

Input to the FIMS Task Force for discussion on RFT objectives

G. Dimino with contributions from Al Kovalik

FIMS objective

The objective of FIMS should be the identification of the components of a framework for interoperable services in media production able to solve current integration and maintenance inefficiencies of IT based systems. The first action will be the issuing of a public Request For Technology.

RFT objectives and content

Use case description and analysis can be a starting point to come to a more structured and sound analysis of the RFT objectives and content (that is the requirements and guidelines we are going to provide to the submitters of technologies).

Use cases

1. A user needs to transcode a batch of media files from format A to format B. He/she needs the result of transcoding be available within 24 hours.
Upon invocation, the Services Discovery service provides a list of transcoding services available: how can the client choose the most convenient one, in terms of features and SLA?
2. A user has a bunch of servers that wants to use to flexibly provide a set of pure software media services (e.g., transcoding, media processing, rendering). The number of running instances for each service should be modulated dynamically depending on the user requests.
3. A user needs a viewer able to play a media file, providing the usual transport controls (Play, Stop, Pause, Fast forward, Rewind). The display can be the user pc screen or a monitor connected to a dedicated video output (SDI) either from the user pc if a proper hardware is installed or from a remote video switcher.
4. A user needs to capture a video feed starting at time X for a duration of Y seconds.

From the analysis of the above cases the following **requirements** can be derived:

- A Services Discovery service that provides to the clients a list of available services on the network is an essential component. It's task includes keeping an updated list of available service types and running instances.
- Use case 2 suggests that the Discovery service must also be able to deal with service factories that can create (and destroy) service instances upon request, depending on hardware availability
- Service description is another key point: to select a proper instance to use, the client needs information about features actually implemented (e.g., not all the services of type "transcoder" can handle the same set of formats) and about the SLA provided (latency time, bandwidth, costs, etc...). In the case where specific hardware is needed to run the service, than the SLA specifications are even more important.
- A standardized services model is needed if we care about interoperability.

- Guidelines must be set to identify useful services, keeping in mind market attractiveness and reusability
- Services that are time aware. Some services may have access to a global clock for "frame accurate" control of devices. This allows a service to do some task @Tx
- Async and sync services. Most services are sync (invoke and wait for response) but async (long jobs) require messaging of some kind. I realize that most ESBs deal with this but we are not demanding an ESB as part of FIMS,
- SLA related, it may be good to include a param for ETOC (estimated time of completion), This way a service caller can know when a job will be done (or before). As our industry moves to deadline processing and away from @Tx, ETOC will become of value. This is really a "Do By Tx" (or before) method.

A generalization of the considerations above can lead to the following incomplete list of practical **objectives** for the RFT:

1. Methods to describe services, including description language, list of attributes for service type definition and for SLA declaration, interface methods and parameters exposition
2. A standardized service model for media production (or a specific area), with focus on market attractive reusable services
3. Identification and description of the structural elements that the framework should provide, including service discovery, system management, messages and media files communication formats and bus, etc ...
4. Interoperability with existing infrastructure, in particular videocentric systems
5. Methods to manage time synchronization