



«Zusammen
MehrWert»

To reuse or not to reuse

05.09.2025, Andreas Rain



NOSER
ENGINEERING

Um was geht es?

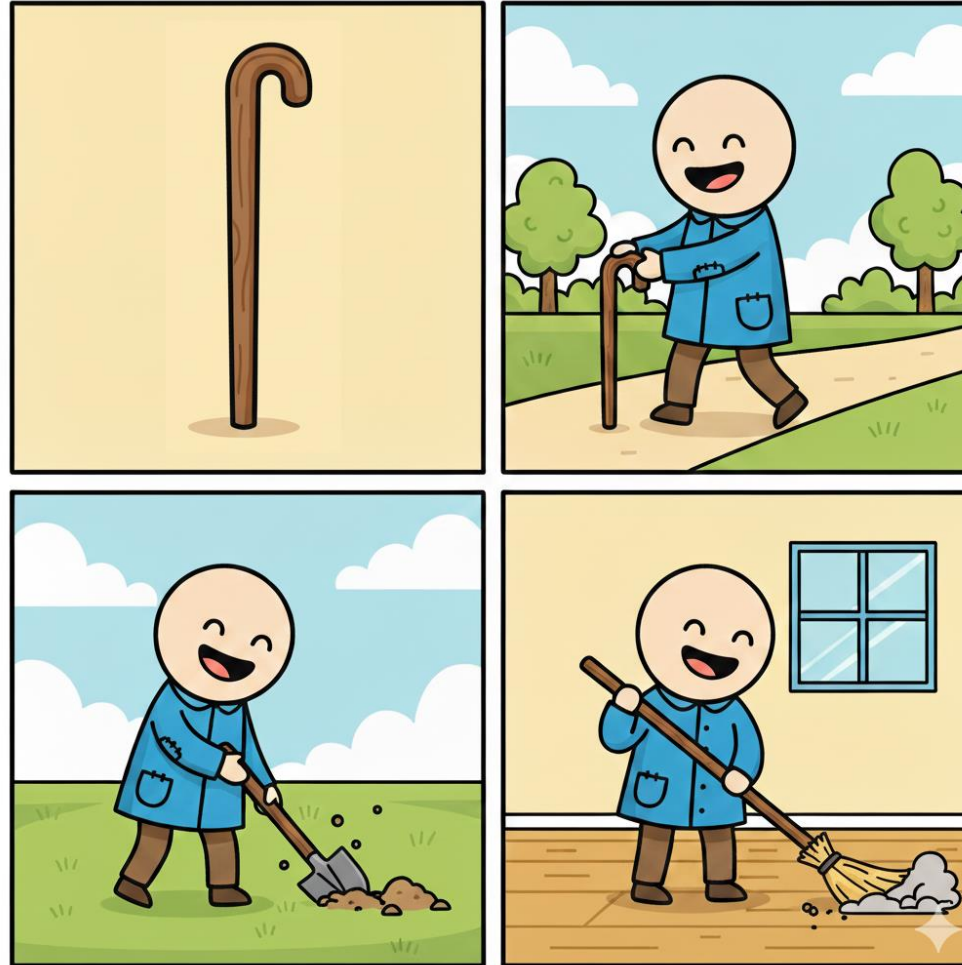


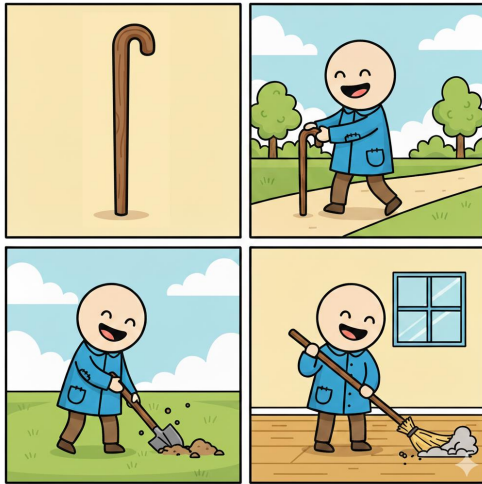
Image generated with the help of AI

Um was geht es?



Image generated with the help of AI

Wo liegt der Unterschied?

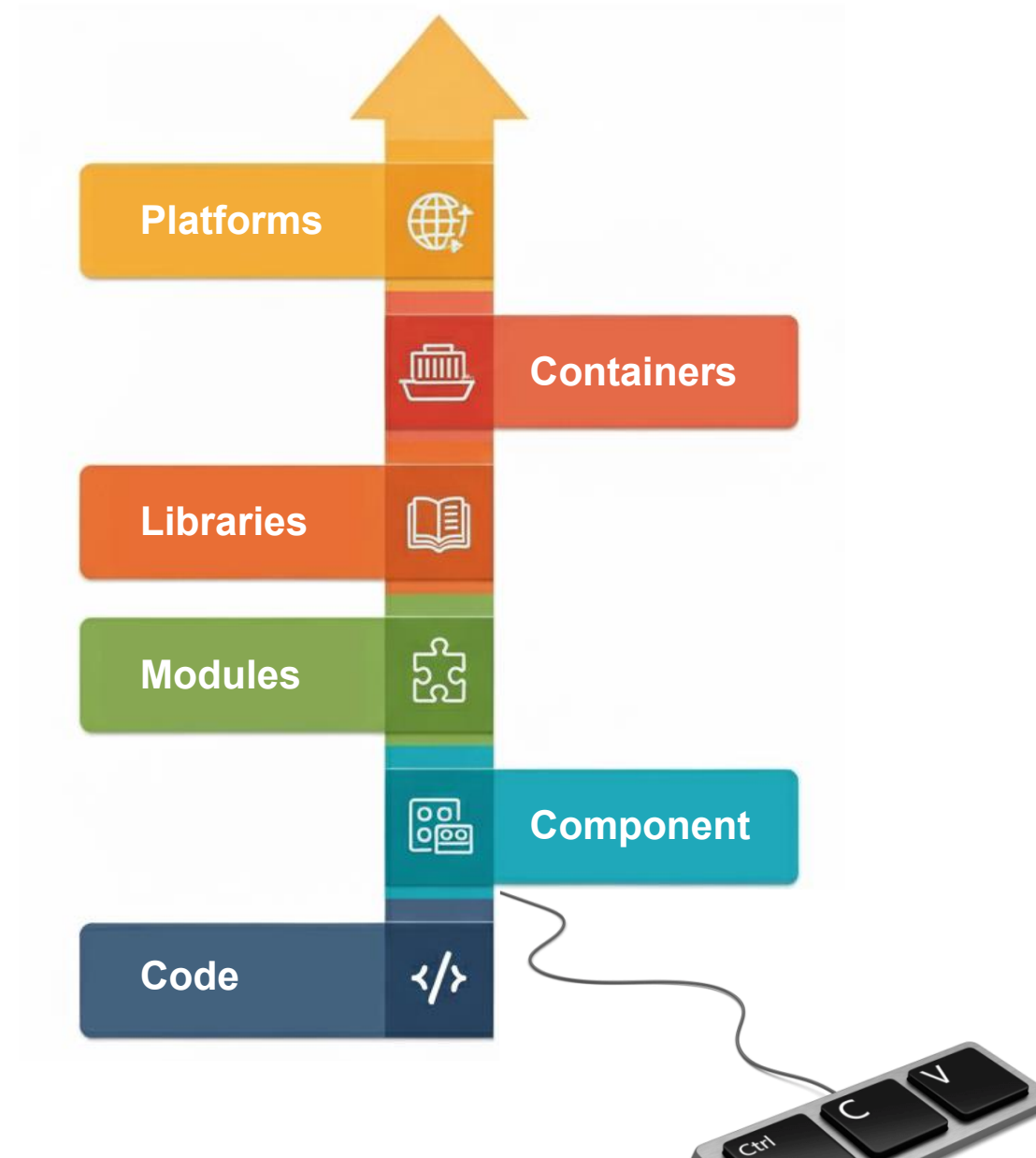


Verallgemeinerte Basis wird für unterschiedliche, jedoch geeignete Einsatzzwecke wiederverwendet



