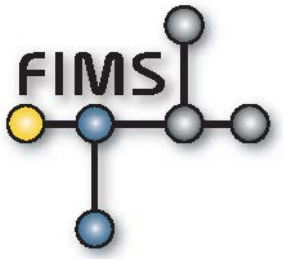
EBU
OPERATING EUROVISION AND EURORADIO

FIMS: THE FRAMEWORK FOR INTEROPERABLE MEDIA SERVICES

Out of the box components for interoperable media workflows



What is FIMS?

FIMS (Framework for Interoperable Media Services) is a joint project of the Advanced Media Workflow Association (AMWA) and the European Broadcasting Union (EBU) with a membership of more than one hundred companies. Suppliers now recognise the value of the proposition and are offering FIMS-compliant products.

Why FIMS?

The current media landscape requires that companies address a constantly accelerating rate of technology change for media creation and consumption, with increasing consumer expectations and a proliferation of delivery platforms and formats. Business models need to be easily adaptable to sustain the change and support content creativity as well as technical innovation, such as the announced shift from file-based to IP stream-based workflows.

How does FIMS work?

FIMS defines open services that are loosely coupled, enabling multi-vendor services to be integrated and creating “best-in-class” media systems. The bottom line is that implementing a FIMS framework provides agility, interoperability, interchangeability and reusability of media-related services. In order to achieve this, great attention is devoted to the decomposition of systems into business processing units that are highly reusable if exposed as services through common interfaces. Two services that expose the same interface can be considered interchangeable. And, equivalently, two services that implement the same business function must expose the same interface. This greatly reduces the number of interfaces that must be implemented to connect different components on a composite system.

Service-Oriented Architecture (SOA) keeps business process logic separate from service implementation. FIMS interfaces can abstract vendor interfaces but vendors can also natively implement FIMS.

FIMS is SOA for media

The FIMS specification has been developed specifically for media and is aware of long-running synchronous or asynchronous processes. Associated job management functionalities include control for prioritisation, status monitoring and fault management. A job request contains all the parameters needed for its execution, optionally a description of the media content on which the command operates and the execution parameters grouped into profiles. It also provides an extension mechanism for the inclusion of vendor/user specific parameters where needed. The Orchestration System calls Media Services via FIMS interfaces over the IT network using a web services protocol such as SOAP, or accessing RESTful HTTP resources, and combines them in agile workflows.

FIMS supports the management of partial content and will soon support "growing" content in the context of live IP production. Beyond the service-based infrastructure, FIMS is also a strong data model for seamless management of technical and descriptive information in the workflow. The FIMS data model is based on EBUCore (see: https://tech.ebu.ch/MetadataEbuCore).

How does the FIMS project progress?

FIMS follows an established and proven process. The FIMS Technical Board defines and realises the specifications. But FIMS also favours open source development via a public github (https://github.com/fims-tv/). The continuity and coherence in the development of successive specifications is ensured by the FIMS Architecture Council.

What is available?

FIMS 1.3 already provides the definition of the following interfaces for basic media services: Transfer service, Transform service, Capture service, Extended Repository service, Quality analysis (QA). It also provides support for Partial and Growing content. Details of these are available at www.fims.tv and github.

FIMS is also a modular data model.

FIMS defines the job lifecycles and state machines including for long during processes.

What are the benefits from adopting FIMS?

With FIMS, there is now an interoperable solution for interfacing orchestration engines and services.

The main short term advantages for users are:

- Clean, consistent and vendor-independent design of component interfaces.
- Ability for vendors to leverage the flexible and generic FIMS interface to expose their products' APIs as a standardized and robust interface that seamlessly integrates with other compliant products for media exchange.

- Minimisation of integration costs and system complexity for media organisations through standardising operations for managing media assets in a vendor independent way.

More substantial advantages are expected in the mid-term with a growing market adoption of FIMS. Standardised interfaces provide ease of maintenance, scalability and repurposing of systems with the freedom to always choose best of breed products.

Extending the contributions to FIMS

FIMS has now introduced a github (https://github.com/fims-tv/) platform to which open source developers are invited to contribute freely. This can be either by proposing new additional features or by implementing an existing part of the specification, including interfaces between widely used applications. FIMS will also publish bounties, calling for developers to compete for a financial reward.

How can I learn more?

Further information is available on the FIMS YouTube channel (search for "Framework for Interoperable Media Services"). This has presentations from FIMS implementers and promoters. Video guidelines are also available.

The official FIMS website provides a wealth of information at www.fims.tv

There is also a LinkedIn FIMS SOA User group, periodically communicating recent news. Visit www.linkedin.com and search for "SOA-FIMS User Group".

Please note

FIMS is operated under an IPR framework assuring that all products can implement FIMS compensation and royalty free. As of today, more than 100 companies have signed the FIMS Participation Agreement, including manufacturers, system integrators, MAM solution providers and broadcasters.

Why not join us now?

For more information, please contact fims_adm@list.ebu.ch.

Key contact at EBU

- Jean-Pierre Evain: Evain@ebu.ch