

What Is Claimed Is:

1. An apparatus to improve network transmission efficiency of a sending peer, comprising:
 - an encoder interface to prevent an encoder from encoding data in a native format to a target format associated with a requesting peer in response to a transmission trigger;
 - a composite interface description (CID) engine to generate architecture agnostic metadata of the data in the native format;
 - a composite layout description (CLD) engine to generate architecture specific metadata of the data in the native format, the architecture specific metadata identifying offset values associated with elements of the data in the native format; and
 - a client payload engine to improve the network transmission efficiency by sending unencoded data combined with the architecture agnostic metadata and the architecture specific metadata to the requesting peer having an architecture different than that of the sending peer.
2. An apparatus as defined in claim 1, wherein the unencoded data includes a storage footprint lower than that of the data in the native format after encoding by the encoder.
3. An apparatus as defined in claim 1, wherein the data in the native format includes a composite associated with a type.
4. An apparatus as defined in claim 3, wherein the type includes at least one of a C-structure or an enumeration.
5. An apparatus as defined in claim 1, further including a pointer manager to identify a first one of the elements in the data in the native format.
6. An apparatus as defined in claim 5, further including a CID file generator to generate a CID file to store the architecture agnostic metadata.
7. An apparatus as defined in claim 6, wherein the CID file generator is to encode element data into the CID file.

8. An apparatus as defined in claim 7, wherein the element data includes at least one of a composite type, a composite name, a composite element name, or a composite element type.
9. An apparatus as defined in claim 5, further including a CLD file generator to generate a CLD file to store the architecture specific metadata.
10. An apparatus as defined in claim 9, further including a payload analyzer to:
identify the offset values associated with the elements of the data in the native format;
and
write the offset values in the CLD file.
11. An apparatus as defined in claim 1, further including a task process compiler to compile the architecture agnostic metadata and the architecture specific metadata into a task executable of the sending peer.
12. A method to improve network transmission efficiency of a sending peer, comprising:
preventing an encoder from encoding data in a native format to a target format associated with a requesting peer in response to a transmission trigger;
generating architecture agnostic metadata of the data in the native format;
generating architecture specific metadata of the data in the native format, the architecture specific metadata identifying offset values associated with elements of the data in the native format; and
improving the network transmission efficiency by sending unencoded data combined with the architecture agnostic metadata and the architecture specific metadata to the requesting peer having an architecture different than that of the sending peer.
13. A method as defined in claim 12, further including identifying a first one of the elements in the data in the native format.
14. A method as defined in claim 13, further including generating a composite layout description (CLD) file to store the architecture specific metadata.

15. A method as defined in claim 14, further including:
identifying the offset values associated with the elements of the data in the native format; and
writing the offset values in the CLD file.
16. A tangible computer readable storage medium comprising computer readable instructions which, when executed, cause a processor to at least:
prevent an encoder from encoding data in a native format to a target format associated with a requesting peer in response to a transmission trigger;
generate architecture agnostic metadata of the data in the native format;
generate architecture specific metadata of the data in the native format, the architecture specific metadata identifying offset values associated with elements of the data in the native format; and
improve the network transmission efficiency by sending unencoded data combined with the architecture agnostic metadata and the architecture specific metadata to the requesting peer having an architecture different than that of the sending peer.
17. A tangible computer readable storage medium as defined in claim 16, wherein the instructions are to cause the processor to transmit the unencoded data in a storage footprint lower than that of the data in the native format after encoding by the encoder.
18. A tangible computer readable storage medium as defined in claim 16, wherein the instructions are to cause the processor to transmit the native format as a binary format.
19. A tangible computer readable storage medium as defined in claim 16, wherein the instructions are to cause the processor to identify a first one of the elements in the data in the native format.
20. A tangible computer readable storage medium as defined in claim 19, wherein the instructions are to cause the processor to generate a composite interface description (CID) file to store the architecture agnostic metadata.

21. A tangible computer readable storage medium as defined in claim 20, wherein the instructions are to cause the processor to encode element data into the CID file.

22. A tangible computer readable storage medium as defined in claim 19, wherein the instructions are to cause the processor to generate a composite layout description (CLD) file to store the architecture specific metadata.

23. A tangible computer readable storage medium as defined in claim 22, wherein the instructions are to cause the processor to:

identify the offset values associated with the elements of the data in the native format;
and

write the offset values in the CLD file.

24. A tangible computer readable storage medium as defined in claim 16, wherein the instructions are to cause the processor to generate a metadata reference associated with the combined architecture agnostic metadata and the architecture specific metadata.

25. A tangible computer readable storage medium as defined in claim 16, wherein the instructions are to cause the processor to compile the architecture agnostic metadata and the architecture specific metadata into a task executable of the sending peer.