

Jiří Kanda

Software development lead, HAVIT, s.r.o.

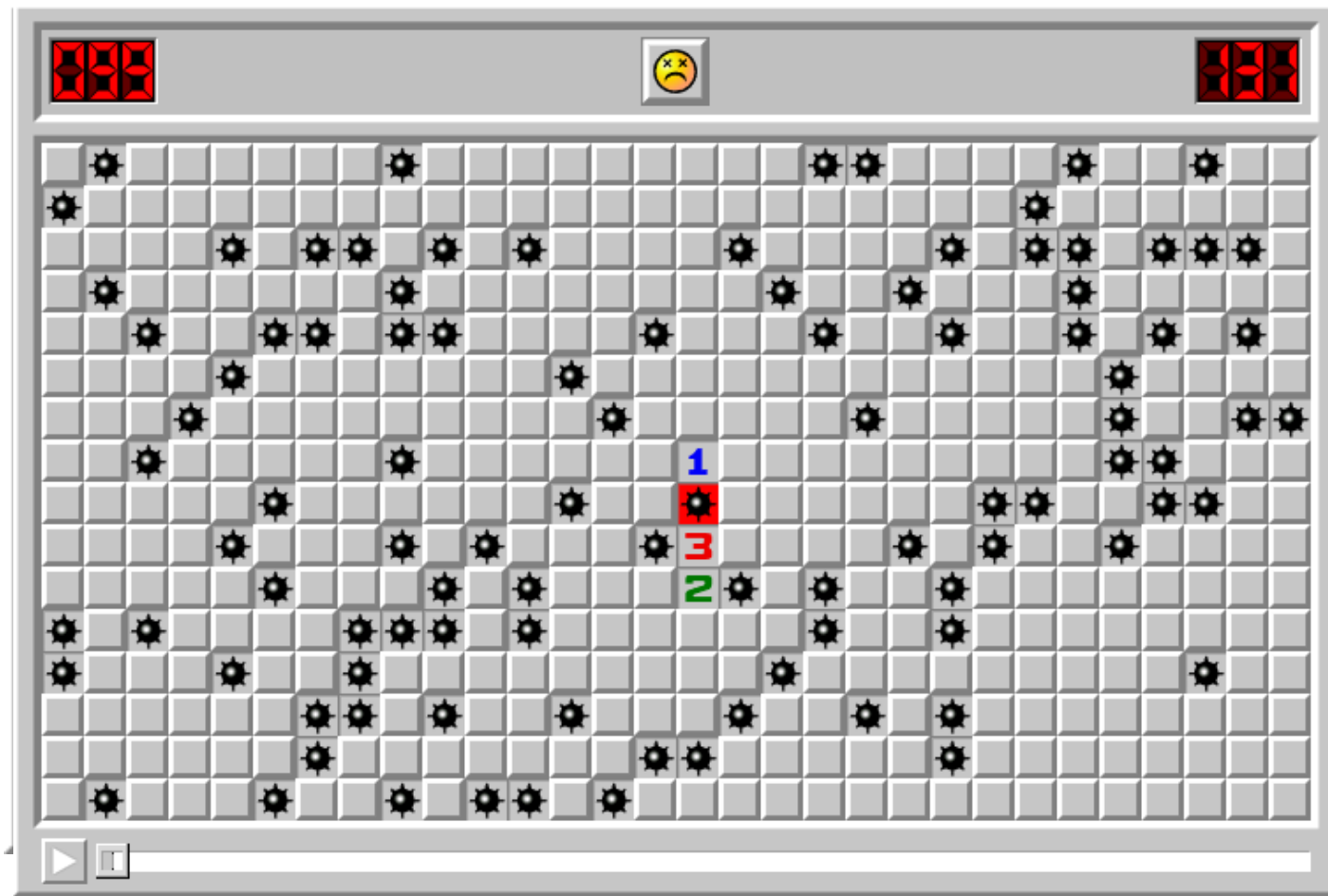
kanda@havit.cz

# Minesweeping

Všechny cesty do pekla jsou dlážděny  
dobrými úmysly.



