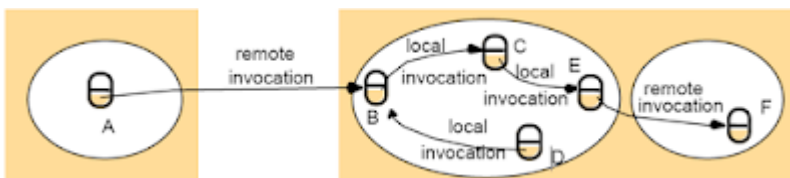


## External data representation

- At language-level data are stored in data structures
- At TCP/UDP-level data are communicated as 'messages' or streams of bytes - hence, conversion/flattening is needed
- Converted to a sequence of bytes
- Problem? Different machines have different primitive data reps,
- Integers: big-endian and little-endian order
- float-type: representation differs between architectures
- char codes: ASCII, Unicode
- Either both machines agree on a format type (included in parameter list) or an intermediate external standard is used:
- External data representation: an agreed standard for the representation of data structures and primitive values
- Marshalling: process of taking a collection of data items and assembling them into a form suitable for transmission.
- Unmarshalling: disassembling (restoring) to original on arrival
- Three alter. approaches to external data representation and marshelling:
- XML (Extensible Markup Language) : defines a textual format for rep. structured data
- First two: marshalling & unmarshalling carried out by middleware layer
- First two: primitive data types are marshalled into a binary form.



## Marshalling

Marshalling is a data presentation conversion, performed according to special rules, usually for network transfer. The following data presentation factors have to be taken into account to perform marshalling.

Different platforms use their own character formats (ASCII or EBCDIC), integers and floats formats (IEEE, VAX, Cray, IBM). For example, the scheme is acceptable - the native numeric representation is two's complement for integers, ANSI/IEEE single/double for floats and also that characters are in ISO Latin/1 (an ASCII superset addressing Western European characters). Byte order is basically dependent on processor. But some processors support different byte orders as required for different software environments. There are different alignment strategies. "Natural alignment" is a policy that the processor controls the alignment of data based on its type. It means that the data of basic types are located in memory from the boundaries divisible by their size. Some CPUs have a maximum alignment requirement of two or four bytes, others have some type-specific exceptions to the normal "alignment == size" rule. "Fixed alignment" ignores data type when establishing alignment; not all processors support such policies. Also, realignment is required for the components of record or structure types, since different CPU's use different rules for positioning structures fields in memory. Linearization is required for data structures that are stored in non-contiguous memory sections, such as dynamically allocated tree structures. So, marshalling procedures have to have information on data format for current platform, convert it into some standard format used for network transfer, get data from network and be able to decode them back from standard network format into this platform one. In this way, marshalling includes two mutually reverse procedures - encode and decode. They have symmetrical algorithms, whose differ just by basic primitives 'put' into and 'get' from some buffer. But decode procedures are always more complex due to the requirement for error-checking and some memory allocation. Further we shall call those procedures marshalling

and unmarshalling, respectively.