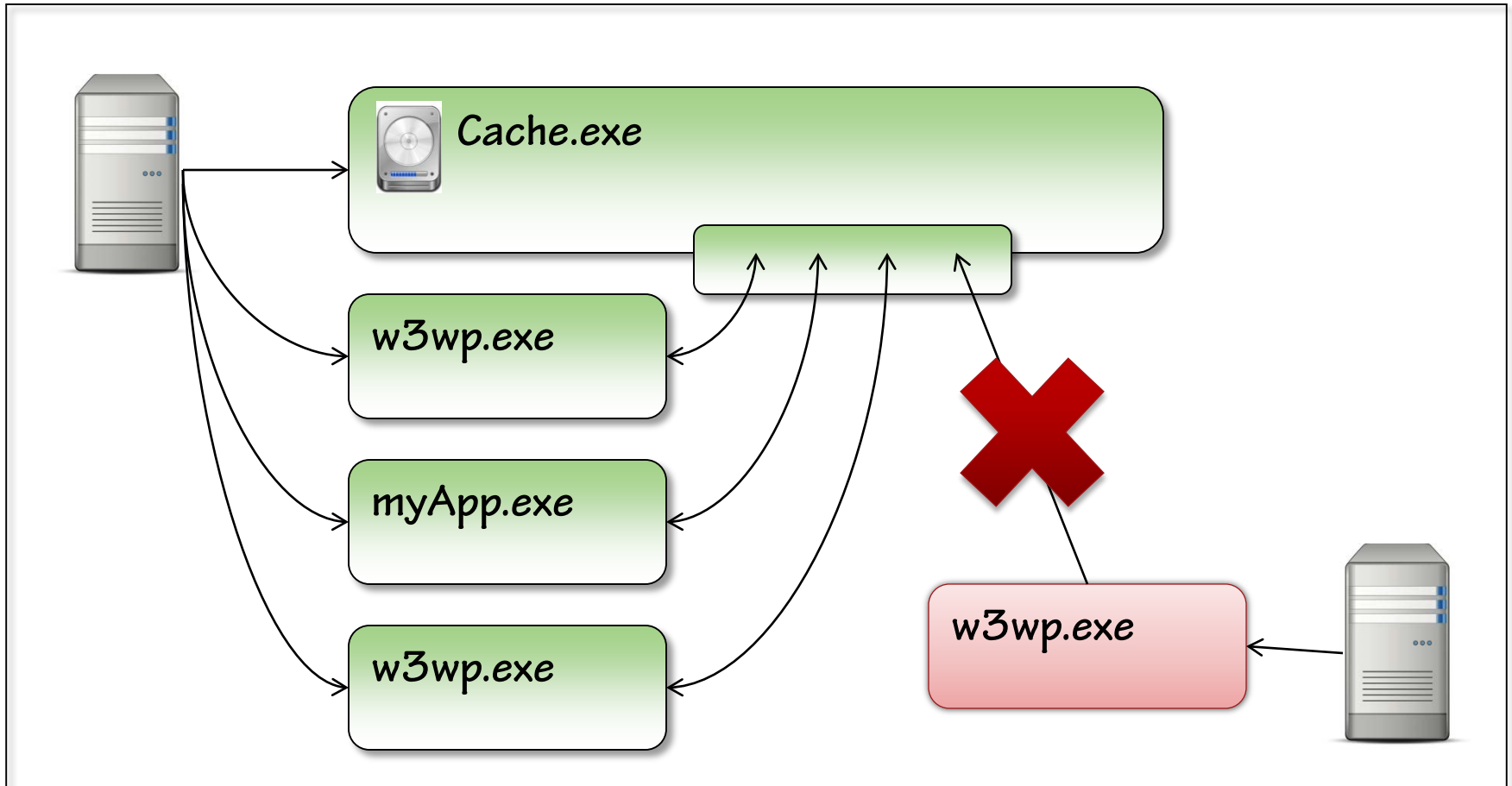
Types of Local Cache Store

- Out-of-process



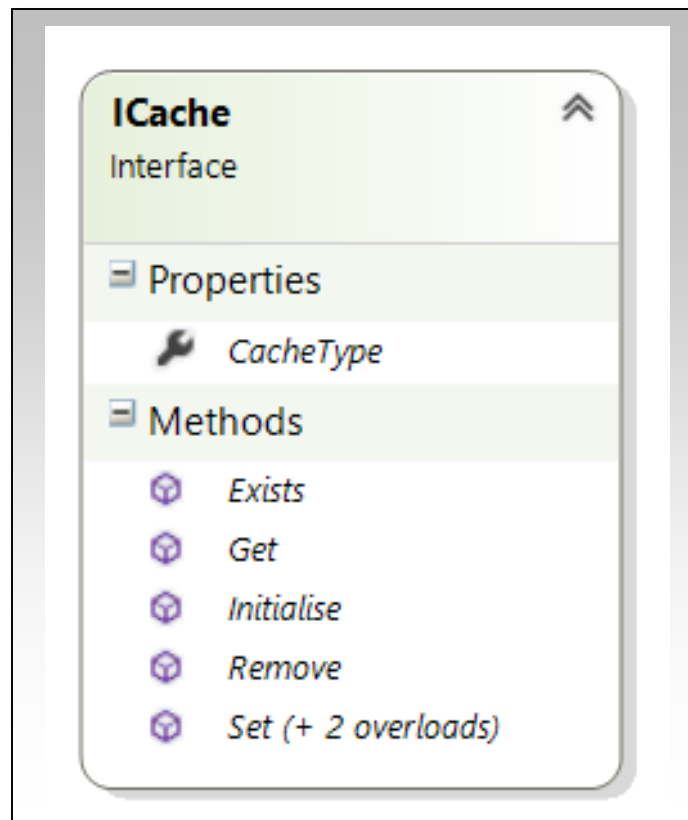
Types of Local Cache Store

- Out-of-process



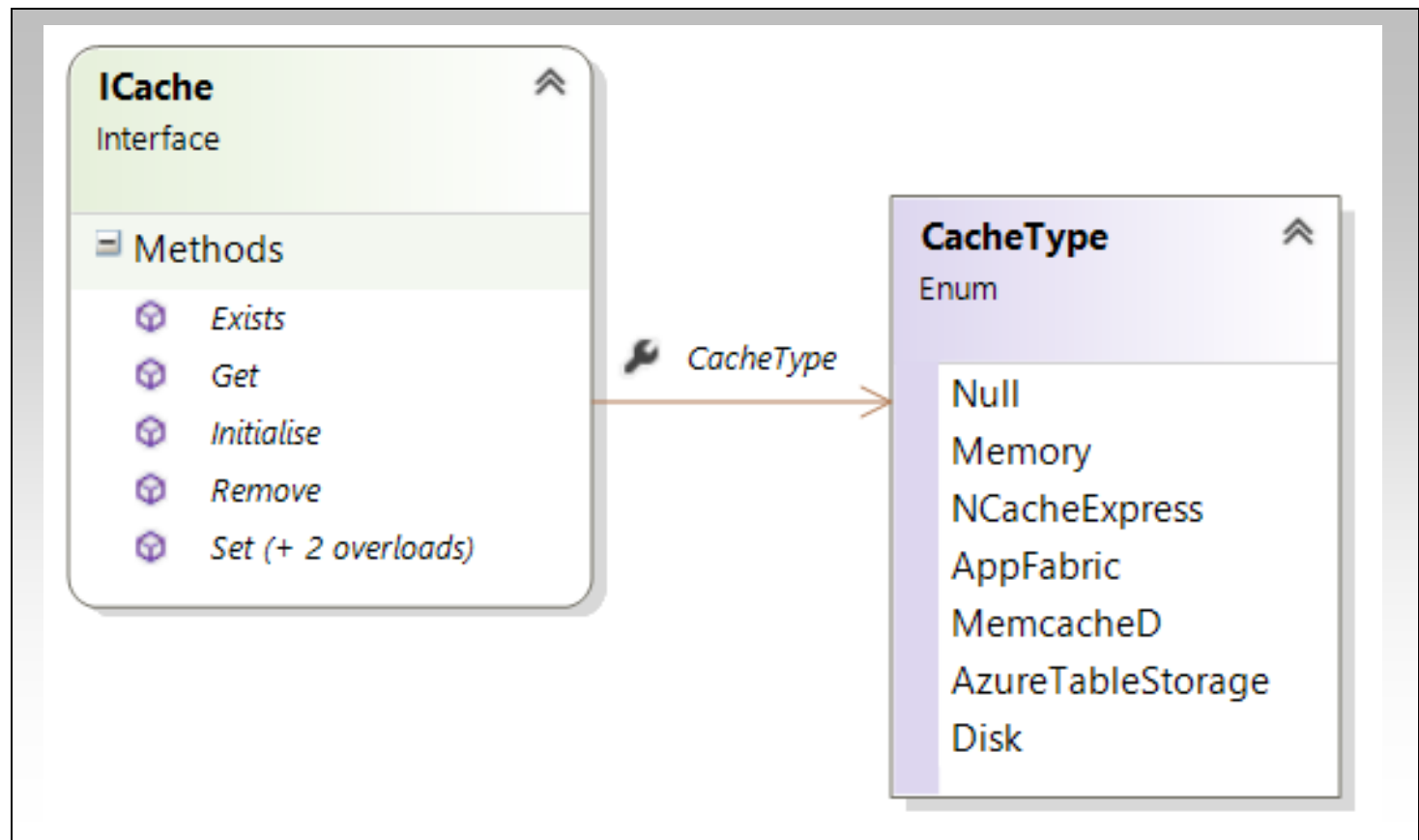
Abstracting the Store from the Solution

- **ICache interface**
 - Common subset of features to support get/check/set



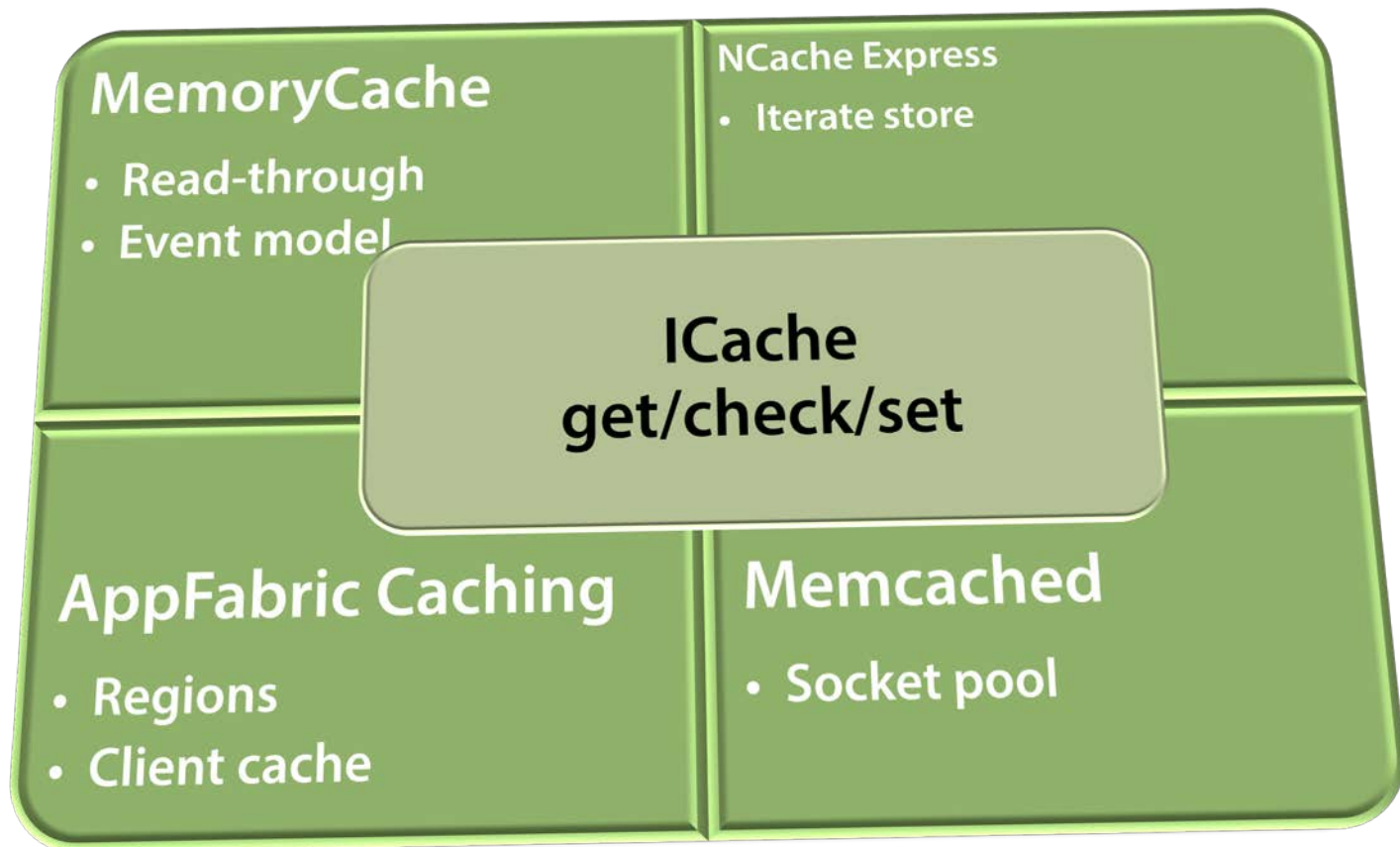
