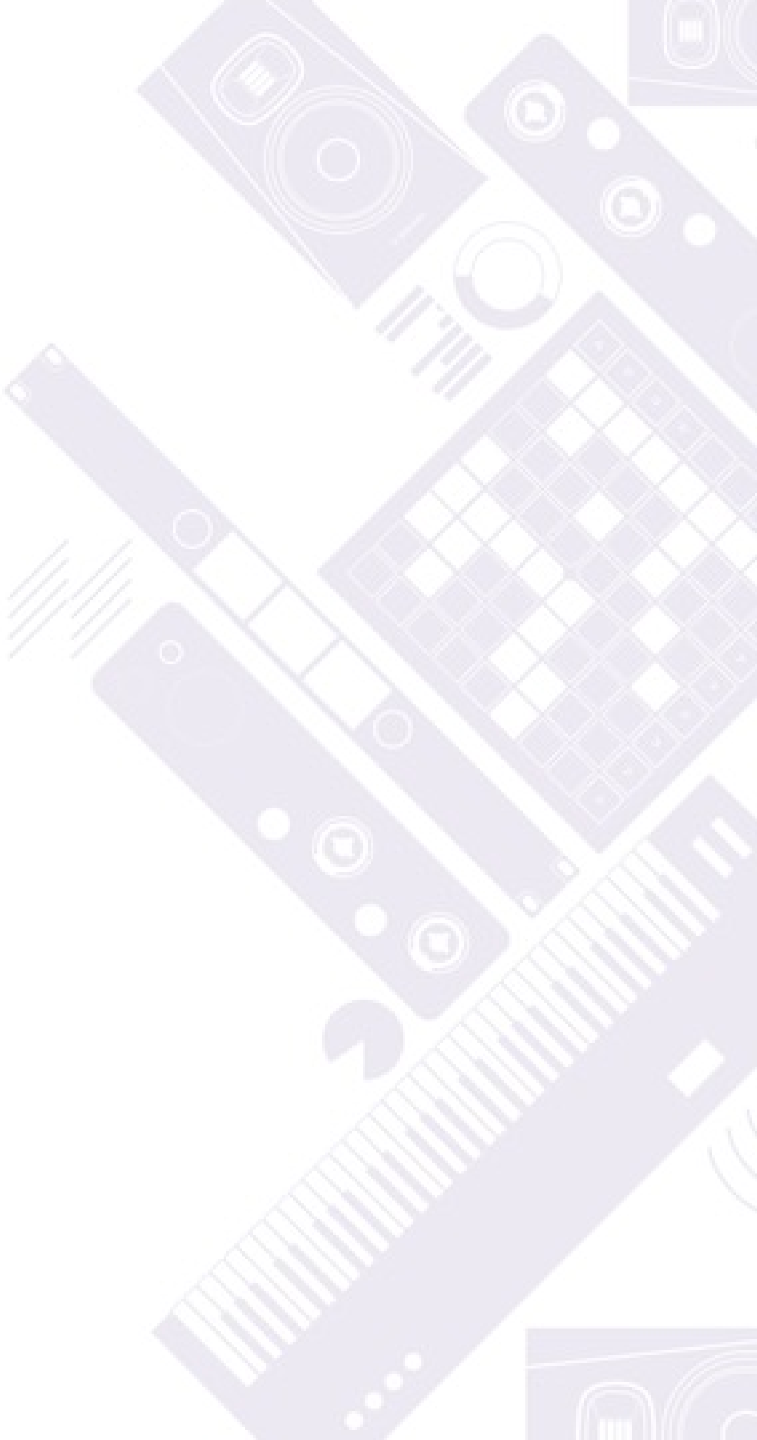
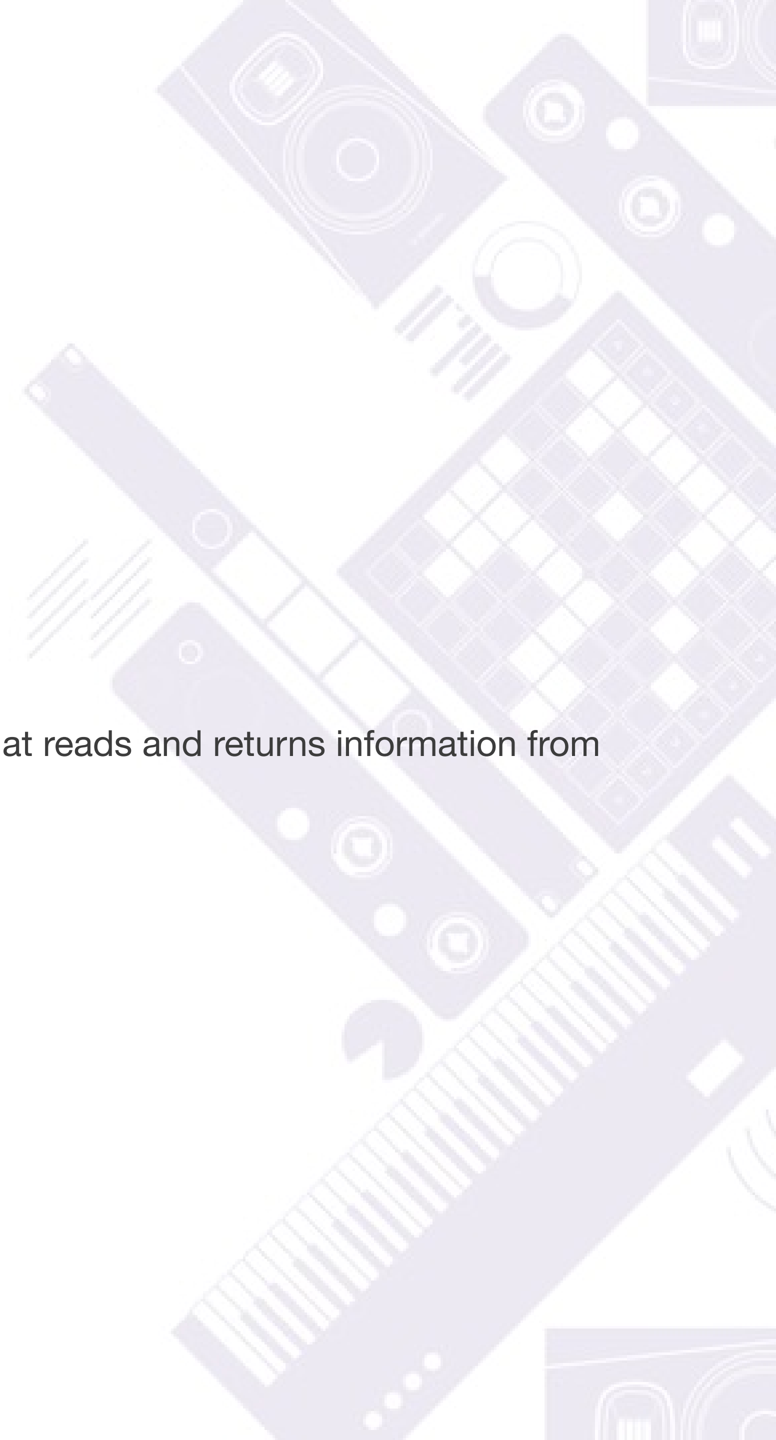
WAV File Parser Challenge



