



PV178 – Cvik010

Agenda

- Synchronizační konstrukty
- **Parallel vs PLINQ**
- **Asynchronní programování**

Upozornění

- Speciální odpovědník „Téma projektu“
- Stručné zadání projektu
- Musí být schváleno mnou před deadline

Interlocked

- Mnohonásobně rychlejší jak `Lock`
- Pouze pro práci s `int` a `long`
- Processor umí udělat některé operace atomicky (jednou instrukcí - ThreadSafe)
- Proměnná musí být předána přes `ref`
- **Metody:** Add, Increment, Read, Exchange, CompareExchange

```
int number = 0;  
Interlocked.Increment(ref number);
```

MemoryBarrier

- Zabraňuje optimalizacím před a po čtení proměnné

```
int result = 10;
bool complete = false;

// Zaručuje, že result bude zapsán dříve než complete
Thread.MemoryBarrier();
complete = true;

// Refresh hodnoty pro další vlákna
Thread.MemoryBarrier();
```

volatile

- Zabraňuje optimalizacím
- Synchronizace s nejnovější hodnotou
- Neposkytuje sync mezi Read a Write
- **NEPOUŽÍVAT !!!**

```
private volatile int x;
```

AutoResetEvent a ManualResetEvent

- Signalizace
- Dává možnost notifikovat, že je něco: hotovo | připraveno | apod.

```
AutoResetEvent autoResetEvent = new AutoResetEvent(false);  
int result = 0;
```

```
Task.Run(() =>  
    {  
        Thread.Sleep(1000);  
        result = 10;  
        autoResetEvent.Set();  
    });
```

```
// Čekání na signál  
autoResetEvent.WaitOne();
```

```
Console.WriteLine($"Result: {computedResult}");
```

Samostatná práce

- Signaling



Parallel vs PLINQ

- Parallel
 - Třída, metody Stop(), Break()
 - NEUMÍ zachovávat jednoduše pořadí
- PLINQ
 - sada rozšiřujících metod
 - lze jednoduše zachovat pořadí prvků

Parralel



Samostatná práce

- ParallelProgramming



Porovnání výkonu LINQ a PLINQ



Asynchronní programování

- Zlepšuje odezvu aplikace
- Zajišťuje responzivitu grafického rozhraní aplikace (GUI)
 - „Uživatel nechce synchronně čekat, on chce asynchronně klikat“
- Vhodné využít například u:
 - přístupu k webu (volání API, ...)
 - práce se soubory
 - zpracování velkých dat (obrazu, videa ...)

Asynchronní programování - demo

- Demo AsyncWFDemo_02



Async



- Klíčové slovo
- Umožňuje asynchronní spuštění metody
- Metoda označená `async` musí vrátet `Task` alebo `Task<T>`
- Async metody nemohou používat `out` a `ref`
- Metoda by se měla jmenovat: `MyMethodAsync(...)`
- Uvnitř metod **by měl** být `await`

```
public static async Task TestMethodAsync()  
{  
    ...  
}
```

Await

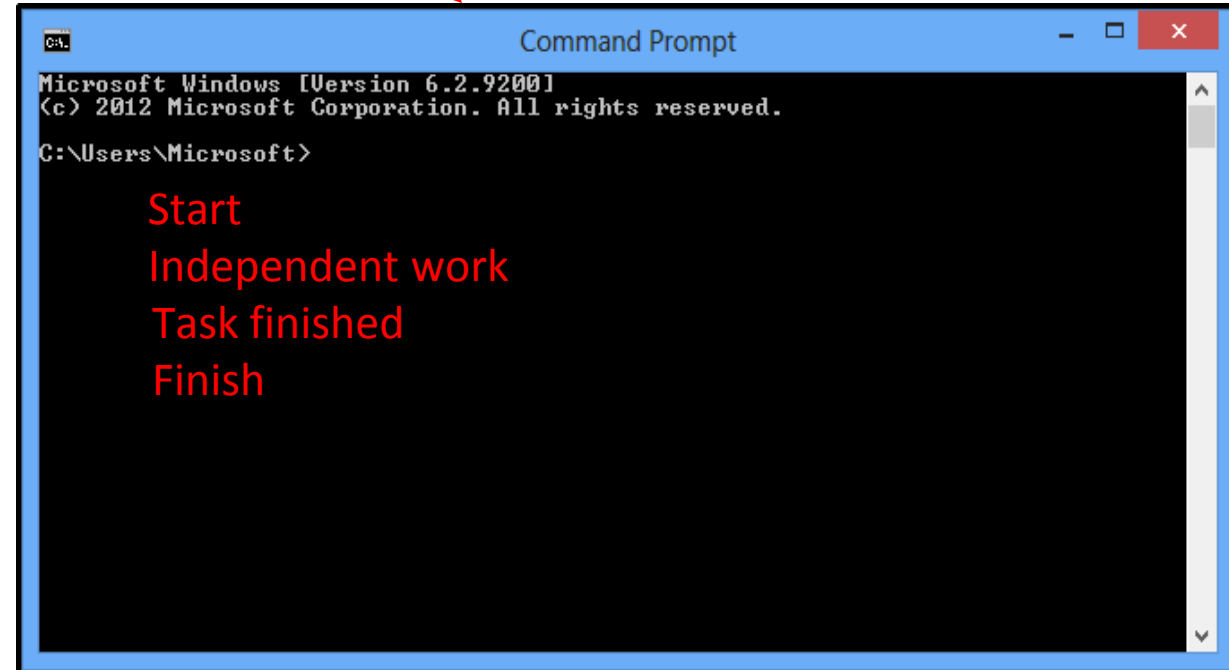
- Klíčové slovo
- Umožňuje asynchronní čekání na metody
- Označují místo čekání
- Čekat pomocí await lze jen na async metodu
- **Měl by** se vykytovat alespoň jednou v asynchronní metodě
- **Nemůže** být v klasické metodě

```
public static async Task TestMethodAsync()  
{  
    await Task.Run(() => { });  
}
```