Abstracting the Store from the Solution

- ICache interface
 - Use all stores in the same way



Abstracting the Store from the Solution

- ICache subset
 - Lose extended functionality of the store



.NET MemoryCache

- **In-process cache**
- **System.Runtime.Caching**
 - .NET 4.0, evolved from System.Web.Caching in .NET 2.0

.NET MemoryCache

- **Demo**

.NET MemoryCache

- **In-process cache**
 - Cache is lost if process ends
 - Multiple processes cannot share caches
- **Configure basic settings**

```
<system.runtime.caching >  
  <memoryCache>  
    <namedCaches>  
      <add name="Sixeyed.Core.Cache"  
          cacheMemoryLimitMegabytes="200"  
          pollingInterval="20"/>  
    </namedCaches>  
  </memoryCache>  
</system.runtime.caching >
```

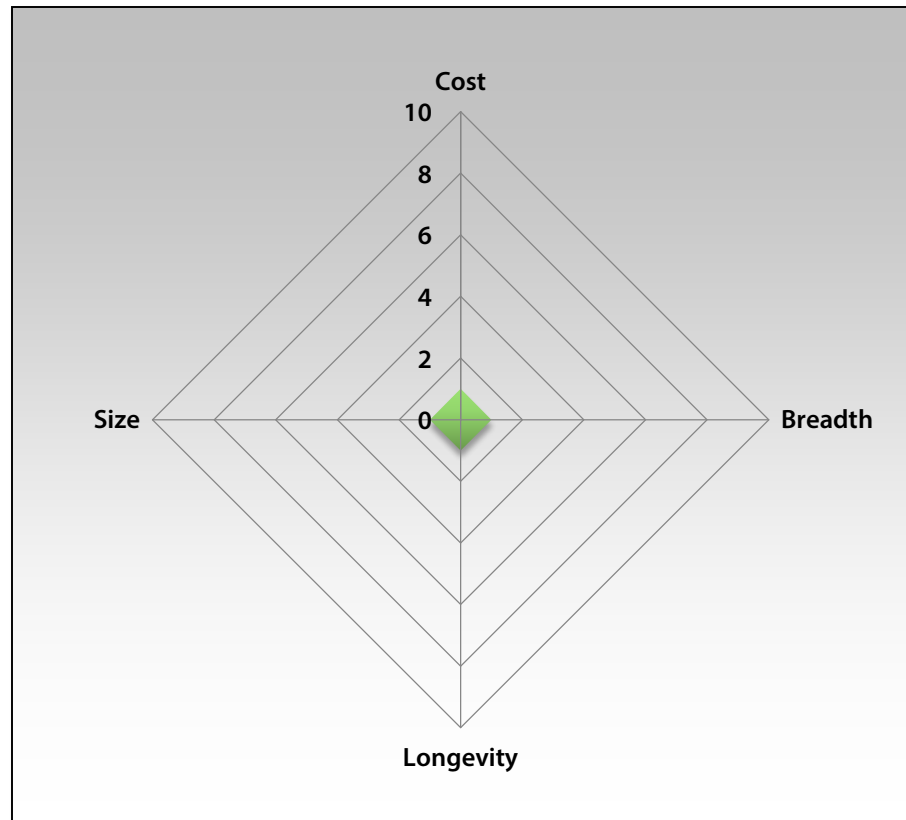
- **No management interface**
- **Advanced features**
 - Event model
 - Change monitors

.NET MemoryCache

- **Suitability applied to the decision matrix**
 - Cost – very fast memory lookup
 - Breadth – only available to the process
 - Longevity – lost when process ends
 - Size – limited, shared with process

