

# TEAMVORSTELLUNG

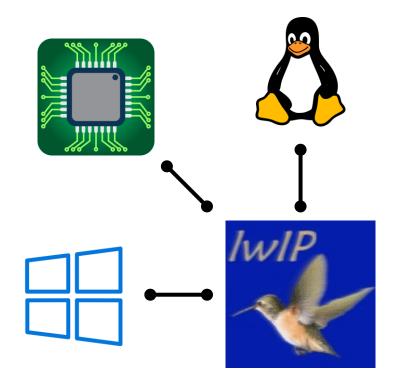
| Name              | Matrikelnummer | Projektrolle         | eMail-Adresse                    |
|-------------------|----------------|----------------------|----------------------------------|
| Lucas Kaczynski   | 9411266        | Techn. Redakteur     | inf20147@lehre.dhbw-stuttgart.de |
| Benjamin Esenwein | 5655208        | Produktmanager       | inf20074@lehre.dhbw-stuttgart.de |
| Maximilian Meier  | 6020212        | Leitender Entwickler | inf20084@lehre.dhbw-stuttgart.de |
| Yannis Plaschko   | 6645234        | Testmanager          | inf20093@lehre.dhbw-stuttgart.de |
| Laura Reeken      | 3811697        | Projektleiterin      | inf20051@lehre.dhbw-stuttgart.de |
| Isabel Schwalm    | 6038831        | Systemarchitektin    | inf20085@lehre.dhbw-stuttgart.de |

#### INHALT

- Was ist lwIP?
- Master Use Case
- Systemarchitektur & Produktübersicht
- Funktionale und Nichtfunktionale Anforderungen
- Module
- Entwicklungsprozess
- Vorgehensweise beim Testen
- Rückblick Prototyp & Live Demo
- Fazit / Ausblick

#### Was ist lwIP?

- kleine unabhängige Implementierung des TCP/IP-Protokolls
- Zielsetzung
  - Reduzierung des Ressourcenverbrauchs
  - Gleichzeitige Bereitstellung eines vollwertigen TCP/IP-Protokolls
- → Geeignet für Embedded Systems
- → Lauffähig auf mehreren Betriebssystemen



#### MASTER USECASE



Ziel: • Websocket Support für lwIP implementieren

• lwIP-Status Seite & REST-API implementieren



Aufgabe: Mängel des vorhandenen Patch #9525 beheben



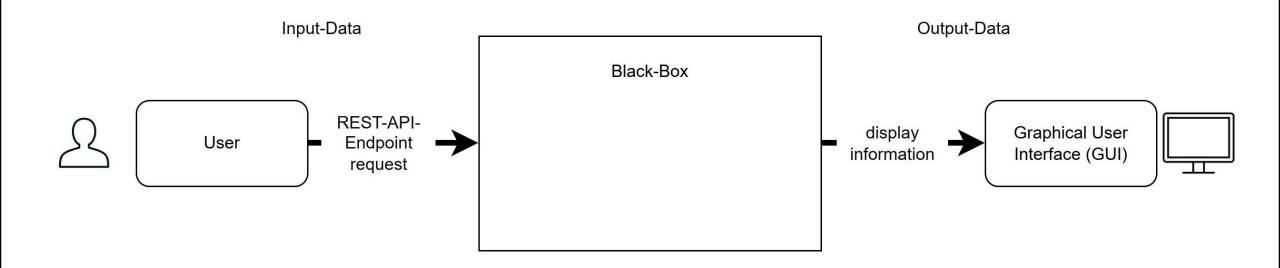
Use Case:

Beteiligte Nutzer: Entwickler von Embedded System

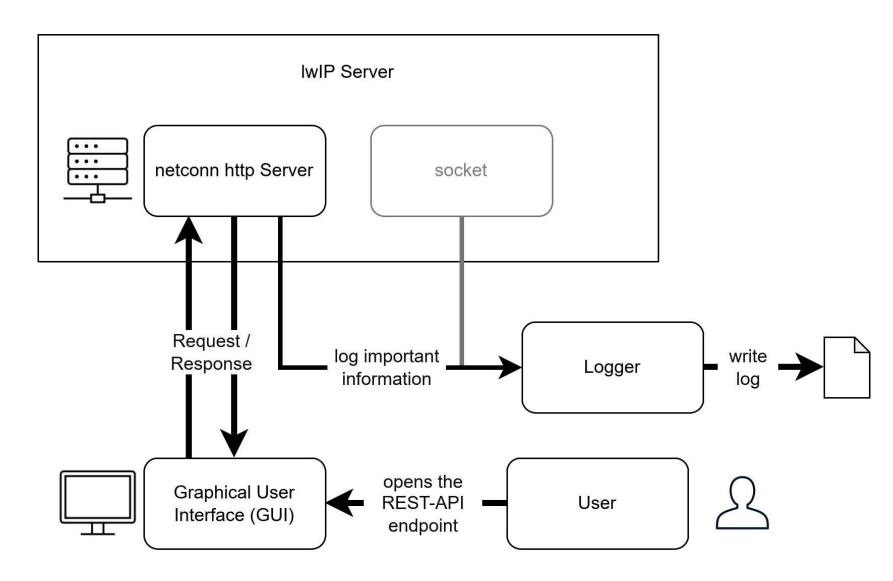
Zielsetzung: • Support für TCP/IP und Websockets

