From FIMS to MCMA
2009 – 2018…2020

Bloomberg, New-York,12-13 February 2020

Jean-Pierre Evain, EBU

Once upon a time, FIMS started in 2009!!

- **Scope**: a common Framework for Interoperable Media Services using a Service Oriented Architecture (SOA) for use in broadcast, production, post-production, media distribution, and media archive applications.
- **Why**: vendor-neutral, loose-coupling, interoperability, scalability, workflow agility, interchangeability and reusability of media specific services.
- **What**: SOAP interfaces with a preliminary view on REST
- **Initial Media Services**: Capture, Transform, and Transfer
- Followed by Automatic Metadata Extraction, Quality Control, Repository Management
- Latest version of the FIMS specifications: EBU Tech 3356 v1.3.1 in October 2017 with code as open-source on github

An AMWA-EBU project sponsored by Sony and IBM

Building on a window of opportunity opened at IBC 2009

FIMS Request for Technology: phase 1 scope

- Phase 1 focuses on the framework
- Define a few services to test framework (e.g. Ingest, Transcode)

FIMS key milestones, moving into MCMA

The original FIMS reference model

The resource manager concept

Long process job lifetime

Capture service and time constraints

The FIMS resource oriented data model

IBC 2011

DEMONSTRATORS
IBM
SONY
avid.com
CiNegy
CUBE-TEC International
radiant GRID

SUPPORTERS
harmonic
HARRIS
TIXEL
ISILON
RED BEE
metaglue
Rai
BROADCAST INTERNATIONAL
mesclado
marquis
glookast
solving the workflow puzzle

Primary Contributors

FIMS 2011 Demo

Existing services can be used with 'Adaptors'

FIMS @ IBC 2012 – IBC Judges Prize

DEMO SPONSORS

Bloomberg TELEVISION

CUBE-TEC International

IBM

SONY

Triskel Software Development Experts

FIMS SUPPORTERS

DVS

Cinegy

FLYING EYE

EMC² ISILON

harmonic

HARRIS

mesclado

メタフロンティア

radiant GRID

Rai

TIXEL

VSN VIDEO STREAM NETWORKS

FIMS BUSINESS BOARD 2014

FIMS MEMBERS 2014

FIMS IBC 2016

Organizations Using FIMS - Fra
https://fims.tv/FIMS_implementers.shtml#Manufacturers_and_Integrators

SOAP: Planned In Progress Completed REST: Planned In Progress Completed

Service:	Capture	Transfer	Transform	Repository	Quality Analysis
Aspera Richard Heitmann		✓	✓		
BCE Sascha Quillet Michael Backes		→		→	
Cognizant John Footen	✓	✓	✓	✓	✓
Cube-Tec International Jorg Houpert		✓	✓		✓
Dalet Stephane Guez Matthieu Fasani	✓	✓ ✓	→ →	○ ○	○ ○
EMC^2 Isilon Storage Division Robert McNeal				○ ○	
File Catalyst Elton Carneiro Alan Atkinson		✓ ○			
Glookast Joost Rovers	✓ →	→ →	→ →		

- Participation Agreement (pdf)
- FIMS Terms of Reference (pdf)
- More Resources

mercredi, 12 février 2020
FRA 10:55 12.02.2020

Organizations Using FIMS - Fra
https://fims.tv/FIMS_implementers.shtml#Manufacturers_and_Integrators

Service:	Capture	Transfer	Transform	Repository	Quality Analysis
Harmonic Chris Braehler					
Hitachi Data Systems Steven Sonnenberg Jay Yogeshwar					
IBM Praveen Dharmavaram Peter Guglielmino Frank Schaffa Ulrich Voigt					
Imagine Communications Vincent Moriaux Huw Dymond					
InfoLibrarian Corporation B. Brewer					
Institut für Rundfunktechnik [IRT] elser@irt.de Peter Altendorf					
Interra Systems Vandan Kaushik					
IPV Limited tscotcher@ipv.com					

mercredi, 12 février 2020
FRA 10:56 12.02.2020

Organizations Using FIMS - Fra
https://fims.tv/FIMS_implementers.shtml#Manufacturers_and_Integrators

Limecraft Dieter Vanrijsselbergen					
Marquis Broadcast Simon Fearn					
Service:	**Capture**	**Transfer**	**Transform**	**Repository**	**Quality Analysis**
Mikros Image Thomas Eskénazi					
Quantel Richard Cartwright					
QStar Technologies massimiliano.finotti@qstar.com					
Rohde & Schwarz DVS GmbH Michael Hackl					
SGT Yves BRETON					
Signiant Ian Hamilton					
SONY Toshiaki Kojima					

FRA 10:57 12.02.2020

Organizations Using FIMS - Fra
https://fims.tv/FIMS_implementers.shtml#Manufacturers_and_Integrators