Code/Komponente wird für einen nicht dafür vorgesehenen Einsatzzweck wiederverwendet

Arten der Wiederverwendung



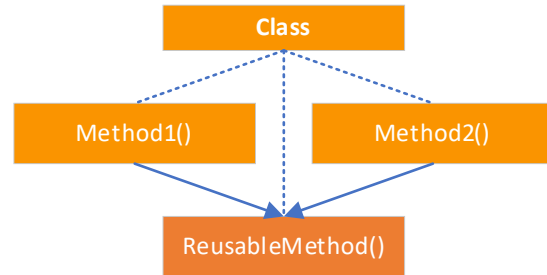
Wo findet Wiederverwendung statt?

Shared Code

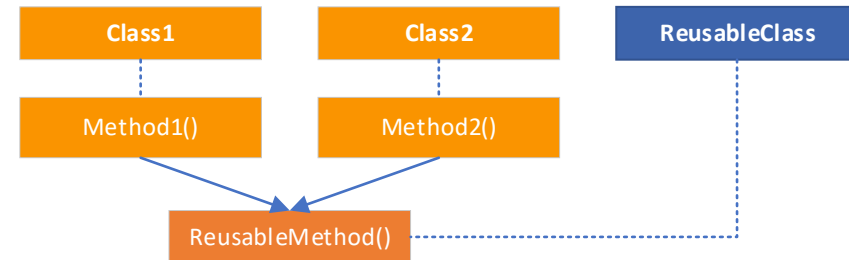
DRY
Infrastruktur
Business Logik
...

Compliance
Logging
ORM
Security
...

Innerhalb von Klassen



Klassen übergreifend von Klassen

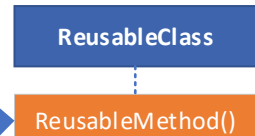


Durch andere Projekte, Module, Bibliotheken

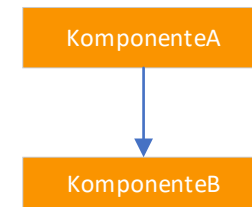
Projekt/Modul



Projekt, Modul,
Bibliothek



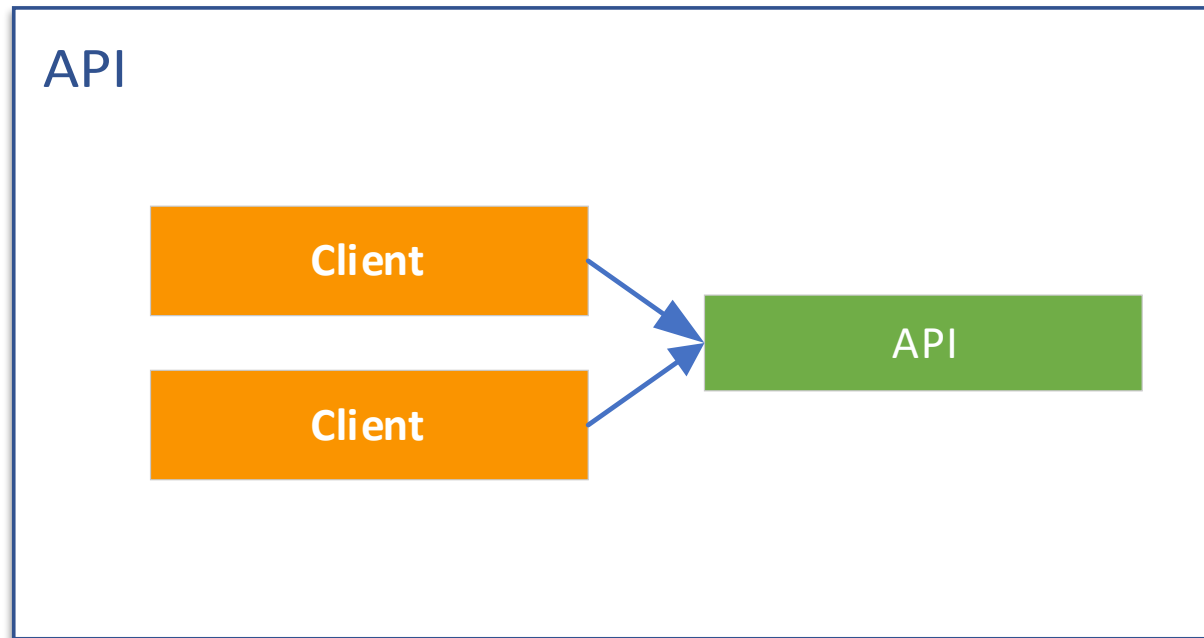
Oder allgemein



Wo findet Wiederverwendung statt?

API

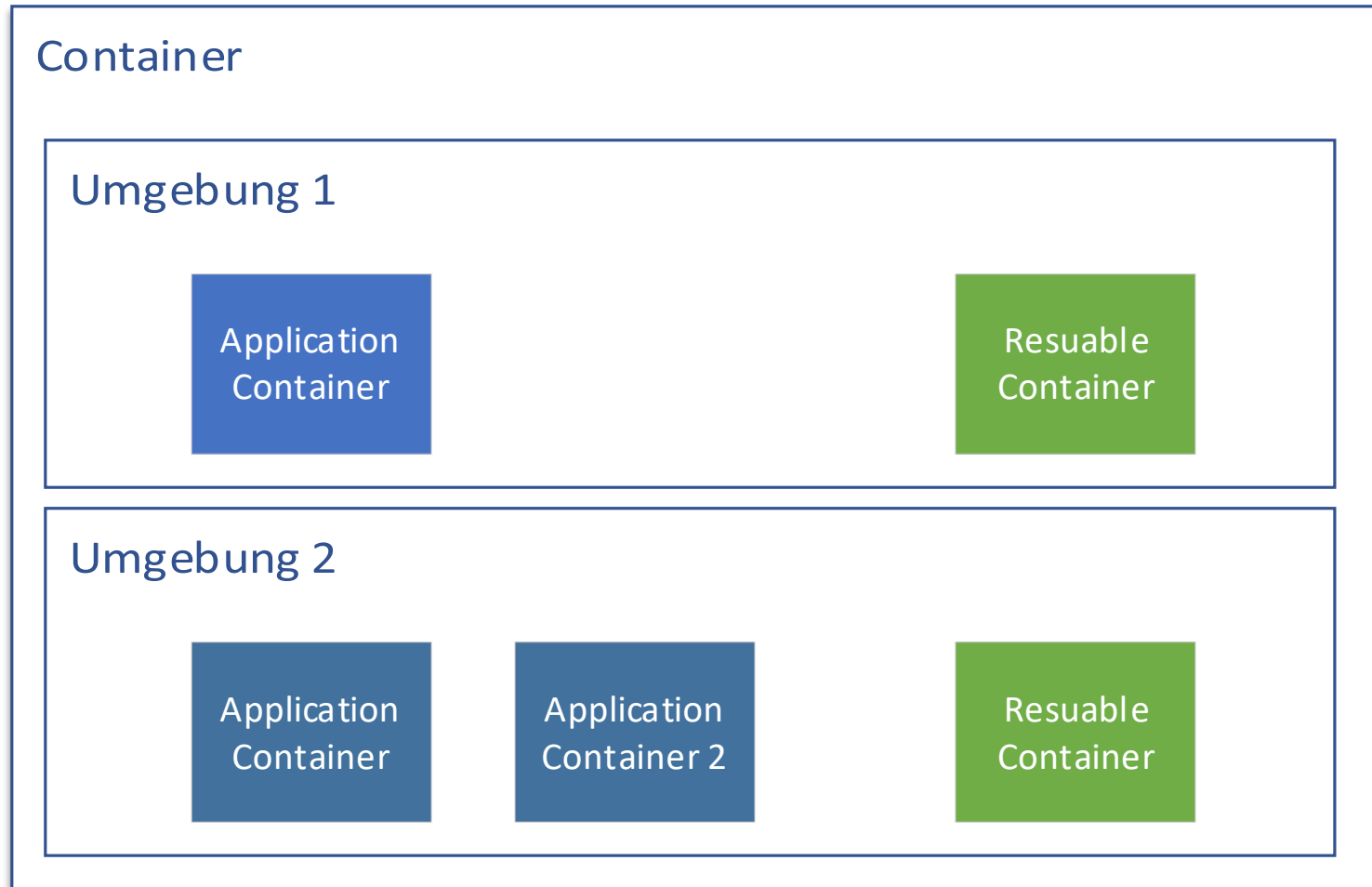
SOA
Microservices
Cloud
...



