

Audiokommunikation im Computer



Worum geht's?

- Wie kommunizieren die Teile einer DAW miteinander?
 - Host ↔ Hardware
 - Host ↔ Plugins
- Was gibt es in der Praxis zu beachten? Wo liegen Gefahren?
- Konkreter:
 - Warum ist ASIO besser als MME?
 - Warum hat Windows einen Mixer und Mac OS nicht?

Worum geht's?

Beispiel Core Audio

Generelles

HAL

Audio Units

AU-Kommunikation

AUGraphs

Folgerungen

Plugins

VST

LADSPA, DSSI, LV2

TDM/RTAS

I/O unter

Linux

Windows

XP

Vista

Bewertung

Core Audio – Generelles

- Bestandteil von OS X
- einzige Schnittstelle für alle Audio-Aufgaben (anders als bei Linux und Windows)
 - Hardware-Kommunikation (Audio und MIDI)
 - Integration von Plugins
- APIs frei verfügbar, Teil der Developer Tools

Worum geht's?

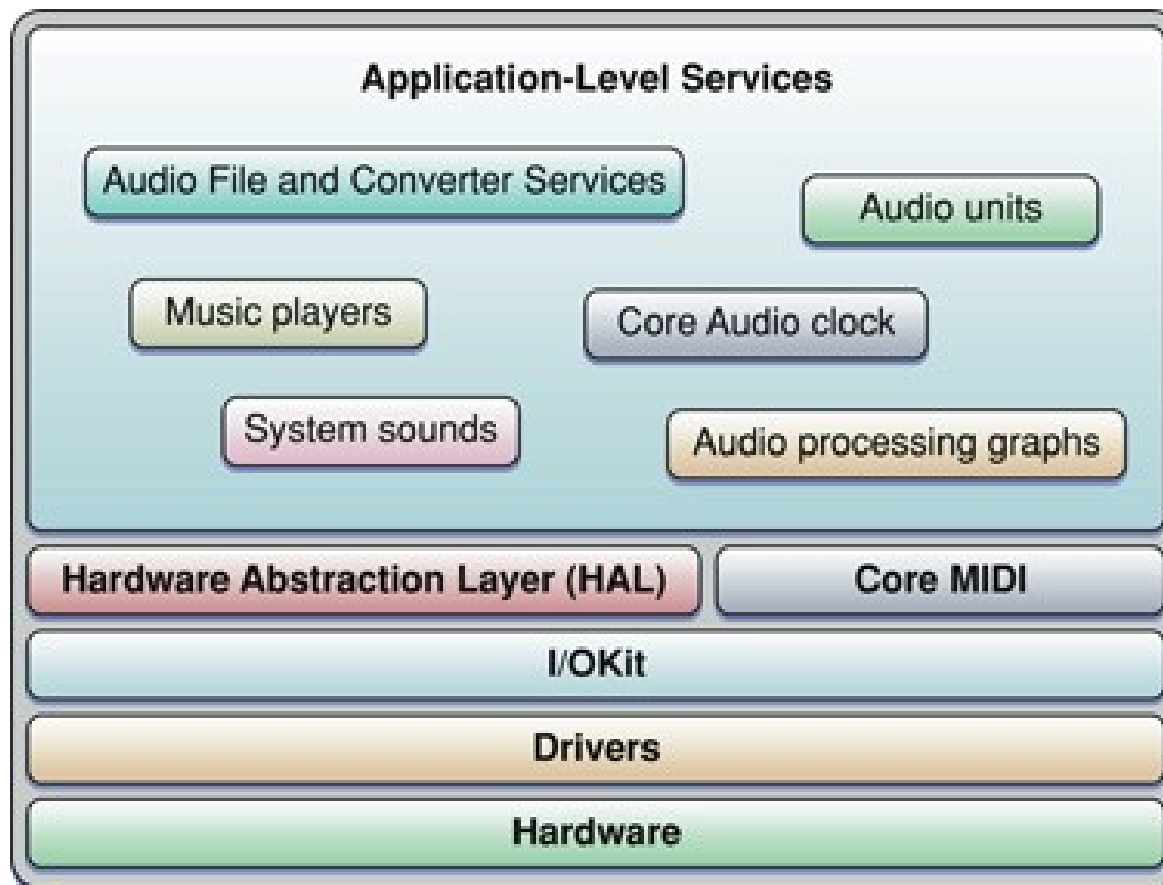
Beispiel Core Audio
Generelles
HAL
Audio Units
AU-Kommunikation
AUGraphs
Folgerungen

Plugins

I/O

Bewertung

Core Audio – Generelles



Worum geht's?