<https://minesweeper.online/>



# Agenda

- Microsoft.Data.SqlClient & async/await
- DevDrive & Web Deployment Package
- Build Time
- Application Insights Profiler
- Blazor & Inputs & Exception handling
- Azure Blob Storage SDK a ASP.NET Core s EnableBuffering
- Central Package Management



# Microsoft.Data.SqlClient & async/await

- <https://github.com/dotnet/SqlClient/issues/593>



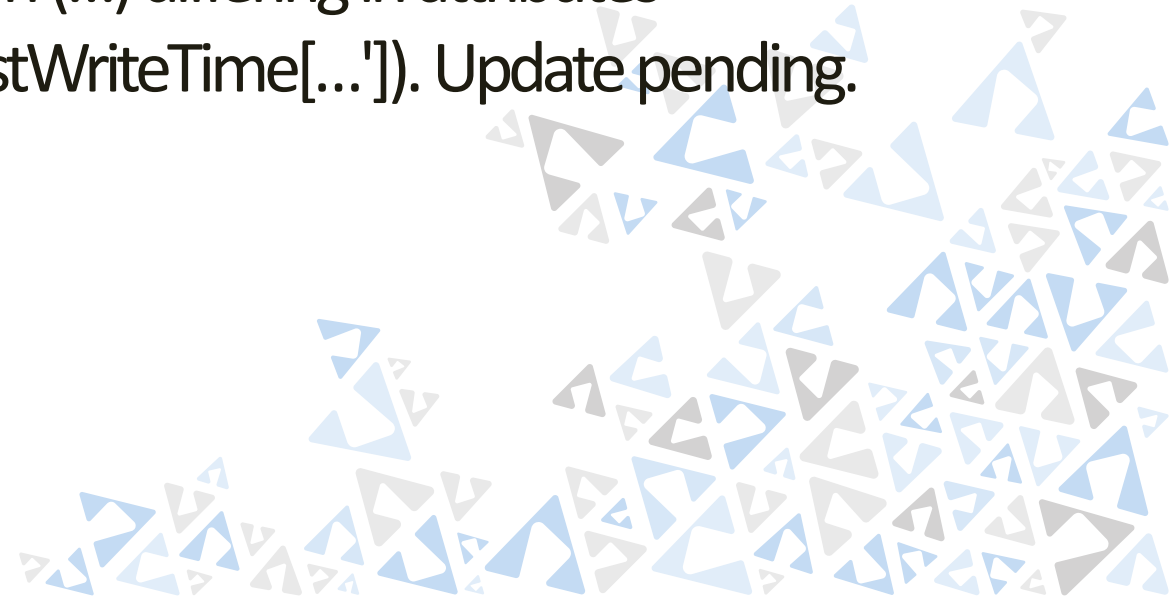
# DevDrive & Web Deployment Package

- [DevDrive](#)



# DevDrive & Web Deployment Package

- Error: The synchronization is being stopped because the maximum number of sync passes '5' has been exceeded even though all the changes could not be applied. This could occur if there are external changes being made to the destination.
- Source filePath (...) does not match destination (...) differing in attributes (`attributes['ReadOnly, SparseFile', 'Archive']`, `lastWriteTime[...]`). Update pending.
- [FileAttributes](#)