Tektronix Andrew Scott Malateshgouda Karegoudar					
Telestream Paul Turner					
TIXEL Lars E. Fuerst					
Triskel Loic Barbou					
Service:	**Capture**	**Transfer**	**Transform**	**Repository**	**Quality Analysis**
Vidcheck Simon Begent					
Vidispine Erik Åhlin Isaak Jonsson					
Vizrt Ximena Araneda Marcus Bergner Arun Krishnaswamy					
VSN Innovation & Media Solutions Manuel Escribano					

FRA 10:57 12.02.2020

Organizations Using FIMS - Fra
https://fims.tv/FIMS_implementers.shtml#Manufacturers_and_Integrators

Wavelet Beam Dirk Hildebrandt					
Xytech Richard Gallagher					
Yangaroo Richard Klosa					
Service:	**Capture**	**Transfer**	**Transform**	**Repository**	**Quality Analysis**

SOAP: Planned In Progress Completed **REST:** Planned In Progress Completed

Back to Top

Media Companies

				Implementing or operating
			Asking for FIMS in Request for Technology, Calls for tender	
		Testing, Proof of concept		
	Investigating			
A&E Use FIMS to standardize asset metadata across systems. Plan to leverage FIMS repository to simplify integration between content management systems. Adam Weiss	✔	✔	✔	✔
BBC	✔	✔		✔

FRA 10:58 12.02.2020

Organizations Using FIMS - Fra
https://fims.tv/FIMS_implementers.shtml#Manufacturers_and_Integrators

BBC Use FIMS to standardize asset metadata across systems. Using FIMS to allow interoperability between internal CMSes (such as our News Production system - Jupiter) and our media movement solutions, allowing us to deliver files between sites and to cloud services. Using services purely within the Media Movement domain to allow non-vendor specific transfers. Investigating the use of FIMS for storage. Charlie Brown Ricardo Oliveira	✔	✔		✔
Bloomberg Implemented Transfer, Transform, Repository interfaces globally for content production and distribution. Design and implemented a workflow engine compatible with FIMS. Standardized Media Asset representation across the organization by leveraging FIMS. Roman Mackiewicz	✔	✔	✔	✔
CBC Has produced a FIMS pilot in order to demonstrate benefits of FIMS and to allow CBC to learn how service-based technology can aid in media workflows. Anthony Caruso	✔	✔		
Deutsche Welle Investigating usage of FIMS to simplify integration between content management systems, transfer, QC and transform services Hubert Czaja	✔	✔		
HBO Investigating FIMS repository implementation. Matt Rose	✔			

FRA 10:59 12.02.2020

PBS Implementing Transfer, Transform as well as Repository interfaces to unify the media systems supporting the production and delivery of content behind a standardized API. Plans to implement the new QC specification and to continue the expansion of its existing implementation. Sylvain Nazef				✔
RAI FIMS Transform and Transfer services have been implemented. They are used in theT3 (Transition To Tapeless) project, the enterprise tool to manage integrated production workflows at RAI. The Repository service is under testing. It will be used to interconnect T3 and the legacy archive digitization system. Giorgio Dimino	✔	✔	✔	✔
RTP Pedro Landeiro		✔		
Turner Broadcasting Dan Shockley	✔			
NRK Investigating FIMS implementation of transfer, QC, transform and repositories. Plans to utilize the standard in future versions of the infrastructure. Tormod Værvågen	✔	✔	✔	
Mediacorp Kelvin Ma Jing	✔			
Hitachi Implemented REST based Repository Services Adaptor for the Hitachi Content Platform - Object Store. Extending this to Block and File Platforms	✔	✔		✔

FRA 11:00 12.02.2020

From SOAP to REST, from spec to github

- SOAP allows defining precise technical specifications, which helps testing conformance.
- However, even if several FIMS members were providing solutions in SOAP and REST, in became clear that SOAP was too complex.
- People were asking for REST.
- And some vendors went back to proprietary solutions or were subject to significant reorganisation…
- On that basis, we changed the FIMS paradigm for what would become the development ground for MCMA sharing code on github.

FIMS 2017

First AWS workflow @NAB

Now the MCMA conformance workflow!

Who?
Bloomberg, Triskel, Glookast, Dalet, Myers
Supported by EBU

In 2018, FIMS becomes MCMA

An EBU project

REST-based!

Who?
Bloomberg, Triskel, NHK, Glookast, EBU

Cloud AI big bang – an opportunity?!