Wo findet Wiederverwendung statt?

Container

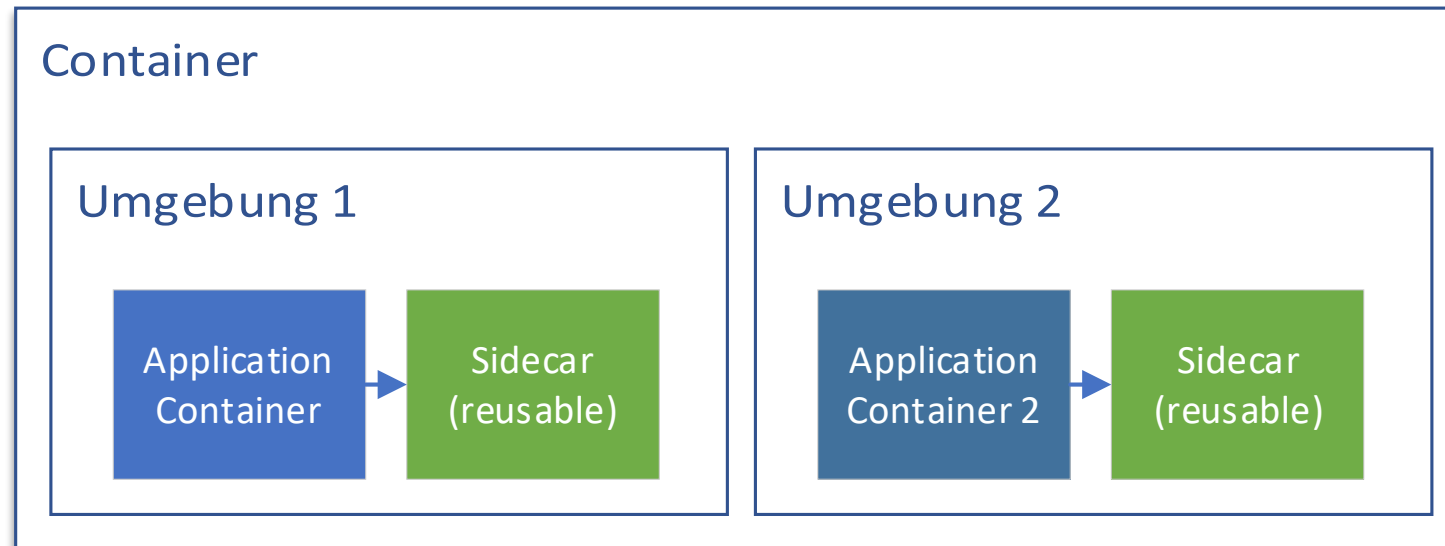
Services
Applikationen
Infrastruktur
...



Wo findet Wiederverwendung statt?

Sidecar
Container

Compliance
Observability
Security
Redundancy
Legacy
...



Unterschiede zu «Verwendung»



Image generated with the help of AI

Unterschiede zu «Verwendung»

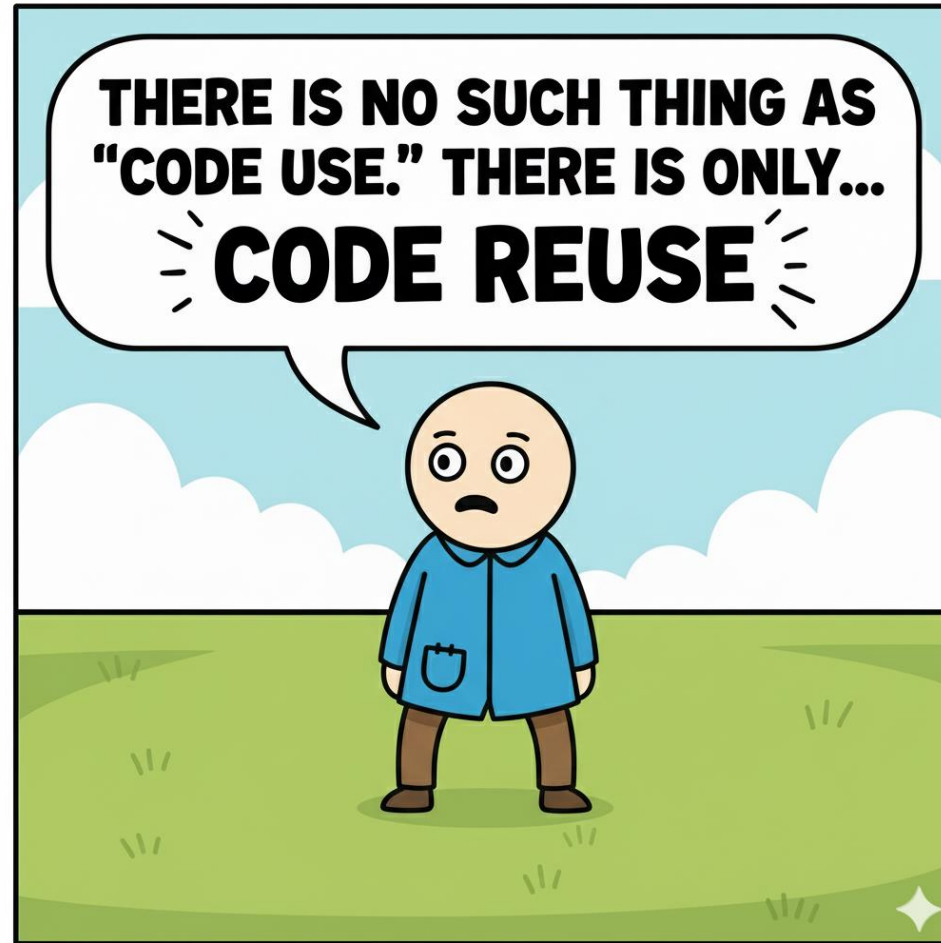


Image generated with the help of AI

Warum wiederverwenden?



