



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Σχεδιασμός και Ανάπτυξη Τμήματος Πληροφοριακού Συστήματος Αυτοματισμού Διαχείρισης Τηλεοπτικού Υλικού Design and Developmend of a Modular Multimedia Management Information System for Television
Όνοματεπώνυμο Φοιτητή	Παναγιώτης Αδαμόπουλος
Πατρώνυμο	Ηρακλής
Αριθμός Μητρώου	ΜΠΠΛ 14001
Επιβλέπων	Χρήστος Δουληγέρης, Καθηγητής

Ημερομηνία Παράδοσης: **Οκτώβριος 2016**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα
Βαθμίδα

Επώνυμο

Όνομα
Βαθμίδα

Επώνυμο

Όνομα
Βαθμίδα

Επώνυμο

ΕΥΧΑΡΙΣΤΙΕΣ

Η εργασία αυτή εκπονήθηκε στο πλαίσιο της μεταπτυχιακής μου διατριβής του ΠΜΣ Πληροφορικής.

Θα ήθελα να ευχαριστήσω τους προϊσταμένους μου αλλά και συναδέλφους μου στο Mega Channel για την βοήθειά τους στην χρήση του πολυμεσικού υλικού του καναλιού αλλά και των υποδομών του. Επίσης θα ήθελα να ευχαριστήσω τη γυναίκα μου Βασιλική για την υπομονή της κατά την εκπόνηση της παρούσας εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα κ. Δημήτρη Κοτσιφάκο για την απρόσκοπτη συνεργασία αλλά και υποστήριξη σε δύσκολες και καίριες στιγμές.

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική διατριβή παρουσιάζει, μέρος ενός αρθρωτού πληροφοριακού συστήματος τηλεοπτικών σταθμών. Πρόκειται για τον πυρήνα ενός συστήματος εννέα (9) εφαρμογών (εκ των οποίων τρεις (3) αποτελούν έτοιμες λύσεις) οι οποίες συνδυαζόμενες μεταξύ τους αυτοματοποιούν πλήρως τη ροή ενός τηλεοπτικού σταθμού από τη στιγμή λήψης βίντεο από την κάμερα ως και τη στιγμή μετάδοσής του στα διάφορα μέσα προβολής, τα οποία εκτός της τηλεόρασης περιλαμβάνουν το διαδίκτυο (youtube, video on demand στην ιστοσελίδα του σταθμού), τη διαχείριση διαφημιστικού υλικού, smart TV και άλλα. Το λογισμικό σέβεται πλήρως και αυτοματοποιημένα τόσο τα δικαιώματα προβολής προγράμματος μέσω των σημάτων καταλληλότητας όσο και την πιθανότητα μη προβολής υλικού σε επιλεγμένα μέσα όπως για παράδειγμα την άδεια αναμετάδοσης προγράμματος στην Ελλάδα αλλά όχι στο εξωτερικό κτλ.

Στην συνέχεια θα περιγραφούν διεξοδικά οι λειτουργικές του δυνατότητες, ενώ θα γίνει αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν. Ενδεικτικά θα αναφέρουμε ότι χρησιμοποιήθηκε για την υλοποίηση του συστήματος η γλώσσα προγραμματισμού Java και τεχνολογίες J2EE. Ως βασικό framework επιλέχτηκε το Spring 4, για την διασύνδεση με την βάση σε επίπεδο ORM(Object Relational Mapping) χρησιμοποιήθηκαν JPA Annotations και ως provider (σύστημα υλοποίησης) το hibernate, ενώ για την παραγωγή RestFul Services το module Spring Services. Χρησιμοποιήθηκαν επίσης Apache Tiles, Ajax τεχνολογίες, Json, Rest, Bootstrap, AngularJS κ.α.

Τέλος, ιδιαίτερη σημασία έχει δοθεί στην ασφάλεια του συστήματος με την χρήση του module Spring Security με Two Factor Authentication, τόσο για την ασφάλεια των πληροφοριών του σταθμού, όσο επίσης και για τα διαφορετικά επίπεδα ρόλων και τις διαφορετικές δυνατότητες σε υπολειτουργίες που διατίθενται αντιστοίχως.

ABSTRACT

This thesis presents a part of a modular information system for TV stations. This is the core of a system of nine (9) applications which together produced a full automation of the flow of a TV station from the video shooting time up to the video transmission. The software is fully automated and respects both the program viewing rights through health marks and the likelihood of non-promotional material on selected media such as a program broadcasting license in Greece but not abroad etc.

