



Media Broadcasting Solutions . SpaceCom . Remote Sensing Technology

VCS Aktiengesellschaft **Integration komplexer Produktionswerkzeuge in vernetzte Produktions- und Sendesysteme**

▶ Autor: Kai Uwe Kaup

© VCS Aktiengesellschaft
Borgmannstrasse 2, 44894 Bochum, Germany
www.vcs.de

© VCS Aktiengesellschaft – subject to change without notice

1



Media Broadcasting Solutions . SpaceCom . Remote Sensing Technology

Integration komplexer Produktionswerkzeuge in vernetzte Produktions- und Sendesysteme

Übersicht

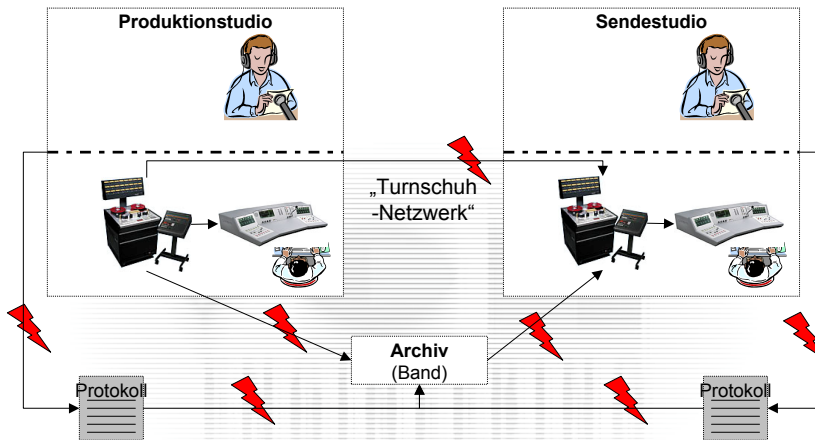
- ▶ Vom Tonband zur vernetzen Produktion
 - ▶ Historie, Gegenwart und Ausblick
- ▶ Aspekte der Integration
 - ▶ Formate und Funktionalität
- ▶ Beispiele und Ausblick

© VCS Aktiengesellschaft – subject to change without notice

2



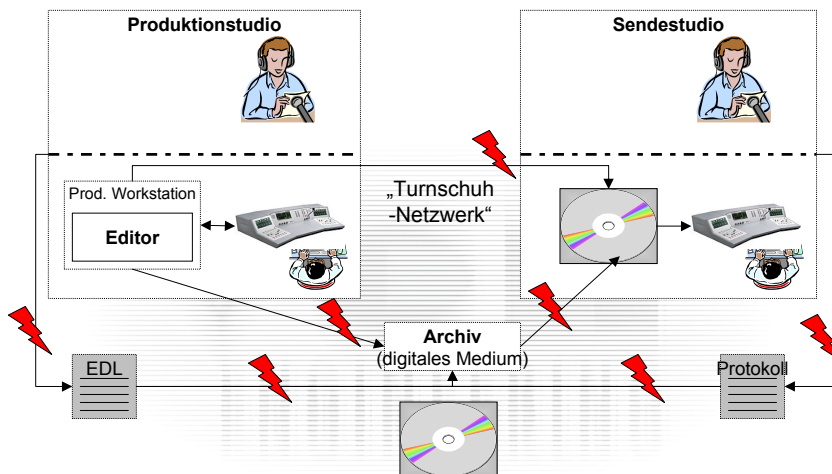
Entwicklung – Vom Tonband zur vernetzten Produktion (1)



