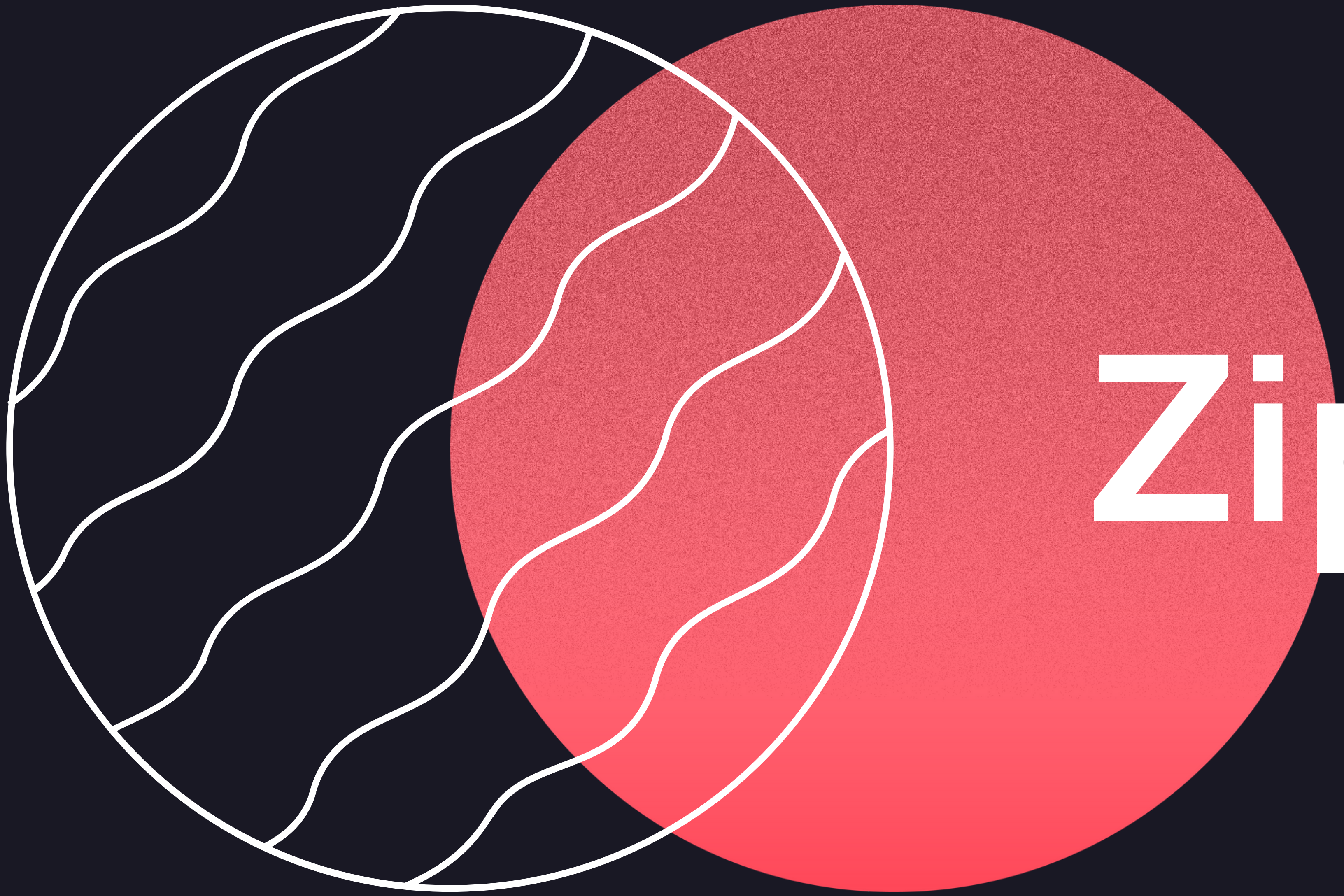
