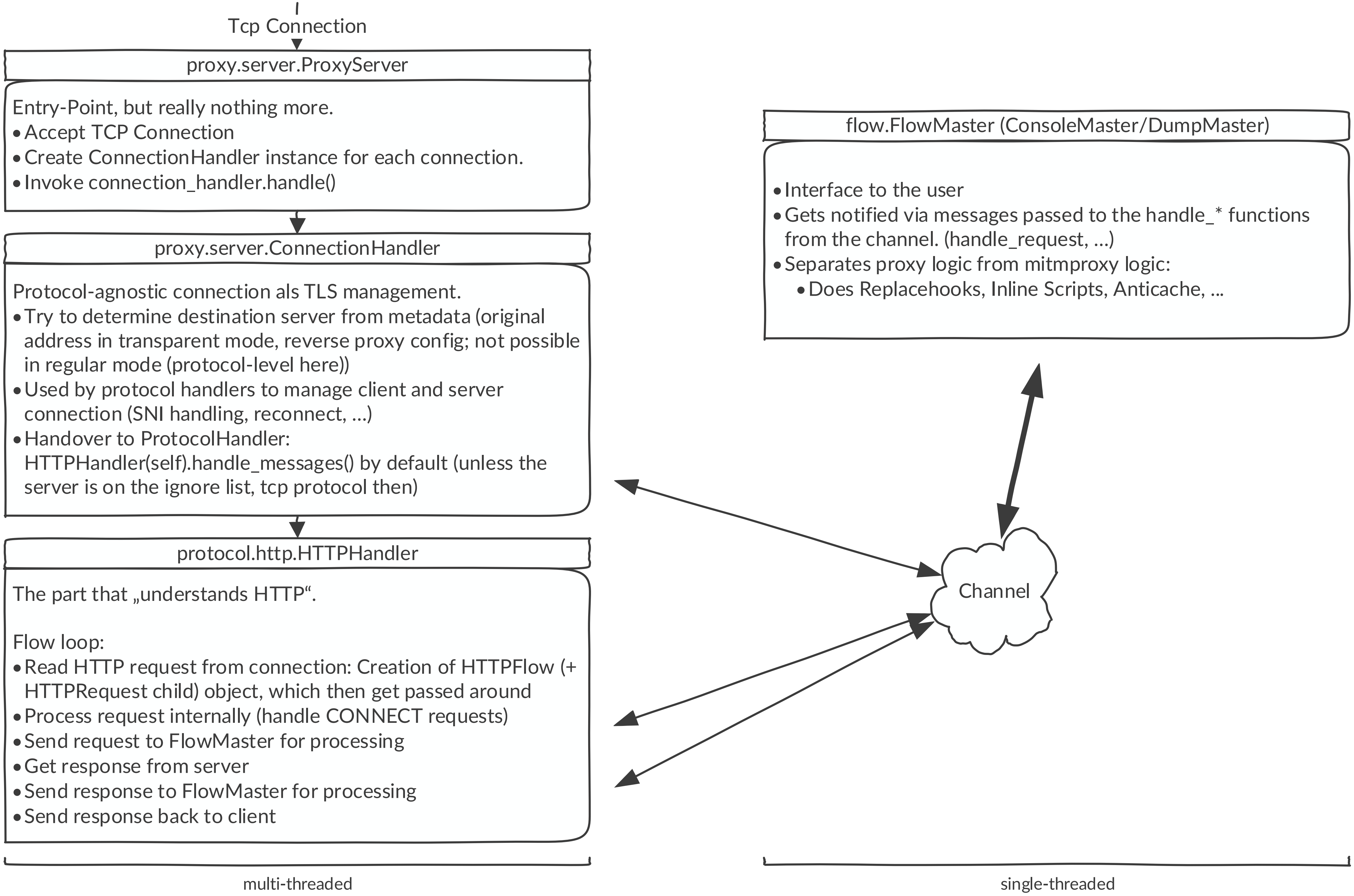

Tcp Connection

proxy.server.ProxyServer

Entry-Point, but really nothing more.

- Accept TCP Connection
- Create ConnectionHandler instance for each connection.
- Invoke connection_handler.handle()

proxy.server.ConnectionHandler

Protocol-agnostic connection als TLS management.

- Try to determine destination server from metadata (original address in transparent mode, reverse proxy config; not possible in regular mode (protocol-level here))
- Used by protocol handlers to manage client and server connection (SNI handling, reconnect, ...)
- Handover to ProtocolHandler: HTTPHandler(self).handle_messages() by default (unless the server is on the ignore list, tcp protocol then)

protocol.http.HTTPHandler

The part that „understands HTTP“.

Flow loop:

- Read HTTP request from connection: Creation of HTTPFlow (+ HTTPRequest child) object, which then get passed around
- Process request internally (handle CONNECT requests)
- Send request to FlowMaster for processing
- Get response from server
- Send response to FlowMaster for processing
- Send response back to client

flow.FlowMaster (ConsoleMaster/DumpMaster)

- Interface to the user
- Gets notified via messages passed to the handle_* functions from the channel. (handle_request, ...)
- Separates proxy logic from mitmproxy logic:
  - Does Replacehooks, Inline Scripts, Anticache, ...