

EMDS: Extensible Multimedia Distribution Service

J. Povedano-Molina
Jose M. Lopez-Vega
Juan M. Lopez-Soler

Signal Theory, Telematics and Communications Department
University of Granada
SPAIN

OMG

Workshop on Real-time, Embedded and
Enterprise-Scale Time-Critical Systems
Arlington, VA

May 25, 2010

Agenda

- 1 Motivation
 - Multimedia content delivery
 - Multimedia scenarios
- 2 EMDS
 - Introduction
 - Architecture
 - Service examples
- 3 Demo
- 4 Conclusions

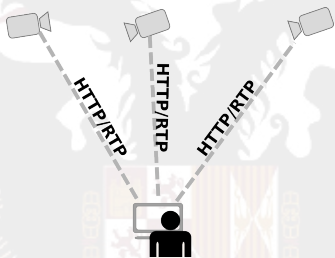
Outline

- 1 Motivation
 - Multimedia content delivery
 - Multimedia scenarios
- 2 EMDS
 - Introduction
 - Architecture
 - Service examples
- 3 Demo
- 4 Conclusions

Multimedia Content Delivery

- Traditionally **Multimedia Services Delivery** adopts client-server paradigms
 - It requires ad-hoc centralized servers
 - It is strongly coupled
 - *in space*: multimedia producers must be located
 - *in application*: clients are not service agnostic
 - *in format*: clients must support the particular stream format (resolution, sampling and frame rate, codecs,...)
 - *in resources*: network bandwidth, processing capabilities, etc.
- Additionally, multimedia data
 - Are continuous (stream based)
 - Are latency and jitter sensitive: data liveliness is very short

Multimedia Scenarios: video surveillance



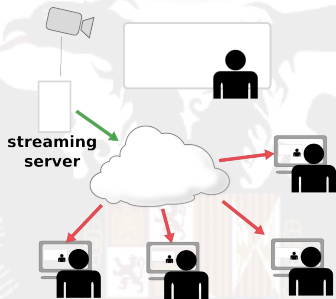
- Highly static scenario
- No audio
- Many-to-One communications
- Challenges on centralized systems:
 - Client application should know the cameras location
 - Mixing disparate cameras from different vendors (different protocols and capabilities) can be difficult

Multimedia Scenarios: teleconference



- Real-time data with audio
- Frequent join and leave operations
- Many to many communications
- Challenges on centralized systems:
 - It requires specialized hardware (e.g. H.323 MCU)
 - It lacks of deployment flexibility: difficulties for extending, customizing or composing new multimedia service

Multimedia Scenarios: event streaming



- Video and audio data
- One-to-many communications
- It relies on centralized servers
- Challenges on centralized systems:
 - To increase the scalability, it requires proxy servers
 - It does not provide tools for users feedback

Outline

- 1 Motivation
 - Multimedia content delivery
 - Multimedia scenarios
- 2 **EMDS**
 - Introduction
 - Architecture
 - Service examples
- 3 Demo
- 4 Conclusions

EMDS Introduction

- What is EMDS?
 - EMDS stands for *Extensible Multimedia Distribution Service*
 - EMDS is a framework for multimedia **services providing** and **multimedia content delivery** based on **DDS**
 - Data-centric approach
 - Automatic discovery
 - Relies on DDS QoS policies
- EMDS is designed with **extensibility** and **flexibility** as main goals
- EMDS eases the deployment and customizing new de-coupled multimedia services
- EMDS is suitable for different scenarios

EMDS Architecture: design issues

- In EMDS each media is published in a different topic (audio and video not multiplexed)
- Media topics are codec agnostic
- New codecs and coding schemes can be easily added
- New services and service customizing can be achieved transparently
- Subscriber applications decide how media are aggregated

EMDS Architecture: content discovery

- EMDS utilizes DDS SDP (Simple Discovery Protocol) for media discovery
- Media capabilities and metadata are distributed alongside with endpoint discovery metadata
 - MIME type
 - Data rates: sample rate (a), bitrate (a/v), framerate (v)
 - Spatial Resolution (v)
 - Decoder configuration
 - Other Specific Parameters (i.e. language (a))
- Media and service capabilities are encoded using JSON
 - Lightweight, structured and human-readable