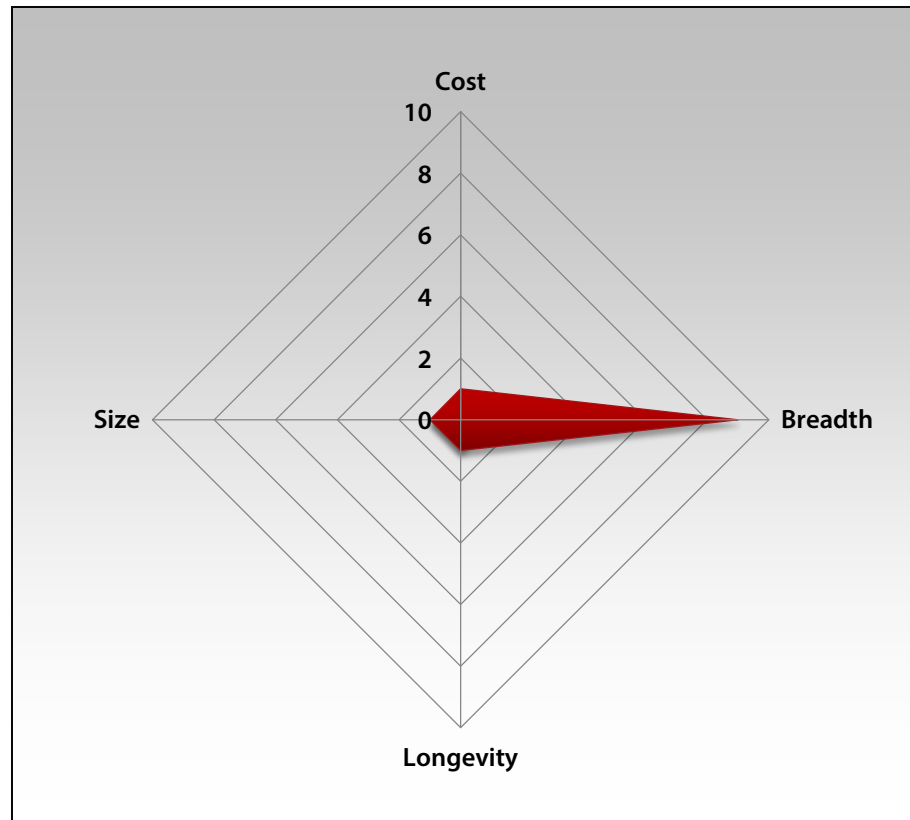
.NET MemoryCache

- Theoretically, limited suitability



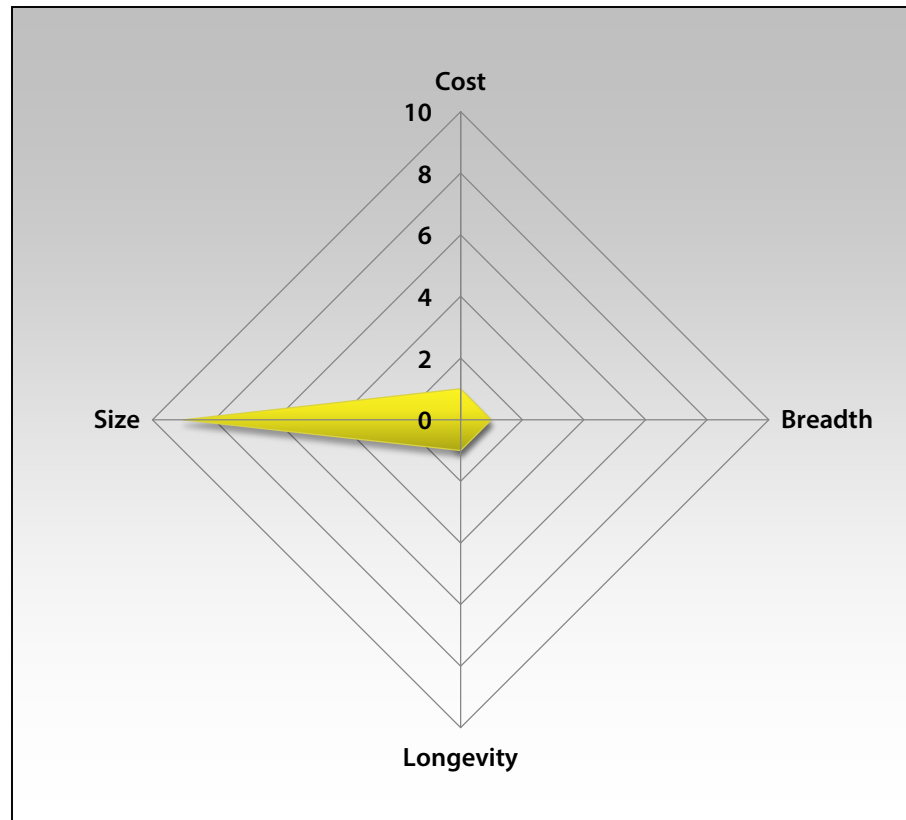
.NET MemoryCache

- Wide breadth, no re-use



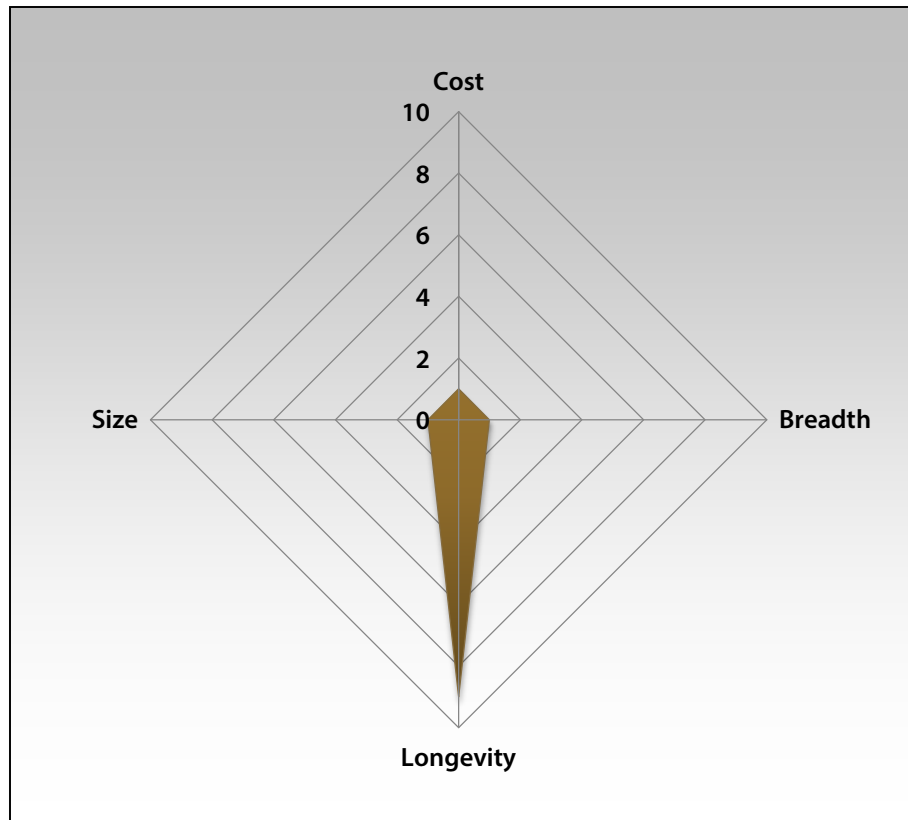
.NET MemoryCache

- Larger objects, cache quickly filled



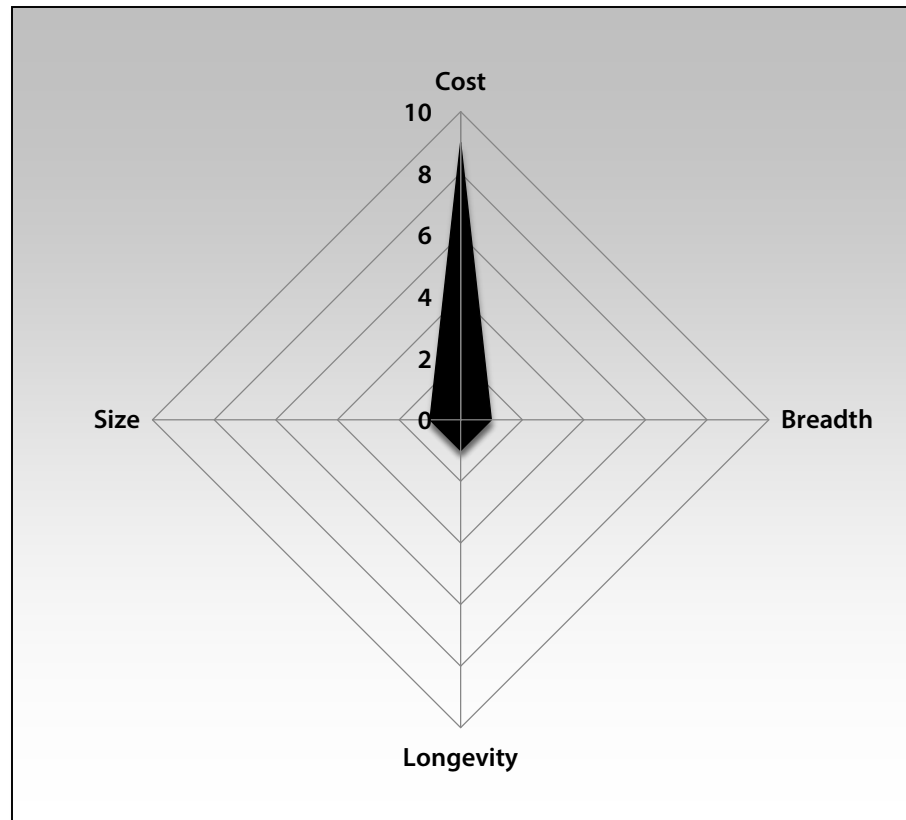
.NET MemoryCache

- Long lived, will expire on reset



.NET MemoryCache

- Large objects, all caches incur fetch cost



NCache Express

- **Out-of-process cache**
 - Windows Service
 - Local or remote
- **NCache product**
 - Free Express edition
 - Limited to 2 nodes & 500Mb size