# DevDrive & Web Deployment Package

- <https://github.com/dotnet/sdk/issues/40562>



# DevDrive & Web Deployment Package

```
$stream = [System.IO.File]::Open($zipFile.FullName, [System.IO.FileMode]::Open, [System.IO.FileAccess]::ReadWrite)
# Open the ZIP file in update mode
$zipArchive = New-Object System.IO.Compression.ZipArchive($stream, [System.IO.Compression.ZipArchiveMode]::Update)
foreach($zipEntry in $zipArchive.Entries)
{
    # C# equivalent: archiveEntry.ExternalAttributes &= ~(int)FileAttributes.SparseFile;
    $zipEntry.ExternalAttributes = $zipEntry.ExternalAttributes -band (-bnot [int][System.IO.FileAttributes]::SparseFile)
}
$zipArchive.Dispose() #_SAVE_ the ZIP archive and close the stream
```

# Build time

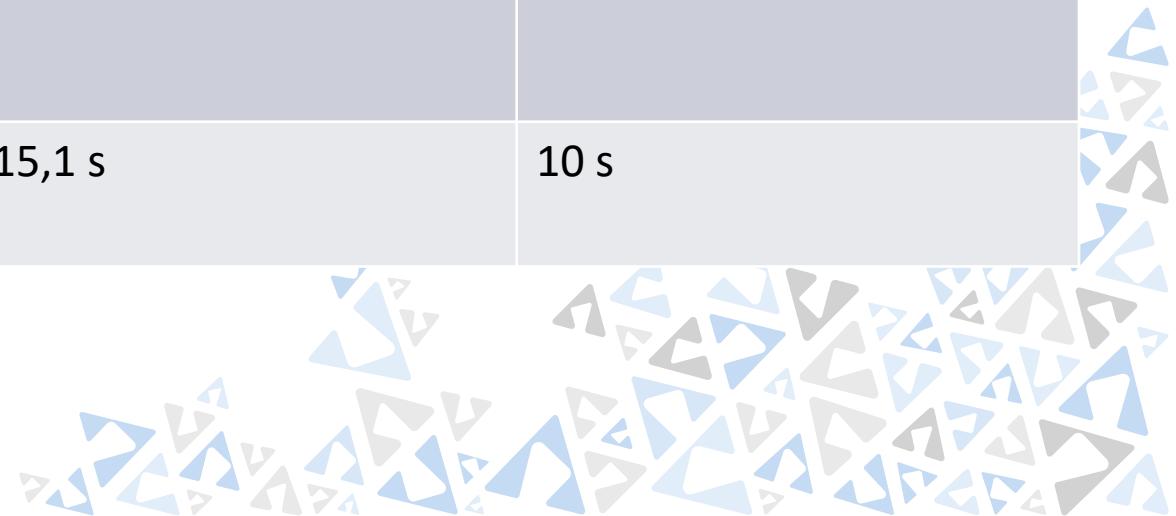
dotnet build

- Trvá nečekaně dlouho.
- Proč?
- <https://github.com/microsoft/vs-threading/issues/1526>



# Build time

	včetně *.Designer.cs	bez *.Designer.cs (smazány)
Kompletní build solution se vším	49,9 s	17,1 s
Entity bez Microsoft.VisualStudio.Threading.Analyzers	24,2 s	16,1 s
Solution bez Microsoft.VisualStudio.Threading.Analyzers	17,4 s	11 s
Úplně bez analyzérů	15,1 s	10 s