Beispiel Core Audio
Generelles
HAL
Audio Units
AU-Kommunikation
AUGraphs
Folgerungen

Plugins

I/O

Bewertung

Core Audio – Generelles

- Treiber, I/OKit und System Sounds hier nicht von Interesse
- Core Audio Clock bietet Synchronisationsmöglichkeiten mit verschiedenen TC-Formaten
- Music Player z.B. zum Abspielen von MIDI-Daten
- Audio File and Converter Services zum Lesen, Schreiben, En- bzw. Decodieren von Audiodateien

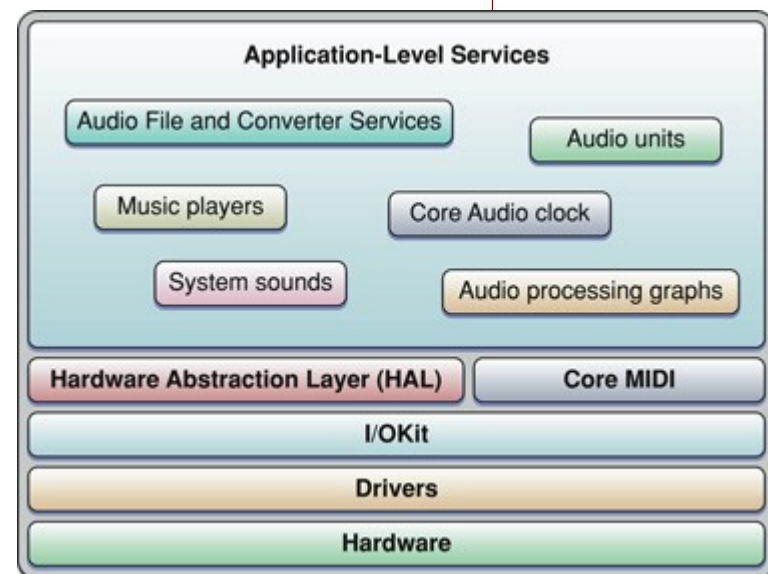
Worum geht's?

Beispiel Core Audio
Generelles
HAL
Audio Units
AU-Kommunikation
AUGraphs
Folgerungen

Plugins

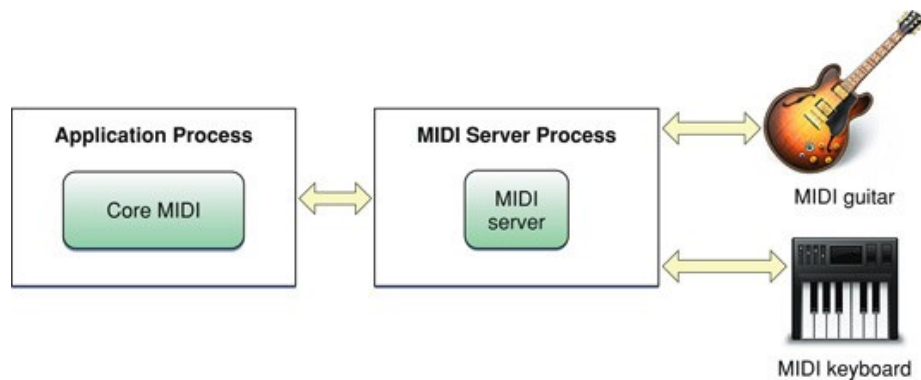
I/O

Bewertung



Core Audio – Generelles

- Core MIDI
 - Einbindung von MIDI-Geräten
 - Synchronisation
 - Zusammenfassung mehrerer Ports zu einem Gerät usw.
- MIDI Server
 - Zusätzlicher Prozess, der die einzelnen MIDI-Verbindungen überwacht und koppelt



