

# Du Podcast à DMS (Descartes Media Suite)

**Patrick de Carné**

DISI, université Paris Descartes  
12 rue de l'École de Médecine, 75006 Paris  
patrick.de-carne@parisdescartes.fr

**Eddy Pelaic**

DISI, université Paris Descartes  
45 rue des Saints-Pères, 75006 Paris  
eddy.pelaic@parisdescartes.fr

**Fabienne Leroy-Carat**

DISI, université Paris Descartes  
12 rue de l'École de Médecine, 75006 Paris  
fabienne.leroy@parisdescartes.fr

**Nicolas Meneceur**

DISI, université Paris Descartes  
45 rue des Saints-Pères, 75006 Paris  
nicolas.meneceur@parisdescartes.fr

**Jérôme Dubois**

DISI, université Paris Descartes  
45 rue des Saints-Pères, 75006 Paris  
jerome.dubois@parisdescartes.fr

**Frédéric Rivalland**

DISI, université Paris Descartes  
12 rue de l'École de Médecine, 75006 Paris  
frederic.rivalland@parisdescartes.fr

## Résumé

*En 2008, l'université Paris Descartes a lancé son projet podcast avec un financement de l'établissement, complété par le Ministère de l'Enseignement Supérieur et de la Recherche puis la Mairie de Paris. La solution est conçue pour être déployable sur l'UNR Paris Ile-de-France.*

*Les limitations de la solution Apple Podcast Producer ont entraîné en 2010 l'abandon de cette filière "tout en un" au profit d'une architecture composée de quatre modules indépendants, reposant sur un développement interne et des produits professionnels, tout en conservant l'architecture matérielle d'origine.*

*DMS (Descartes Media Suite) est une suite logicielle composée de quatre briques applicatives :*

- *Descartes Capture, industrialisation des captations et diffusions via interface web,*
- *Descartes Broadcast, infrastructure de diffusion à très haute disponibilité,*
- *Descartes Engine, infrastructure d'encodage vidéo haute performance,*
- *Media<sup>2</sup> Descartes, plateforme de publication et d'administration des ressources multimédia.*

*30 points de captation sont déployés à l'université Paris Descartes. Le service est utilisé pour la PAES (Première Année d'Étude en Santé) qui a totalisé plus de 1000 heures de cours captés et diffusés lors de sa première année de production en 2010/2011.*

*La suite constitue une chaîne de production et de diffusion de média en masse industrialisable et mutualisable. Elle se prête parfaitement à un déploiement en mode SaaS (Software as a Service) au sein d'un cloud universitaire privé.*

## Mots clefs

Podcast, grille d'encodage, téléamphi, vidéo à la demande, cluster de streaming, médiathèque.

# 1 Introduction

Un des enjeux de l'Université est de répondre à la grande difficulté à gérer individuellement le rythme d'apprentissage des étudiants, dans un contexte de formation de grand nombre. Le podcast est une des solutions.

La production de podcast s'appuie sur quatre métiers :

- la captation qui produit massivement et automatiquement des contenus multimédia,
- la diffusion qui transmet, en direct ou en différé, le média à destination d'un très large public,
- l'encodage qui optimise les médias pour une visualisation sur des périphériques variés (pc, tablettes, smartphones, ... ),
- la publication qui permet de soumettre les médias à diffusion, de les entreposer, les classer, les organiser, les indexer, gérer leurs droits d'accès et les rechercher.

Autour de ces quatre axes, le projet podcast de l'université Paris Descartes apporte de nombreux avantages dans le domaine pédagogique en produisant massivement des ressources numériques en appui à l'enseignement présentiel. Cofinancé par la sous-direction des TICE du Ministère (SDTICE), ce projet qui a pour vocation d'être mutualisé à l'échelle inter-universitaire a reçu également une dotation de la ville de Paris via l'Université Numérique Régionale (UNR) Paris Ile-de-France.

## 2 Mise en œuvre de la solution « tout en un » Apple Podcast Producer

### 2.1 Historique

En 2008, l'université Paris Descartes a souhaité mettre en place un système automatisé de production de podcast, basé sur la solution *Apple Podcast Producer*, pour l'enregistrement et la mise en ligne des cours. Chaque point de captation, amphithéâtre ou salle de cours, faisant l'objet d'un enregistrement quasi continu des cours s'y déroulant, la solution avait été choisie pour aboutir à l'industrialisation de la production massive de contenus depuis la captation jusqu'à la mise en ligne.

La solution *Podcast Producer* s'appuie sur une architecture répartie d'agents enregistreurs. Chaque agent est relié aux ressources audiovisuelles du point de captation et produit des enregistrements qu'il soumet à un système centralisé de production de podcast constitué d'un SAN pour le stockage et de serveurs qui réalisent tous les traitements nécessaires à la mise en ligne des contenus. Cette architecture rend le dispositif facilement extensible et autorise une production massive de contenus.

La première tranche du projet podcast a été réalisée en 2008/2009. Elle comportait 8 points de captation mono-source répartis sur 7 sites de l'université et une infrastructure de production centralisée, constituée d'un SAN de 12 To et de 6 serveurs XServe, dont un serveur frontal pour l'interface de service, deux contrôleurs pour le SAN et trois nœuds d'encodage pour traiter les enregistrements provenant des agents. Sur la base de 10h de cours par jour et par point de captation, cela correspond à une capacité de production théorique maximale de 80h/jour de captation.

