WAVESABRE

A case study in 64k synthesis

Jake «Ferris» Taylor / Youth Uprising

The Gathering 2013

INTRODUCTION

WHO AM I?

INTRODUCTION

- Jake Taylor
- Ferris / Youth Uprising
- «Real-Time Graphics Engineer» at Outracks Technologies
  - (think RenderProgrammer++)
- Age 21
- But none of this matters. On to synthy things.

WHAT IS A 64K SYNTH, EXACTLY?

INTRODUCTION

- Sound synthesizer
- Generates music on-the-fly for 64k intros
  - Also used in exemusic etc.
- They sound pretty good by now

WHY DID I DO MY OWN?

INTRODUCTION

- I’ve wanted to since 2009
- I like making/using my own tools
- There are simple techniques that aren’t available in existing synths
- I wanted to make music I thought was cool
  - Not necessarily something so general-purpose
- **Because I can.**

WHY A CASE STUDY?

INTRODUCTION

- It’s not really a science
- My system is fundamentally different
  - (Except for Mvx)
- It’s not necessarily complicated, but there are lots of small pieces

WHY SHOULD YOU CARE?

INTRODUCTION

- I have no idea. Not my problem.

TERMINOLOGY
INTRODUCTION

- **WAVESABRE:**
  - My synth :)

- **64K SYNTH** (ambiguous):
  - The whole 64k sound system
  - A specific part of the system dealing with sound generation (SYNTHesis)

- **CHANNEL/TRACK**:
  - A stream for audio from A to B
  - Can also have effects and/or sum/split from/to other channels

- **OSCILLATOR:**
  - Original source of an audio signal (sine etc)

TERMINOLOGY

INTRODUCTION

- **DEVICE:**
  - A signal processor/source

- **SAMPLE** (ambiguous):
  - The elementary unit of an audio signal
  - Represents speaker position at a time value

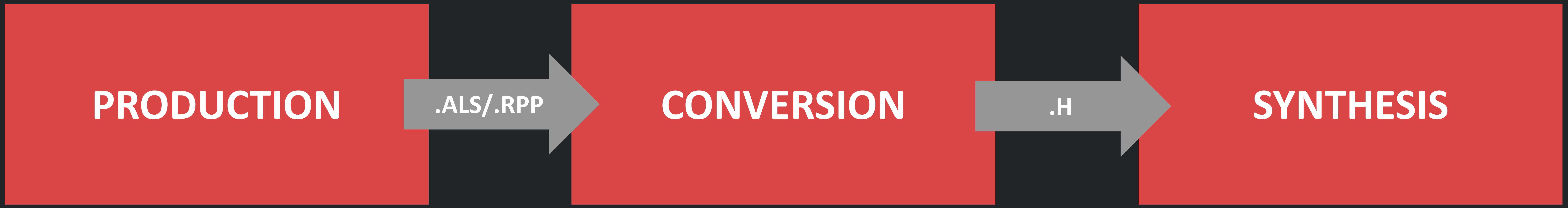
- **BUFFER**:
  - A storage/transmission medium for an audio signal
    - (An array of samples)

