

# Python Protokoll Projekte

**Abgabe** Donnerstag 8. Januar 2026, 23h00

**Auswahl** Zwei Projekte stehen zur Auswahl. Projekt Ping-Pong ist grundlegend (Level 1), Projekt Chat-Service ist anspruchsvoller (Level 2)

**Notengebung** Diese Note zählt 40% der Gesamtnote. Im Allgemeinen gilt je mehr Features implementiert wurden, desto mehr Punkte. Die Note wird in einem Gespräch/Interview/Verhör bestimmt. Die Dokumentation (wenig ist mehr, ein Bild sagt mehr als 1000 Worte) zählt maximal **+0.5** Punkte.

**Rahmenbedingungen** Bitte befolgt diese Anforderungen:

- Tragt euren Namen und git-url in die Datei `python-protokoll.xlsx` (ms-teams) ein.
- Angabe des git urls: `git clone <url>` funktioniert!
- Regelmässiges gitten (mindestens einmal `git push` nach jedem Lektionsblock),
- Klare, konzise Beschreibung, wie das Program zu nutzen ist, (`md` format im git)
- `python <your-code.py>` funktioniert

**Metabetrachtungen** Die Aufgabe ist gewollt und also auch bewusst offen gehalten. Das erhöht den Fun-Faktor und die Selbstverantwortung. Das Projekt ist anspruchsvoll, die Zeit und die Noten sind grosszügig ausgelegt.

**Ziele** - Selbstverantwortliches erarbeiten

- eines Protokolls
- einer Implementation des Protokolls in Python
- Erkennen und implementation von

- Prozessen (not OS/CPU related)
- Methoden
- Funktionen
- Strukturen
- Praktische Anwendung von git, IDE, terminal, GNU/Linux/Debian und KISS in Projektarbeit
- Selbstständiges Erlernen
  - von communication endpoints (sockets:)
  - und deren Funktionen und Methoden
- Soziale Kompetenzen, Lernkompetenzen:
  - Teamarbeit mit Aufgabenteilung
  - rechtzeitig Rückfragen
  - Arbeit mit Papier und Bleistift, IDE

## 1 Projekt Ping Pong (Level 1)

**Projektbeschreibung** Implementiere ein UDP/TCP basierendes Ping-Pong Protokoll in Python. Ein Ping wird gesendet und mit einem Pong beantwortet. Jede Erweiterung des Service (siehe unten für die Erweiterungen) gibt zusätzliche Punkte (siehe Notenschlüssel). Jeder Erweiterung in einer separaten Datei oder git branch.

**Notengebung** Siehe oben für grundlegende Notengebung

- Implementation des Basic-Ping-Pong Service : **4-5**
- Jedes weitere Feature: **+0.3**

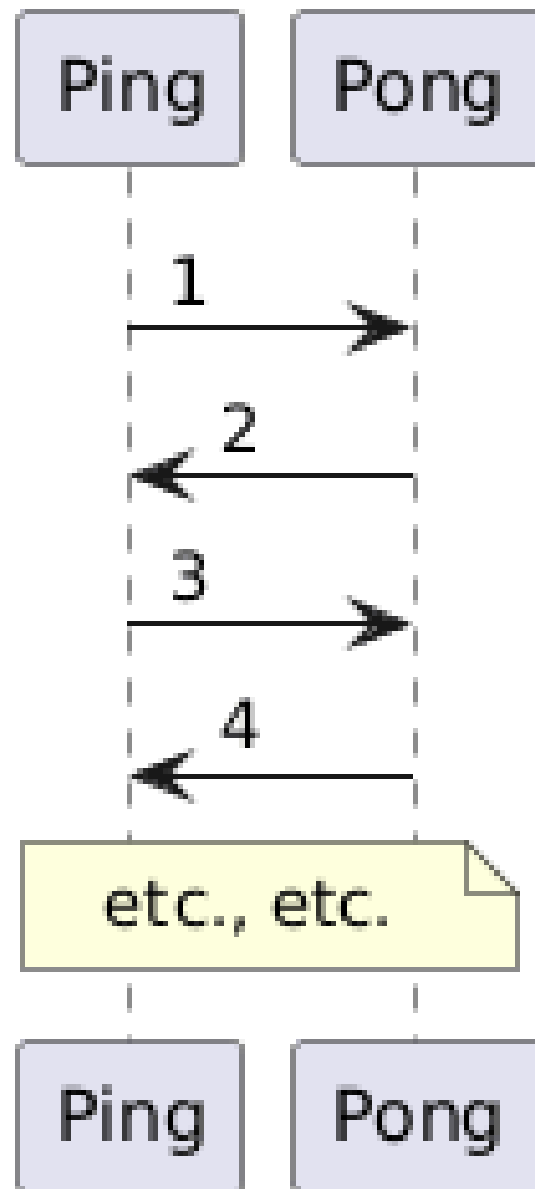
**Features** Siehe untenstehende Liste von Ping-pong Services/Architektur