FOCUSRITE GROUP

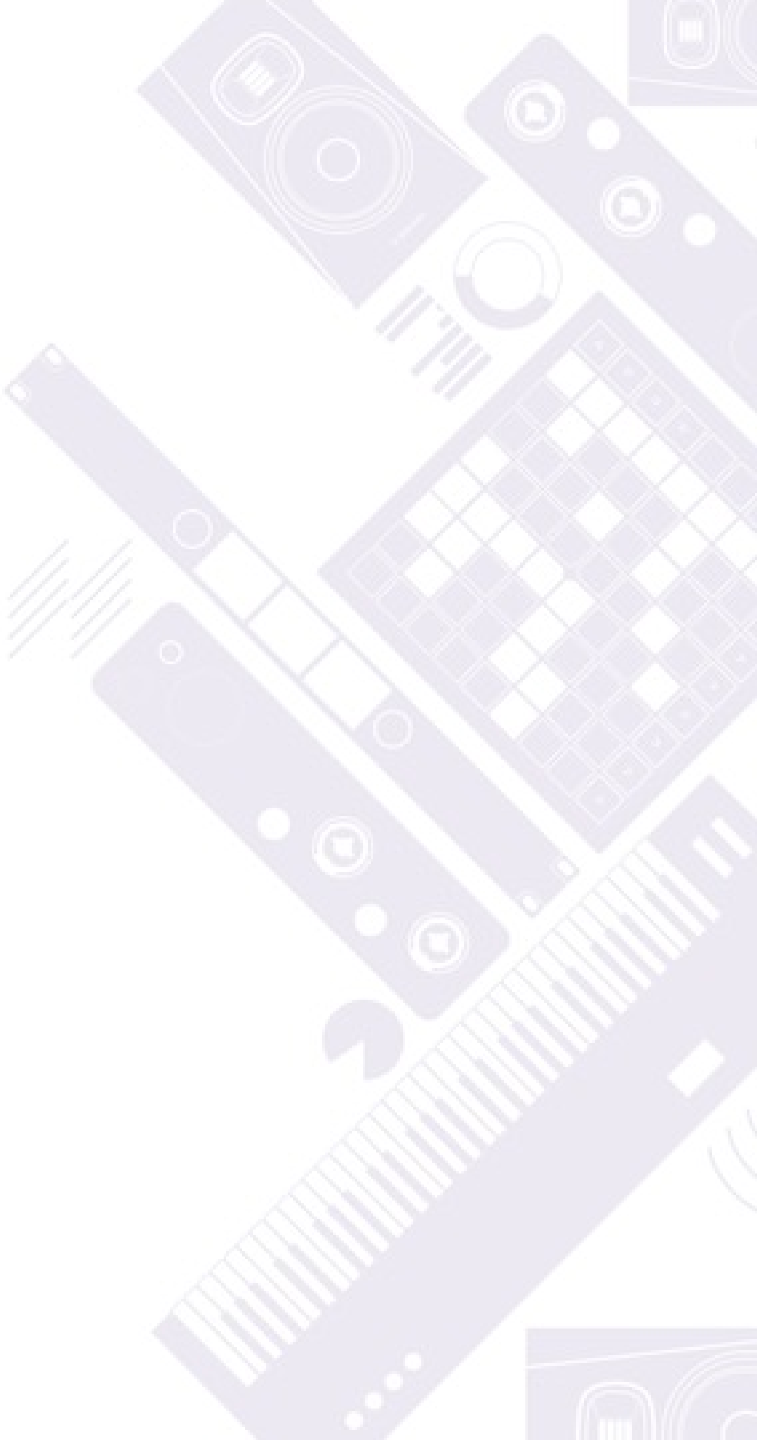
Set Up

Welcome to the WAV File Parser Challenge. Your task is to create a robust, efficient WAV file parser that reads and returns information from the file's header.



What Are We Looking For?