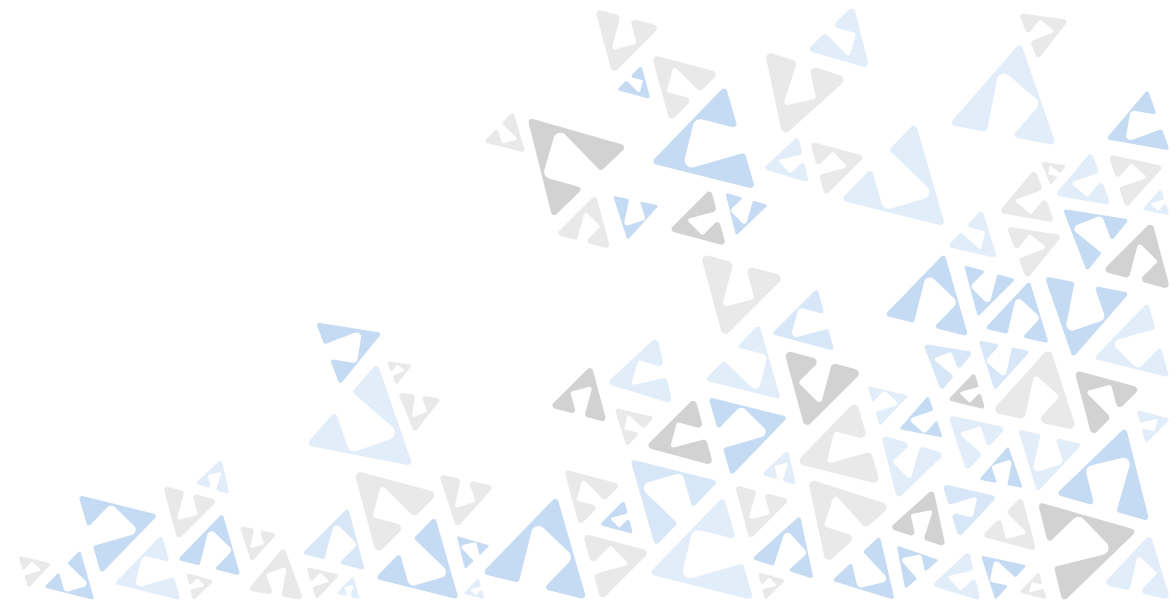
# Application Insights Profiler

- ASP.NET Core aplikace
- Azure App Service
- Píše se rok 2022...

Časový limit na provedení operace vypršel.

