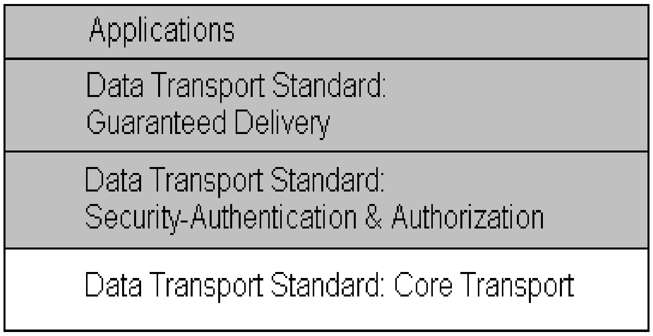
# Data Transport Core Standard

This is a protocol specification only and must be easily implementable in Java and .Net.

Data Transport Standard is a layered protocol definition, with each higher layer building on the one below it. This provides DTS implementors with manageable segments to implement, and will make managing the implementation and maintenance easier.


Applications
Data Transport Standard: Guaranteed Delivery
Data Transport Standard: Security-Authentication & Authorization
Data Transport Standard: Core Transport

Core Transport is defined within the WSDL document. This XML document defines the required set of information to invoke a SOAP connection from a client to a server.

(Put WSDL here)

The body of the SOAP message will include exactly one element named "payload" with zero attributes which is a Base 64 encoded, Zlib compressed CDATA section.

Each request and response must contain the following header elements:

```
Sender - Type core:EntityType
Recipient - Type core:EntityType
TransportUUID - UUID of type xs:String
TransmitDateTimeGMT - Type xs:dateTime
```

Any implementation of the core protocol must be able to handle additional unknown header elements.

Any request that is received with any of these header elements missing will be rejected and a SOAP Fault will be returned.

Every distinct file type will have a standard SOAP method name to use. The method name will closely correlate to the Email and FTP filenames currently defined by ESC.

All method names will follow the following naming convention:

- [submit/request] - submit is used when pushing data to the trading partner. request is used when requesting the trading partner to return data.
- [File Type] - CRC, CAM, CL. All Upper Case
- [Version] - two digit number with the version of file being sent/requested
- [File Sub Type] - Camel Case of the current sub type
- {Batch} - If the response is not expected immediately then this request is to be considered a "Batch Mode" request. If the response is expected in real-time, then no suffix will be used.

As an example, CRC Version 1 Application Send request expecting a real-time response will be named submitCRC01AppSend. A similar request with an expected response at a later time would be named submitCRC01AppSendBatch

There will be a centralized LDAP server to act as a registry for all SOAP methods supported. This centralized registry will contain each participating entity. Each entity will contain business and technical contact information. A list of all supported transaction types and the public X.509 key of the entity will also be stored.