THE LIFE OF A WAVESABRE SONG

WAVESABRE
TG2013

THE LIFE OF A WAVESABRE SONG

OVERVIEW



PRODUCTION

.ALS/.RPP

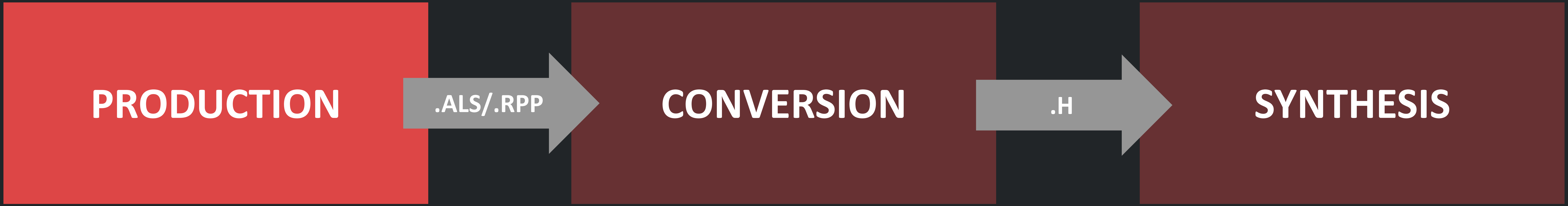
CONVERSION

.H

SYNTHESIS

THE LIFE OF A WAVESABRE SONG

PRODUCTION



PRODUCTION

.ALS/.RPP

CONVERSION

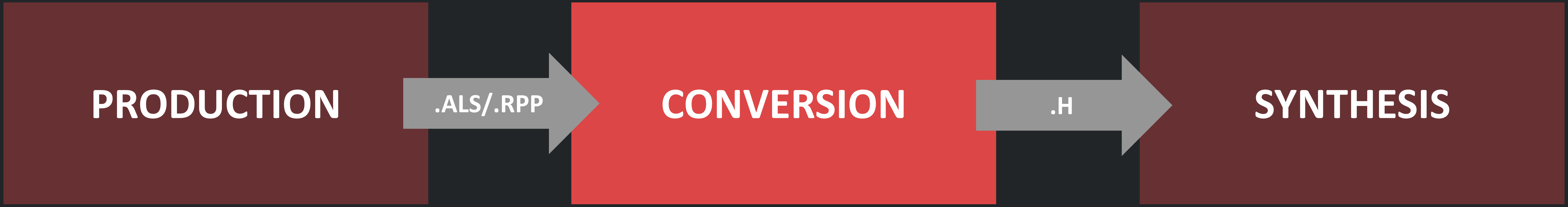
.H

SYNTHESIS

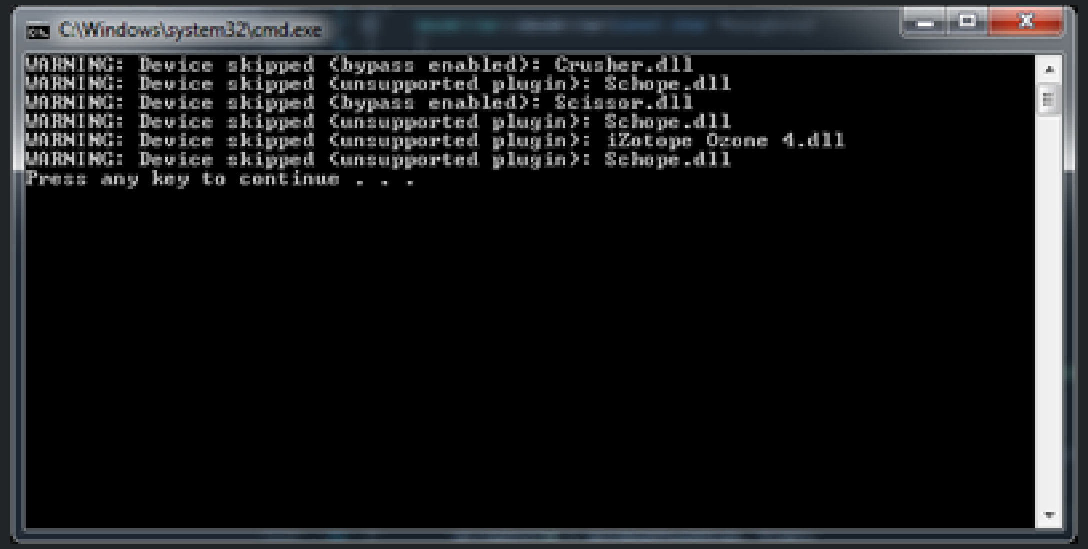
- Song is produced in DAW
- Very few limitations are actually imposed on the producer

THE LIFE OF A WAVESABRE SONG

CONVERSION



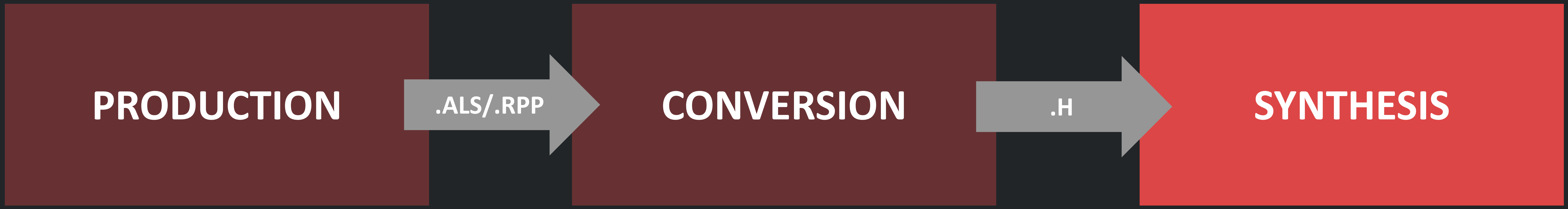
- Project file is parsed and converted to simple device graph
- Little to no sanity checks
- Data is «massaged» and serialized
- Binary exported to .h



```
WARNING: Device skipped (bypass enabled): Crusher.dll
WARNING: Device skipped (unsupported plugin): Schope.dll
WARNING: Device skipped (bypass enabled): Scissor.dll
WARNING: Device skipped (unsupported plugin): Schope.dll
WARNING: Device skipped (unsupported plugin): iZotope Ozone 4.dll
WARNING: Device skipped (unsupported plugin): Schope.dll
Press any key to continue . . .
```