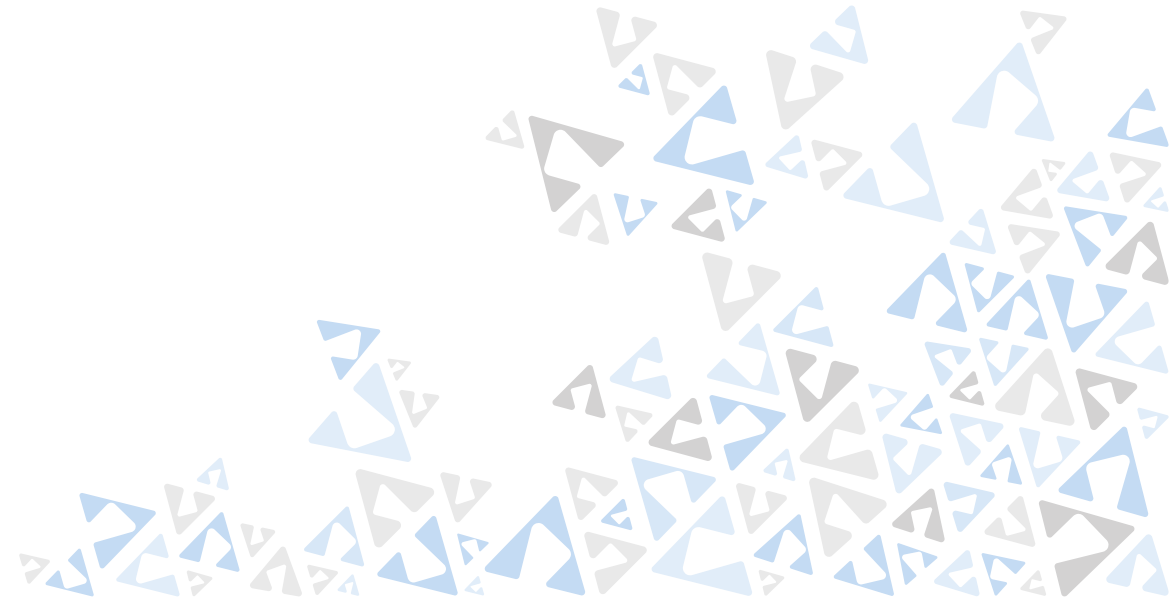


- 0-3 výskyty / den



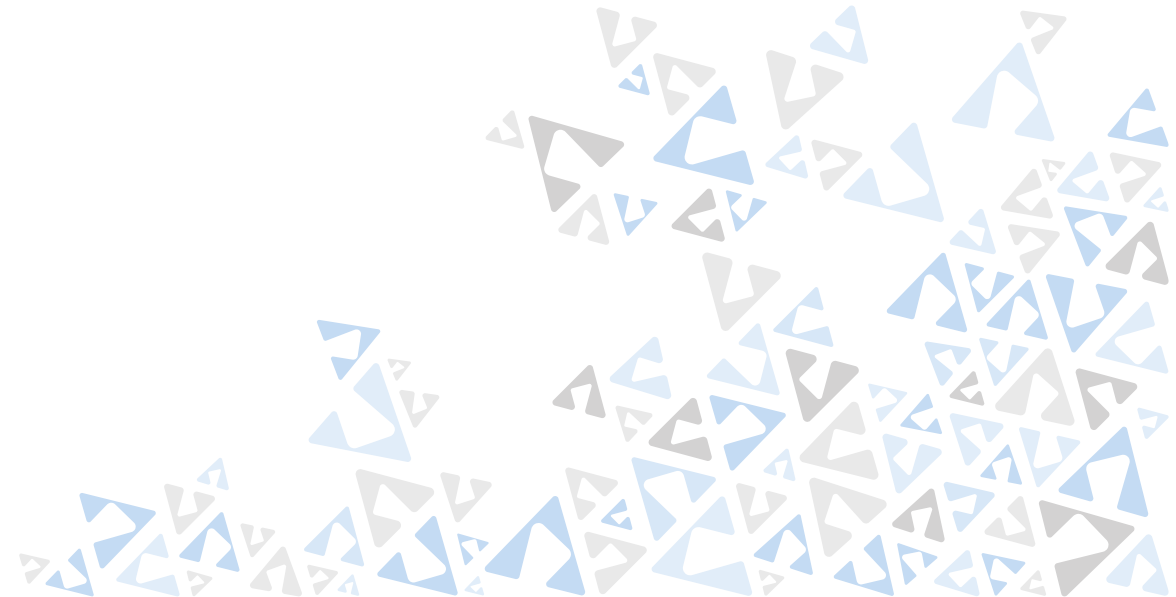
# Application Insights Profiler

- Spojení, metro, výtah...
- Měření rychlosti internetu
- Neoptimální execution plan, upgrade Azure SQL Elastic Pool, ...
- CPU intensive období (zpochybněno)