Un ETP ingénieur système a été dédié à temps plein pour le suivi des prestations d'installation des points de captation et de l'infrastructure serveur. Le système a été utilisé à titre expérimental et la recette fonctionnelle a été honorée.

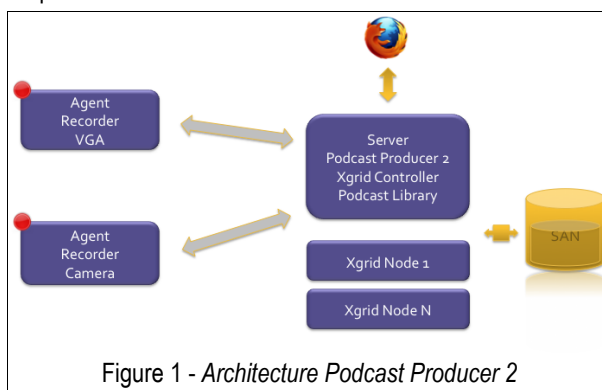


Figure 1 - Architecture Podcast Producer 2

La deuxième tranche du projet podcast en 2009/2010 était une extension du système déployé : 6 nœuds ont été rajoutés à la grille d'encodage pour traiter 22 points de captation supplémentaires. Le système, composé au total de 12 serveurs et de 30 points de captation répartis sur les 12 sites de l'université, avait une capacité de production théorique maximale de 300h/jour de captation.

En parallèle, nous avons migré vers *Podcast Producer 2* qui apportait des évolutions majeures et attendues, en particulier le *dual source*<sup>1</sup> qui a conduit au déploiement d'un second agent enregistreur sur chaque point de captation.

Toujours avec un ETP ingénieur système, la conduite du chantier d'extension du nombre de points de captation et la mise à niveau sur le nouveau schéma des points déjà déployés ont entraîné un retard non négligeable sur les tests de charge et la recette finale. Nous avons fait appel à un prestataire spécialisé dans l'intégration des produits Apple à hauteur de 40 jours ingénieur pour nous aider sur le déploiement et la mise au point de la nouvelle configuration. Un ETP développeur a également été affecté au projet pour effectuer l'intégration au système d'information.

<sup>1</sup> Fonctionnalité permettant de produire des contenus enrichis mixant deux sources, par exemple la vidéo de l'enseignant et son écran d'ordinateur.

À la fin de l'année universitaire 2009/2010 et en phase d'intégration de la solution Apple au système d'information, l'université s'est engagée pour le concours de la Première Année en Étude de Santé (PAES), à enregistrer tous les cours et à les retransmettre en téléamphi. Le protocole fonctionnel, applicable à la rentrée 2010, était d'enregistrer en triple exemplaire chaque cours capté/diffusé (40 cours par semaine en moyenne). Il s'est avéré que la solution basée sur *Podcast Producer 2* posait des problèmes techniques, organisationnels et logistiques qui n'auraient pas permis son exploitation pour la PAES.

## 2.2 Limitations de la solution Podcast Producer

Malgré nos efforts, la mise au point de la filière de captation Apple n'a pas permis d'atteindre les objectifs et l'industrialisation de la production des contenus à cause des limitations de la solution :

- Architecture complexe de la captation dans une configuration multi-sources : les cours d'anatomie se font en alternance diaporama / tableau / visuel de l'enseignant. *Podcast Producer 2* dispose d'un mode *dual source* mais sur deux agents enregistreurs séparés et le montage (commutation de sources) se fait sans contrôle via un algorithme de détection de changement d'activité lors de la phase d'encodage sur le serveur. Il n'y a aucun moyen (hors postproduction manuel) de gérer le résultat de la vidéo finale.
- Intégration difficile du service dans le système d'information de l'établissement : la phase d'intégration de *Podcast Producer 2* avait démontré la nécessité de développer une couche applicative intermédiaire faisant office d'adaptateur entre le système d'information de l'université et le système d'Apple. Cette couche devait être entre autres le frontal «fonctionnel» des cas d'utilisation interne (protocole PAES par exemple) et pas uniquement une intégration basique (authentification, SSO, charte graphique, ...). Cette phase nécessitait de réécrire certaines sections du code applicatif du serveur podcast (scripts Ruby On Rails). Cela engendrait les problèmes de maintenance des modifications vis à vis des mises à jour système de Mac OS X et de *Podcast Producer*.
- Manque de fiabilité des agents de captation : les tests effectués avec les dernières versions de la solution Apple ont fait apparaître une dégradation de la fiabilité (en particulier des arrêts imprévisibles sur les enregistrements) ; ce qui était incompatible avec le niveau de service de haute disponibilité que l'on nous demandait d'assurer.
- Problèmes de performance de la grille d'encodage XGrid : les temps d'encodage sur des simulations de production équivalentes au maximum de la capacité de notre infrastructure de captation ont révélé des problèmes d'ordonnement et d'efficacité. Nous avons fait une étude comparative interne avec un produit professionnel du secteur, *Episode Engine* de TeleStream qui fournit un service comparable afin de déterminer l'écart de performance. Lors de ces tests, nous avons constaté que *Podcast Producer 2* sollicitait sur plus de la moitié du temps d'encodage le GPU de la carte graphique au lieu des 16 cœurs CPU des XServe qui sont munis d'une carte graphique de base avec seulement 256 Mo de mémoire vidéo.

Suite à l'accumulation de demandes fonctionnelles en grande partie incompatibles avec la solution retenue, et de l'échéancier de mise en œuvre pour la PAES, nous avons décidé de reprendre le projet podcast en repartant sur une architecture en phase avec nos nouveaux besoins. Une contrainte était de reprendre l'infrastructure existante. En effet, au vu de l'investissement consenti, et avec les contraintes de temps et de budget imposées, il n'était pas question de remplacer l'infrastructure matérielle et de commander une autre solution en parallèle.