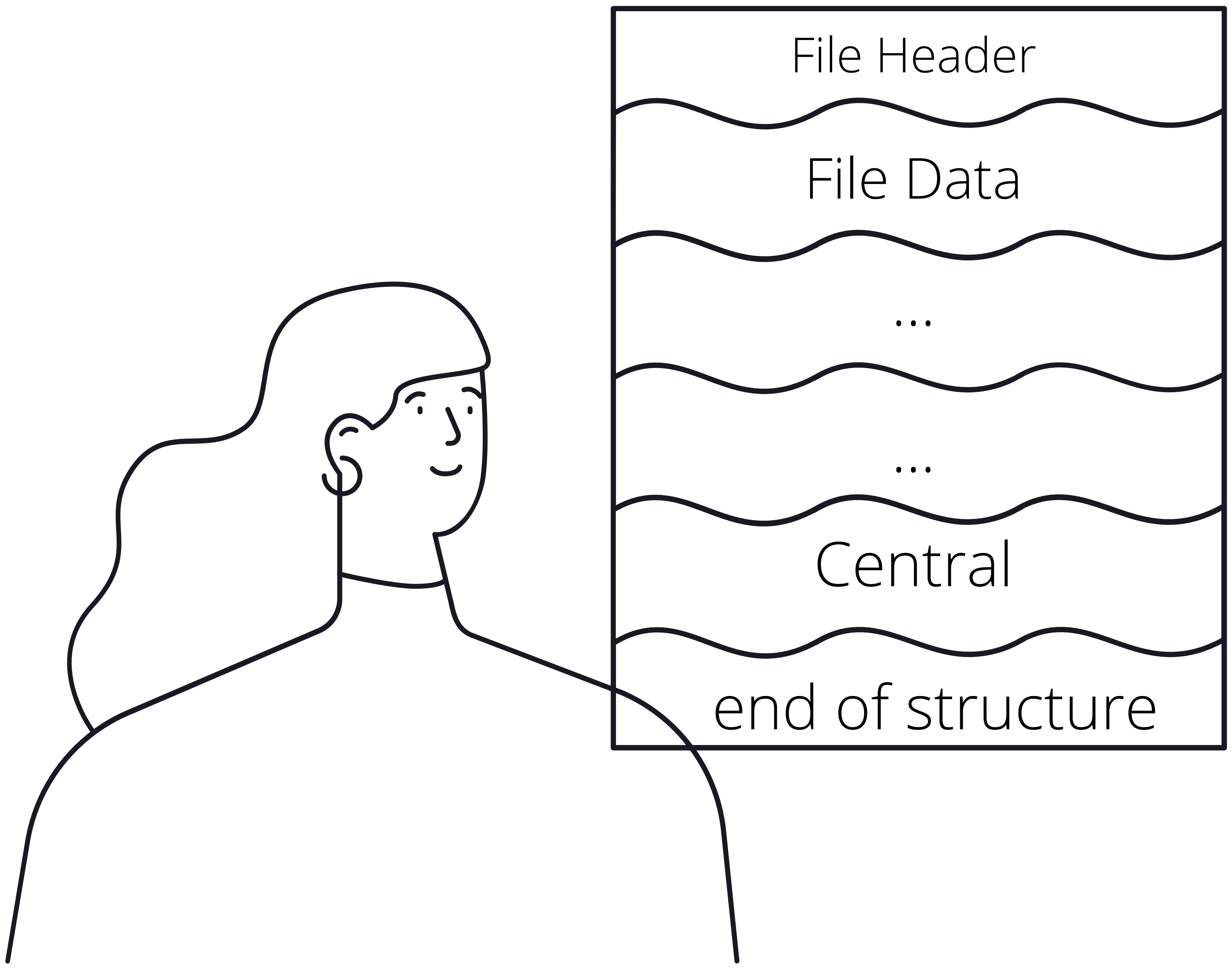