Then the functional capabilities will be described in detail and reference will be made to the technologies used. Indicatively mention that was used for the implementation of the Java programming language and J2EE technology system. As a basic framework was selected in Spring 4, the interface with the base level in ORM (Object Relational Mapping) JPA Annotations used as provider (delivery system) to hibernate, while production RestFul Services the module Spring Services. Also used Apache Tiles, Ajax technologies, Json, Rest, Bootstrap, AngularJS etc.

Finally, particular importance is given to the safety of the system using the module Spring Security with Two Factor Authentication, both for the safety of the station information, as well as different levels of roles and different possibilities in subfunctions allocated respectively.

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ	8
2	ΑΝΤΙΚΕΙΜΕΝΟ ΚΑΙ ΔΟΜΗ ΤΗΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΑΤΡΙΒΗΣ	12
2.1	ΥΠΗΡΕΣΙΕΣ ΔΙΑΔΙΚΤΥΟΥ	12
2.2	ΣΥΝΟΠΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	14
2.3	ΕΜΠΛΕΚΟΜΕΝΟΙ ΦΟΡΕΙΣ	15
2.4	ΔΙΑΔΙΚΑΣΙΑ ΚΑΙ ΠΗΓΕΣ ΠΑΡΑΓΩΓΗΣ VIDEO	15
2.5	ΜΗΧΑΝΙΣΜΟΙ ΚΑΙ ΔΙΑΔΙΚΑΣΙΕΣ.....	16
2.6	Η ΚΑΙΝΟΤΟΜΙΑ ΤΟΥ MEDIASTACK – ΥΠΟΔΟΜΗ	17
2.7	ΥΛΟΠΟΙΗΣΗ ΒΑΣΙΚΗΣ ΥΠΗΡΕΣΙΑΣ	18
2.8	SERVER STACK DIAGRAM.....	19
2.9	MEDIASTACK SERVERS IMPLEMENTATION	20
2.10	MEDIASTACK SOFTWARE IMPLEMENTATION.....	20
2.11	MEDIA STACK DEVELOPMENT TOOLS	21
2.12	ΕΝΣΩΜΑΤΩΣΗ ΔΟΚΙΜΑΣΤΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΕΛΕΓΧΟΣ ΛΕΙΤΟΥΡΓΙΑΣ	21
3	ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ	23
3.1	JAVA	23
3.2	SPRING MVC	24
3.2.1	<i>DispatcherServlet</i>	25
3.3	SPRING TOOL SUITE.....	26
3.4	POSTGRESQL	27
3.5	JBOSS	27
3.6	MAVEN	28
3.7	JAVASCRIPT	29
3.8	ANGULARJS	30
3.9	HTML.....	31
3.10	CASCADING STYLE SHEETS.....	32
3.11	BOOTSTRAP.....	32
4	ΤΕΧΝΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ	33
4.1	RESPONSIVE DESIGN.....	33
4.2	MODEL – VIEW - CONTROLLER.....	33
4.3	SPRING MVC + ANGULAR	34
4.4	VIEW LAYER	34
4.5	CONTROLLER LAYER.....	34
4.6	FRONTEND SERVICES LAYER	34
4.7	BACKEND.....	35
4.8	ROUTER LAYER	35
5	ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΟΣ	36
5.1	ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΑΣ – ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ.....	36
5.2	ΒΑΣΙΚΕΣ ΟΝΤΟΤΗΤΕΣ	37
5.2.1	<i>Στόχος του συστήματος</i>	37
5.2.2	<i>Domain Model</i>	37
5.2.3	<i>Data Model</i>	38
5.2.4	<i>Video content</i>	38
5.3	ΑΡΧΙΤΕΚΤΟΝΙΚΗ	39
5.4	ΔΙΑΔΙΚΑΣΙΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΤΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ VIDEO	41
5.5	ΠΑΡΑΓΩΓΗ ΠΡΩΤΟΓΕΝΩΝ ΑΡΧΕΙΩΝ VIDEO	41
5.6	ΔΙΑΧΕΙΡΙΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΔΗΜΟΣΙΕΥΣΕΩΝ - «PROGRAM MANAGER»	43