NCache Express

- Demo

NCache Express

- **Out-of-process cache**

- Multiple processes can share the cache
- Windows Service

- **Configure cache setup**

```
<cache-config name="Sixeyed.Core.Cache" inproc="false">  
  <storage cache-size="200mb"/>  
  <cleanup interval="60sec"/>  
  <eviction-policy default-priority="normal"  
    eviction-ratio="10%"  
    eviction-enabled="true"/>
```

- **Command line management tools**

- listcaches; clearcache

- **Extra features**

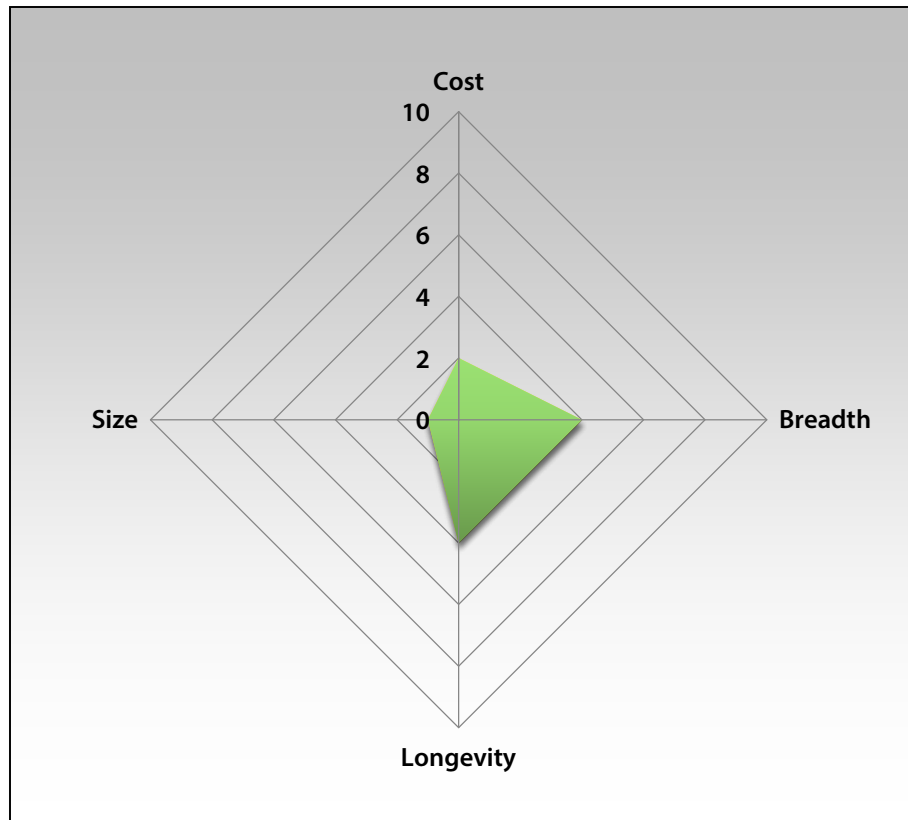
- Perfmon counters

NCache Express

- **Suitability applied to the decision matrix**
 - Cost – fast binary communication across processes
 - Breadth – available across processes on the machine
 - Longevity – machine uptime
 - Size – limited to 500Mb

NCache Express

- Wider suitability, still limited



AppFabric Caching

- **Windows Server AppFabric**
 - "Windows Server AppFabric Cache" – Jon Flanders
- **Windows Server AppFabric**
 - "Application Server"
 - Caching
 - WCF and WF hosting – deployment, monitoring and persistence
 - Minimum OS: Vista, Server 2003
- **Out-of-process cache**
 - Windows Service with PowerShell management
 - Local or remote