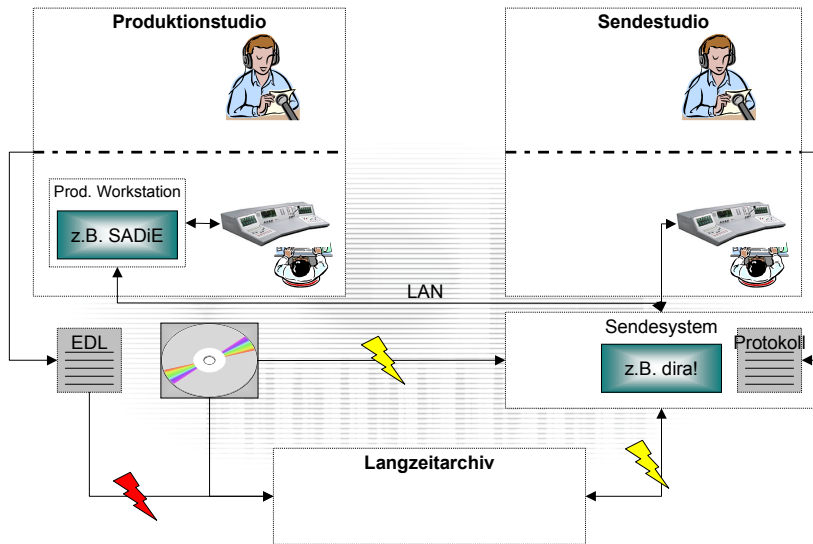
3



Entwicklung – Vom Tonband zur vernetzten Produktion (2)



4



Produktionseditoren (z.B. SADiE)

- ▶ optimiert auf “Craft” Editing
- ▶ ausgereifte, hocheffiziente Produkte
- ▶ i.d.R. werden als Host vernetzbare PC-Plattformen eingesetzt

Sendesysteme (z.B. dira!)

- ▶ optimiert auf Zuverlässigkeit und redaktionellen Workflow
- ▶ ausgereifte, hocheffiziente Produkte
- ▶ stellen Audioeditoren für den Desktop zur Verfügung, jedoch kein “craft editing”



Anforderungen – Aspekte der Integration

Projekt- und Audioformate

- ▶ Wave und Wave64
- ▶ native Projektformate, AES31-3, AAF

Metadaten

- ▶ Nachhalten der Verwendung von Quellmaterial für die spätere Protokollierung (Genealogy)
- ▶ Metadaten können bei einer Vollintegration auch beim "Roundtrip" durch den Produktionseditor erhalten werden
- ▶ Es können produktions-bezogene Metadaten hinzugefügt werden.



7



Anforderungen – WAVE64

WAVE64 = Sony Digital Pictures (TM)

- ▶ Ursprünglich von Sonic Foundry ins Leben gerufen
- ▶ Unterstützung von Mehrkanal-Produktionen
- ▶ Praktisch keine Größenbeschränkung
- ▶ Marker Chunk
- ▶ Problemlose Unterstützung von hersteller-eigenen Chunks durch Verwendung von weltweit eindeutige GUIDs
- ▶ Von VCS um BWF ähnlichen Hüllkurven-Chunk erweitert, um das schnelle Laden in Editoren zu ermöglichen



8



WAVE64 – Format Definition

„The GUID STATIC_AVRIF (defined below)
 A LONGLONG giving the length
 The GUID GUID_W64_WAVE (defined below)

RIFF64 headers are a GUID followed by a LONGLONG cbSize.

RIFF64 chunks are byte-aligned on 8 byte boundaries, but their chunk lengths do not include the pad if it was necessary. RIFF64 chunk headers are a GUID followed by a DWORD cbSize.

To save W64 markers/regions, a RIFF64 header is written, followed by the marker count as follows:

The GUID GUID_MARKER (defined below)
 A LONGLONG header size, always is (sizeof(GUID) + sizeof(LONGLONG))
 The marker count, as a DWORD value

Individual W64 markers/regions are saved as follows:

DWORD dwID
 LONGLONG ccPosition
 LONGLONG ccLength // -1 for markers
 DWORD cbNameSize
 WCHAR string (Unicode) containing the marker name text

Summary info looks like it is saved with a RIFF64 header with a GUID of GUID_SUMMARYLIST, followed by a DWORD summary item count. Each summary item is saved similar as it is for WAV, with the same FOURCC code.“

Heidi Ilgaray (HeidiI@SonicFoundry.com)