## 3 Évolution vers une architecture modulaire « Descartes Media Suite »

Compte tenu des limitations précédentes, il a été décidé en 2010 d'abandonner la filière Apple au profit d'une architecture composée de quatre parties indépendantes, correspondantes aux quatre métiers (captation, encodage, diffusion et publication), reposant sur un développement interne et des produits professionnels, tout en conservant l'architecture matérielle d'origine.

Descartes Media Suite (DMS) est composée des quatre briques suivantes :

- Le serveur « Descartes Capture » gère la logique de captation/diffusion de l'ensemble des sites, notamment par un système de réservation et de délégation de gestion.
- La grille d'encodage « Descartes Engine » transforme à la demande les médias, notamment ceux captés par les agents podcast, pour les encoder dans un ou plusieurs formats adaptés à leur diffusion.
- Le cluster « Descartes Broadcast », reposant sur une architecture haute performance et haute disponibilité, diffuse massivement les événements en direct et les contenus en VoD (Video on Demand).
- La médiathèque « Media<sup>2</sup> Descartes », sous forme d'entrepôt de médias, gère l'espace de publication des ressources multimédia sur laquelle pointent les plateformes de publication tierces (Dokeos, Moddle, iTunesU, ...) pour afficher les ressources dans leurs pages.

Ces quatre briques sont séparées pour faire évoluer indépendamment chacune d'elle en gardant une approche de couplage faible pour réduire les dépendances.

### 3.1 Descartes Capture

Descartes Capture est la clé de voûte de DMS. Cette application est le frontal qui commande l'enregistrement des captations et leur diffusion par streaming IP sur le réseau du campus et sur l'internet.

L'architecture retenue pour Descartes Capture est basée, comme pour *Podcast Producer*, sur une architecture répartie serveur/agents. Plutôt que de réinventer la roue sur la partie audiovisuelle (enregistrement, commutation de sources), nous avons décidé d'exploiter sur les agents un logiciel commercial de WebTV très fiable, *Wirecast* de Telestream, qui a fait ses preuves à la fois dans le monde professionnel et au sein même de Descartes. Du côté du serveur, nous sommes repartis de notre développement interne visant à s'interfacer avec *Podcast Producer*. Nous avons revu le modèle de données afin de l'adapter aux besoins fonctionnels tout en gardant comme objectif de maintenir la simplicité et le côté intuitif du démarrage et de l'arrêt de la captation, ce qui était l'un des avantages de la solution Apple.

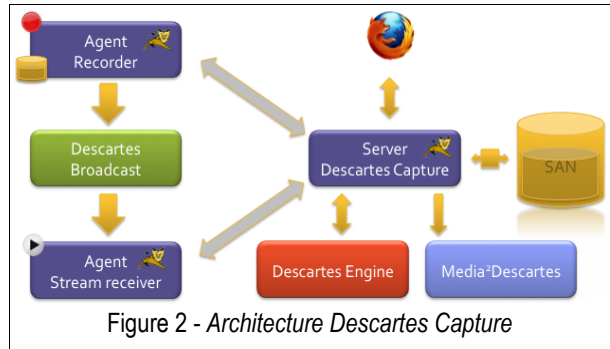


Figure 2 - Architecture Descartes Capture

Voici les principales fonctionnalités de Descartes Capture :

- gestion centralisée web, réservation,
- captation, commutation de sources,
- télé-amphi, diffusion streaming,
- délégation de gestion,
- gestion des encodages,
- export vers publication

Descartes Capture utilise un système de réservation des points de captation. Le système permet la délégation de gestion tant sur les droits de réservation que sur les opérations techniques liées à la captation et au téléamphi. Il est possible de piloter depuis une interface web les événements à enregistrer et/ou à diffuser, ainsi que de choisir les propriétés d'affichage (screencast, picture-in-picture, écran d'attente). La simplicité du système est tel que le téléamphi est déclenché par une seule action pour l'ensemble des points de réception sélectionnés lors de la réservation.

Le serveur utilise les dernières technologies Java JEE6. Il propose une interface web JSF2/Richfaces4 avec des composants Ajax, un protocole de commande HTTP/XML pour l'interface tactile en cours de développement et un système de suivi du workflow (dépôt, encodage et publication) depuis une interface multi-utilisateurs. Il gère le retour image depuis les points de captation et la coordination des commandes envoyées aux agents. Il sert d'entrepôt de stockage temporaire des fichiers vidéo sources et encodés. Le serveur Descartes Capture s'interface avec Descartes Broadcast pour la diffusion, Descartes Engine pour l'encodage et Media² Descartes pour la publication.