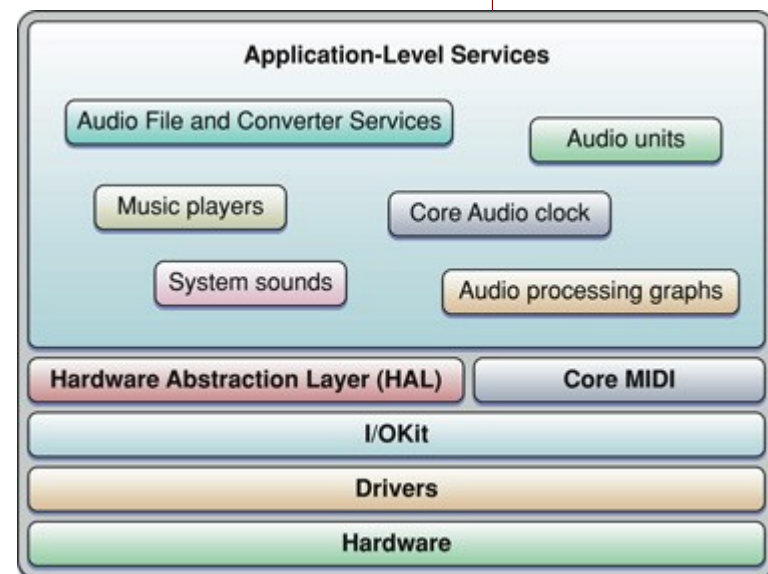
Worum geht's?

Beispiel Core Audio
 Generelles
 HAL
 Audio Units
 AU-Kommunikation
 AUGraphs
 Folgerungen

Plugins

I/O

Bewertung



Core Audio – HAL

- HAL (Hardware Abstraction Layer)
 - einheitliche Schnittstelle zwischen Hard- und Software
 - Transport des Datenstroms mit möglichst wenig Overhead und unbeeinflusstem Signal
 - Synchronisation
 - mehrere Clients pro Gerät
 - Veränderungen von Einstellungen am Gerät, z.B. Sample Rate und Lautstärke
 - ermöglicht Aggregate Devices, also die Zusammenfassung von mehreren physikalischen Geräten zu einem Logischen

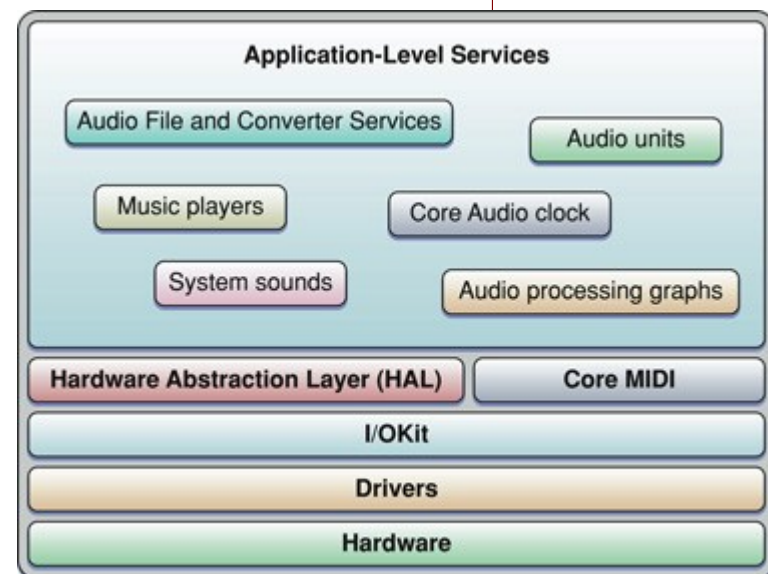
Worum geht's?

Beispiel Core Audio
Generelles
HAL
Audio Units
AU-Kommunikation
AUGraphs
Folgerungen

Plugins

I/O

Bewertung



Core Audio – Audio Units

- „kleine“ Programme, die einer Host-Software eine bestimmte Audiofunktion bieten
- selbst nicht lauffähig
- Aufgaben
 - DSP
 - Abspielen und von Dateien, Senden und Empfangen über Netzwerke, Ausgabe auf Hardware
 - Der gesamte Audiodatenfluss geschieht über die Verkettung der Audio Units, Host-Software übernimmt nur die Steuerung