### 1.1 Ping-Pong Services

#### 1.1.1 Basic Ping-Pong

Ping sendet eine Zahl  $n$  (=spin), und Pong antwortet mit  $n + 1$ .

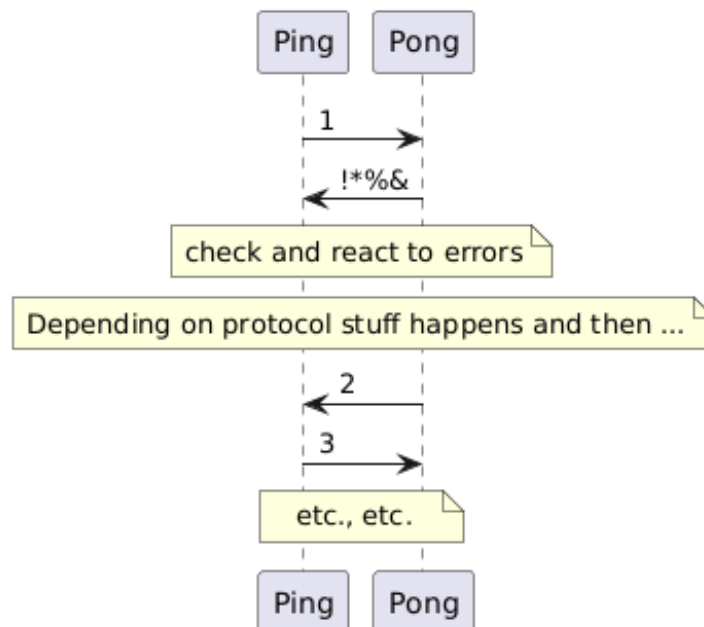
## Ping Pong Service



### 1.1.2 Ping-Pong mit UDP Fehlerbehandlung

Manchmal können Daten-Fehler entstehen. Ping-Pong mit Fehlerbehandlung kann damit umgehen.

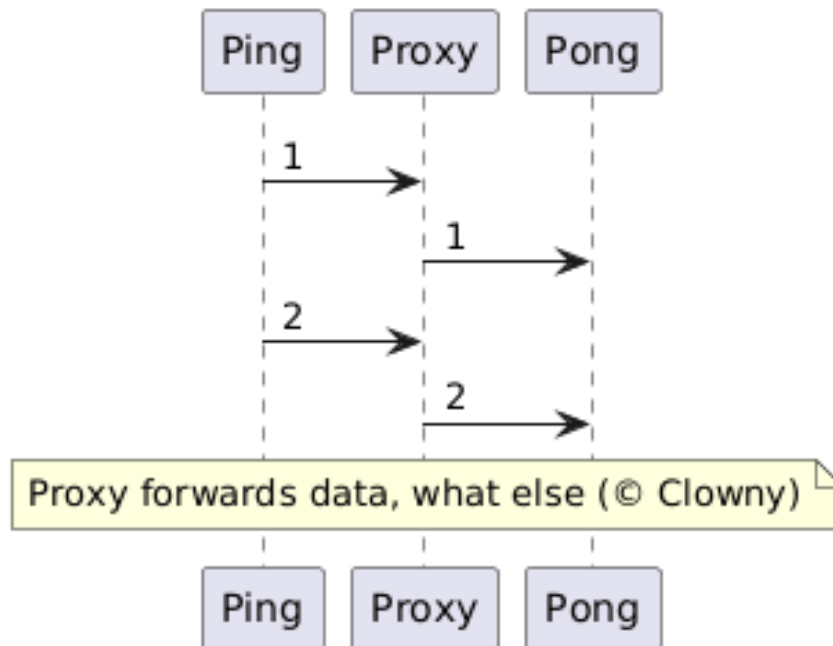
#### Ping Pong Service with error detection/correction



### 1.1.3 Ping-Pong mit einem Ping-Pong Proxy

Der proxy nimmt den ping-pong-pall und verlängert die Flugbahn ohne den Spin zu ändern.