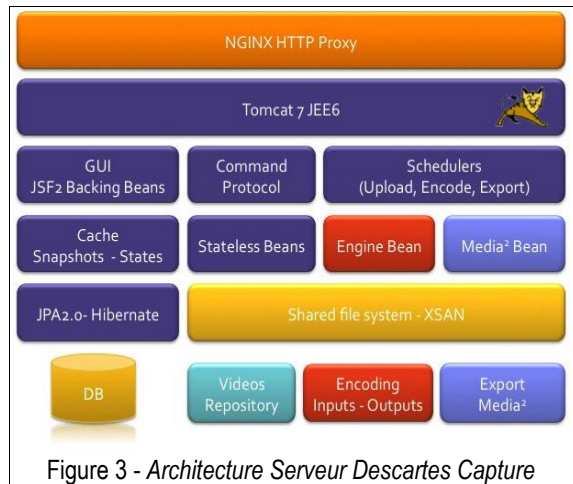


Figure 3 - Architecture Serveur Descartes Capture

L'agent joue deux rôles : l'agent de captation enregistre la vidéo et la diffuse en streaming, les agents récepteurs reçoivent le flux vidéo. L'agent est développé en Java et s'interface avec le système d'exploitation (MacOS) et l'application de captation (*Wirecast*) et de réception (*QuickTime*). L'interface *HardwareLayer* permet d'être indépendant du système d'exploitation et d'utiliser toute autre application qui fournit les services de base comme l'enregistrement, la commutation de sources et l'émission d'un flux vers un serveur de streaming. L'agent est totalement indépendant du serveur, il se contente d'exécuter les ordres du serveur. Il possède un protocole simple de commande sur HTTP et une interface web de paramétrage des prises de vue comme la configuration du picture-in-picture (taille, position, ratio).

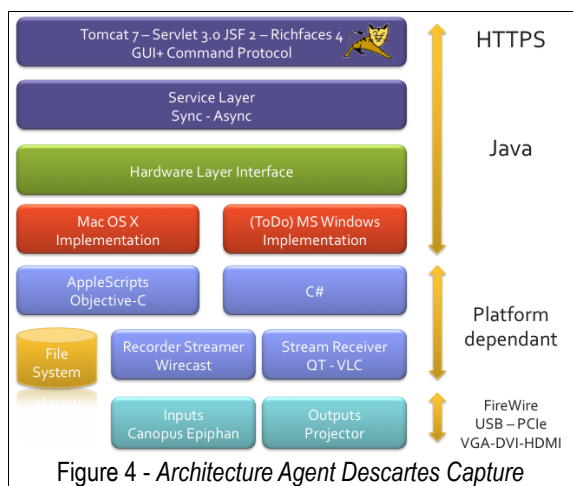


Figure 4 - Architecture Agent Descartes Capture

### 3.2 Descartes Broadcast

Descartes Broadcast est le nom donné à notre infrastructure qui assure la retransmission en direct et en différé des cours ou des événements diffusés entre amphithéâtres (téléamphi) et vers l'Internet (webcast).

Initialement, l'architecture de diffusion s'appuyait sur deux serveurs de streaming indépendants QTSS (QuickTime Streaming Server), l'un dédié au téléamphi et l'autre au webcast. Ces serveurs, limités à quelques centaines de connections simultanées, ne permettaient plus de répondre à la montée en charge des diffusions, en particulier lors de la retransmission d'événements à forte audience hébergés par l'université (tutoJRES, ESUP-Days, JNum, ...).

Dans le cadre de DMS, l'architecture a évolué vers quatre serveurs de streaming redondants et en répartition de charge, autorisant la diffusion vers un très large public et avec une très haute disponibilité. La solution retenue est basée sur le logiciel Wowza Media Server, un serveur de streaming performant, extensible et très populaire chez les fournisseurs de CDN (Content Delivery Network). Un serveur Wowza permet la diffusion du format H.264 sur divers protocoles de transport (RTSP/RTP, RTMP, MPEG-TS, HTTP et HLS). Il supporte le transcodage et permet de distribuer les contenus vers tous les périphériques existants même les plus exotiques (smartphones, tablettes, set-top box, ...).

Plusieurs serveurs Wowza peuvent être configurés en cluster pour distribuer de façon optimale le flux vidéo. Dans cette configuration, l'un des serveurs (serveur *Origin*) agit pour équilibrer la charge des noeuds de diffusion (serveur *Edge*). Si un client demande un flux vidéo alors le serveur *Origin* le redirige vers le noeud le moins chargé. Tout serveur Wowza dans le cluster peut être configuré comme un serveur *Origin*, un serveur *Edge* ou les deux. L'ajout de nouveaux serveurs *Edge* au serveur *Origin* permet d'étendre la diffusion. Le cluster rend ainsi le service de diffusion extensible et la maintenabilité plus aisée.

L'architecture de diffusion de Paris Descartes est constituée d'un cluster de quatre serveurs Wowza. Deux serveurs QTSS indépendants, dédiés exclusivement au téléamphi sont encore exploités pour nos besoins internes.

Le cluster de diffusion Wowza est décrit sur la figure 5 :

- deux serveurs *Origin*, configurés en *Load Balancer Listener*, reçoivent les flux émis par des encodeurs, les redistribuent aux noeuds et gèrent la répartition de la charge de la diffusion aux clients,
- deux serveurs *Edge*, configurés en *Load Balancer Sender*, signalent leur taux de disponibilité aux serveurs *Origin* et distribuent aux clients les flux reçus des serveurs *Origin* en s'adaptant au protocole de transport .