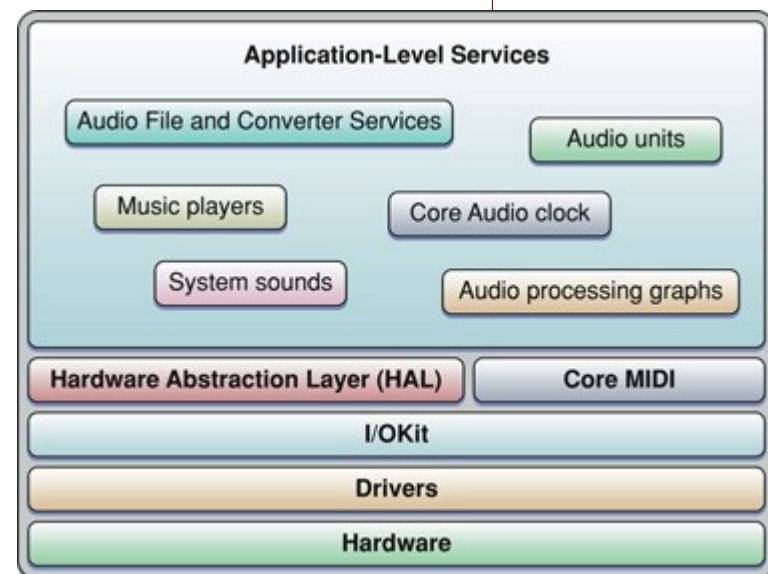
Worum geht's?

Beispiel Core Audio
Generelles
HAL
Audio Units
AU-Kommunikation
AUGraphs
Folgerungen

Plugins

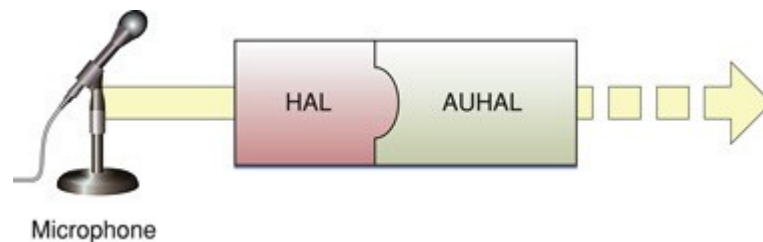
I/O

Bewertung



Core Audio – Audio Units

- Spezialisierte Unit: die AUHAL
 - Audio Unit zur Kommunikation mit der Hardware
 - weitere Abstraktion von der HAL und ihrem komplizierteren Zugriff
 - übernimmt Kanalzuweisung, Datenratenkonvertierung etc.



Worum geht's?

Beispiel Core Audio
Generelles
HAL
Audio Units
AU-Kommunikation
AUGraphs
Folgerungen

Plugins

I/O

Bewertung

Core Audio – AU-Kommunikation

- Kommunikation der Units untereinander
 - jede Unit muss nur die ihr vorgeschaltete kennen
 - Datenverarbeitung über Pull-Model: letzte Unit in der Kette wird von Host gerufen. Hat sie keine Daten mehr, fordert sie welche von der ihr vorhergehenden an usw.

Worum geht's?

Beispiel Core Audio
Generelles
HAL
Audio Units
AU-Kommunikation
AUGraphs
Folgerungen