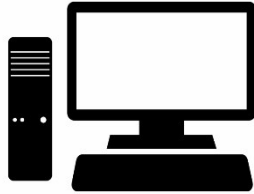


Asynchronní programování – flow

```
public static async Task Main()  
{  
    [redacted]  
    [redacted] (_ => {  
        [redacted]  
        [redacted]  
        [redacted]  
        [redacted]  
        [redacted]  
    })  
}
```



Asynchronní programování a uživatel



```
public static void Main()
{
    Console.WriteLine("Start");
    Task t = Task.Run(_ => {
        Thread.Sleep(15000);
        Console.WriteLine("Task finish");
    });
    Console.WriteLine("Independent work");
    await t;
    Console.WriteLine("Finish");
}
```

- Uživatel čeká – nemůže klikat ($< 1 \mu\text{s}$)
- Uživatel čeká – nemůže klikat (15s), hlavní vlákno je blokováno
- Uživatele nezajímá – běží asynchronně (souběžně s hlavním vláknem)
- Uživatel nečeká – může klikat, hlavní vlákno není blokováno

Asynchronní metody s void

```
public async void MyMethodAsync()  
{  
    await Task.Delay(5);  
}  
// await MyMethodAsync() nejde
```

- Znemožňuje čekání na metodu
- Speciální použití (obsluha událostí ve WinForms)
- Kromě speciálních případů nepoužívat!



Samostatná práce

- Asynchronous Programming



ValueTask

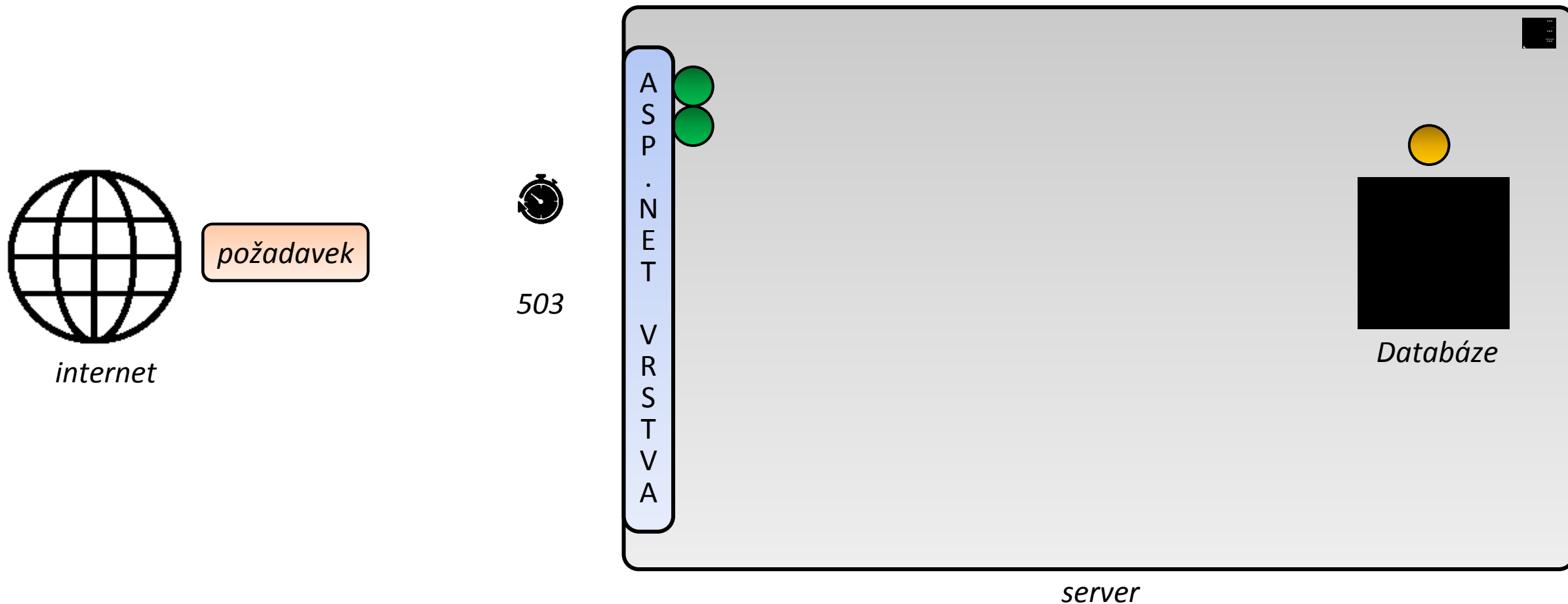
- Použití jako Task
- Zlepšuje výkon
- Od C# 7.0
- Vyšší flexibilita
- Demo: `ValueTaskExample.cs`