Effizienz



Zuverlässigkeit



Einsparungen



Compliance



Komplexität

**«Wiederverwendbare
Komponenten
schreiben ist schwer»**

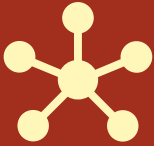
Warum NICHT wiederverwenden?



Anfälligkeit für Bugs



Annahmen über
Zukunft



Abhängigkeiten



Komplexität



Entwicklungsdauer



Fehleinschätzungen



Integrationsaufwand



Business aus den
Augen verlieren

Warum NICHT wiederverwenden?

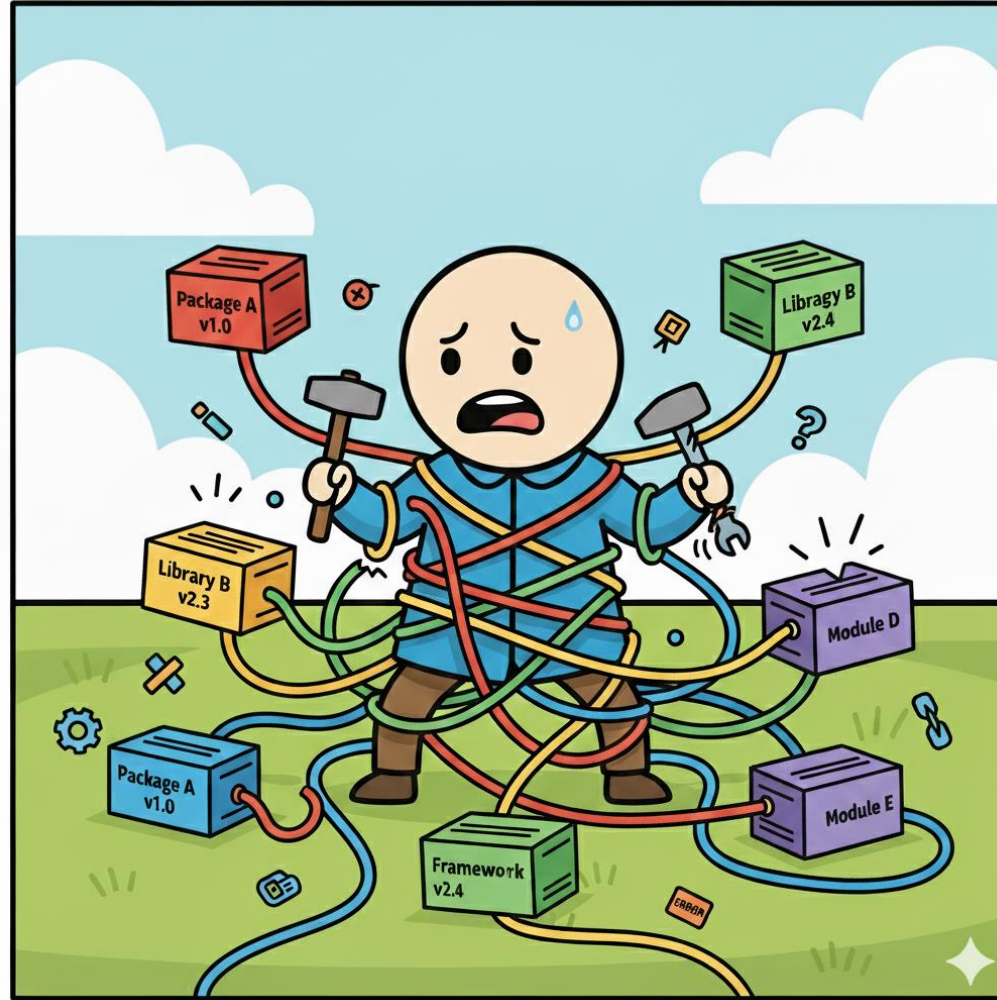
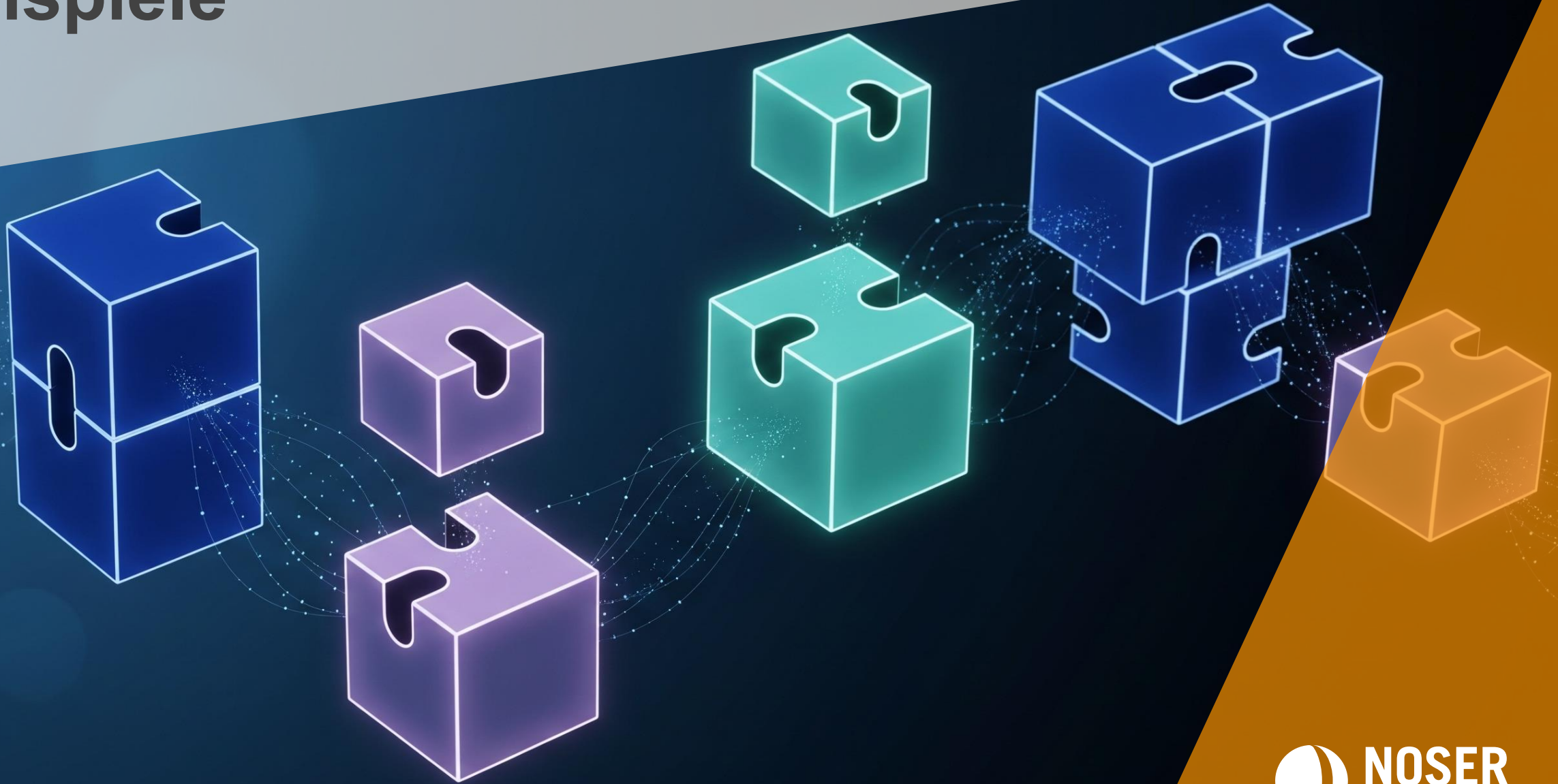


Image generated with the help of AI

Beispiele



Beispiele – Tabellen Komponente

Einfache Tabelle in Blazor die für unterschiedliche Zwecke wiederverwendet werden kann.