- Problem-solving ability
- High quality, efficient, and maintainable code
- Code in a GitHub repository
- Clean and unit tested code
- Please, do not use the web-audio API for this task



Parsing the WAV Header

Implement a feature to access the headers of a WAV file with the following requirements:

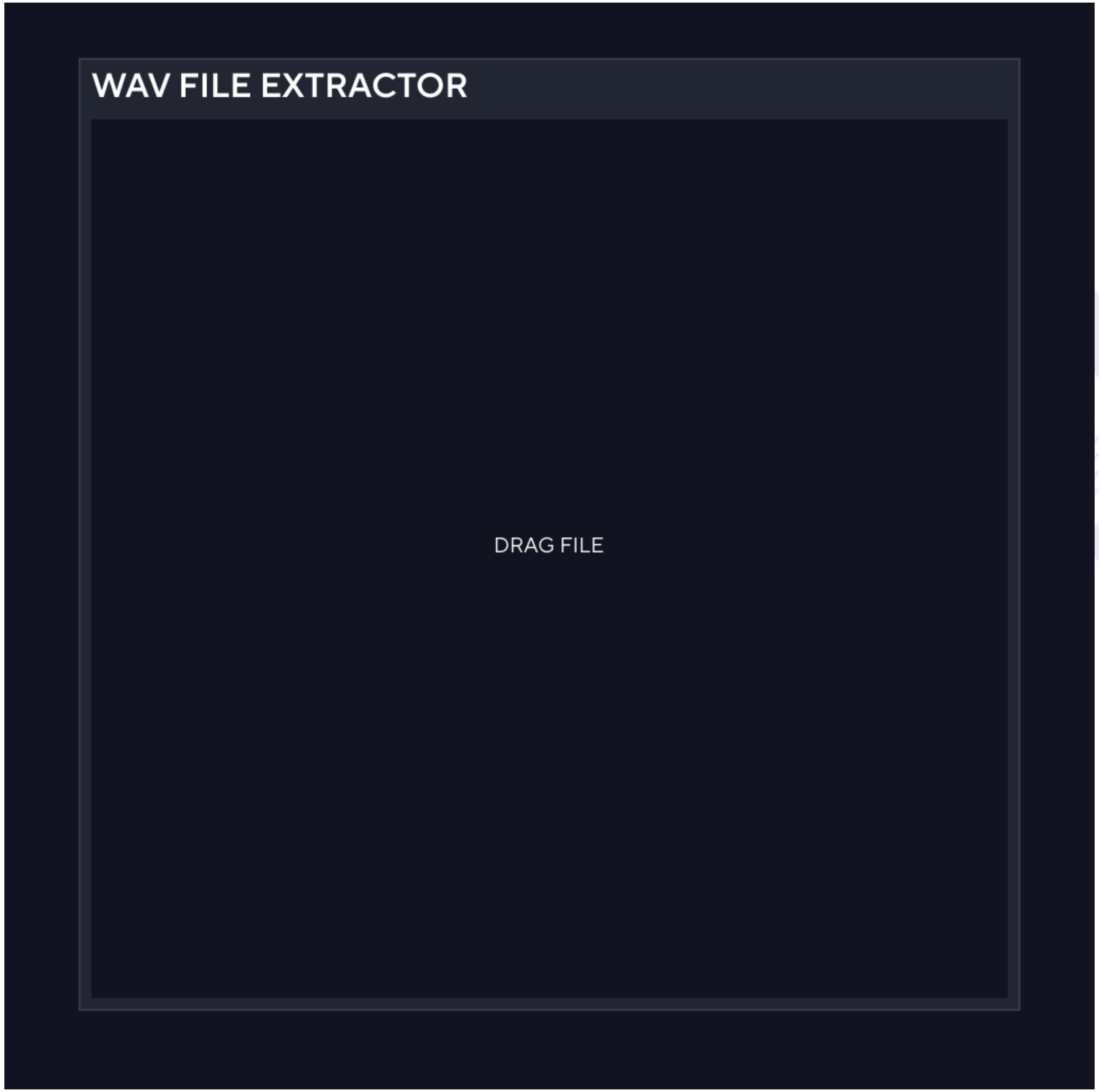
- The file is dragged and dropped onto the page
- Validate that the file is in the .wav format
- Parse the header of the WAV file in the browser
- Retrieve and return the information in the design

To help you get started, we've provided a Sample WAV File alongside this document and the WAVE PCM soundfile format with the designs below.