EMDS Architecture: content discovery

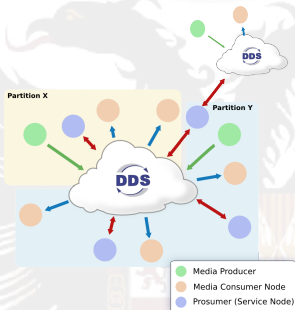
- Non-compatible publications will be filtered
- EMDS includes two levels filtering:
 - 1 Publications and subscriptions are matched whenever both
 - requested and offered capabilities are compatible and
 - QoS settings are compatible
 - 2 Filtering by metadata (keywords)
- Discovery can be customized by using domains and partitions
 - Isolating different multimedia profiles

EMDS Architecture: content discovery

```
{  
  "mime-type" : "video/h-263++",  
  "format" : { "framerate" : 25.0 ,  
               "resolution" : "320x240",  
               "bitrate" : 64000 },  
  "keywords" : ["sport", "soccer", "barcelona", "arsenal",  
               "champions", "league"],  
  "description" : "Champions League Final: F.C. Barcelona - Arsenal"  
}
```

Listing 1: Discovery Metadata

EMDS Architecture: entities

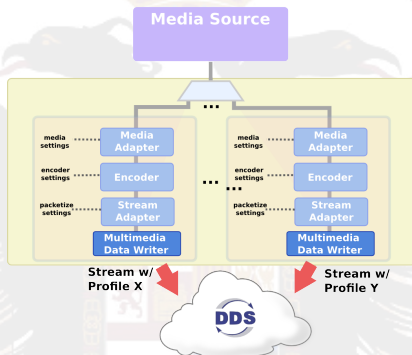


- Media Producers
 - 1 or many data publishers
 - i.e. surveillance camera
- Media Consumers
 - 1 or many data subscribers
 - i.e. surveillance station
- Hybrid Entity
 - *Prosumers*
 - They provide services extending/adapting multimedia topics
 - i.e. teleconferencing, transcoding services,...

EMDS Architecture: multimedia handling

- Media Producers send multimedia streams produced according to certain encoding settings (different partitions can be used)
 - Each video producer can publish streams with multiple profile settings
 - Profile settings are disseminated during discovery
- Media Consumers receive multimedia streams
 - Decode and render pipelines are built according to encoding settings found in the discovery
 - Appropriate decoders are chosen

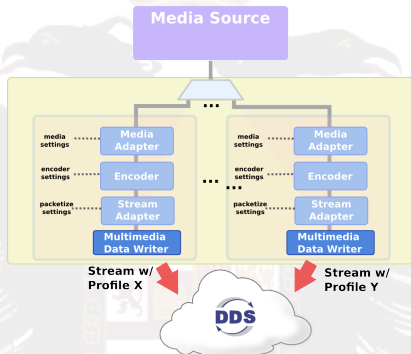
EMDS Architecture: media producers



- Encoder block:

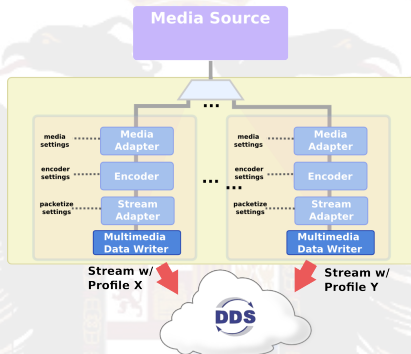
- Media Adapter: resolution, framerate, ...
- Encoder: bitrate, ...
- Stream Adapter: packet splitting
- Multiple encoder blocks