Vorteile:

- Einheitlicher Stil
- Clean Code
- Usw.



```
@if (people == null)
{
    <p><em>Loading data...</em></p>
}
else
{
    <Table Items="people">
        <HeaderTemplate>
            <th>ID</th>
            <th>Full Name</th>
            <th>Email Address</th>
            <th>Age</th>
        </HeaderTemplate>
        <RowTemplate>
            <td>@context.Id</td>
            <td>@context.Name</td>
            <td>@context.Email</td>
            <td>@context.Age</td>
        </RowTemplate>
    </Table>
}
```

Beispiele – Personentabelle

Idee: Entwicklung eines CRM. Personentabelle wird als eigene wiederverwendbare Komponente entwickelt, weil sie an verschiedenen Stellen im System dargestellt wird.

```
@if (persons == null)
{
    <p><em>Loading data...</em></p>
}
else
{
    <PersonTable Persons="persons" />
}
```

Beispiele – Personentabelle

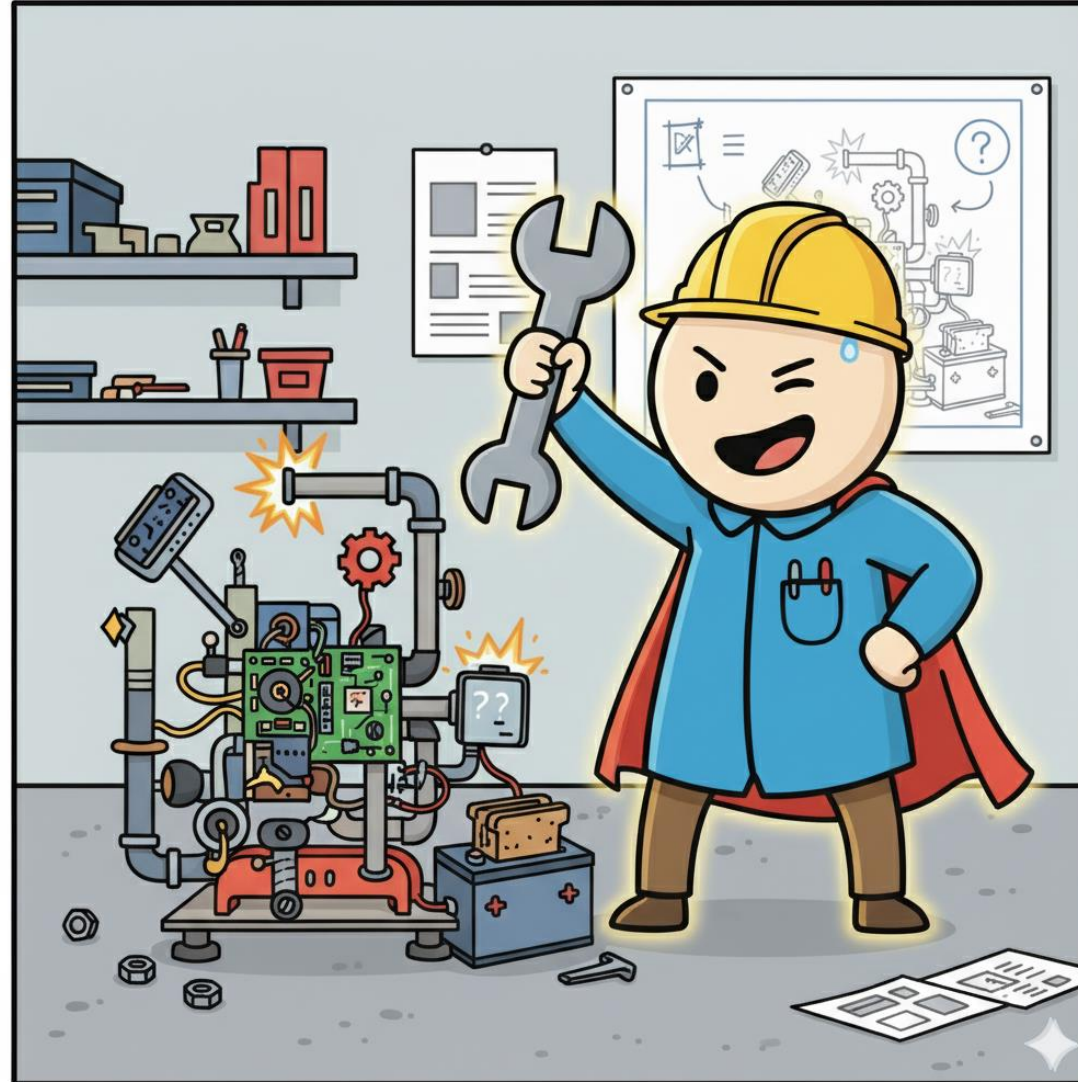
Weiter: Nun möchte man die Tabelle auch gerne an anderen Stellen im System anzeigen, allerdings nicht genau die gleichen Spalten.

```
@if (persons == null)
{
    <p><em>Loading data...</em></p>
}
else
{
    <PersonTable Persons="persons" VisibleColumns="visibleColumns" />
}
```

Beispiele – Personentabelle

Weiter: Nun soll sowohl die **Reihenfolge** der Spalten, als auch die **Darstellung des Profilbilds** etwas unterschiedlich sein, da die Personen an unterschiedlichen Stellen dargestellt werden.

Auch das ist kein Problem, immerhin kann die ursprüngliche Komponente erweitert werden.



Beispiele – Personentabelle

Weiter: Nun soll sowohl die **Reihenfolge** der Spalten, als auch die **Darstellung des Profilbilds** etwas unterschiedlich sein, da die Personen an unterschiedlichen Stellen dargestellt werden.