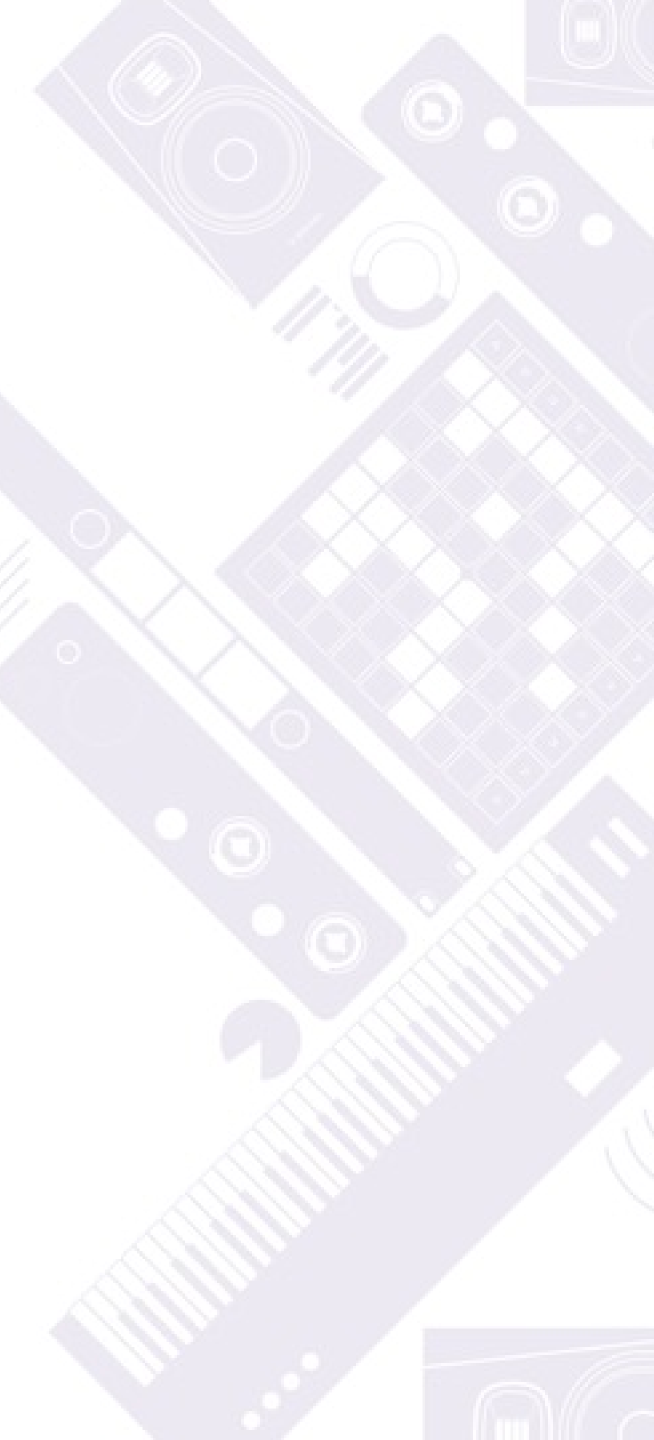
Required Styling and design

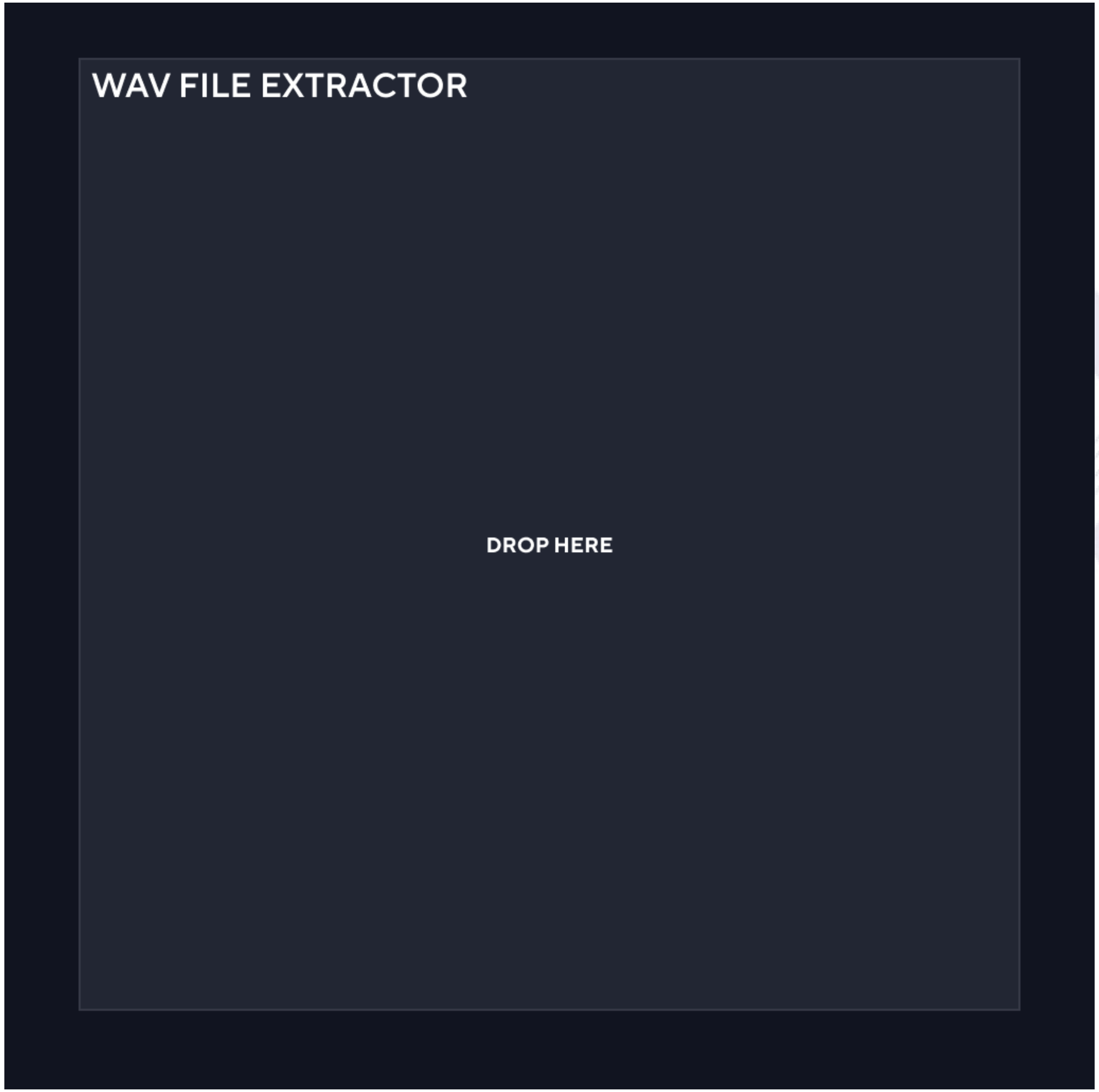
To ensure a cohesive user experience, your web app should follow the following styling guide and design:

- **Primary Font:** TT Norms Pro, sans-serif
- **Colors:** #FFFFFF, #111420, #222633, #363946