EMDS Architecture: media producers



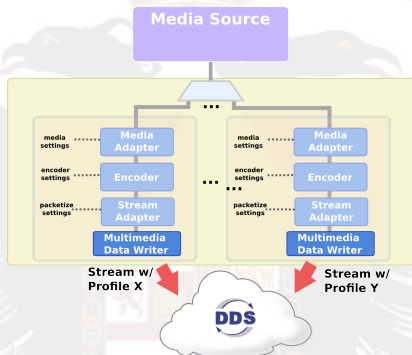
- Encoder block:
 - Media Adapter: resolution, framerate, ...
 - Encoder: bitrate, ...
 - Stream Adapter: packet splitting
- Multiple encoder blocks

EMDS Architecture: media producers



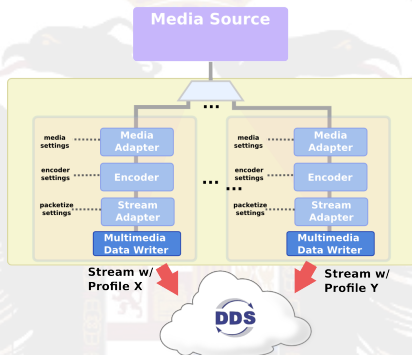
- Encoder block:
 - Media Adapter: resolution, framerate, ...
 - Encoder: bitrate, ...
 - Stream Adapter: packet splitting
- Multiple encoder blocks

EMDS Architecture: media producers



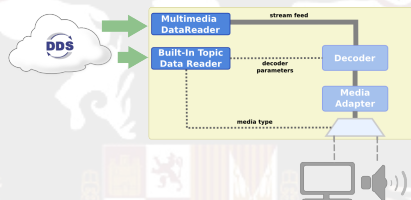
- Encoder block:
 - Media Adapter: resolution, framerate, ...
 - Encoder: bitrate, ...
 - Stream Adapter: packet splitting
- Multiple encoder blocks

EMDS Architecture: media producers



- Encoder block:
 - Media Adapter: resolution, framerate, ...
 - Encoder: bitrate, ...
 - Stream Adapter: packet splitting
- Multiple encoder blocks

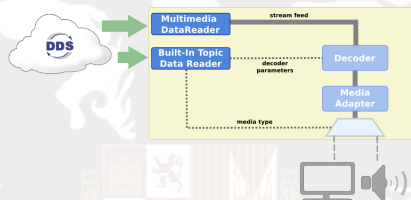
EMDS Architecture: media consumers



• Decoder block

- Configure decoder according to metadata acquired in the discovery phase
- Feed the decoder with topic samples
- Adapt the decoded stream to the output device

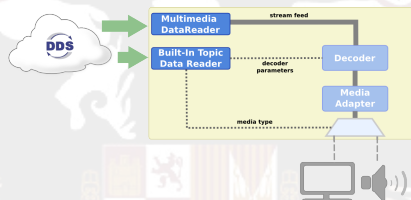
EMDS Architecture: media consumers



- Decoder block

- Configure decoder according to metadata acquired in the discovery phase
- Feed the decoder with topic samples
- Adapt the decoded stream to the output device

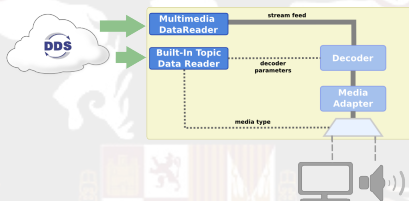
EMDS Architecture: media consumers



- Decoder block

- Configure decoder according to metadata acquired in the discovery phase
- Feed the decoder with topic samples
- Adapt the decoded stream to the output device

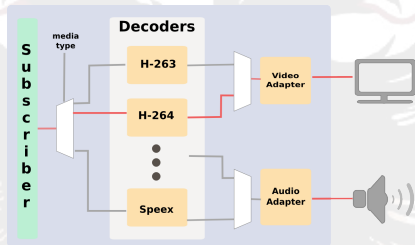
EMDS Architecture: media consumers



- Decoder block

- Configure decoder according to metadata acquired in the discovery phase
- Feed the decoder with topic samples
- Adapt the decoded stream to the output device

EMDS Architecture: multimedia handling (subscriber)