M for 'Media': our core business, understanding long jobs and processes and **building on lessons learnt from FIMS**

C for 'Cloud': understanding the value of serverless deployment and surfing on the hype around fast deploying cloud platforms

M for '(micro-)service': people who didn't get the notion of service ''seem'' to grasp of micro-services associated with cloud platforms (AWS, Azure, Google…)

A for 'architecture' but could also be 'A' for AI, surfing on the AI hype also core to micro-services proposed on cloud platforms

Speech-to-Speech demo

Jean-Pierre Evain, EBU

Actually more than speech-to-speech

MCMA Multi-cloud AI workflow

s3.amazonaws.com/ch.ebu.mcma.ai-wf.us-east-1.nyc.website/index.html#/monitor

Apps | Cours Franc suisse... | Mail – evain@ebu.ch | Société Générale | Maintenant à la TV,... | Google | EBU WebEx | France Inter en dire... | Portail Orange | Off... | Gmail

_ORF_00_18_09_conv.mp4
_ORF_00_18_09_conv.mp4
_ORF_00_18_09_conv.mp4
_ORF_00_18_09_conv.mp4
_ORF_00_18_09_conv.mp4
_ORF_00_18_09_conv.mp4

Title: jpenyc

Description: jpenyc

File Name: 2015_GF_ORF_00_18_09_conv.m

Speech to Speech:

AI Workflow Status: Completed

AWS | Azure | Google

Transcript | STT Benchmarking | Translation | Speech to Speech | Celebrities

"That is what we love "way. Love about Eurovision. The whole world's participating and on to make sure the world remains the a great place that it is, we're is. We're hosting this song contest as a green events. Millions of events, millions, our families and friends in and 45 countries watching together for one big party like this very big family of 10,000 party people here in Vienna. Vienna be insured on. And I've got some

FRA 15:48 12.02.2020

MCMA from scratch

Jean-Pierre Evain, EBU

Initial personal developer knowledge :--)

- Not much outside data processing (xml, json, rdf)
- Some Java-EE on Eclipse for proof of concepts using local apache servers
- NodesJs and surrounding framework: none
- Terraform, gradle: none
- Angular: none (and desperatly still none)
- Github: more like an encounter of the third kind
- AWS: none

In other words: not very encouraging as a start :--)

But I understood that I would have to do it if I wanted it

With a “little” help of our Joost

- Basic developer framework setup
  - Java JRE or JDK
  - NodeJs (nvm may not work on some versions of Windows)
  - Terraform
  - Github -> set Gradle settings with platforms’ credentials, clone project, deploy and run website

- AWS Console key services
  - S3 to monitor and manage the storage buckets
  - ECS to monitor tasks of docker services (e.g. sttbenchmarking)
  - Step functions to visualise workflows
  - Cloudwatch to visualise logs
    - Workflow step logs
    - Service log: e.g. ibc-aws-ai-service-worker

- Step-by-step deployment of a basic AWS Polly speech-to-text service
  - Just look at the code for this specific purpose – practical and effective – no general considerations
    - How to develop step function and customise lambdas
    - How to define a workflow and setup the deployment environment
    - Some introductory debugging hints

A video was recorded but not needed if you attended the webex

Get a good editor and dive into code

- Github: create a new branch for your own developments
- Define a first step function to initiate a job (lambda function)
  - Code is 90% common. Easy to customise a new job call
  - Important to well identify inputs and output buckets. This is through what lambdas will access/exchange resources.
- Define your lambda e.g. to access AI tools
  - This is where customisation takes place. See the use of ffmpeg and STT-Benchmarking (Docker image on ECS).
- Define a step function to process service outputs
  - Collect data from services, process as needed, store results in the right S3 bucket and move to the step and job
- Update gradle settings and build,
- Update workflow definitions for terraform deployment
- Gradle deploy
- Access to the resulting website URL in the deployment region, select content and run workflow
- Use the AWS console for monitoring and debugging (good code documentation helps)

Lessons learnt

- It is necessary to identify simple loosely coupled jobs, not only for accessing AI services.
- An apparently simple task may require several chained steps and jobs.
- Tricky but effective: JS promisify
- Deploying a Docker image (STT Benchmarking) on AWS ECS – effective in terms of cost and processing
- In comparison to the rest, Angular doesn’t seem like easy to improve the front-end (I didn’t touch it!)
- But, more than anything, many things I didn’t need to look at code running transparently in the background (service registry, authentication, etc.)

The real challenges are not about MCMA

- Manage files and data -> management of buckets, association with BMContent, exposure as web resource

- Interrupt, resume, separate workflows steps e.g. to edit STT
  - *To be investigated… Currently only inserting corrected files in the workflow for PoC purposes*

- STT editors are missing -> e.g. upgraded subtitle editors?
  - *Subtitle: fragmented sentences when translator needs full sentences…*

- Polly / ssml – not easy – more time consuming than traditional dubbing unless if you just want an idea about what is being said / subject /topic – then go for simple Polly text-to-speech.
  - *Markup to introduce "pauses" (e.g. derived from STT or SRT timing but not accurate) or modulate speech speed for sync purposes – not available in all regions*
  - *Other problem: not enough voices in various languages.*
  - *Currently monochord voice-over à la Polish*