AppFabric Caching

- Demo

AppFabric Caching

- **Out-of-process cache**

- Multiple processes can share the cache
- Windows Service

- **Shared configuration**

```
<cache consistency="StrongConsistency" name="Sixeyed-Core-Cache">  
  <policy>  
    <eviction type="Lru" />  
    <expiration defaultTTL="10"  
      isExpirable="true" />  
  </policy>  
</cache>
```

- **PowerShell management tools**

- get-cache; get-cachestatistics; restart-cachecluster

- **Extra features**

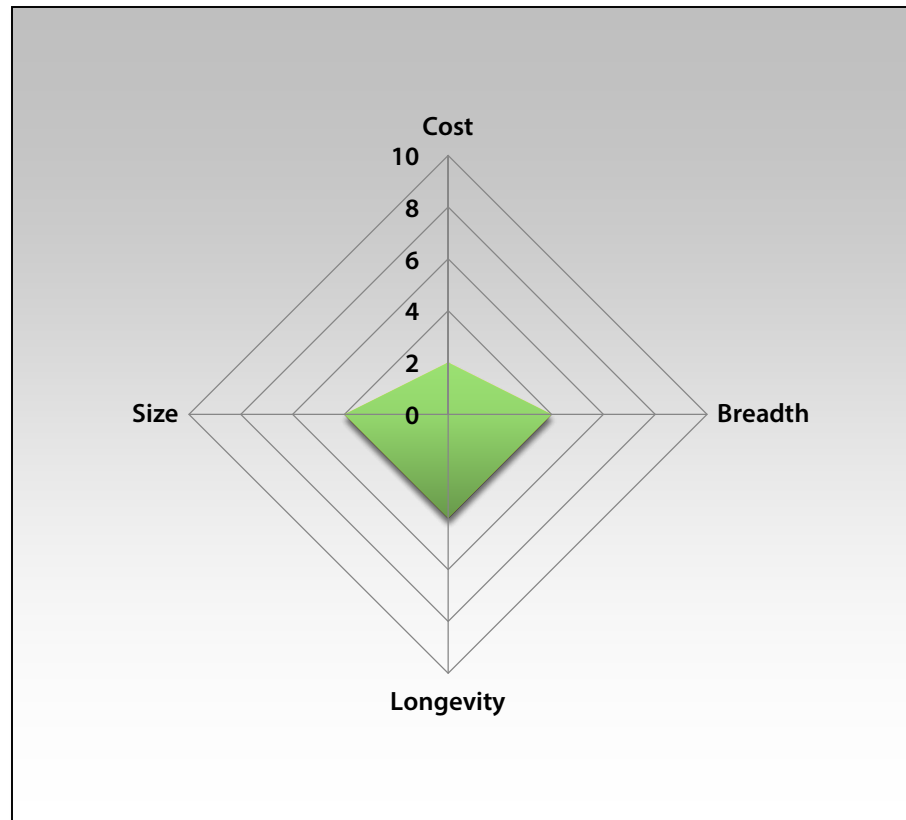
- HA, concurrency, read-through, write-through

AppFabric Caching

- **Suitability applied to the decision matrix**
 - Cost – fast binary communication
 - Breadth – available across processes
 - Longevity – machine uptime
 - Size – limited by physical memory

AppFabric Caching

- Widest



The NullCache

- **NullCache implements ICache**
 - Replaces concrete cache stores if not available
 - Does nothing
- **Caching as a non-essential non-functional component**
 - Failure in the cache should not cause system failure
- **Graceful degradation**
 - CacheBase wraps all access in try/catch
 - Solution functionally the same
 - Performance will be impacted

Summary

- **Types of local cache store**
 - In-process
 - Out-of-process
- **Local caches**
 - .NET MemoryCache
 - NCache Express
 - Windows Server AppFabric Caching
- **Abstracting the store from the solution**
 - ICache interface
 - NullCache and CacheBase

References

- **Ncache Express**

- Download

- <http://www.alachisoft.com/download.html#ncache>

- Documentation

- <http://www.alachisoft.com/ncache/ncache-express.html>

- **AppFabric 1.1 for Windows Server**

- Download

- <http://www.microsoft.com/en-us/download/details.aspx?id=27115>

- Documentation

- <http://msdn.microsoft.com/en-us/windowsserver/ee695849.aspx>