Evolving MAM Systems in the Cloud

Julián Fernández-Campón
Business Solutions Director@Tedial
Traditional MAM Systems

• Complex Deployment
• Monolithic
• Hardcoded / Complex Logic / Difficult to Configure
New Business Needs

• Rapid Deployment
• On-Line Services
• Flexibility and Adaptability
• CI/CD (Continuous Improvement/Continuous Deployment)
Are these situations familiar?

• Q: When is the go-live date?
  • A: In one month

• Q: When will the new delivery specs be effective?
  • A: Next week
Evolution

- Cloud Ready vs. Cloud Native
- Conceptual Change
- Re-Architect the System
Evolution

• **Infrastructure.** Must be deployable in the Cloud:
  • Virtualized. A must
  • Servers: Specific Instances
  • Database. SQL vs NoSQL
  • Storage: Cloud Native Vendor Storage (such as S3)
  • Improved Security: Ports, Networks

• **Architectural**
  • Modules -> Micro-Services
  • Independency: No centralized dependencies such as DB

• **Functional:**
  • New On-Line services needed
  • Use New Features available in the Cloud: AI
Evolution: Specific Broadcast Needs

- Media Management. Not only data
- Efficiency. Quick Content Processing and Delivery
- Flexibility and Adaptability
  - New Requirements
  - Agility
- Multiple Use Cases:
  - Fully in the Cloud
  - Hybrid (+ Multisite)
Catalogue of Services

- Two Level Approach
- Core Services
  - Content Management
  - Workflow
  - Storage
- Building Blocks
  - Ingest
  - QC
  - Edit
  - Versioning
  - Delivery

<table>
<thead>
<tr>
<th>Content Management Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata</td>
</tr>
<tr>
<td>Index</td>
</tr>
<tr>
<td>ACLs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workflow Management Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow Data</td>
</tr>
<tr>
<td>Engine</td>
</tr>
<tr>
<td>Media Processing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage Management Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operations</td>
</tr>
<tr>
<td>Locations</td>
</tr>
<tr>
<td>Archival</td>
</tr>
</tbody>
</table>

Ingest | Quality Control | Edit | Archival | Delivery
Core Services

• Basic MAM Services
• Need to be Self-Contained
  • No central DB
• All interaction with others via API
• Independent:
  • Errors
  • Upgrades
Building Blocks

• One level up in Abstraction
• Self-Contained Operational Units
• Used to Deploy New Services Quickly
• Use the MAM Core Services
• One Step further in Workflows Orchestration in the Cloud
• Flexibility:
  • Dynamic Notifications
  • Parametrization
Building Blocks

Ingest
- Search
- Flow Data
- Locations
- Metadata
- Engine
- Operations
- Index
- Connectors
- Storage Plugins
- ACLs
- Media Processing

Quality Control
- Search
- Flow Data
- Publish
- Metadata
- Engine
- Operations
- Streaming
- Connectors
- Storage Plugins
- Media Processing

Editing
- Search
- Flow Data
- Publish
- Metadata
- Engine
- Operations
- Connectors
- Storage Plugins
- Media Processing

Archival
- Metadata
- Publish
- Operations
- Storage Plugins
- Archival
Building Block: Ingest

- **Input**
  - Source File(s): Video / Audio / Subs / Package / Folder, ...

- **Output**
  - Ingested Content in the MAM

- **Parameters**
  - Supported Formats and Transformations
  - File naming conventions

- **Notifications:**
  - Ingested Content (Optional)
Building Block: Quality Control

- **Input**
  - UUID

- **Output**
  - QC Result: Accepted / Rejected / Needs Editorial

- **Parameters**
  - AutoQC System
  - Test Plan
  - Manual QC Conditions

- **Notifications**
  - QC Result
Building Block: Editing

• Input
  • UUID

• Output
  • Edit Result: Accepted / Rejected / Needs Review

• Parameters
  • Editing System
  • Replacement/New Version

• Notifications
  • Edit Result
Building Block: Publishing

- Input
  - UUID

- Output
  - Status

- Parameters
  - Delivery Profile:
    - Video/Audio/Subs
    - Additional Components (Artwork, Trailers, etc.)
    - Transformations (including Assembly on the Fly)
Deploy My New Service

• Architectural Decisions:
  • Cloud vs Hybrid (+Multi-Site)

• Core MAM. Dimensioned according to each user needs
  • Connected Users
  • Streaming Services
  • Incoming Media

• Building Blocks:
  • Ingest,
  • QC
  • Edit
  • Publish

[....]
Business Benefits

• Rapid Deployment
  • System Dimensioning
  • Catalogue of Building Blocks

• Elasticity. New services can be spun up on demand

• Measurement and Predictability

• Flexibility. Quick changes

• Evolution.
  • Upgrades. New Releases
  • New Features: Core Features, Integrated Systems, Building Blocks
Business Benefits

Content Management Services
- Metadata
- Search
- Index
- Streaming
- ACLs
- Housekeeping

Workflow Management Services
- Flow Data
- Flow Config
- Engine
- Connectors
- Media Processing
- Flow Housekeeping

Storage Management Services
- Operations
- Library
- Locations
- Storage Plugins
- Archival
- Publishing
Case Study: Distributed Editing for News

- Architecture
  - Hybrid Cloud
  - 12 Sites
- Core MAM
  - 500 concurrent connected users
- Building Blocks
  - Ingest
  - QC
  - Edit: SAM Quantel Integration
  - Publishing
Case Study: Distributed Editing for News

Diagram:
- Private Cloud
  - Node-1
    - Editing Stations
    - PAM
  - Node-12
    - Editing Stations
    - PAM
Case Study: Content Prep and Delivery

- **Architecture**
  - AWS, Native S3

- **Core MAM**
  - 100 concurrent connected users

- **Building Blocks**
  - **Ingest:**
    - Video, Audio and Subs
    - Localization
    - Leader Insertion
  - **QC**
  - **Publishing**
    - 50 Destinations
    - Assembly on the Fly
Case Study: Content Prep and Delivery
Conclusions

• Traditional MAM Systems must be Re-Architected to be efficiently deployed in the Cloud

• Difference between Cloud Ready and Cloud Native

• The Use of Building Blocks drastically simplifies and optimizes the deployment of Services in the Cloud

• The Services that can be provided need to be flexible and tailored to customer needs
Conclusions

Be Open to Evolve

Be Ready for the new incoming requests to maintain the quality of the platform!

No one wants to start building a nice brand new system and end in a Frankenstein
Thanks!