THE LIFE OF A WAVESABRE SONG

SYNTHESIS



PRODUCTION

.ALS/.RPP

CONVERSION

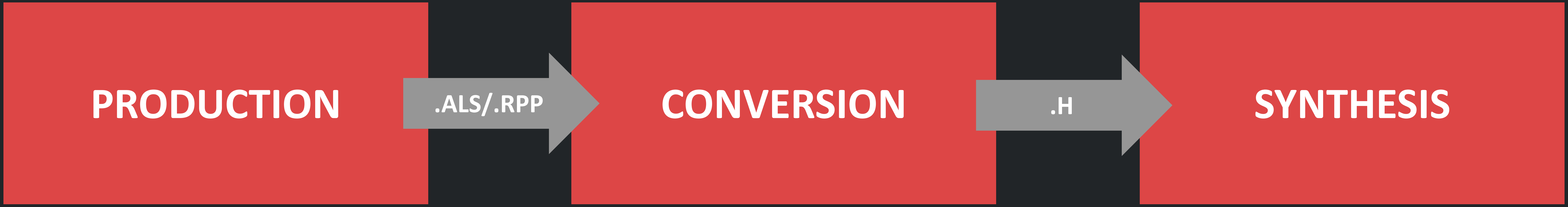
.H

SYNTHESIS

- Device graph is deserialized
- AND THEN WE BLAST LIKE THERE’S NO TOMORROW.

THE LIFE OF A WAVESABRE SONG

OVERVIEW



- In short, we have 3 components:
  - Production tools
  - Converter
  - Player

- So, let’s look at each in more detail.

PRODUCTION TOOLS

PRODUCTION TOOLS

OVERVIEW

- Could’ve gone two ways
  - Make an entire tool
    - Tried and failed
    - Takes AGES to do UI
      - Which I hate
  - Make one (or more) plugs
    - Musicians are used to plugins
    - Leverage existing DAW’s
    - Saves tons of design work

PRODUCTION TOOLS

LET’S TALK ABOUT PLUGINS

- Monolithic VST
  - Most 64k synths work this way
  - Allows system to be entirely DAW-independent
  - Track routing is very annoying
  - Very tedious to add/remove features
  - I hate this

PRODUCTION TOOLS

LET’S TALK ABOUT PLUGINS

- Modular system
  - Implement synth as a set of plugins
  - Musicians are used to working this way
  - Must be able to access track routing info
    - In this case, I parse project files
  - Locks system in to few DAW’s
  - My advantage:
    - *I’m* the musician!
    - Cater to my own needs
  - Very easy to add/remove bits and pieces
  - I don’t hate this