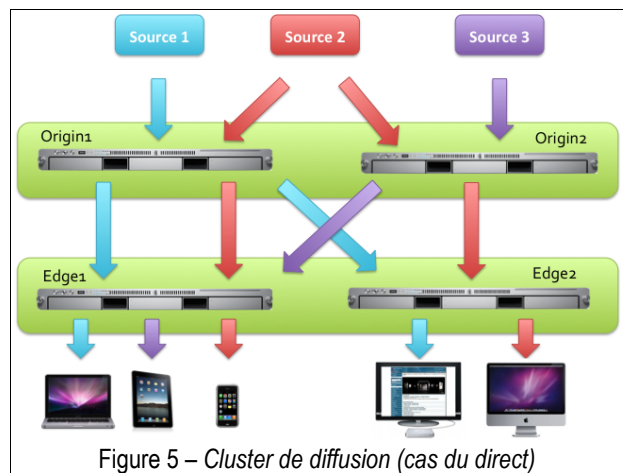


Figure 5 – Cluster de diffusion (cas du direct)

Le principe de la diffusion d'un événement en direct est le suivant :

- l'encodeur diffuse à destination des deux serveurs *Origin*,
- l'utilisateur se connecte à un proxy web qui s'adresse à l'un des deux serveurs *Origin* pour connaître le serveur *Edge* le moins chargé,
- le proxy renvoie une requête RTMP (Flash) ou HTTP (iOS) ou RTSP (VLC) au serveur *Edge* identifié précédemment,
- à la réception de cette requête, le serveur *Edge* récupère le flux auprès d'un des deux serveurs *Origin*,
- le serveur *Edge* sert le flux selon les 3 protocoles simultanément.

Avec cette architecture, le service reste disponible même en cas de maintenance ou de panne d'un serveur *Edge* et d'un serveur *Origin*.

Les capacités d'un noeud dans le cluster ont été mesurées lors de simulations de diffusion en direct et en VoD et sont présentées sur la figure 6.

Pour étendre la diffusion, il suffit de rajouter le nombre de noeuds nécessaires au cluster de diffusion.

TEST DE DIFFUSION EN DIRECT		
depuis un flux source encodé en H.264 à 1 Mbit/s		
Connexions max (par noeud)	Origine de la saturation	
1000	Débit de l'interface Ethernet (1 Gbit/s)	

TEST DE DIFFUSION VOD		
depuis des fichiers vidéos encodés en H.264 à 500 kbit/s		
X fichiers en lecture partagé	Connexions max (par noeud)	Origine de la saturation
fichiers stockés sur disque local	X=1	Débit de l'interface Ethernet (1 Gbit/s)
	X=10	Accès concurrentiel sur le disque local
fichiers stockés sur disque SAN	X=250	Débit de l'interface Ethernet (1 Gbit/s)
	X=500	Nombre de Threads Java (1500)

Figure 6 - Capacité d'un noeud (serveur *Edge*) dans le cluster

### 3.3 Descartes Engine

Dans une optique de standardisation des médias publiés (générique de début et de fin, watermark, déclinaisons multi-formats), et ce avec les différents outils de production de contenu, il aurait été fastidieux d'adapter les outils de captation à chaque changement de la charte de publication au niveau de l'institution, de ses composantes ou d'événements ponctuels, et encore plus d'unifier les processus de postproduction par les opérateurs multimédia.

L'activité d'encodage étant très gourmande en ressources matérielles et en temps, il était donc de-facto indispensable de l'automatiser et par la même occasion, de mutualiser les ressources matérielles coûteuses mais aussi de libérer les opérateurs multimédia de cette tâche à faible valeur ajoutée, afin qu'ils puissent se concentrer sur la production de média de qualité.

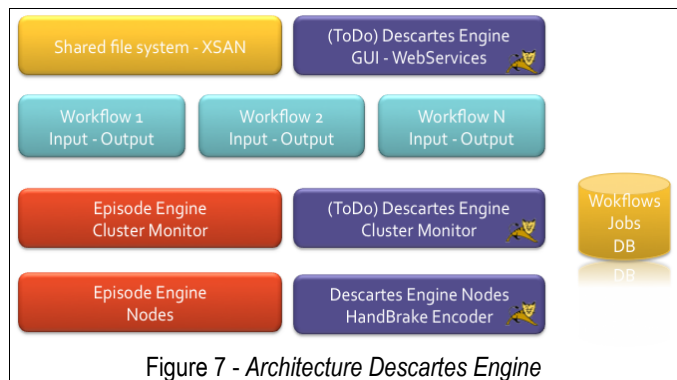
Une grille d'encodage avec parallélisation des traitements est nécessaire pour industrialiser la production des contenus et pour les mettre en ligne le plus rapidement possible. Pour prendre un exemple concret, l'information captée par Descartes Capture est très lourde et se mesure en dizaines de gigaoctets de données brutes hebdomadaires. Les fichiers vidéo en 1024x768 générés par les agents de captation sont trop volumineux pour être publiés, ils ont besoin d'être réencodés avec un générique de début et de fin. Le format de publication H.264 en 640x480 est privilégié en particulier pour être compatible avec la publication sur iTunesU.

Les fichiers produits par Descartes Capture sont poussés dans l'une des files d'attente correspondant à un workflow. Chaque workflow est constitué d'une file d'attente en entrée et d'un conteneur en sortie. Actuellement, nous utilisons comme encodeur le logiciel commercial *Episode Engine* de TeleStream ou le logiciel Open Source *HandBrake*. L'un des objectifs de Descartes Engine est de pouvoir intégrer selon les besoins différents encodeurs selon leurs performances ou bien les formats qu'ils proposent et cela depuis une seule interface.