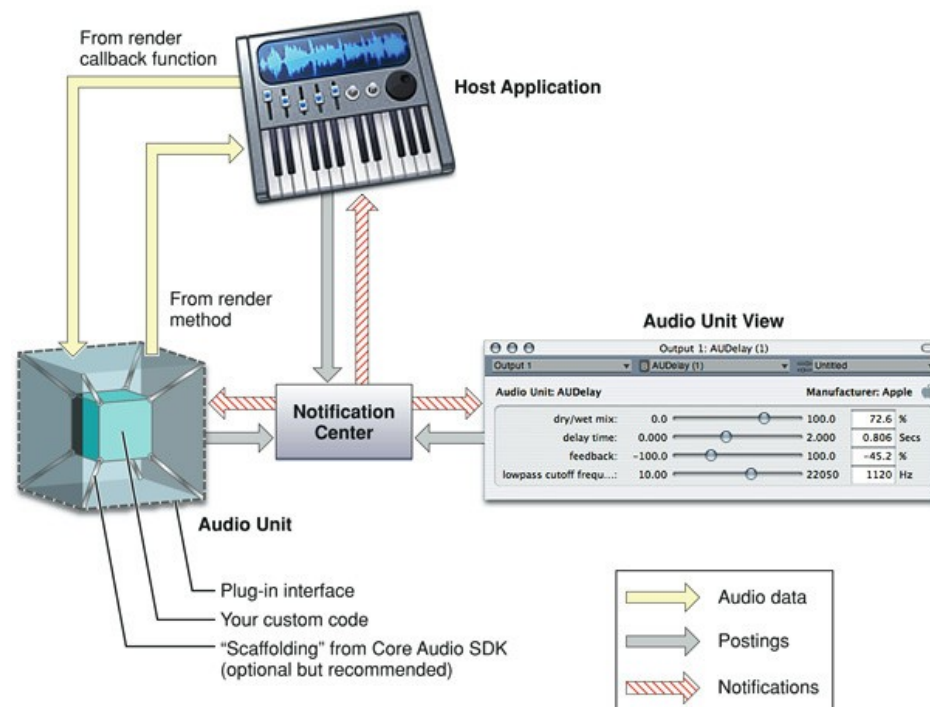
Plugins

I/O

Bewertung

Core Audio – AU-Kommunikation

- Kommunikation zwischen View, Unit und Host
 - View: die Darstellung, die der Nutzer als „das Plugin“ sieht
 - Unit kann über bestimmte Parameter gesteuert werden, so können z.B. Faktoren der DSP-Berechnung verändert werden
 - Kommunikation über sog. Notification Center: ändert ein Beteiligter einen Parameter, wird das an die anderen beiden weiter gegeben – dadurch wird z.B. Automation möglich



Worum geht's?

Beispiel Core Audio
 Generelles
 HAL
 Audio Units
 AU-Kommunikation
 AUGraphs
 Folgerungen

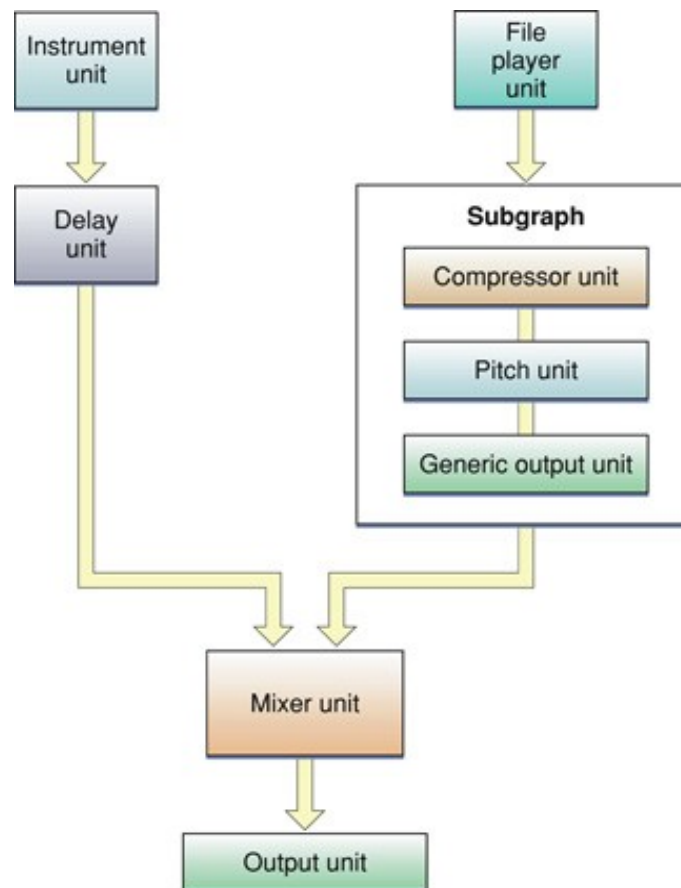
Plugins

I/O

Bewertung

Core Audio – AUGraphs

- Verkettung mehrerer Units, dann Steuerung des gesamten Verbunds möglich
- kann im laufenden Betrieb Units einbinden oder entfernen sowie Verbindungen verändern
- Bildung von Subgraphen innerhalb eines AUGraphs möglich



Worum geht's?