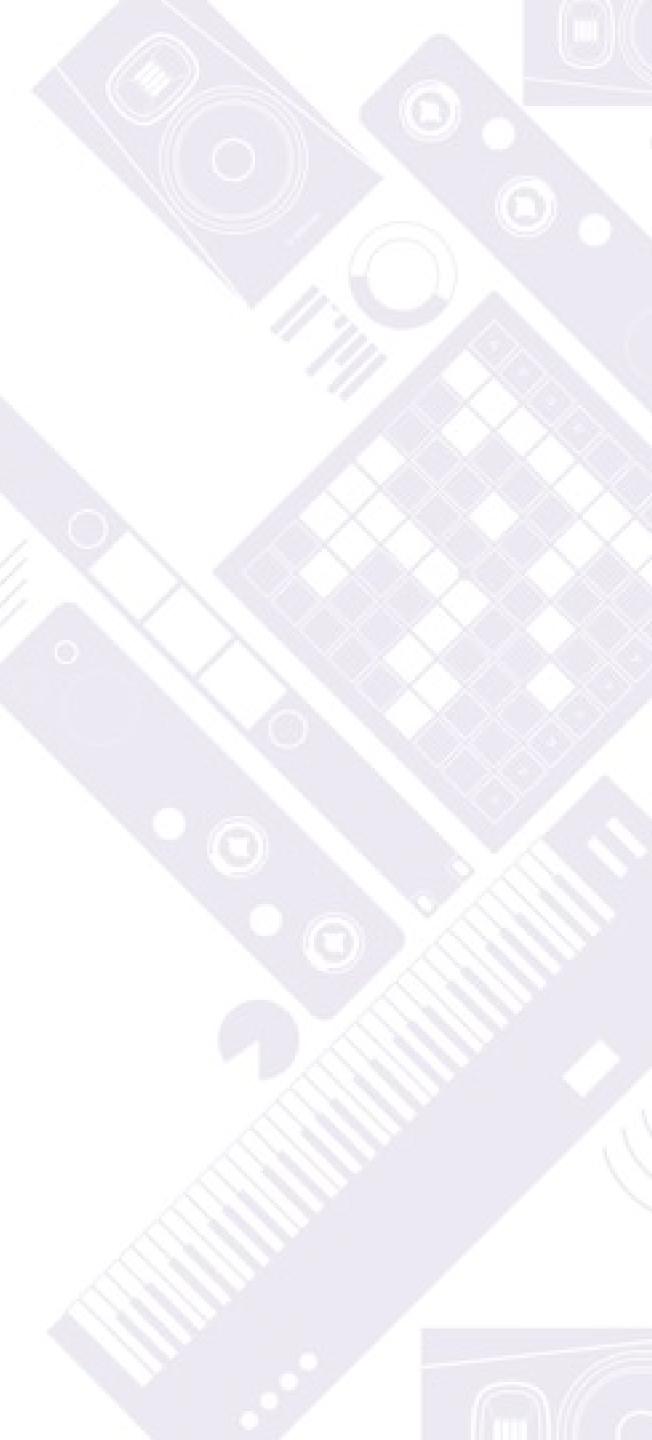
WAV FILE EXTRACTOR

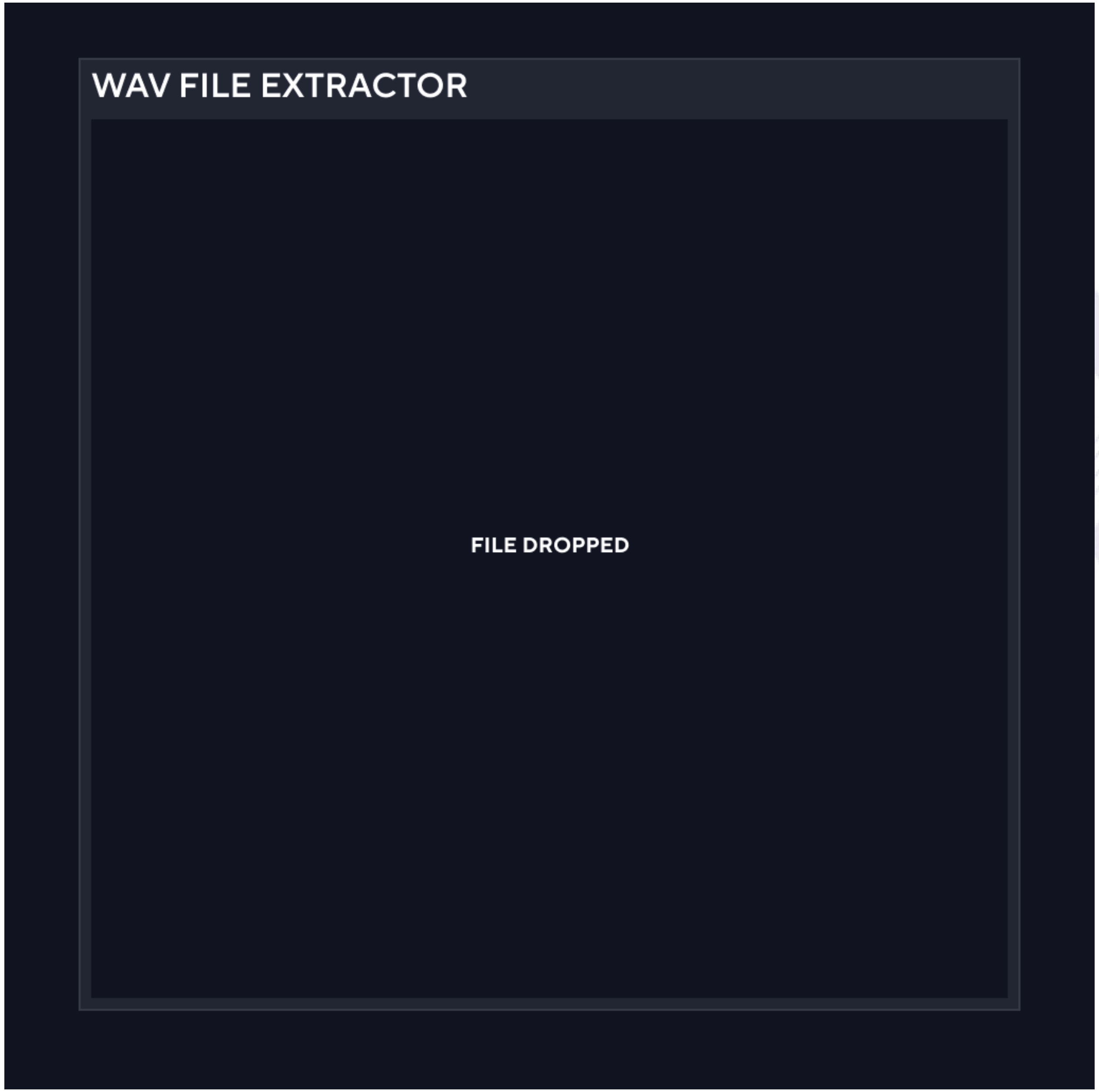




WAV FILE EXTRACTOR

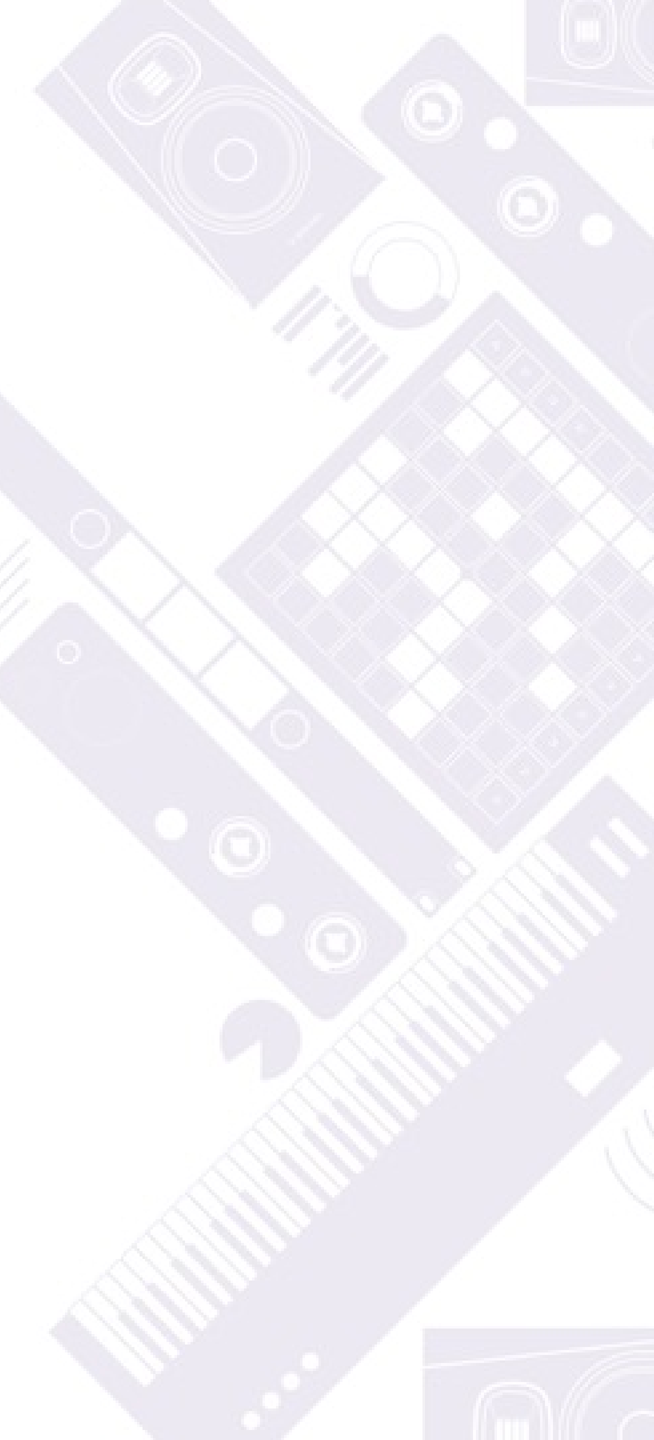
DROP HERE

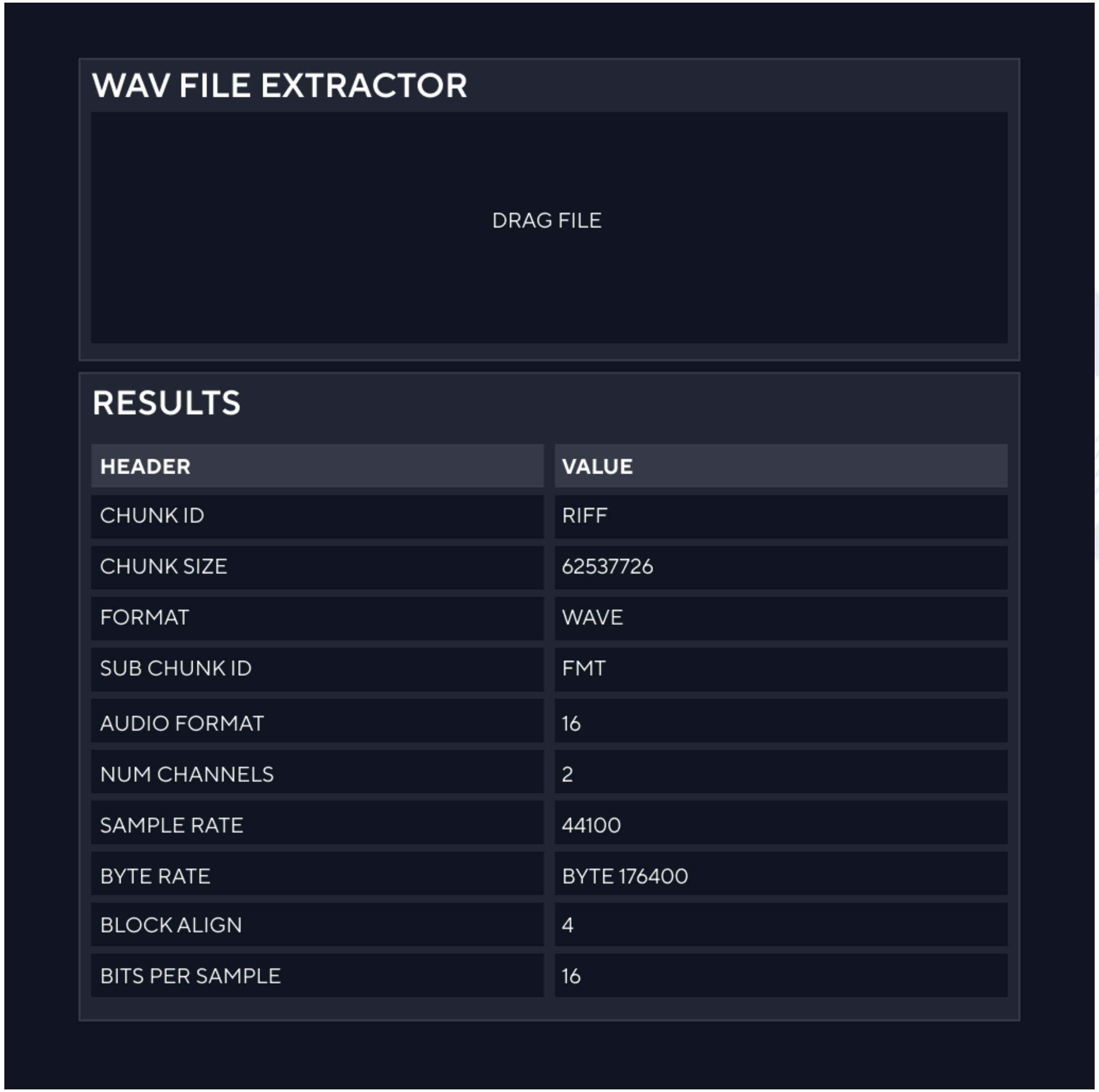




WAV FILE EXTRACTOR

FILE DROPPED



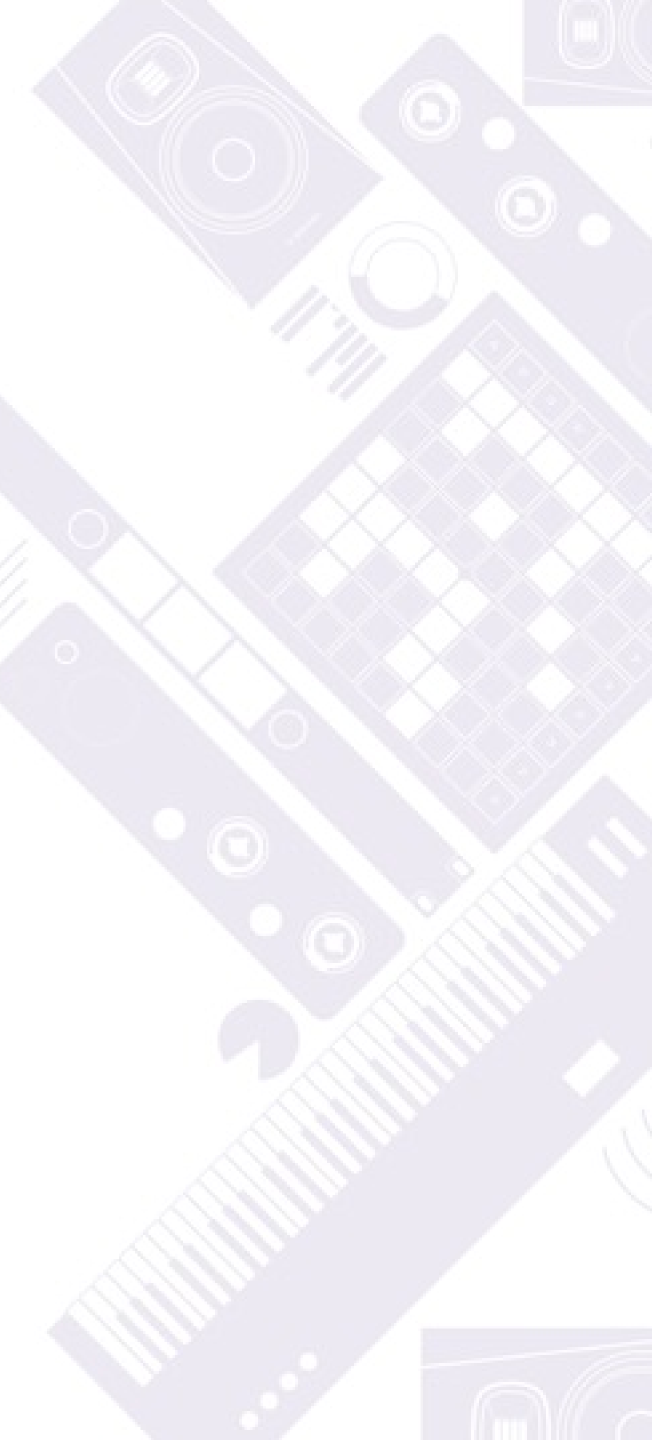


WAV FILE EXTRACTOR

DRAG FILE

RESULTS

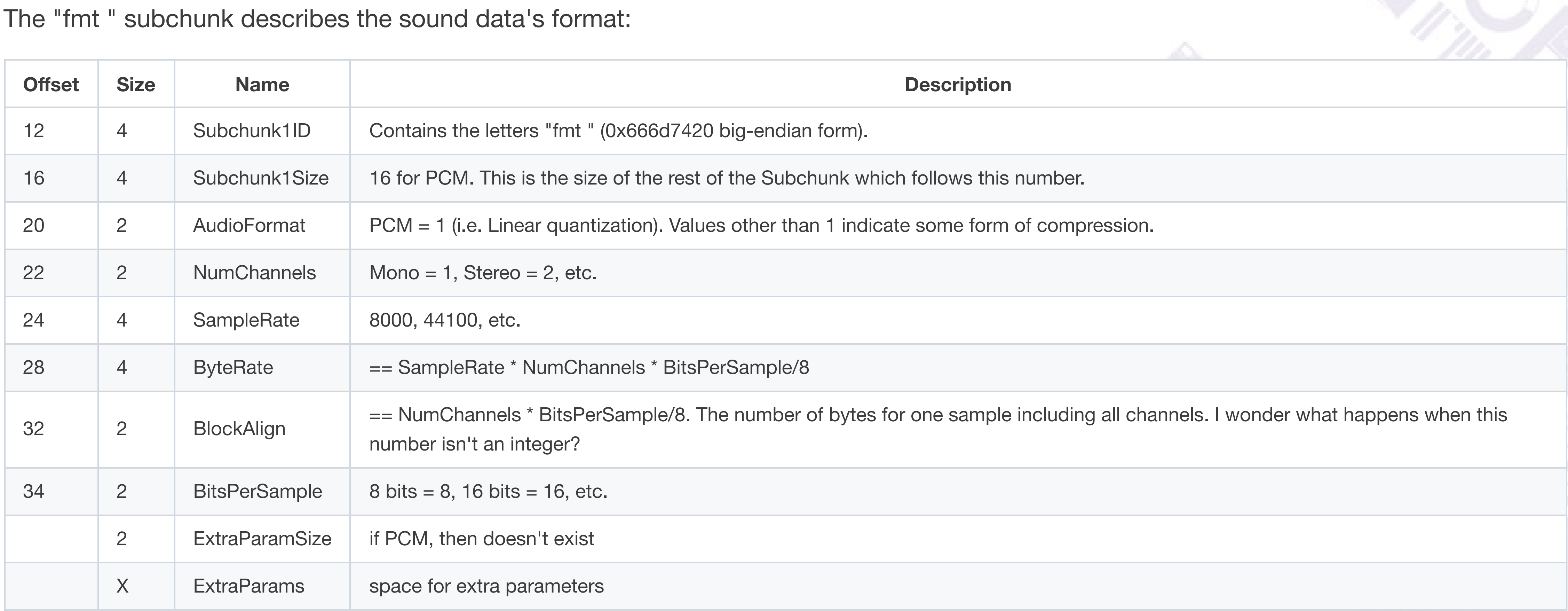
HEADER	VALUE
CHUNK ID	RIFF
CHUNK SIZE	62537726
FORMAT	WAVE
SUB CHUNK ID	FMT
AUDIO FORMAT	16
NUM CHANNELS	2
SAMPLE RATE	44100
BYTE RATE	BYTE 176400
BLOCK ALIGN	4
BITS PER SAMPLE	16



WAVE PCM soundfile format

