Tareq Aziz

Executive (Software Engineering) at PETRONAS

WP. Kuala Lumpur, Federal Territory of Kuala Lumpur

PETRONAS Digital Sdn Bhd

Show more

Start a post

Video Photo Write article

Sort by: **Top**

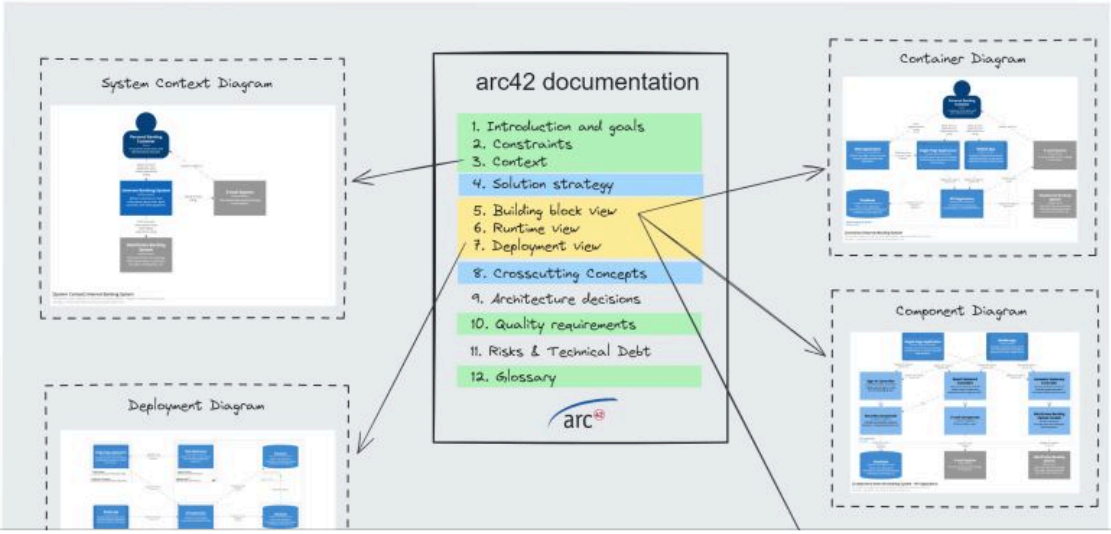
Milan Jovanović likes this

Dr Milan Milanović • 2nd
Chief Roadblock Remover and Learning Enabler | Software Developme...
View my newsletter
16h •

Follow

Documenting Software Architectures with arc42

Software architecture is the process of designing and organizing the overall ...more



Ashraf Alam and 79 other connections follow **Confluent**

...

Confluent
598,386 followers
Promoted

Follow

Explore a better way to build real-time streaming pipelines for databases, data warehouses, and mainframe integration.

Anton Martyniuk • Following
Microsoft MVP | Helping 30K+ Software Engineers Improve .NET S...
View my newsletter
38m •

During my career from 2014 I have worked on various projects

Today I am sharing the story about one project 👇 ...more

Ashfaqur Rahman • 1st
Principal Research Scientist at CSIRO
1h •

I am delighted to inform that we have a total of six papers accepted in #ICML2025 (A* conference with acceptance rate 26.93%) where StatML group (CSIRO's Data61) staff were co-authors. Among them two are spotlight papers (top 2.59% pape ...mor

24 3 comments

Suggested

Dmitry Hanziuk • 2nd
C# | .Net | Software Engineer
23h •

Follow

C# Streams — Small, Silent, Superpowered

Most developers use streams. Few really understand how much power and flexibility they offer.

Let's change that.

What is a Stream?

In .NET, a Stream is a powerful abstraction for reading and writing bytes — whether it's a file, memory, network, or even compression.

Think of it like a pipeline:

You don't care where the data comes from — only how to process it efficiently.

Practical Uses

Read large files without loading everything in memory

Chain streams: compress → encrypt → send

Write to memory as if it were a file

Final Thoughts

Prefer streams when working with large or dynamic data

Use MemoryStream to avoid I/O overhead

Chain streams (e.g., GZipStream) for even more control

Streams are a quiet hero of .NET — learn them once, and they'll serve you forever.

Streams in .NET — Simplified Examples to Show What Matters

❌ Inefficient: Full memory load

```
// ❌ Loads the whole file into memory before processing
var lines = File.ReadAllLines("uploaded-file.csv");

foreach (var line in lines)
{
    var model = Parse(line);
    dbContext.Records.Add(model);
}

dbContext.SaveChanges();
```

Caption: Loads everything first — risky with big files. Might crash due to memory overload.

✅ Streaming on the fly (simplified)

```
// ✅ Streams the file and processes it line-by-line
using var stream = uploadedFile.OpenReadStream();
using var reader = new StreamReader(stream);

string? line;
while ((line = reader.ReadLine()) ≠ null)
{
    var model = Parse(line);
    dbContext.Records.Add(model);
}

dbContext.SaveChanges();
```

Caption: Processes and saves without loading the full file. More scalable and memory-safe.

Dmitry Hanziuk

44

6 comments 3 reposts