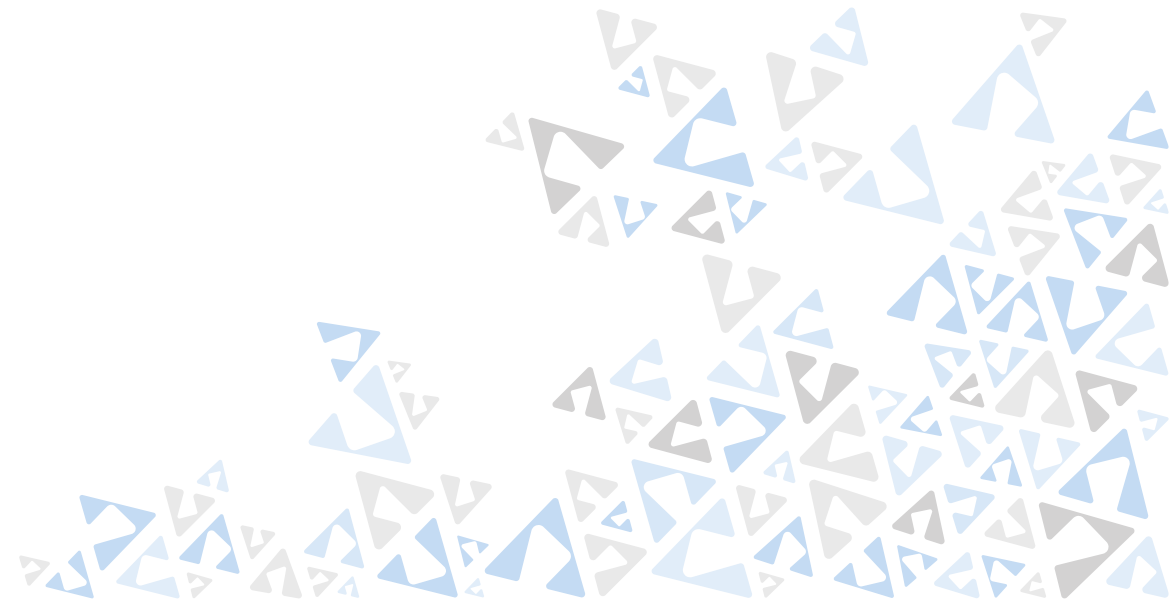
# Application Insights Profiler

- žádný async/await kód
- Application Insight
  - Request, který má jen SQL závislosti, každá závislost kratší než 100ms
  - Ale mezi závislostmi občas existuje prodleva 10 sekund
  - (4x OK, pak 10 sekund prodleva, atp.)



# Application Insights Profiler

- Některé requesty, které začnou v lehce jiný čas (rozdíl jednotek sekund), se zaseknou a skončí víceméně stejně. Jakoby se requesty odšpuntovaly.
- To by chtělo Application Insights Profiler
- Je zapnutý!



Framework/Library AsyncMethodBuilderCore.Start	System.Private.CoreLib.il	42007.77 ms
LoanCaseHypotecniBankaFacade+ <PatchLoanCaseAsync>d__22.MoveNext	hav.it.uverovaplatforma.facades	42007.77 ms
HypotecniBankaLoanCasePatcher.PatchLoanCaseAndSaveAsync	hav.it.uverovaplatforma.services	22900.27 ms
PartnerAuthorizationVerifier.GetAndVerifyLoanCase	hav.it.uverovaplatforma.services	18862.88 ms
DbRepository`1.GetObject	hav.it.data.entityframeworkcore.patterns	18836.31 ms
Framework/Library EntityFinder`1.Find	microsoft.entityframeworkcore	18836.16 ms
CILJit::compileMethod	clrjit	18806.44 ms
jitNativeCode	clrjit	18806.44 ms
Compiler::compCompile	clrjit	18806.44 ms
Compiler::compCompileHelper	clrjit	18806.44 ms
Compiler::compCompile	clrjit	18806.44 ms
Compiler::fgImport	clrjit	18806.44 ms
Compiler::implImport	clrjit	18806.44 ms
Compiler::implImportBlock	clrjit	18806.44 ms
Compiler::implImportBlockCode	clrjit	18806.44 ms
Compiler::implImportCall	clrjit	18806.44 ms
Compiler::impMarkInlineCandidateHelper	clrjit	18806.44 ms
Framework/Library CEEInfo::runWithErrorTrap	coreclr	18806.44 ms
<lambda_7935d81526369d076ecfcd6ed9223d0d>::operator	clrjit	18806.44 ms
Framework/Library CEEInfo::canInline	coreclr	18806.44 ms
Waiting		18804.99 ms
CPU Time		1.45 ms
dynamicClass.lambda_method2553	Anonymously Hosted DynamicMethods Assembly	21.23 ms