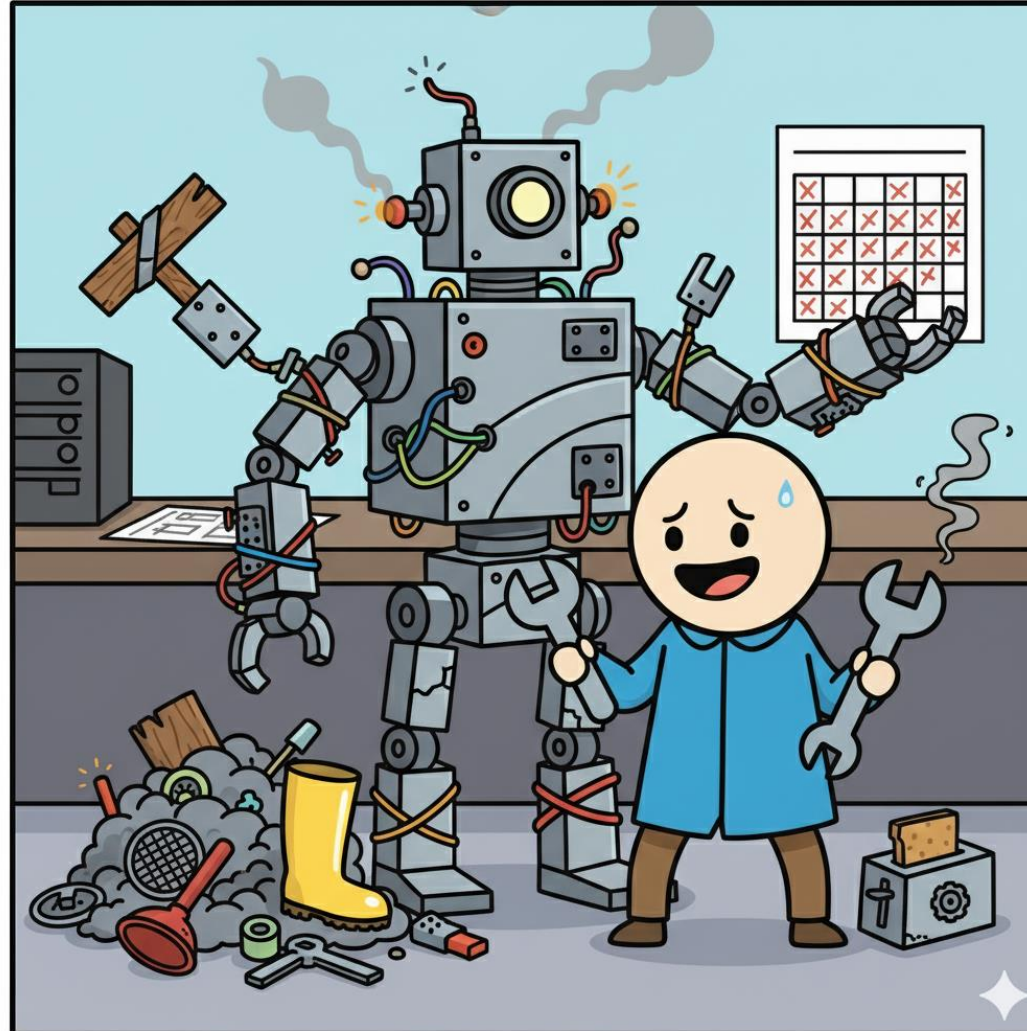
Auch das ist kein Problem, immerhin kann die ursprüngliche Komponente erweitert werden.

```
@if (persons == null)
{
    <p><em>Loading data...</em></p>
}
else
{
    <PersonTable Persons="persons" VisibleColumns="visibleColumns" ColumnsOrder="columnsOrder">
        <ProfileImageTemplate ...>
            ...
        </ProfileImageTemplate>
    </PersonTable>
}
```

Beispiele – Personentabelle

Problem: Leider hat der Kunde festgestellt, dass an einer anderen Stelle im System das Profilbild nicht mehr korrekt dargestellt wird.

Wir müssen also einen Hotfix machen.



Beispiele – Die Kehrseite der Medaille

```
// --- KOMPONENTE 1: Umsatz-Statistik-Karte ---
// Hat einen Titel, einen Wert und eine Beschreibung
function UmsatzKarte({ umsatz, zeitraum }) {
  const cardStyle = {
    border: '1px solid #e0e0e0',
    borderRadius: '8px',
    padding: '20px',
    margin: '10px',
    width: '250px',
    boxShadow: '0 2px 5px rgba(0,0,0,0.05)',
    backgroundColor: '#fff',
  };

  const titleStyle = {
    fontSize: '1.2em',
    fontWeight: 'bold',
    marginBottom: '10px',
    color: '#333',
  };

  const valueStyle = {
    fontSize: '2.5em',
    fontWeight: 'bold',
    color: '#007bff', // Blauer Akzent
    marginBottom: '5px',
  };

  const descStyle = {
    fontSize: '0.9em',
    color: '#777',
  };

  return (
    <div style={cardStyle}>
      <h3 style={titleStyle}>Gesamtumsatz</h3>
      <p style={valueStyle}>€{umsatz.toLocaleString('de-DE')}</p>
      <p style={descStyle}>Im letzten {zeitraum}</p>
    </div>
  );
}
```

```
// --- KOMPONENTE 2: Benutzer-Statistik-Karte ---
// Hat ebenfalls einen Titel, einen Wert und eine Beschreibung, aber
// mit minimalen Änderungen
function BenutzerKarte({ anzahlBenutzer, aktivitaet }) {
  const cardStyle = {
    border: '1px solid #e0e0e0',
    borderRadius: '8px',
    padding: '20px',
    margin: '10px',
    width: '250px',
    boxShadow: '0 2px 5px rgba(0,0,0,0.05)',
    backgroundColor: '#fff',
  };

  const titleStyle = {
    fontSize: '1.2em',
    fontWeight: 'bold',
    marginBottom: '10px',
    color: '#333',
  };

  const valueStyle = {
    fontSize: '2.5em',
    fontWeight: 'bold',
    color: '#28a745', // Grüner Akzent
    marginBottom: '5px',
  };

  const descStyle = {
    fontSize: '0.9em',
    color: '#777',
  };

  return (
    <div style={cardStyle}>
      <h3 style={titleStyle}>Aktive Benutzer</h3>
      <p style={valueStyle}>{anzahlBenutzer.toLocaleString('de-DE')}</p>
      <p style={descStyle}>{aktivitaet}</p>
    </div>
  );
}
```

```
// --- KOMPONENTE 3: Support-Tickets-Statistik-Karte ---
// Wieder das gleiche Muster, nur andere Daten und Farben
function SupportKarte({ offeneTickets, status }) {
  const cardStyle = {
    border: '1px solid #e0e0e0',
    borderRadius: '8px',
    padding: '20px',
    margin: '10px',
    width: '250px',
    boxShadow: '0 2px 5px rgba(0,0,0,0.05)',
    backgroundColor: '#fff',
  };

  const titleStyle = {
    fontSize: '1.2em',
    fontWeight: 'bold',
    marginBottom: '10px',
    color: '#333',
  };

  const valueStyle = {
    fontSize: '2.5em',
    fontWeight: 'bold',
    color: '#dc3545', // Roter Akzent
    marginBottom: '5px',
  };

  const descStyle = {
    fontSize: '0.9em',
    color: '#777',
  };

  return (
    <div style={cardStyle}>
      <h3 style={titleStyle}>Offene Tickets</h3>
      <p style={valueStyle}>{offeneTickets}</p>
      <p style={descStyle}>{status}</p>
    </div>
  );
}
```

Beispiele – Unix Pipe

- **Entkopplung:** grep weiß nichts über cat oder wc. Es kennt nur Input und Output. Dadurch kann grep mit jedem beliebigen Programm kombiniert werden, das Text ausgibt.
- **Flexibilität:** Dieselben Werkzeuge können für völlig andere Aufgaben neu kombiniert werden. Möchten Sie die 10 häufigsten IP-Adressen in einem Webserver-Log finden?



```
cat access.log | cut -d' ' -f1 | sort | uniq -c | sort -nr | head -n 10
```

Beispiele – Formatierung von Text / Ausgaben

```
var formattingPipeline = new Pipeline<string>();

// Bausteine zusammensetzen
formattingPipeline
    .AddStep(new ToUpperCaseStep())
    .AddStep(new ConsoleLogStep("ToUpperCase")) //
Wiederverwendbarer Logging-Baustein
    .AddStep(new ReplaceWhitespaceStep())
    .AddStep(new ConsoleLogStep("ReplaceWhitespace")) //
Derselbe Baustein, an anderer Stelle
    .AddStep(new AppendTimestampStep());
```