Zip Structure

Structure



File Header

File Data

...

...

Central

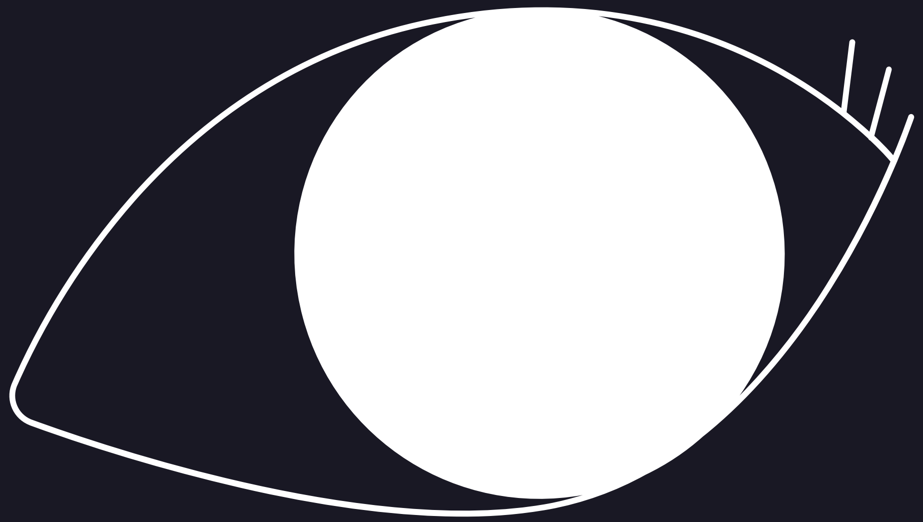
end of structure

Header Component

- Signature
- Version
- Flags
- Compression Size … etc

POINT TO TALK ABOUT

- Create Zip File
- Extract Zip File
- Problems
- Work with real application



EVERY STRUCTURE USED IN ZIP
HAVE MADE BY HANDS

CREATE ZIP FILE

we have to choose file and folder to add to zip file , we read files by stream

Create Entries : we will read every file and folder and make entry for it (each entry has : file header - file data - central header) and choose right encoder

Write to Memory : we will read every entry we have created then write it file header and file data after finish write them we will write centrals then write tail (end of zip structure)

EXTRACT ZIP FILE

we need first to read zip file then we need path to extract

we will still read file header and file data until we start to read central we will stop

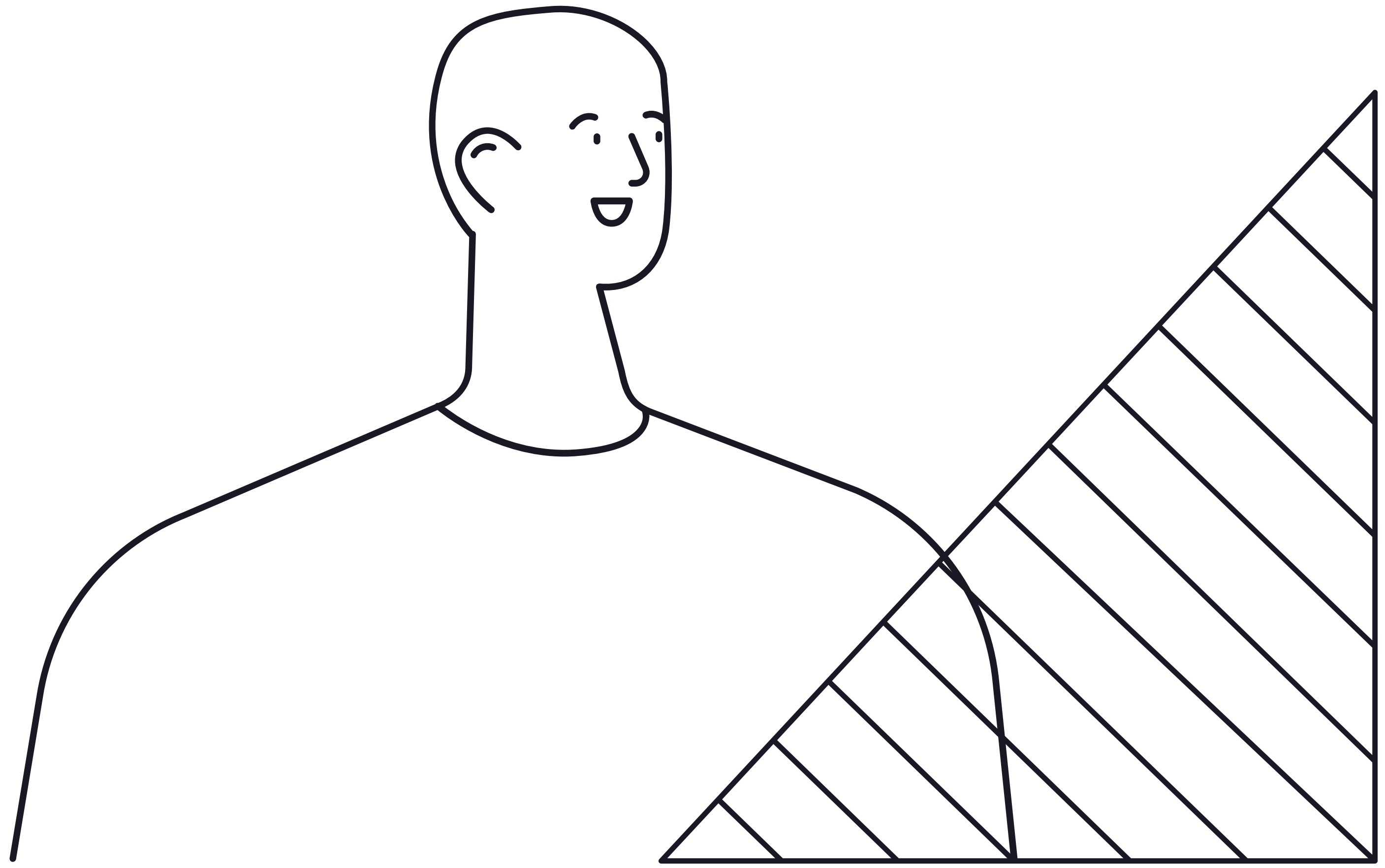
when we read file header we decide what decoder to use after we read compression method in file header then write data in stream that create file on chossen path

PROBLEMS & SOLUTION

problem : how should read and write data type ?
solution : use little endian encode and decode .

