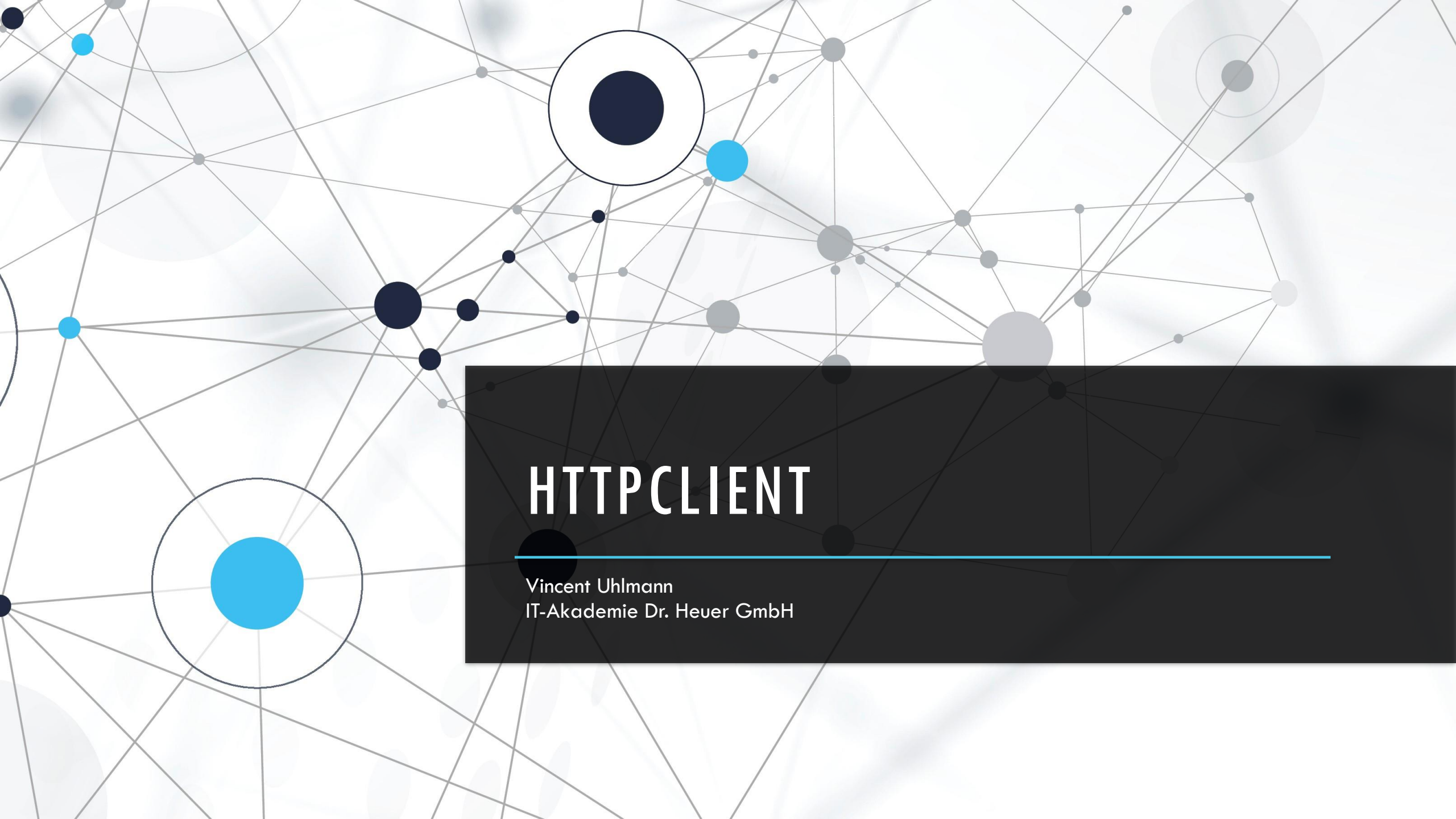

HTTPCLIENT

Vincent Uhlmann
IT-Akademie Dr. Heuer GmbH

WAS IST DER C# HTTPCLIENT?