▼ 🔥 JsonSerializerInternalReader.PopulateObject	newtonsoft.json	⊗	22630.46 ms
▼ 🔥 JsonSerializerInternalReader.SetPropertyValue	newtonsoft.json	⊗	22629.19 ms
▼ 🔥 JsonSerializerInternalReader.CreateValueInternal	newtonsoft.json	⊗	11359.12 ms
🔥   JsonSerializerInternalReader.CreateList	newtonsoft.json	⊗	11359.12 ms
▼ 🔥 JsonSerializerInternalReader.PopulateList	newtonsoft.json	⊗	11359.12 ms
▼ 🔥 JsonSerializerInternalReader.CreateValueInternal	newtonsoft.json	⊗	11358.12 ms
🔥   JsonSerializerInternalReader.CreateObject	newtonsoft.json	⊗	11358.12 ms
▼ 🔥 JsonSerializerInternalReader.PopulateObject	newtonsoft.json	⊗	11358.12 ms
▼ 🔥 JsonSerializerInternalReader.SetPropertyValue	newtonsoft.json	⊗	11357.11 ms
🔥   ExpressionValueProvider.SetValue	newtonsoft.json	⊗	11357.11 ms
🔥   ExpressionReflectionDelegateFactory.CreateSet	newtonsoft.json	⊗	11357.11 ms
▼ 🔥 Framework/Library LambdaCompiler.Compile	system.linq.expressions.il	⊗	11357.11 ms
▼ 🔥 CILJit::compileMethod	clrjit	⊗	11351.41 ms
🔥   jitNativeCode	clrjit	⊗	11351.41 ms
🔥   Compiler::compCompile	clrjit	⊗	11351.41 ms
🔥   Compiler::compCompileHelper	clrjit	⊗	11351.41 ms
▼ 🔥 Compiler::compCompile	clrjit	⊗	11351.41 ms
▼ 🔥 Compiler::fgInline	clrjit	⊗	11343.86 ms
🔥   InlineResult::Report	clrjit	⊗	11343.86 ms
▼ 🔥 Framework/Library CEEInfo::reportInliningDecision coreclr	coreclr	⊗	11343.86 ms
🔥 Waiting			11343.10 ms