lwIP ist auf CPU lauffähig

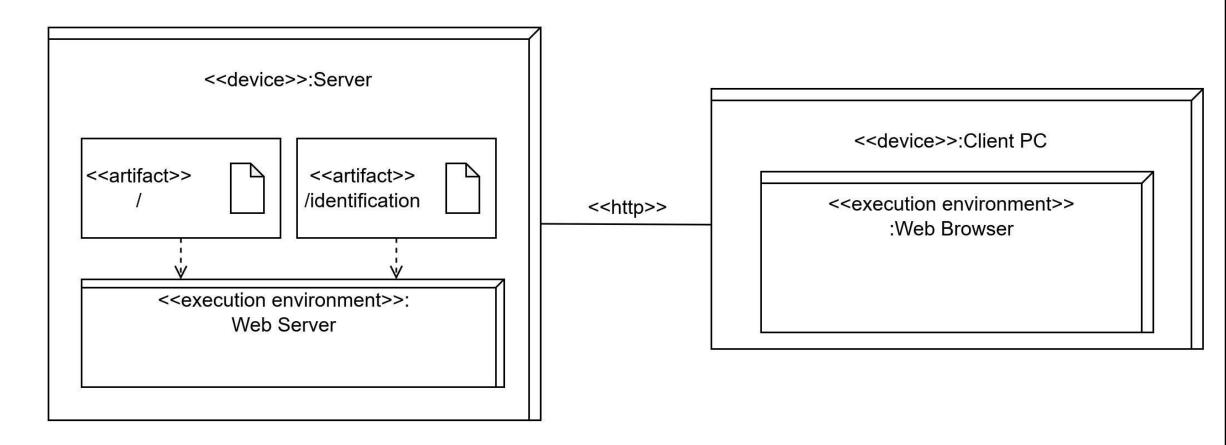
#### SYSTEMARCHITEKTUR



### Produktübersicht



### Produktübersicht



# FUNKTIONALE ANFORDERUNGEN

/F10/ Extra Funktion für base64 encoding

/F20/ Implementation der WebSocket API

/F30/ Schreiben eines GUI-basierten Test-Client

/F40/ http Webserver für Testzwecke

### NICHTFUNKTIONALE ANFORDERUNGEN

| /NF10/ Zusätzlich geschriebener Code sollte bei einem Minimum gehalten werden |  |
|---|--|
| /NF20/ Effizienter Code   |  |
| /NF30/ Intuitive GUI für Test-Client  |  |
| /NF40/ Bestehen des Approval Process /NF41/ Informationsweitergabe            |  |
|   |  |
| /NF50/ Executable Program   |  |

#### Module

```
/MOD.001/ Graphical User Interface

/MOD.002/ HTTP-Server

/MOD.003/ WebSocket Server
```

## Entwicklungsprozess

- Visual Studio 2022
- WIN SDK 10.0
- Analyse & Einarbeitung Patch#9525
- Debugging
- Alternative WebSocket Server









#### Bildquellen:

https://logos.fandom.com/wiki/Microsoft\_Visual\_Studio#2021.E2.80.93present

https://cmake.org/

https://www.winpcap.org/docs/docs\_412/html/

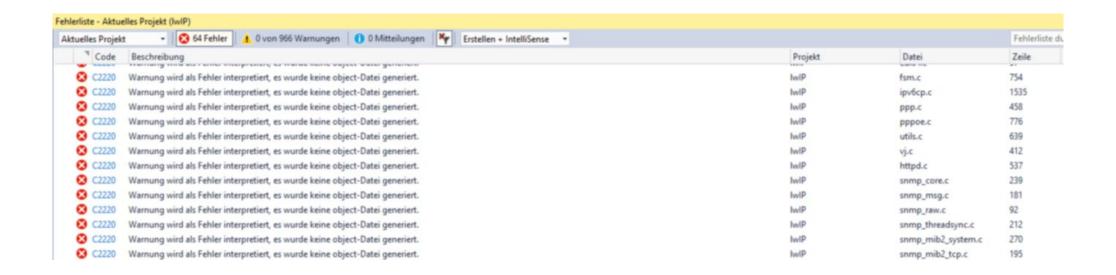
Npcap: Windows Packet Capture Library & Driver

## VORGEHEN BEIM TESTEN

- LwIP ist sehr umfangreich und komplex
  - Kein White-Box-Testing nicht möglich
- Black-Box-Testing
- GUI als POC daher kaum Testdaten
- POO aus Sicht des Users
- Testplanung
  - Verifikation er Lauffähigkeit
  - Erzeugen und Zerstören von Verbindungen

### RÜCKBLICK AUF PROTOTYP

- Rückmeldung der lwIP-Community stand aus
- Code-Abhängigkeits-Fehler



# LIVE-DEMO

Anwendung öffnen

#### **FAZIT**