- **HttpClient** ist eine Klasse in .NET, die verwendet wird, um HTTP-Anfragen zu senden und HTTP-Antworten von einer Ressource zu empfangen, die durch eine URI identifiziert wird
- Entwickelt für die Kommunikation mit Web-APIs und das Herunterladen von Daten aus dem Internet
- Unterstützt asynchrone Programmierung

- **Wofür nutzt man den HttpClient?**
- Abrufen und Senden von Daten an Web-APIs
- Ermöglicht die Kommunikation zwischen verteilten Anwendungen über HTTP
- Integration externer Dienste und APIs in eigene Anwendungen

ERSTELLEN DES HTTPCLIENT

- **Einzelne Instanz**
- Eine einzelne Instanz von HttpClient wird wiederverwendet
- Vermeidet Probleme mit Port- und Socket-Auslastung

```
private static readonly HttpClient client = new HttpClient();
```

ERSTELLEN DES HTTPCLIENT

- **HttpClientFactory**
- Empfohlen wenn viele HttpClient-Instanzen benötigt werden
- Verwaltet Lebenszyklen und Konfiguration von HttpClient-Instanzen effizient

```
private readonly HttpClient _client;

public MyService(IHttpClientFactory httpClientFactory)
{
    _client = httpClientFactory.CreateClient();
}
```

GET REQUEST MIT HTTPCLIENT

- Abrufen von Daten von einer angegebenen URL
- Die Antwort wird als String gelesen und ausgegeben

```
private static readonly HttpClient client = new HttpClient();

public static async Task Main()
{
    var getResponse = await client.GetAsync("https://api.example.com/data");
    string getResponseContent = await getResponse.Content.ReadAsStringAsync();
    Console.WriteLine(getResponseContent);
}
```

POST REQUEST MIT HTTPCLIENT

- Senden von Daten an eine URL
- Die Daten werden als JSON-String im Body der Anfrage gesendet. Die Antwort wird als String gelesen und ausgegeben

```
private static readonly HttpClient client = new HttpClient();

public static async Task Main()
{
    var postData = new StringContent("{\"name\":\"value\"}", Encoding.UTF8, "application/json");
    var postResponse = await client.PostAsync("https://api.example.com/data", postData);
    string postResponseContent = await postResponse.Content.ReadAsStringAsync();
    Console.WriteLine(postResponseContent);
}
```

PUT REQUEST MIT HTTPCLIENT

- Aktualisieren von Daten an einer angegebenen URL
- Die Daten werden als JSON-String im Body der Anfrage gesendet. Die Antwort wird als String gelesen und ausgegeben

```
private static readonly HttpClient client = new HttpClient();

public static async Task Main()
{
    var putData = new StringContent("{\"name\":\"newValue\"}", Encoding.UTF8, "application/json");
    var putResponse = await client.PutAsync("https://api.example.com/data/1", putData);
    string putResponseContent = await putResponse.Content.ReadAsStringAsync();
    Console.WriteLine(putResponseContent);
}
```

DELETE REQUEST MIT HTTPCLIENT

- Aktualisieren von Daten an einer angegebenen URL
- Die Daten werden als JSON-String im Body der Anfrage gesendet. Die Antwort wird als String gelesen und ausgegeben

```
private static readonly HttpClient client = new HttpClient();

public static async Task Main()
{
    var deleteResponse = await client.DeleteAsync("https://api.example.com/data/1");
    string deleteResponseContent = await deleteResponse.Content.ReadAsStringAsync();
    Console.WriteLine(deleteResponseContent);
}
```

VERSCHIEDENE CONTENT-TYPEN

- **StringContent**
- Senden von z.B. JSON-Daten

```
var content = new StringContent("{\"name\":\"value\"}", Encoding.UTF8, "application/json");
```

- **ByteArrayContent**
- Senden von binären Daten

```
var byteArray = Encoding.UTF8.GetBytes("Hello, world!");
var content = new ByteArrayContent(byteArray);
```

- ...

SETZEN VON HEADERN

- Setzt Standard-Header für alle Anfragen, die mit diesem HttpClient gesendet werden

```
private readonly HttpClient _client = new HttpClient();

private async Task DoSomething()
{
    _client.DefaultRequestHeaders.Add("Content-Type", "application/json");
}
```

SETZEN VON HEADERN

- Setzt Header für die aktuelle Anfragen, die mit diesem HttpClient gesendet wird

```
private readonly HttpClient _client = new HttpClient();

private async Task DoSomething()
{
    HttpRequestMessage httpRequestMessage= new HttpRequestMessage(HttpMethod.Get, "URL");
    httpRequestMessage.Headers.Add("content-type", "application/json");

    HttpResponseMessage httpResponseMessage = await _client.SendAsync(httpRequestMessage);
    string json = await httpRequestMessage.Content.ReadAsStringAsync();
}
```

LESEN VON HEADERN

- Überprüfen und lesen eines spezifischen Headers aus der Http-Antwort

```
private readonly HttpClient _client = new HttpClient();

private async Task DoSomething()
{
    HttpRequestMessage httpRequestMessage= new HttpRequestMessage(HttpMethod.Get, "URL");
    httpRequestMessage.Headers.Add("content-type", "application/json");

    HttpResponseMessage httpResponseMessage = await _client.SendAsync(httpRequestMessage);
    httpRequestMessage.Headers.TryGetValues("content-type", out var result);
}
```