Beispiele – Domänen Objekte



```
public class Customer
{
    // Properties are public in C# to be
    public long Id { get; set; }
    public string Name { get; set; }
    public string Address { get; set; }
    public decimal CreditLimit { get; set; }
    public bool Active { get; set; }

    // Constructor
    public Customer(long id, string name,
    {
        Id = id;
        Name = name;
        Address = address;
        CreditLimit = creditLimit;
        Active = active;
    }

    public void HasOpenInvoices()
    {
        ...
    }
}
```

```
public class CustomerRepository
{
    ...

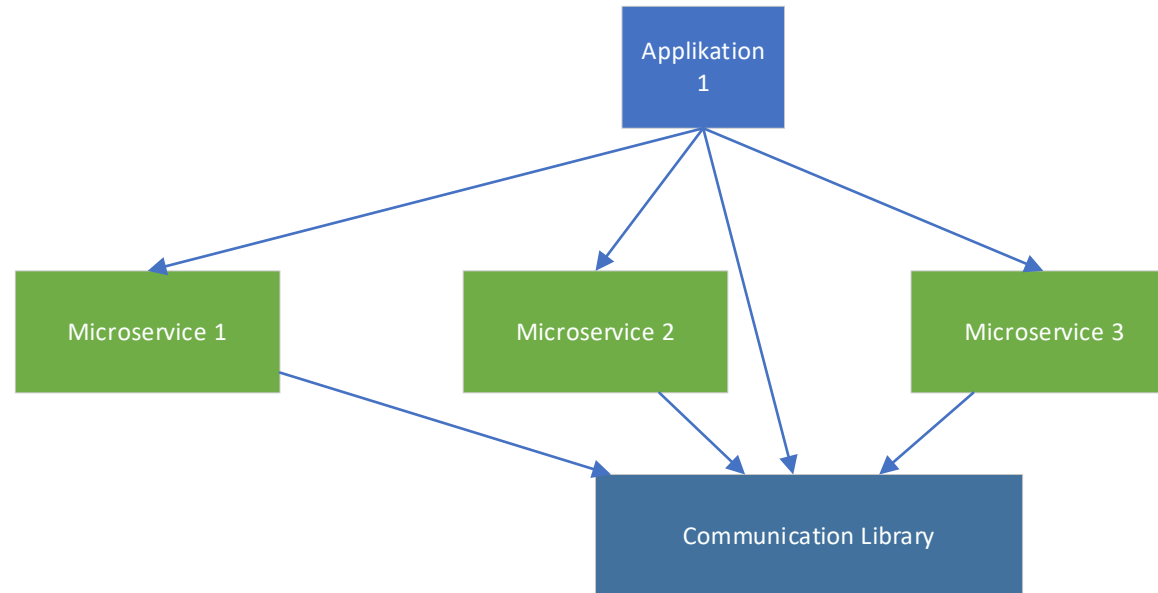
    public Customer? FindById(long id)
    {
        _customers.TryGetValue(id, out var customer);
        return customer;
    }

    ...

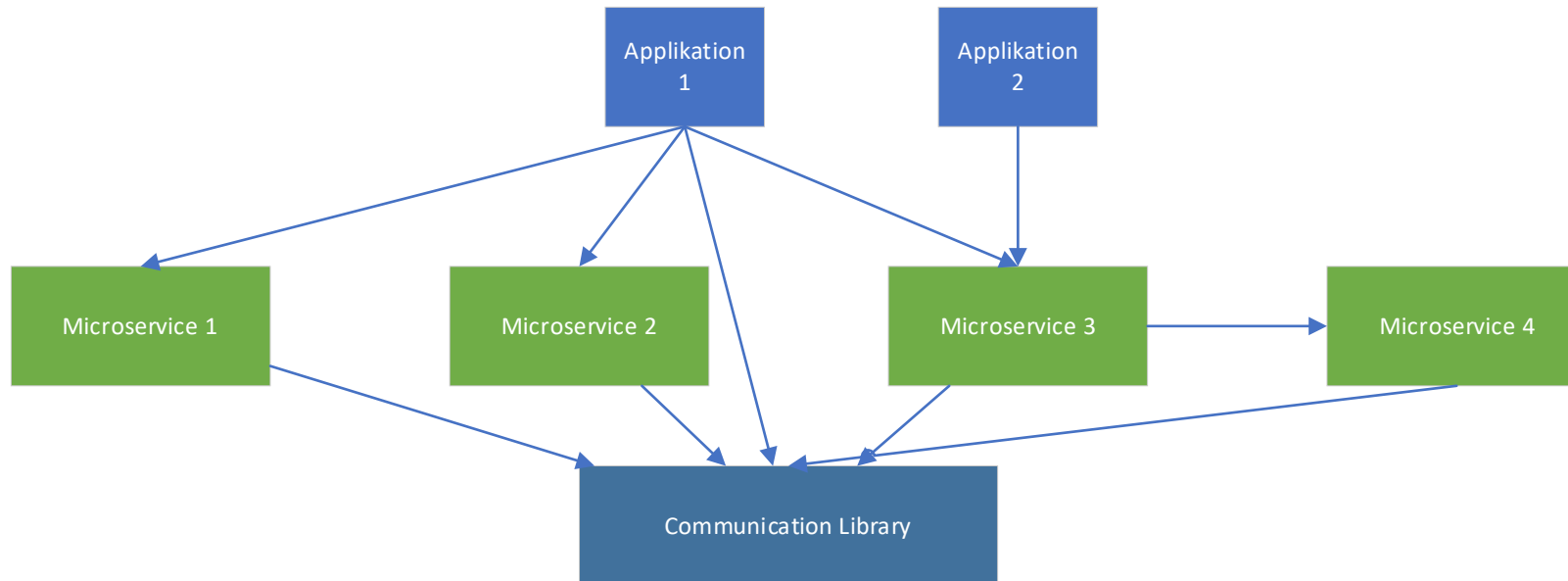
    app.MapGet("/api/customers/{id}", (long id, CustomerRepository customerRepository) => {
        // Fetch the full, internal domain object from the repository
        var customer = customerRepository.FindById(id);

        // Return the domain object directly to the client
        if (customer != null)
        {
            return Results.Ok(customer);
        }
        return Results.NotFound();
    });
}
```