Avec l'encodeur *Episode Engine*, une étude réalisée en interne nous a prouvé que le matériel (notamment les processeurs à 16 cœurs), était utilisé à 100% de ses capacités sur tous les nœuds de la grille lors des tests à forte charge. Ces tests nous ont permis de définir précisément le paramètre correspondant au nombre d'encodages simultanés par nœud avant de les passer en « job pending ». Cela nous a permis de confirmer la capacité de l'infrastructure à encoder avec une garantie du temps de sortie max (H+9) des médias, inférieur à la commande (J+1). Le problème est simple, l'infrastructure d'encodage doit à la fin du premier créneau horaire de cours (8h-9h30) encoder et gérer la file d'attente jusqu'à 20h30-21h. Elle doit impérativement encoder la totalité des vidéos de J-0 avant la fin des premiers cours de J+1, sinon cela voudrait dire qu'elle commence à cumuler du retard. Avec la solution Apple nous arrivions à des temps d'encodage de plus de 24 heures.

Ces temps de sortie sont obtenus pour un scénario où nous poussons en simultané 30 vidéos de 1h30 (environ 30Go) qui doivent être déclinées en trois formats différents avec un watermark soit au total 90 encodages (30 sources x 3 formats). Les premières vidéos sortent au bout de 2h et les dernières en 9h sur une grille composée de 4 nœuds. Dans ce type de scénario *Episode Engine* répartit les trois déclinaisons d'une même source sur plusieurs nœuds, mais chaque déclinaison est traitée entièrement sur le même nœud. Il est possible de descendre à moins de 15 minutes par déclinaison en parallélisant l'encodage d'une même déclinaison sur plusieurs nœuds. Cette option correspond plus à une utilisation lors de production de médias en mode événementiel, par exemple quand il faut enregistrer une conférence et la publier dans les meilleurs délais en VoD 15 à 20 minutes après la fin de la conférence.

L'infrastructure de stockage utilisée est basée sur une baie promise Vtrak E610 et 2 commutateurs Fibre Channel QLogic avec le système Apple Xsan qui profite de la technologie *Multipathing* pour la redondance. L'ACFS (Apple Cluster File System) est le système de fichiers en cluster utilisé sur Mac OS X 10.6 avec Xsan 2.2. Xsan, le système de fichiers SAN (Storage Area Network) pour Mac OS X permet une optimisation des accès. Deux contrôleurs de métadonnées en redondance gèrent les demandes d'accès au système de fichiers des différents serveurs connectés au Fibre Channel. Seize disques de 750Go composent la baie podcast et le SAN s'appuie sur un LUN (Logical Unit Number) de deux disques en miroir pour les métadonnées plus le journal et deux LUNs pour les données en RAID5. Deux disques de rechange avec fonctionnalité *Revert Global Spare* sont disponibles en stock en cas de besoin.





### 3.4 Media<sup>2</sup> Descartes

Media<sup>2</sup> est la plateforme de publication des ressources multimédia de Paris Descartes. Elle permet à chaque membre d'exposer ses productions sur le web : cours filmés, tutoriels, conférences, séminaires, ressources numériques documentaires et imageries scientifiques (lames virtuelles).

Les fonctionnalités sont nombreuses :

- modération des publications,
- signature du contrat d'auteur et d'image en ligne pour les membres de Paris Descartes,
- gestion des accès (public ou restreint à un groupe du référentiel de l'établissement),
- outil de statistique,
- minutage pour les cours filmés,
- indexation des ressources,
- podcast,
- abonnement par flux RSS,
- moteur de recherche,
- fonctionnalités d'intégration pour afficher les médias sur d'autres sites,
- affichage des vidéos dans différents formats suivant l'outil de lecture utilisé.

L'application Media<sup>2</sup> est composée de quatre principaux modules : Consultation, Administration, Contribution et Espace Juridique.

Chaque module est indépendant ce qui autorise une grande souplesse d'évolution. Développée entièrement à Paris Descartes sous la technologie Java WebObject 5.4.3, une base de données MySQL, un serveur Web sous Apache intégrant un module de pseudo-streaming H.264. L'application s'appuie sur un annuaire OpenLDAP pour la gestion des utilisateurs et l'authentification via le service CAS.

L'architecture est multi-tiers avec un serveur d'application java WebObject et un nombre d'instances à la demande. Tout client navigateur Web est supporté (Firefox, Chrome, Safari, Internet Explorer, iTunesU, etc.).

- Media<sup>2</sup> contribution

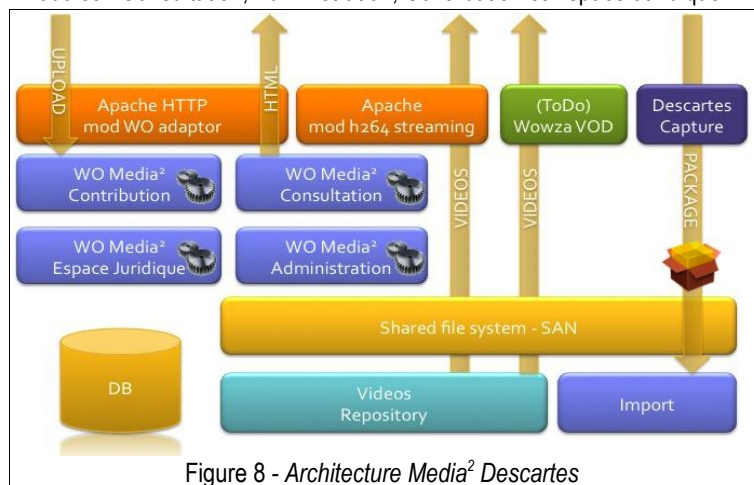


Figure 8 - Architecture Media<sup>2</sup> Descartes

Le module effectue le suivi de la contribution jusqu'à la publication des fichiers multimédia. Il propose une indexation qui peut être simplifiée au dépôt et enrichie à la publication. Les auteurs peuvent signer leur contrat de cession des droits d'auteur et d'image, avant ou au moment du dépôt. Le module alerte les administrateurs des nouveaux dépôts et de la signature des contrats.