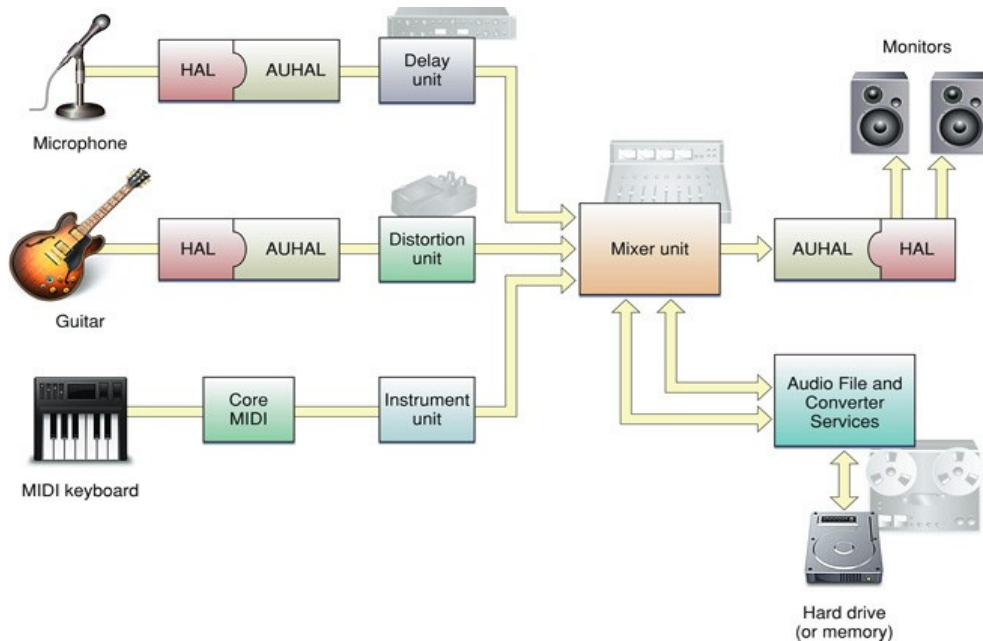
Beispiel Core Audio
 Generelles
 HAL
 Audio Units
 AU-Kommunikation
 AUGraphs
 Folgerungen

Plugins

I/O

Bewertung

Core Audio – Folgerungen



- Alle Komponenten müssen angesprochen und (beliebig) miteinander verknüpft werden können
- Viele Aufgaben kommen immer wieder vor
- Kompatibilität mit Produkten von Drittanbiern nötig
- realisiert durch definierte Funktionen und Parameter der einzelnen Komponenten (vgl. Java Interfaces)
- Was benötigt wird, sind also viele genau definierte Schnittstellen

Worum geht's?

Beispiel Core Audio
 Generelles
 HAL
 Audio Units
 AU-Kommunikation
 AUGraphs
 Folgerungen

Plugins

I/O

Bewertung

Core Audio – Folgerungen

- Kriterien für Audio-Schnittstellen
 - Entwickler
 - Handhabbarkeit (modularer Aufbau)
 - Kompatibilität
 - Zuverlässigkeit
 - Anwender
 - Geschwindigkeit – besonders bei I/O
 - Qualität der Übertragung

Worum geht's?

Beispiel Core Audio

Generelles

HAL

Audio Units

AU-Kommunikation

AUGraphs

Folgerungen

Plugins

I/O

Bewertung

Plugins

- Plugin-Schnittstellen grundsätzlich weniger abhängig vom Betriebssystem
- Trotzdem Zuordnung
 - Mac: AU
 - Windows: VST
 - Linux: LADSPA, DSSI, LV2 (geplant)
- Digidesign-Standards TDM und RTAS
- Emulation über Wrapper möglich
- Alle Konzepte sehr ähnlich (z.B. Integration grafischer Oberflächen) und grundsätzlich gleichwertig
- Generell ist die Plugin-Kommunikation unproblematischer als I/O-Kommunikation

Worum geht's?

Beispiel Core Audio

Plugins

VST

LADSPA, DSSI, LV2

TDM/RTAS

I/O

Bewertung

I/O – Linux

- ALSA
 - Ziel: professionelles Audio-Niveau für Linux-Anwendungen
- PortAudio, PulseAudio und OSS als portable Alternativen
- Qualitative Aussagen schwer zu treffen, da nicht kommerziell und unter ständiger Weiterentwicklung

Worum geht's?

Beispiel Core Audio

Plugins

I/O unter

Linux

Windows

XP

Vista

Bewertung

I/O – Windows XP

- Treiberformat: WDM
- Schwachstelle: WDM Audio beinhaltet K Mixer im Betriebssystemkern
 - Aufgaben
 - Konvertieren und Mischen von Audio-Streams
 - einfaches DSP
 - Probleme
 - zusätzliche Latenz (ca. 30ms)
 - u. U. Veränderungen im Bitmuster
 - versucht, alle Audio-Streams zu mischen (z.B. auch codierte Streams, die über Digitalausgänge ausgegeben werden sollen)

Worum geht's?