WAVE64 – GUID Definitionen (1)

GUID definitions:

/// These are standard GUIDs to identify RIFF64 chunks and LIST chunks

```
static const GUID STATIC_AVRIF =
{ 0x66666972L, 0x912E, 0x11CF, { 0xA5, 0xD6, 0x28, 0xDB, 0x04, 0xc1, 0x00,
0x00 } };
```

```
static const GUID STATIC_AVLIST =
{ 0x7473696CL, 0x912F, 0x11CF, { 0xA5, 0xD6, 0x28, 0xDB, 0x04, 0xC1, 0x00,
0x00 } };
```

```
// 'wave' chunk
// {65766177-ACF3-11d3-8CD1-00C04F8EDB8A}
static const GUID GUID_W64_WAVE =
{ 0x65766177, 0xacf3, 0x11d3, { 0x8c, 0xd1, 0x0, 0xc0, 0x4f, 0x8e, 0xdb,
0x8a } };
```

```
// 'fmt' chunk
// {20746D66-ACF3-11d3-8CD1-00C04F8EDB8A}
static const GUID GUID_W64_FMT =
{ 0x20746d66, 0xacf3, 0x11d3, { 0x8c, 0xd1, 0x0, 0xc0, 0x4f, 0x8e, 0xdb,
0x8a } };
```

```
// 'fact' chunk
// {74636166-ACF3-11d3-8CD1-00C04F8EDB8A}
static const GUID GUID_W64_FACT =
{ 0x74636166, 0xacf3, 0x11d3, { 0x8c, 0xd1, 0x0, 0xc0, 0x4f, 0x8e, 0xdb,
0x8a } };
```



WAVE64 – GUID Definitionen (2)

GUID definitions:

```

/// continued

// 'data' chunk
// {61746164-ACF3-11d3-8CD1-00C04F8EDB8A}
static const GUID GUID_W64_DATA =
{ 0x61746164, 0xacf3, 0x11d3, { 0x8c, 0xd1, 0x0, 0xc0, 0x4f, 0x8e, 0xdb,
0x8a } };

// 'junk' chunk for padding
static const GUID GUID_W64_JUNK =
{ 0x6b6e756a, 0xacf3, 0x11d3, { 0x8c, 0xd1, 0x0, 0xc0, 0x4f, 0x8e, 0xdb,
0x8a } };

// {90020EF4-392D-11d2-86C7-00C04F8EDB8A}
static const GUID GUID_MARKERLIST =
{ 0x90020ef4, 0x392d, 0x11d2, { 0x86, 0xc7, 0x0, 0xc0, 0x4f, 0x8e, 0xdb,
0x8a } };

// {ABF76256-392D-11d2-86C7-00C04F8EDB8A}
static const GUID GUID_MARKER =
{ 0xabf76256, 0x392d, 0x11d2, { 0x86, 0xc7, 0x0, 0xc0, 0x4f, 0x8e, 0xdb,
0x8a } };

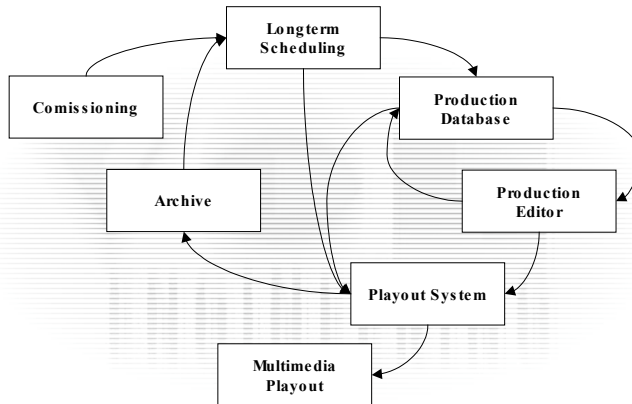
// {925F94BC-525A-11d2-86DC-00C04F8EDB8A}
static const GUID GUID_SUMMARYLIST =
{ 0x925f94bc, 0x525a, 0x11d2, { 0x86, 0xdc, 0x0, 0xc0, 0x4f, 0x8e, 0xdb,
0x8a } };