- Media<sup>2</sup> administration

Le module administration gère les utilisateurs, les ressources (modération, dépublication) et les droits d'accès. Il s'appuie sur le système d'information de l'établissement via un annuaire Open LDAP. La création et l'organisation des rubriques peuvent être déléguées à différents administrateurs. Le module est doté d'un outil de statistiques.

- Media<sup>2</sup> espace juridique

Le module juridique permet aux auteurs de signer leur contrat en ligne pour une année ou un semestre au regard de la déclaration du programme des enseignements. Pour les auteurs qui signeraient leur contrat sous la forme « papier », les administrateurs ont la possibilité de valider la signature du contrat papier. L'application peut donc vérifier à chaque dépôt la signature effective des contrats. Les administrateurs peuvent consulter l'ensemble des auteurs qui ont signé un contrat. Enfin, les auteurs peuvent suivre leur publication suivant leur état : en préparation, proposée à la publication et publiées.

- Media<sup>2</sup> consultation

Le module de consultation offre des fonctionnalités de navigation par minutage des cours filmés et la prise en compte de formats variés dont le H264. Les ressources sont indexées pour une recherche plus performante. Toutes les vidéos peuvent être podcastées et les internautes peuvent s'abonner par flux RSS aux collections qui les intéressent. L'ensemble de la base est en plus indexé par le moteur professionnel SINEQUA, ce qui autorise des recherches raffinées parmi les 6700 publications. L'interface comprend des exports possibles pour exposer les ressources sur d'autres plateformes comme Moodle, Dokeos et iTunesU.

## 4 Indicateurs sur le projet de DMS

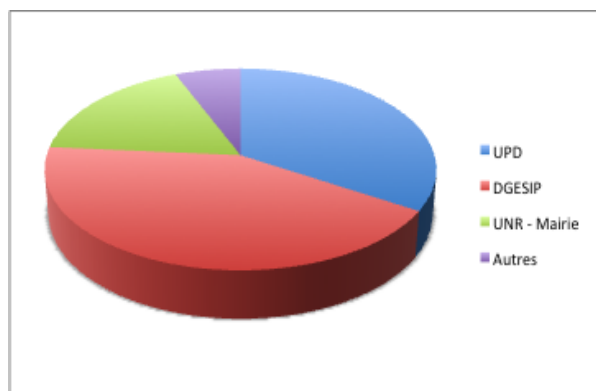
Les cours à l'université Paris Descartes sont captés et diffusés intensivement depuis septembre 2010 pour les 3600 étudiants de la Première Année d'Étude en Santé (PAES). Ainsi, au cours du premier semestre 2010, 1000 heures de cours ont ainsi été captées et diffusées dans six amphithéâtres tous les jours de 8h à 21h.

Le tableau ci-dessous donne des indicateurs sur le projet :

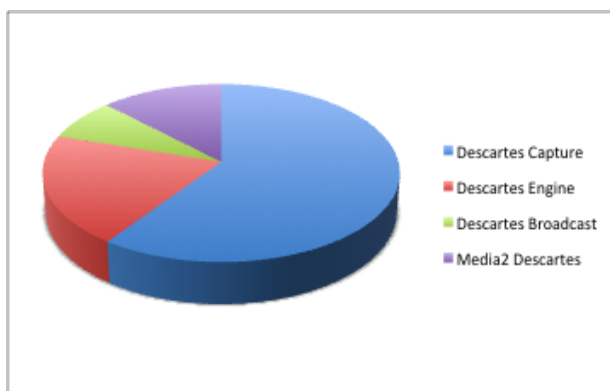
	<i>Descartes Capture</i>	<i>Descartes Engine</i>	<i>Descartes Broadcast</i>	<i>Media<sup>2</sup> Descartes</i>	<i>Total</i>
<b>Ressources 2011</b>	1 développeur	1 ingénieur système	1 ingénieur système	3 développeurs (mi-temps)	<b>Charge projet * 7,5 années.hommes</b>
<b>Temps passé (depuis 2008)</b>	Infra 24 mois Dev 12 mois	Infra 24 mois Dev 1 mois	Infra 6 mois Dev 1 mois	Dev 18 mois	
<b>LOC = Lines of Code (hors tests, HTML et JS) sous licence CeCILL</b>	Serveur 15 000 LOC Agent 13 000 LOC Interface tactile 4 900 LOC	HandBrake Encoder 1 000 LOC	Site Direct & VOD 1 400 LOC	4 modules 30 000 LOC	<b>65 300 LOC</b>
<b>Investissement</b>	-	-	-	-	<b>575 400 €</b>
<b>Répartition des dépenses</b>	59,76 %	20,13 %	7,63 %	12,46 %	
<b>Dépenses **</b>	319 400 €	107 600 €	40 800 €	66 600 €	<b>534 400 €</b>

\* Charge du projet sur ressources internes Descartes hors prestataire externe

\*\* Estimations, hors mise à disposition de l'infrastructure existante de l'université Paris Descartes



Répartition des investissements



Répartition des dépenses

## 5 Projets d'évolutions de DMS

Les évolutions globales de DMS sont :

- la mise en place d'un ESB (Enterprise Service Bus) Multimédia qui est à l'étude,
- le développement d'une interface de pilotage intégrant la fédération d'identité Shibboleth qui est nécessaire dans le contexte de la mutualisation des ressources multimédia à l'échelle de l'UNR,
- le packaging qui est essentiel pour la distribution de la solution et qui intégrera des interfaces utilisateurs personnalisées.