Beispiel Core Audio

Plugins

I/O unter
Linux
Windows
XP
Vista

Bewertung

I/O – Windows XP

- MME / Wave API
 - noch aus Windows 3.11-Zeiten
 - Probleme durch KMixer
- DirectSound
 - Teil des DirectX-Frameworks
 - eingeführt mit Windows 95
 - geringere Latenz als MME
 - hohe Kompatibilität
 - ebenfalls Probleme durch KMixer

Worum geht's?

Beispiel Core Audio

Plugins

I/O unter
Linux
Windows
XP
Vista

Bewertung

I/O – Windows XP

- ASIO
 - Audio Stream Input/Output
 - von Steinberg entwickeltes Protokoll
 - Ziel: direktere Kommunikation zwischen Soft- und Hardware (ab Version 2 z.B. Steuerung des Input-Monitorings der Hardware)
 - umgeht den KMixer: geringere Latenz und bit-identischer Output
 - Möglichkeiten zur Synchronisation, Mehrkanaleigenschaften
- GSIF
 - Giga Sampler Interface
 - Qualität ähnlich ASIO, umgeht ebenfalls den KMixer
 - nur Wiedergabefunktionen

Worum geht's?

Beispiel Core Audio

Plugins

I/O unter

Linux

Windows

XP

Vista

Bewertung

I/O – Windows XP

- WDM Kernel Streaming
 - direkter Zugriff der Software auf Audiotreiber
 - Umgehen des KMixers
 - geringere Latenzen
 - Umgehen der Windows-Lautstärkeregelung
 - In einigen Anwendungen verwirklicht, außerdem Plugins für einige Audio-Player

Worum geht's?

Beispiel Core Audio

Plugins

I/O unter
Linux
Windows
XP
Vista

Bewertung

I/O – Windows Vista

- Audio-Sektion vollständig überarbeitet
 - Möglichkeit zur Priorisierung von zeitkritischen Prozessen
 - WDM Audio inkl. KMixer fällt weg
 - Windows Audio Session API
 - Shared Mode
 - für Standard-Anwendungen
 - Emulationen für DirectSound und MME
 - Betriebssystem-Mixer (ähnlich KMixer) inkl. Resampling usw.
 - Exclusive Mode
 - Konzept ähnlich Kernel Streaming
 - Umgeht den System-Mixer
 - Neues Treibermodell: WaveRT
 - verspricht niedrige Latenzen und geringen CPU-Bedarf

Worum geht's?

Beispiel Core Audio

Plugins

I/O unter
Linux
Windows
XP
Vista

Bewertung

Bewertung

- OS X
 - einfache Konfiguration
 - problemlos
 - teilweise fehlende Features wie Mixer
- Linux
 - professionelles Niveau, wenn man weiß, was man tut
- Windows XP
 - Pro Audio wurde von Microsoft völlig vernachlässigt, möglich erst durch „Aufbohren“ der Technologie durch Dritthersteller
- Windows Vista
 - sehr hohes Potential, offensichtlich hat man aus den Fehlern bei XP gelernt
 - Umsetzung durch Hersteller steht noch aus

Worum geht's?

Beispiel Core Audio

Plugins

I/O

Bewertung

Worum geht's?

- Wie kommunizieren die Teile einer DAW miteinander?
 - Host ↔ Hardware
 - Host ↔ Plugins
- Was gibt es in der Praxis zu beachten? Wo liegen Gefahren?
- Konkreter:
 - Warum ist ASIO besser als MME?
 - Warum hat Windows einen Mixer und Mac OS nicht?

Worum geht's?

Beispiel Core Audio

Plugins

I/O

Bewertung

Quellen

- Core Audio
 - Core Audio Overview
 - <http://developer.apple.com/documentation/MusicAudio/Conceptual/CoreAudioOverview/>
 - Audio and MIDI on Mac OS X
 - <http://developer.apple.com/audio/pdf/coreaudio.pdf>
 - Audio Unit Programming Guide
 - <http://developer.apple.com/documentation/MusicAudio/Conceptual/AudioUnitProgrammingGuide>
- Windows
 - www.staudio.de/kb/english/drivers/
 - en.wikipedia.org/wiki/Windows_audio_components
 - en.wikipedia.org/wiki/Technical_features_new_to_Windows_Vista
- Generell
 - Wikipedia-Einträge zu den entsprechenden Produktnamen