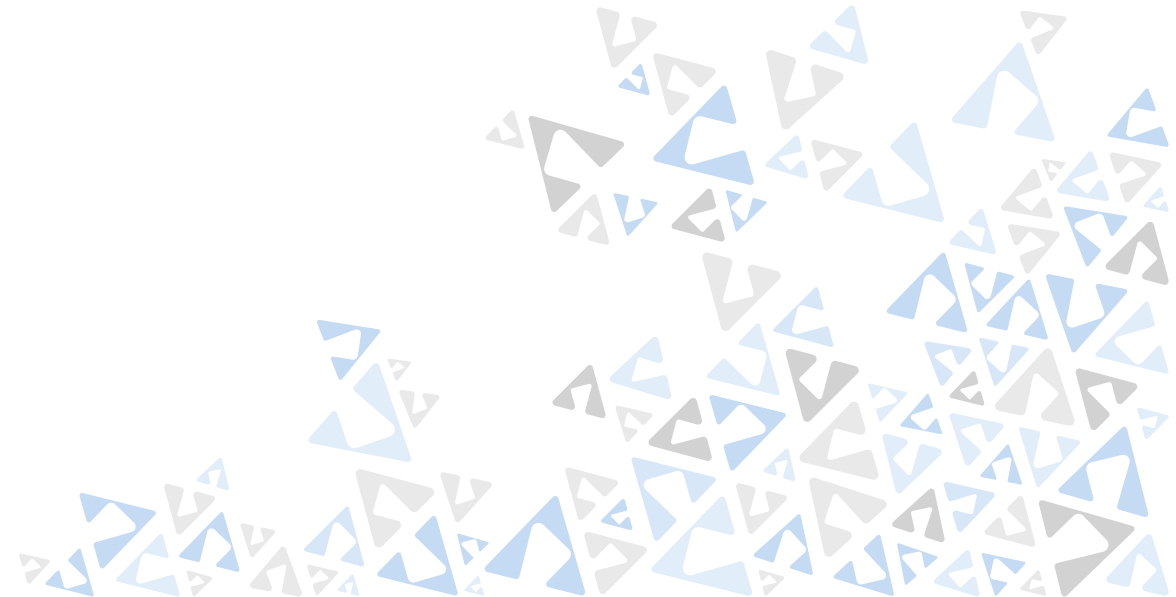
# Application Insights Profiler

- Úpravy našeho kódu
- Hypotéza okolo System.Text.Json (extrémně podobná issue, avšak nesouvisí)
- Hypotézy okolo tiered compilation
- RH využívá znalosti XY v Microsoftu, který se věnuje těmto „internals“
- *Vypada to, že ten deadlock se může stát jenom, když je zapnutý “Profiler ReJIT”. “Profiler ReJIT” si zapíná ApplicationInsights.*
- Vypnut Application Insights Profiler



# Blazor & Inputs & Exception handling

- [InputBase](#)
- <https://github.com/dotnet/aspnetcore/issues/44105>



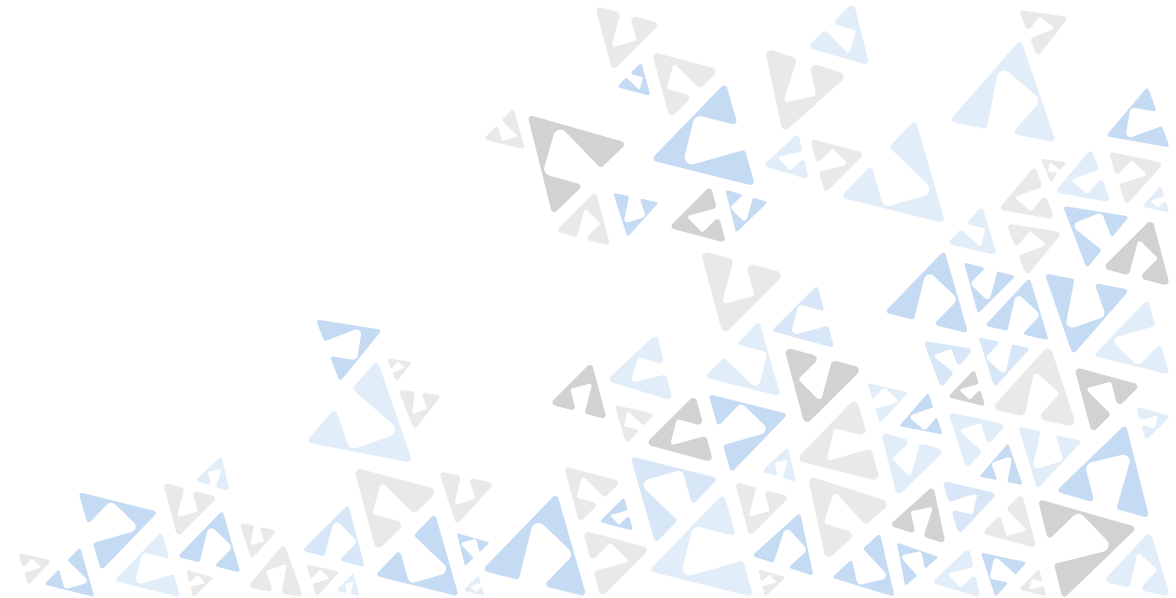
# Azure Blob Storage SDK a ASP.NET Core s EnableBuffering

- [FileBufferingReadStream](#)



# Central Package Management

- Zjednodušuje správu verzí jednotlivých nuget balíčků v solution.
- Ale...



# Recap

- Microsoft.Data.SqlClient async/await
- Můžete změřit dobu trvání analyzérů
- Výjimky v Blazor input komponentách
- Jak funguje EnableBuffering & FileBufferingReadStream
- Central Package Management