```



Anforderungen – Metadatenaustausch

Example for exchange of material and metadata





Broadcast 1 (1992)

Interview	Clip	Interview	Music
-----------	------	-----------	-------

Broadcast 2 (2002)

Live sound	Clip	Music	Interview
------------	------	-------	-----------

Broadcast 3 (2006)

Interview	Clip	Music	Interview
-----------	------	-------	-----------

From: Integrated Metadata in the Broadcast Environment.
Bull, Joe (SADiE); Kaup, K. U. (VCS AG) (2004).
Figure [1]. An example of Genealogy



13



Produktionsdatenbanken

- ▶ Speichern von “work in progress” (Projekten)
 - ▶ check-in / check-out Funktionalität
- ▶ ermöglicht die Weiterbearbeitung auf unterschiedlichen Produktions-Workstations
- ▶ Quellmaterial kann sowohl aus dem Sendesystem (z.B. Jingles), als auch aus dem digitalen Langzeitarchiv einfach in die Produktion eingebracht werden
- ▶ Anwender können auf bekannte Werkzeuge zurückgreifen, die komfortablen Zugang zum Audiomaterial und den Metadaten bieten (Datenbank-Client)
- ▶ Es können die gleichen Zuspeler (Sendeabwicklung, CartPlayer), wie im Sendesystem für die Produktion verwendet werden.



14



Ausblick – SADiE BB2 & dira!Highlander

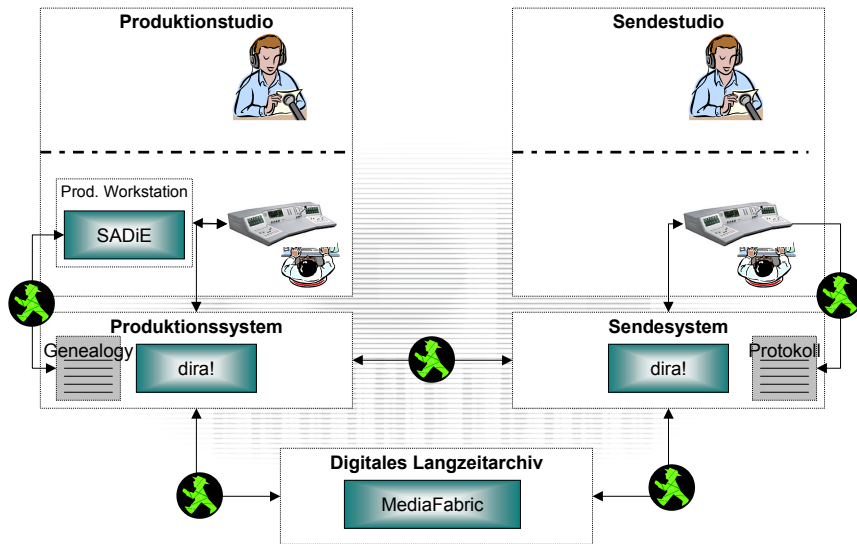
15



Ausblick – Beispiel für Produktionsmetadaten

Take data card					
Titel		Autor		Planned Duration	
Subtitle		Herkunft		Einsetz	
Broadcast Title		Producer		First Broadcasted	
Prod No	Production Type	Production Team		Last Broadcasted	
Production Date	Fee Approved	Department		Länge	
Bearb.-zeit	ZZ	Network		Expires	
Work Status		Folder		Created by	
Account Number	Asset ID	Ext. Archive0	ID	Changed by	
<input type="button" value="OK"/> <input type="button" value="Cancel"/> <input type="button" value="Apply"/>					

16



Vielen Dank für Ihre Aufmerksamkeit!

Kontakt:

- ▶ Joe Bull, joe.bull@sadie.com
- ▶ Kai Uwe Kaup, kai.kaup@vcs.de