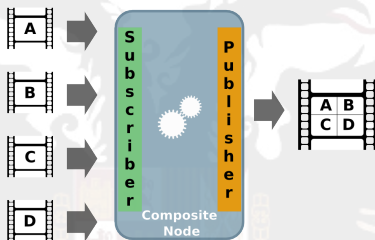


- Multimedia data are processed according to metadata.
- Pipeline is generated dynamically.
 - 1 MimeType is analyzed
 - 2 Appropriate decoder is chosen
 - 3 Media is adapted to the device format and sent to the appropriate output

EMDS services examples

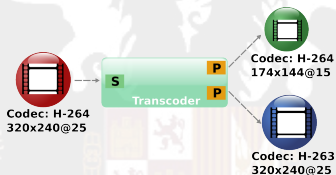
- Multiple services can be implemented on top of EMDS
- With data-centric approach no modifications are necessary on producer nor consumers
 - Services can be added in a transparent manner
 - Network can be extended
- Extensibility: new features are easily provided
- Flexibility: choosing appropriate services to fit a given scenario

EMDS services examples: compositing



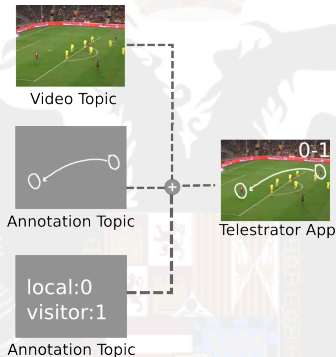
- EMDS mixes several streams
- EMDS reduces the subscriber data processing requirements
- Processing is done at intermediate (*prosumer*) nodes

EMDS services examples: transcoding



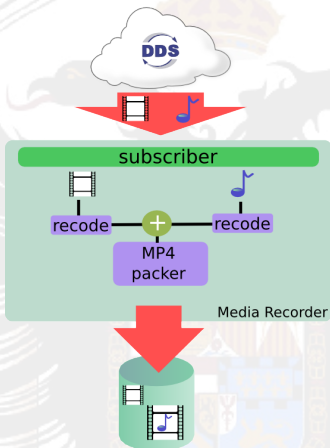
- EMDS allows multimedia streams transformations to adapt subscribers capabilities
 - Codec
 - Bitrate
 - Resolution
- The transcoding (*prosumer*) node subscribes to media streams, and re-codes them using different encoding parameters
- The new streams are re-published by the *prosumer*
- Network resources can be preserved by publishing the different streams in appropriate domain partitions

EMDS services examples: video annotation



- Sketching in video like a *telestrator* does
- Video can be annotated in a different node
- Subscriber application merges video and annotation topics
- Annotations are published in different topics
 - i.e. use a white-board protocol
- No video recoding

EMDS services examples: recording



- Multimedia sessions can be stored in recording nodes
- Multiple media topics are stored in unique multimedia container
- Multimedia containers can include multiple streams
 - video
 - multiple audio languages
 - subtitles
- Example application: recording a seminar

Outline

- 1 Motivation
 - Multimedia content delivery
 - Multimedia scenarios
- 2 EMDS
 - Introduction
 - Architecture
 - Service examples
- 3 Demo
- 4 Conclusions

EMDS Demonstration



Outline

- 1 Motivation
 - Multimedia content delivery
 - Multimedia scenarios
- 2 EMDS
 - Introduction
 - Architecture
 - Service examples
- 3 Demo
- 4 Conclusions

Conclusions

- EMDS is a highly flexible system for multimedia content delivery
- Automatic discovery removes dependency on directory services
- Filtering during discovery phase optimizes resources
- EMDS de-couples the multimedia content production and consumption, as a result new multimedia services can be transparently added (or customized) with minimal impact

Q&A

The background of the slide features a large, faint watermark of the University of Granada seal. The seal is circular, with the word 'GRANATA' at the top and '1492' at the bottom. In the center is a shield with various heraldic symbols, including a crown on top and a lion on the right side. The shield is flanked by two lions.

Thank You!

Acknowledgment: this work has been partially supported by the Ministerio de Ciencia e Innovacion of the Spanish Government under the project TIN2009-13992-C02-02, and the Plan Propio of the University of Granada, Spain.