Asynchronní programování – procvičení na doma

- Exercises_XX.cs



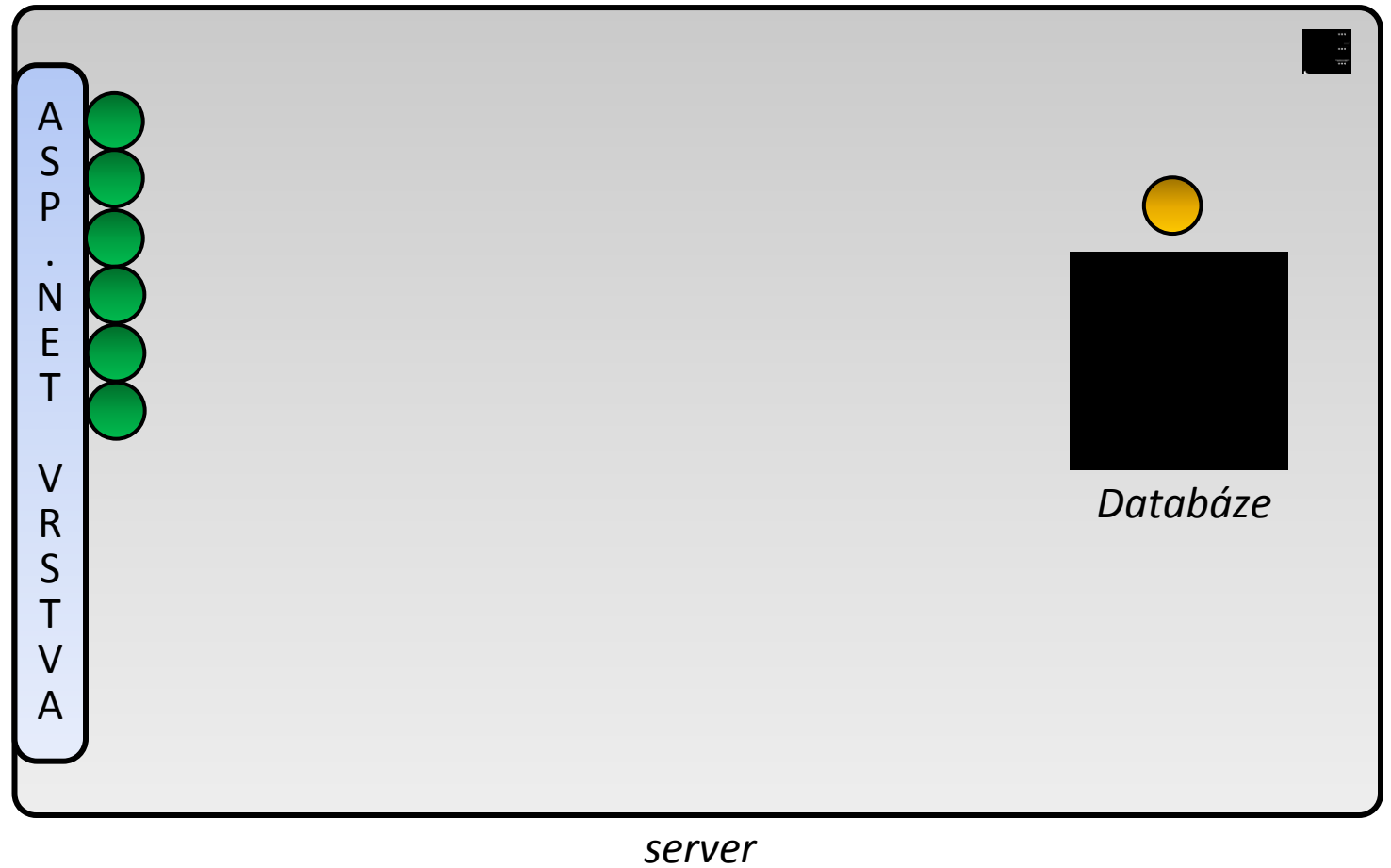
ASP.NET - bez async/await



● ASP.NET vlákno

● Dedicované SQL vlákno (obsluhuje pouze DB)