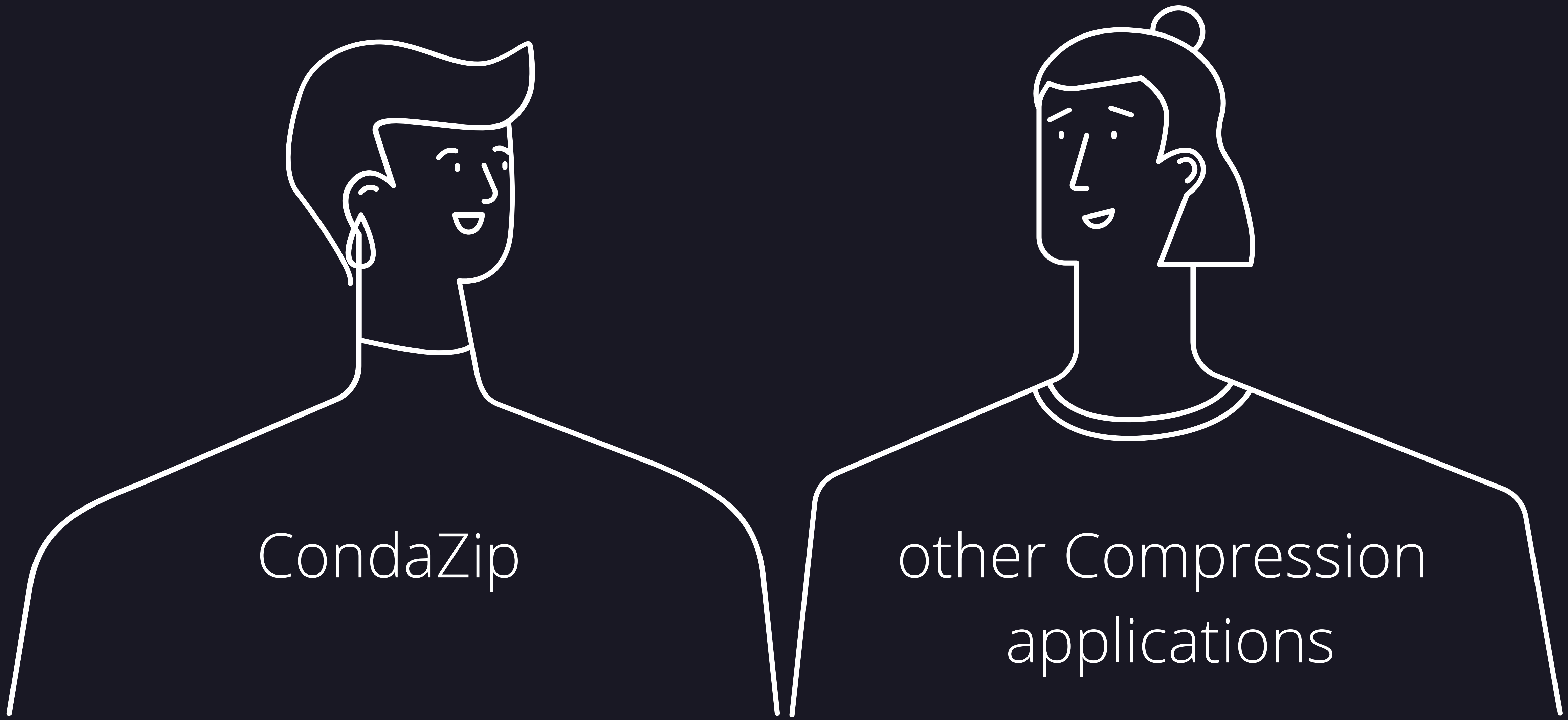
problem : how to know where pointer now in stream ?
solution : count file header length + compressed size

problem : how to choose en/decode dynamic ?
solution : use factory (from design pattern)



WORKS WITH REAL APPLICATION

when we read and write in standard type that mean we speak compression utility language



CondaZip

other Compression applications