Offset	Size	Name	Description
0	4	ChunkID	Contains the letters "RIFF" in ASCII form (0x52494646 big-endian form).
4	4	ChunkSize	36 + SubChunk2Size, or more precisely: 4 + (8 + SubChunk1Size) + (8 + SubChunk2Size). This is the size of the rest of the chunk following this number. This is the size of the entire file in bytes minus 8 bytes for the two fields not included in this count: ChunkID and ChunkSize.
8	4	Format	Contains the letters "WAVE" (0x57415645 big-endian form).

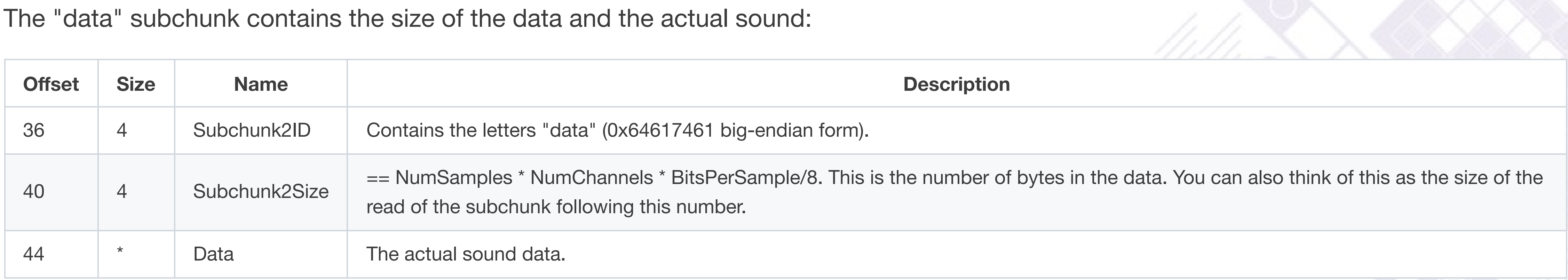
The "WAVE" format consists of two subchunks: "fmt " and "data"



The "fmt " subchunk describes the sound data's format:

Offset	Size	Name	Description
12	4	Subchunk1ID	Contains the letters "fmt " (0x666d7420 big-endian form).
16	4	Subchunk1Size	16 for PCM. This is the size of the rest of the Subchunk which follows this number.
20	2	AudioFormat	PCM = 1 (i.e. Linear quantization). Values other than 1 indicate some form of compression.
22	2	NumChannels	Mono = 1, Stereo = 2, etc.
24	4	SampleRate	8000, 44100, etc.
28	4	ByteRate	== SampleRate * NumChannels * BitsPerSample/8
32	2	BlockAlign	== NumChannels * BitsPerSample/8. The number of bytes for one sample including all channels. I wonder what happens when this number isn't an integer?
34	2	BitsPerSample	8 bits = 8, 16 bits = 16, etc.
	2	ExtraParamSize	if PCM, then doesn't exist
	X	ExtraParams	space for extra parameters

FOCUSRITE GROUP



The "data" subchunk contains the size of the data and the actual sound:

Offset	Size	Name	Description
36	4	Subchunk2ID	Contains the letters "data" (0x64617461 big-endian form).
40	4	Subchunk2Size	== NumSamples * NumChannels * BitsPerSample/8. This is the number of bytes in the data. You can also think of this as the size of the read of the subchunk following this number.
44	*	Data	The actual sound data.

Good Luck!