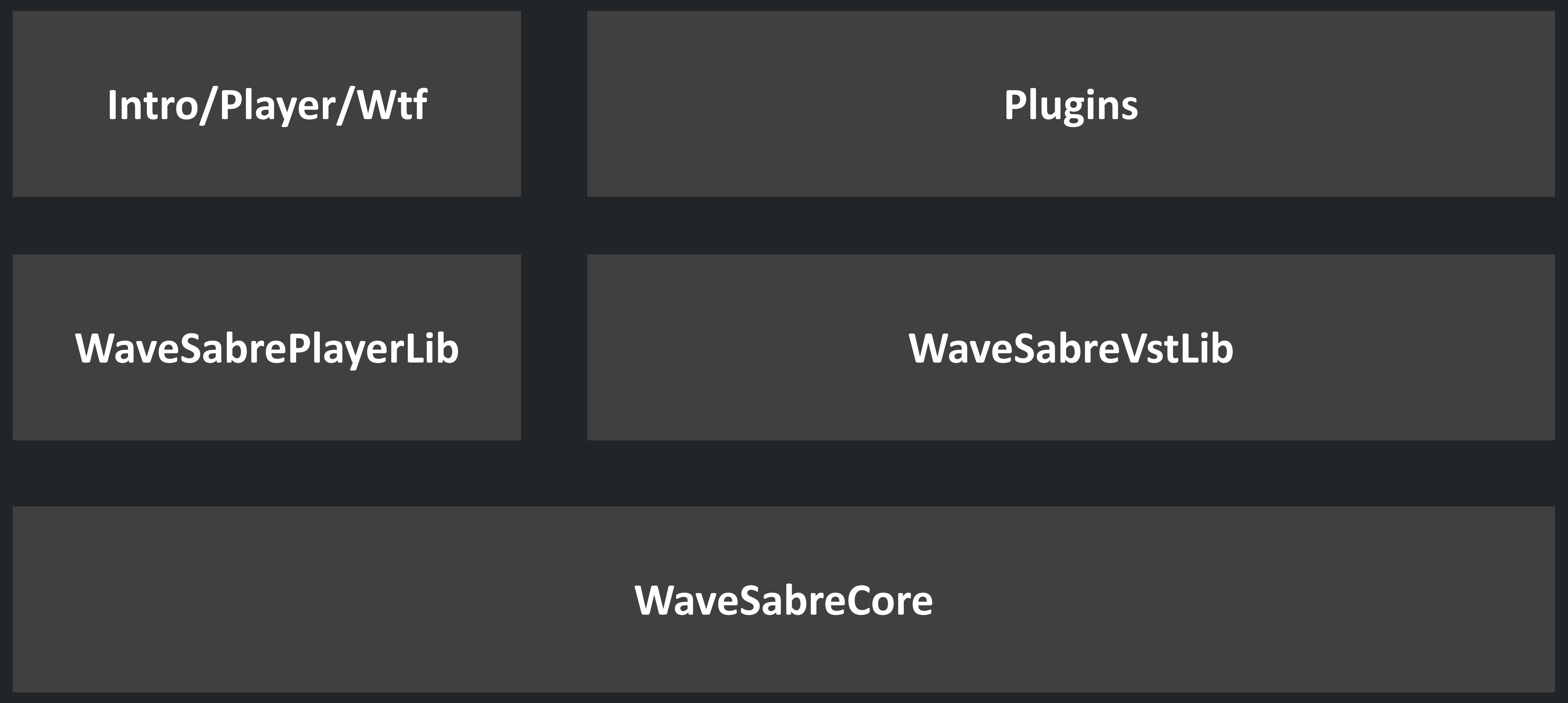
PRODUCTION TOOLS

TOOL DESIGN

- Things I wanted:
  - Use as many existing software layers as possible
    - VST sdk
    - VSTGUI
    - Ripped graphics :D
  - Make it modular
  - Lots of thin layers
  - Use the same audio processing code in both tool and player
  - More gory details to come.

PRODUCTION TOOLS

SOFTWARE STACK



Intro/Player/Wtf

Plugins

WaveSabrePlayerLib

WaveSabreVstLib

WaveSabreCore

CONVERTER

CONVERTER

MAIN IDEA

- Song project (.als, .rpp) in one end
- C/C++ header out the other

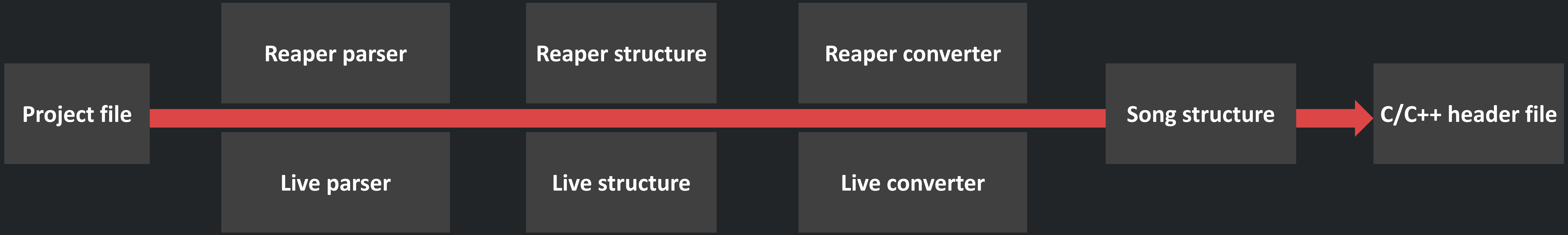
CONVERTER

THINGS I WANTED

- Command-line tool
- C#
  - Best language in the world
- Do as much work here as possible
  - Track rendering order
  - Removing «unreachable» sections of device graph
  - Removing disabled plugins
- Skip unrecognized plugs, not throw up

CONVERTER

DATA FLOW



SYNTHESIS

FINAL WORDS

FINAL WORDS

CLOSING

- This was super fun to make
- I love the sound
- It’s a system designed to last
- There are still bugs, but minor ones
  - So it seems :D

THANK YOU!
Q&A