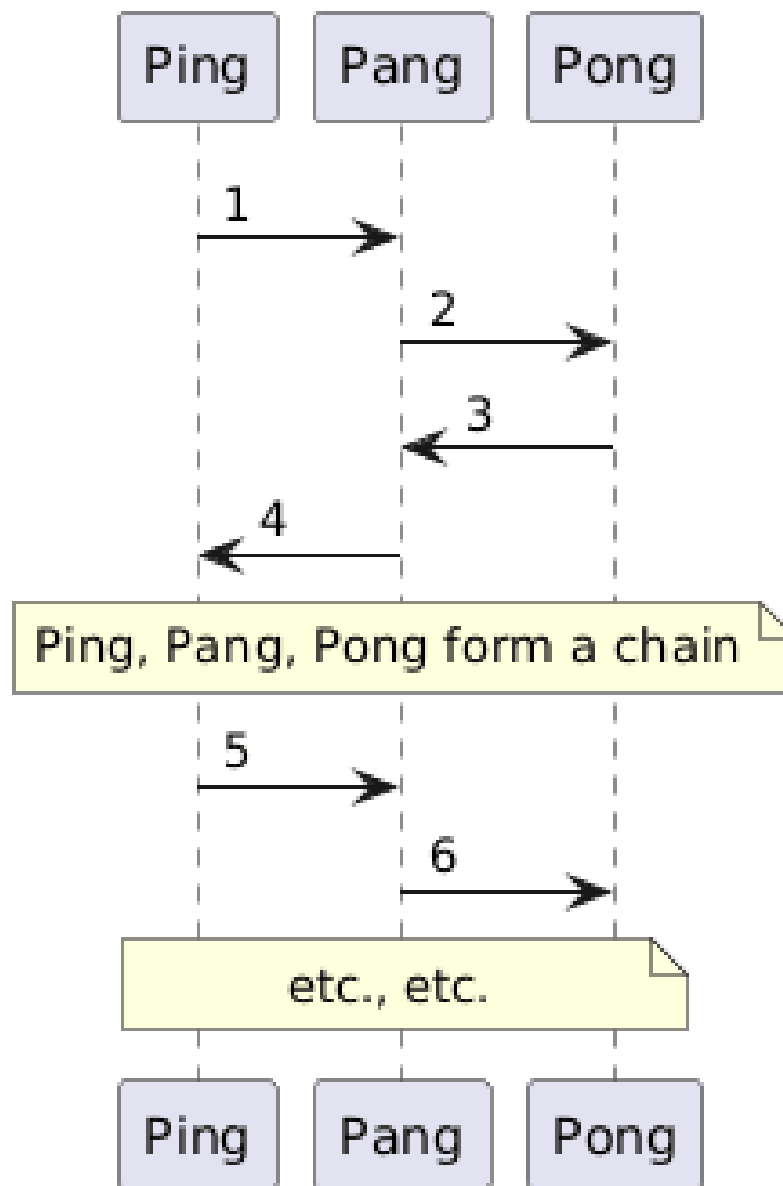
### Ping Pong Service with Proxy



#### 1.1.4 Kette von Ping-Pongs

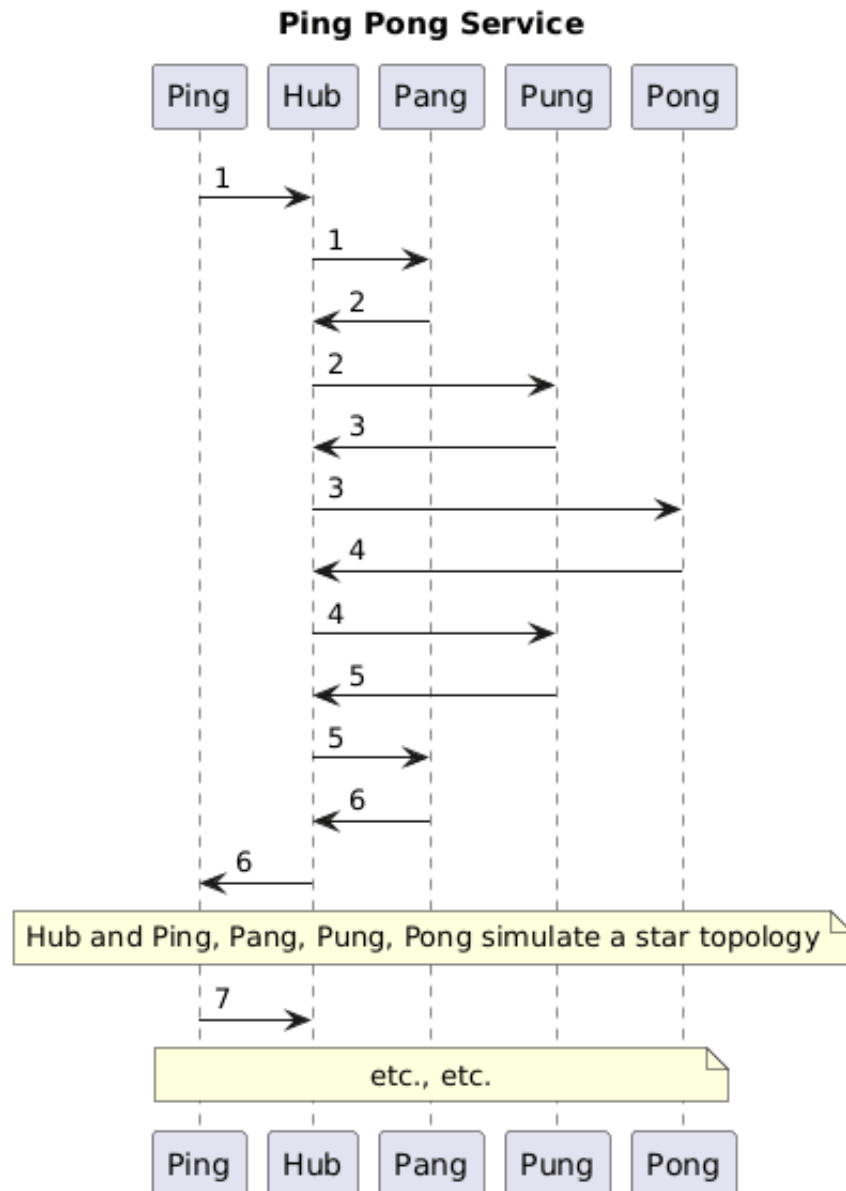
Generalisierung des Ping-Pong-Service: Mehrere PP service provider (PPSP) bilden eine Kette.

## Chained Ping Pong Service



### 1.1.5 Ping-Pong in einer Stern Topologie

Implementiert ein eigene statische Stern Topologie, der Hub mach das Stern-Routing.



### 1.1.6 Ping-Pong in einer vermaschten Topologie

Due to technical difficulties no image is provided.

Consider a Mesh-Topology (like the internet) and implement a meshed network of ping-pong services on top of it.

## 2 Projekt Chat Server (Level 2)

**Projektbeschreibung** Implementiere ein UDP/TCP basierendes Chat Protokoll in Python (KISS=TUI, not GUI). Jede Erweiterung des Service (siehe unten für die Erweiterungen) gibt zusätzliche Punkte (siehe Notenschlüssel). Jeder Erweiterung in einer separaten Datei oder git branch, so dass jede Erweiterung unabhängig getestet werden kann.

**Note/Punkte** Siehe oben für grundlegende Notengebung

- Implementation eines Chat Service mit einem Raum : **4-5**
- Implementation eines multi-threaded Chat Service mit mehreren Räumen: **5-5**
- Jedes weitere Feature: **+0.2**

**Features/Befehle** Einige Befehlsideen für den Chat service:

- server : print server information
- ls : list chat rooms
- cd : change chat room
- users : list users in room
- connect : connect to server
- secure connect : connect in a secure manner to server
- ...