

# CLEO Protocol Comparison Guide

Protocol flexibility becomes increasingly important as your network of trading partners and customers grows. Protocols enable file transfers by outlining a standard procedure for regulating the data exchange between businesses. Depending on your business needs and trading partner requirements, one or many protocols may be appropriate for you.

## The Consideration List

### The Security Checklist

At the core of evolving protocols is the expectation for heightened security around the data being transferred. We've broken down the functional elements of protocol security into the following categories:

#### ✓ Privacy

Privacy means that a transaction between two points cannot be viewed or disrupted by an outside party. Encryption is used to make sure business-critical transactions are kept private. Encryption levels can be further segmented to give two layers of protection to your data – transport and payload.

*Transport Encryption* – Secures the entire connection so that all data contents cannot be viewed.

*Payload Encryption* – Ensures the data being transferred is encrypted prior to sending.

#### ✓ Authentication

Requiring identity credentials before providing access to a system allows businesses to control their content and protect their data. There are three elements related to protocol authentication:

*Username & Password* – Assigning users a unique name and password before allowing connections to data provides the first level of authentication.

*Client Authentication* – Using keys or certificates provides the identifying credentials for clients adding a higher level of authentication. Digital certificates are often signed by a certificate authority – further increasing security.

*Session Authentication* – Client Authentication is provided within the transport layer. Session authentication makes use of keys or certificates within protocol commands to further authenticate parties prior to exchanging any payload.

#### ✓ Integrity

The two qualifiers for integrity are:

1. The data that is sent is identical to the data that is received.
2. The ability to prove that two different copies of a file are identical or not.

Message digest or hashing is often used to ensure data integrity. These methods perform an operation to calculate a number from any type of data. The resulting number will be the same when calculated from the same set of data, regardless of the device used. Any changes to the file will result in a new number being generated.

#### ✓ Non-Repudiation

It's critical to know when discrepancies are present in the data being transferred. Non-repudiation ensures that a trading partner cannot dispute having sent or received a file. There are two parts to non-repudiation that are commonly used together.

*Receipts* – File transfers are confirmed with a returned document or message giving the status of the file transfer. In many cases, receipts also confirm integrity checks as well.

*Signing* – Digital certificates are used to provide the user a way to "sign" a file or receipt. Public and private keys ensure that signatures can be verified by the receiving side and prohibit file transfers from unauthorized parties.

## • The Protocol List Defined

### • Standard File Transfer Protocols

- AS2 ▶ Applicability Statement 2
- AS3 ▶ Applicability Statement 3
- ebMS 2.0 ▶ ebXML Messaging Service (ebXML is Electronic Business Using eXtensible Markup Language)
- FTP ▶ File Transfer Protocol
- FTPs ▶ File Transfer Protocol Secure (Also Known as FTP over SSL)
- SSH FTP ▶ Secure Shell File Transfer Protocol
- HTTP ▶ Hypertext Transfer Protocol
- HTTPs ▶ Hypertext Transfer Protocol Secure (Also Known as HTTP over SSL)
- MLLP ▶ Minimal Lower Layer Protocol
- OFTP ▶ Odette File Transfer Protocol
- OFTP2 ▶ Odette File Transfer Protocol 2
- RNIF ▶ RosettaNet Implementation Framework
- SMTP ▶ Simple Mail Transfer Protocol
- SMTPs ▶ Simple Mail Transfer Protocol Secure (Also Known as SMTP over SSL)
- WS ▶ Web Services

### • Proprietary Protocols

- fasp™ ▶ Aspera Software's High-Speed File Transfer Protocol
- IBM® ▶ IBM® WebSphere® MQ

## • CLEO Software Solutions to Support Protocol Requirements

- ▶ **CLEO LexiCom™**  
*secure file transfer client*
- ▶ **CLEO VLTrader™**  
*enterprise managed file transfer*





## Additional Protocol Considerations . . .

Understanding your current AND future file transfer needs will help you to select the best protocols for standardization and security. Top considerations include file size, volume, frequency and the associated protocol functionality required.

### Message Size

Understanding the range of file sizes that support is needed for will help to better identify an appropriate protocol to meet your needs.

### Compression

Before sending a document or data set to a recipient, compression reduces the size of the file. This reduces the bandwidth usage on both sides and can reduce the total file transfer time.

### Speed

Less overhead on a protocol increases speed. Network conditions, bandwidth and latency all have a compounding impact on protocol performance. It's important to evaluate all of these factors in relation to your desired speed requirements.

### Restart

Very important for large file transfers, restart capabilities allow a protocol to reinstate a failed or interrupted file transfer from the point it was stopped as opposed to restarting the transfer from the beginning.

### Certification

Certification with governing bodies or interoperability testing groups ensures the promised performance of a protocol and reduces setup time when both the sending and receiving parties are utilizing the same specifications and optional protocol functionality. (The Drummond Group is the major global certification body for interoperable secure communication of AS2, AS3 & ebMS protocols.)

### Firewall Friendly

This measurement helps to qualify general ease of use when configuring protocols to talk through the firewall. Protocols that change ports or use port ranges often require greater firewall configuration. Those protocols identified as "Firewall Friendly" tend to provide easier firewall configuration.

### Architecture

Protocol architecture determines how a protocol communicates from point-to-point.

*Peer-to-Peer* – Sometimes referred to as push/push, these protocols work by allowing simultaneous traffic flow from sender to receiver and file transfers are always initiated from the side that has the data. Peer-to-Peer protocols allow for real-time automation on both ends of the connection and require both sides to be in a constant listening state to receive files real time.

*Client-Server* – Sometimes referred to as push/pull, these protocols and associated file transfers are always initiated from the client side. This requires the server to be in a constant listening state, but does not require the client to do the same. Client-Server protocols require fewer configurations on the client side and are more widely used for date/time or batch based file transfers.