Le tableau ci-dessous donne quelques projets d'évolution des différentes briques :

<p style="text-align: center;">Évolution de Descartes Capture</p> <ul style="list-style-type: none"> <li>• Interface tactile (en cours)</li> <li>• Intégration visioconférence dual-source (en cours)</li> <li>• Intégration Terracotta (en cours)</li> <li>• Agent sous OS Windows (étude)</li> <li>• Intégration ADE</li> <li>• Chapitrage / Pagination automatique</li> <li>• Intégration domotique des points de captation</li> </ul>	<p style="text-align: center;">Évolution de Descartes Engine</p> <ul style="list-style-type: none"> <li>• Cluster Monitor pour encodeurs maison (étude)</li> <li>• Encodeur FFMpeg</li> <li>• WebServices Workflows</li> <li>• WebServices Upload</li> <li>• Indicateurs de consommation des ressources</li> </ul>
<p style="text-align: center;">Évolutions de Descartes Broadcast</p> <ul style="list-style-type: none"> <li>• WebServices Visioconférence (en cours)</li> <li>• WebServices LIVE B2B (étude)</li> <li>• Cache VoD</li> </ul>	<p style="text-align: center;">Évolutions de Media<sup>2</sup> Descartes</p> <ul style="list-style-type: none"> <li>• Intégration streaming VoD Wowza (en cours)</li> <li>• WebServices contribution B2B</li> <li>• Affichage Chapitrage - Édition vidéo</li> <li>• Intégration Descartes Engine au module Contribution</li> </ul>

## 6 Intégration, distribution et offre de service SaaS

L'un des objectifs de DMS est de proposer tout ou partie de ses fonctionnalités sous forme de services consommables à la demande, de les instancier ou de les distribuer, mais encore de mutualiser ces ressources afin d'en réduire les coûts d'acquisition et d'exploitation.

Un chantier a été démarré afin de répondre aux différentes configurations possibles de l'utilisation de DMS. Pour couvrir les différentes prestations, les composants liés aux interactions utilisateurs seront paramétrables pour s'intégrer dans un Système d'Information tiers (aspect graphique, base utilisateurs, groupes, etc.) et la topologie de l'infrastructure sera ré-étudiée afin de passer du MAN au WAN pour une consolidation des services de proximité (répartition topologique des serveurs *Edge* dans le cas d'une diffusion nationale par exemple).

Le couplage faible entre ces modules permet de remplacer en fonction des besoins, un module par un autre. Il est tout à fait envisageable pour une entité de définir l'orchestration d'un processus multimédia, composé à la fois de certaines briques de DMS et d'autres gérées en interne.

Les quatre briques de DMS sont distribuables ou consommables « on demand » ou « hosting » (figure 9). Les tests effectués sur les différents modules permettent de définir précisément le dimensionnement de l'infrastructure en fonction du niveau de service demandé, ces qualités font de DMS un excellent candidat à une intégration vers du Cloud Computing de type « SaaS » (Software as a Service).

Application	Services	Type
<i>Descartes Capture</i>	Hébergement + déploiement locale agents Options : • Encodage • Diffusion Live • Publication	Instance + datas
<i>Descartes Broadcast</i>	Live Streaming Options : • VOD (si datas hébergés)	Infra redondante Haute disponibilité.
<i>Descartes Engine</i>	Workflows personnalisés Encodage	Capacité CPU Haute performance.
<i>Media2Descartes</i>	Hébergement Options : • VOD	Instance + datas

Figure 9 - Offre de Service SaaS de DMS

Dans le cadre d'une Université Numérique en Région (UNR) ou d'un pôle de recherche et d'enseignement supérieur (PRES), il serait avantageux de disposer d'un Cloud universitaire privé afin de mutualiser le coût d'acquisition et d'exploitation d'une telle infrastructure. En effet, tous les établissements ne disposent pas des moyens financiers et humains requis pour acquérir, intégrer et maintenir ce type de services. De plus, pour des raisons de coûts, il serait tentant de se limiter à une infrastructure minimale, qui serait alors rapidement inadaptée si l'usage en devenait plus conséquent.

La mise en place d'une architecture « scalable » sur le long terme nécessite l'acquisition d'une base pérenne et évolutive qui permettrait de s'adapter à la demande par l'ajout de ressources matérielles et logicielles pré-packagées en vue d'une industrialisation du déploiement de façon automatique ou semi-automatique.

L'accès aux services pourrait-être conditionné par la participation active à l'infrastructure et au développement de DMS (mise à disposition de moyens matériels/financiers, d'ingénieurs systèmes/développeurs) ou bien par le paiement en fonction de l'usage des services. Pour cela, il faudrait mettre en place un système de monitoring d'accès aux services pour effectuer une tarification au plus juste.

## 7 Bibliographie

Livre blanc sur le podcast à l'Université, MESR-SG-STSI <http://www.anstia.fr/sites/default/files/livre-blanc-podcast-dec08b.pdf>

SOA in Practice The Art of Distributed Sytem Design Nicolai M. Josuttis, Edition O'REILLY ISBN : 978-0-596-52955-0

The Art of Scalability Abbott, Fisher, Edition Addison-Wesley ISBN : 978-0-13-703042-2

Présentation de DMS par F. Cadé à EUNIS 2011, Dublin - Irlande <http://www.eunis.ie/outline.html#full>