

High-Level Design

- **CLI Interface** (`osi_explorer.py`)
 - Entry point using `argparse`.
 - Commands `simulate`, `pcap`, `compare`.
- **Core Models**
 - Header represents protocol headers with optional fields.
 - PDU protocol Data Unit abstraction containing headers + payload.
- **Visualization Layer**
 - ASCII shows encapsulation/decapsulation steps.
 - Comparison table for OSI vs TCP/IP
- **PCAP Generation**
 - Basic mode: Synthetic Ethernet/IP/UDP packet.
 - Advanced mode: Realistic Ethernet/IP/TCP/UDP headers via Scapy.
- **Artifacts**
 - Generated `.pcap` files stored locally.
 - Console output provides visualization logs.

Data Flow

1. **Input Payload** → CLI argument.
2. **Encapsulation/Decapsulation** → Construct or strip headers step by step.
3. **Visualization** → ASCII diagrams printed to console.
4. **PCAP Generation** → Scapy builds packet → written to `.pcap`.
5. **Analysis (optional)** → PCAPs can be opened in Wireshark.

Dependencies

- **Python 3.9+**
- **Scapy** (optional, required for PCAP generation)
- Standard library modules: argparse, dataclasses, textwrap, etc.

Deployment

- Python CLI.
- Can be containerized with Docker
- Strictly offline.