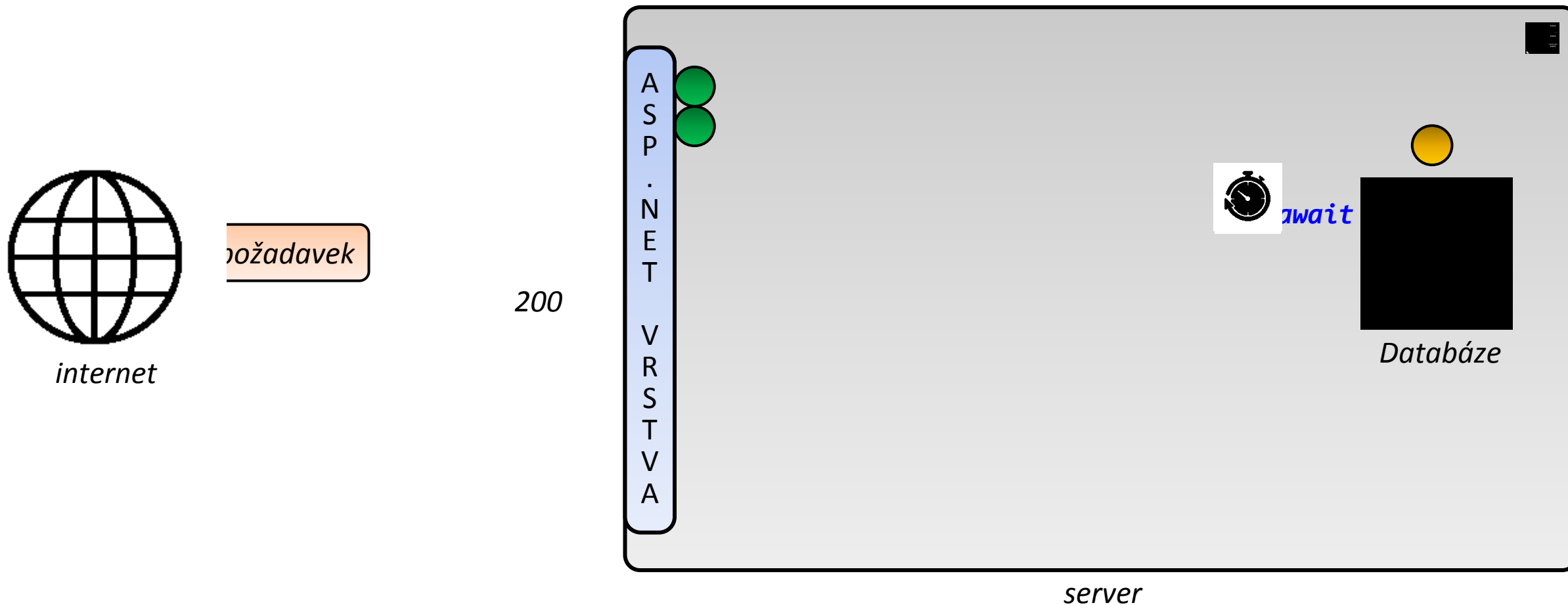
ASP.NET - bez async/await (řešení?)



 ASP.NET vlákno

 *Dedikované SQL vlákno (obsluhuje pouze DB)*

ASP.NET - s async/await



 ASP.NET vlákno

 Dedikované SQL vlákno (obsluhuje pouze DB)