ZeroMQ Message Transport Protocol (ZMTP)

return to the de facto Standards area

Source: ØMQ - The Guide

Table 1: Data Sheet for ZeroMQ Message Transport Protocol (ZMTP)

Characteristic	Value
Original author(s)	Pieter Hintjens
Developer(s)	iMatrix
Initial release	v 3.0
Stable release	None
API Documentation	https://rfc.zeromq.org/spec:23/ZMTP/
Repository	https://github.com/zeromq/zmtp
Written in	C
Example Languages	C
Operating system	
Guide	https://rfc.zeromq.org/spec:23/ZMTP/
Available in	English
Type	Transport Protocol
License	LGHPLv3
Website	http://zeromq.org/

Abstract

The ZeroMQ Message Transport Protocol (ZMTP) is a transport layer protocol for exchanging messages between two peers over a connected transport layer such as TCP. This document describes ZMTP 3.0.

Goals

The ZeroMQ Message Transport Protocol (ZMTP) is a transport layer protocol for exchanging messages between two peers over a connected transport layer such as TCP. This document describes version 3.0 of ZMTP. ZMTP solves a number of problems we face when using TCP carry messages:

- *TCP carries a stream of octets with no delimiters, but we want to send and receive discrete messages. Thus, ZMTP reads and writes frames consisting of a size and a body.*
- *We need to carry metadata on each frame (such as, whether the frame is part of a multipart message, or not). ZMTP provides a flags field in each frame for metadata.*
- *We need to be able to talk to older implementations, so that our framing can evolve without breaking existing implementations. ZMTP defines a greeting that announces the implemented version number, and specifies a method for version negotiation.*

- *We need security so that peers can be sure of the identity of the peers they talk to, and so that messages cannot be tampered with, nor inspected, by third parties. ZMTP defines a security handshake that allows peers to create a secure connection.*
- *We need a range of security protocols, from clear text (no security, but fast) to fully authenticated and encrypted (secure, but slow). Further, we need the freedom to add new security protocols over time. ZMTP defines a way for peers to agree on an extensible security mechanism.*
- *We need a way to carry metadata about the connection, after the security handshake. ZMTP defines a standard set of metadata properties (socket type, identity, etc.) that peers exchange after the security mechanism.*
- *We need to write down these solutions in a way that is easy for teams to implement on any platform and in any language. ZMTP is thus specified as a formal protocol (this document) and made available to teams under a free license.*
- *We need guarantees that people will not create private forks of ZMTP, thus breaking interoperability. ZMTP is thus licensed under the GPLv3, so that any derived versions must also be made available to users of software that implements it.*