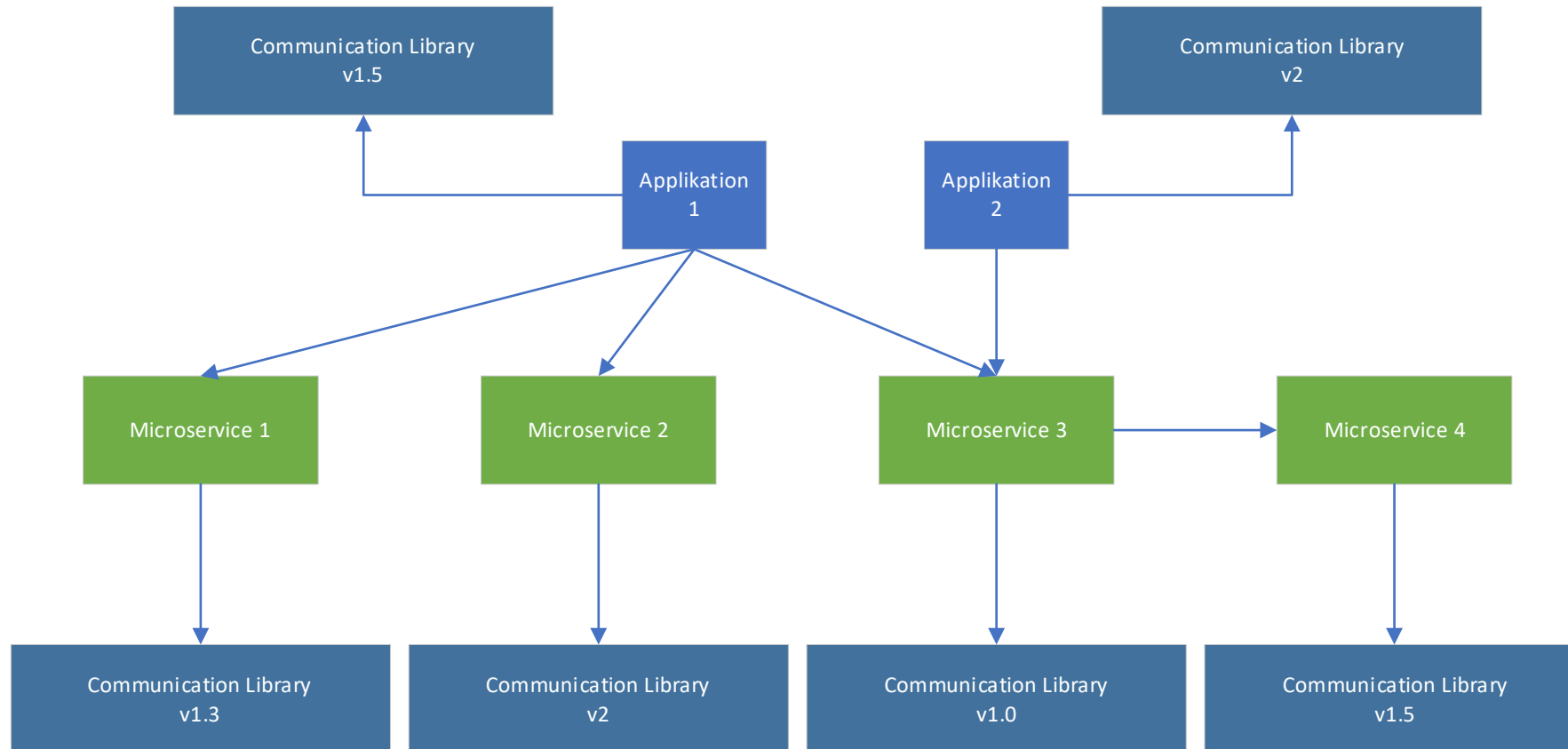
Beispiele – Accidental Hell



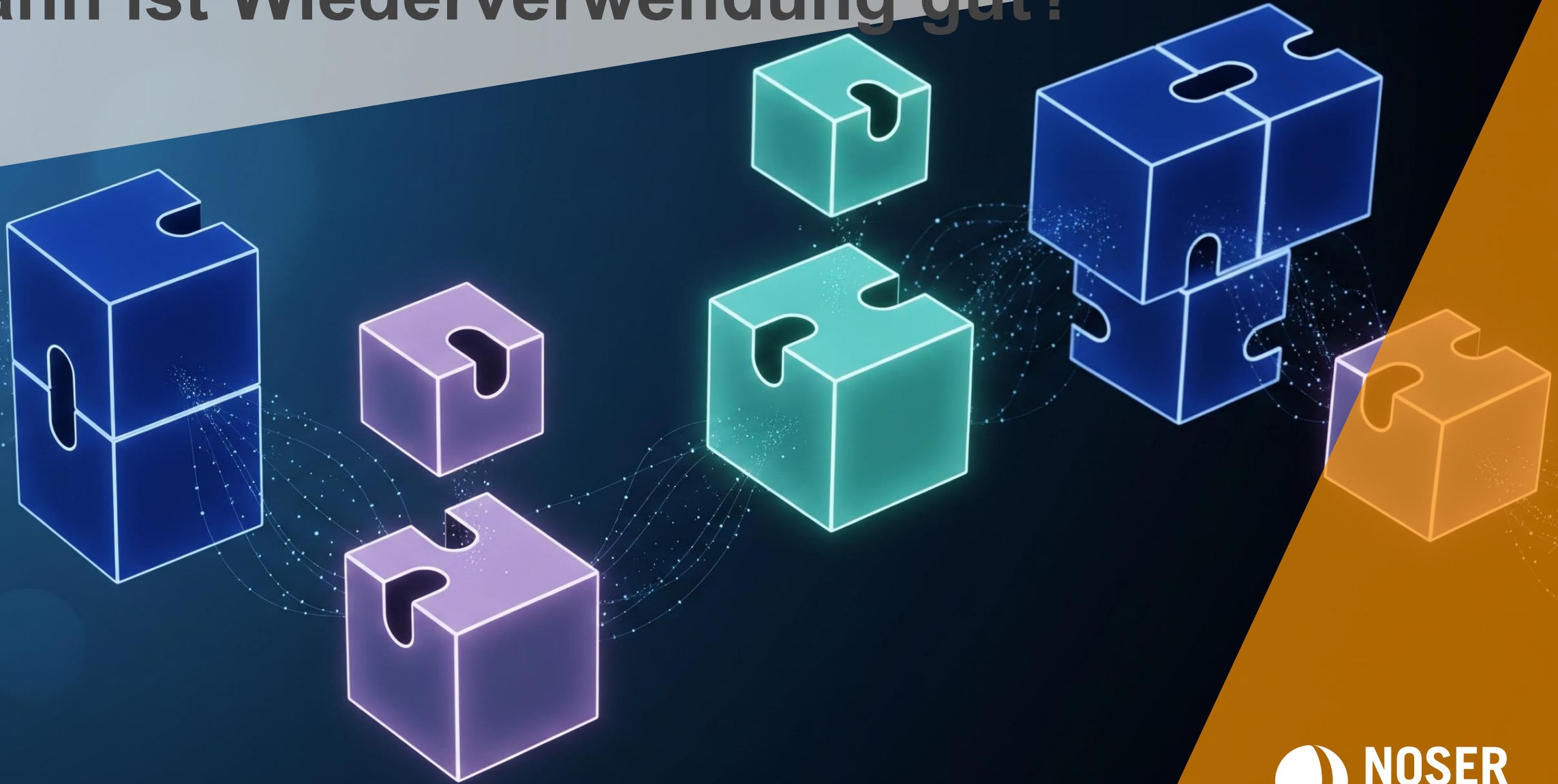
Beispiele – Accidental Hell



Beispiele – Accidental Hell



Wann ist Wiederverwendung gut?



Wann ist Wiederverwendung gut?

Wir haben folgende Ziele:

- Anwendungsfälle verstehen und kennen
- Abhängigkeiten abwägen
- Komponente ist vollständig testbar
- Keine fachlichen Logiken abbilden

Wann ist Wiederverwendung gut?

- Klare Definition der Schnittstelle und Funktionsweise
- Technische Sachverhalte
 - Dumb vs. Smart Component
- Geringe Änderungshäufigkeit
- Intrinsische Motivatoren
- Skalierung der Lösung ist wirtschaftlich vorteilhaft

Danke!

Vielen Dank für eure Aufmerksamkeit! Hier sind ein paar Links:

<https://udidahan.com/2009/06/07/the-fallacy-of-reuse/>

<https://blog.christianposta.com/design/the-cost-of-code-reuse-abuse/>

<https://dev.to/pluralsight/reusable-code-the-good-the-bad-and-the-ugly-3do6>

<https://blog.ndepend.com/code-reuse-not-good-goal/>