5.7	ΕΠΙΚΟΙΝΩΝΙΑ PROGRAM MANAGER ΜΕ MEDIASTACK	46
5.7.1	Υλοποίηση κανόνων	46
5.8	VIDEO CM - USER INTERFACES	47
5.9	ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	57
5.10	ΑΝΑΛΥΣΗ ΠΙΝΑΚΩΝ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.....	58
5.11	ΠΕΡΙΓΡΑΦΗ ΑΥΤΟΜΑΤΙΣΜΩΝ	61
5.12	FFMPEG	61
5.13	ALFRESCO.....	62
5.14	RABBIT MQ	64
5.15	HAZELCAST.....	65
5.16	MEDIA OBJECT SERVER COMMUNICATION PROTOCOL	67
5.17	ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ	68
6	ΡΟΗ ΕΦΑΡΜΟΓΗΣ.....	69
6.1	ADMINISTRATOR	69
6.2	ΔΙΑΧΕΙΡΙΣΤΗΣ	72
6.3	ΥΠΕΥΘΥΝΟΣ ΡΟΗΣ	75
6.4	ΧΡΗΣΤΗΣ	76
6.5	ΤΕΧΝΙΚΟΣ ΕΞΟΠΛΙΣΜΟΣ	77
6.6	ΘΕΑΤΗΣ.....	77
7	ΒΙΒΛΙΟΓΡΑΦΙΑ	82

1 ΕΙΣΑΓΩΓΗ

Η τηλεόραση στην Ελλάδα ξεκίνησε για πρώτη φορά τη λειτουργία της το 1960,[1] με κάποιες πειραματικές εκπομπές που δεν μονιμοποιήθηκαν, ενώ ξεκίνησε να εκπέμπει τακτικά από το 1966 από δημόσιο τηλεοπτικό σταθμό και από δεύτερο σταθμό που λειτούργησε από τις Ένοπλες Δυνάμεις. Κατά τη διάρκεια της Χούντας των Συνταγματαρχών παγιώθηκε η δομή του προγράμματος με μείγμα ψυχαγωγίας και ενημέρωσης, η οποία ήταν τότε λογοκριμένη. Ο κρατικός έλεγχος στην τηλεόραση διατηρήθηκε και στη μεταπολίτευση, οπότε η ενημέρωση ήταν περισσότερο ελεύθερη, αλλά όχι πάντα αντικειμενική.[2]



Εικόνα 1: Η Τ.Ε.Δ., η πρώτη μορφή τηλεόρασης κοντά στα σημερινά δεδομένα

Το 1989, ενώ ήδη είχαν αρχίσει να εκπέμπουν ιδιωτικοί ραδιοφωνικοί και τηλεοπτικοί σταθμοί, επιτράπηκε η λειτουργία ιδιωτικών τηλεοπτικών σταθμών, οι οποίοι σταδιακά επικράτησαν σε τηλεθέαση και επιρροή και επιδόθηκαν σε έντονο ανταγωνισμό στον ενημερωτικό και στον ψυχαγωγικό τομέα. Το τηλεοπτικό τοπίο, όπου κατά καιρούς λειτουργούσαν γύρω στα 150 κανάλια σε όλη τη χώρα, παρέμενε όμως άναρχο[3] λόγω των καθυστερήσεων του κράτους να χορηγήσει μόνιμες άδειες.

Μέχρι και το καλοκαίρι του 2013 υπήρχαν τρία δημόσια κανάλια, η ET1 και η NET σαν εξέλιξη των δύο αρχικών, και η ET3 που εξέπεμπε από το 1988, και τα οποία σταμάτησαν να εκπέμπουν όταν έκλεισε η EPT για να δημιουργηθεί νέα εταιρεία, η Νέα Ελληνική Ραδιοφωνία, Ίντερνετ και Τηλεόραση (NEPIT)· μέχρι να ξεκινήσει το πρόγραμμα της η τελευταία, τον ρόλο του δημόσιου ραδιοτηλεοπτικού φορέα επιτελούσε η μεταβατική Δημόσια Τηλεόραση. Η EPT επανασυστάθηκε και λειτουργεί εκ νέου από τα μέσα του 2015. Υπάρχουν ακόμα δεκάδες ιδιωτικά κανάλια, πανελλαδικής και τα περισσότερα τοπικής και περιφερειακής εμβέλειας.[4] Λόγω της οικονομικής κρίσης πολλά κανάλια έκλεισαν ή περιόρισαν σημαντικά τις δραστηριότητές τους. Ακόμη υπάρχουν επίγεια συνδρομητικά δίκτυα, ενώ για ένα διάστημα αναμεταδίδονταν και κάποια ξένα δορυφορικά. Στην Ελλάδα δεν υπάρχουν καλωδιακά δίκτυα,[5] όμως κάποια ιδιωτικά και η EPT εξέπεμπαν δορυφορικό πρόγραμμα, κυρίως για τους απόδημους Έλληνες. Περιορισμένες είναι οι προσπάθειες για αποκλειστικά ιντερνετικό περιεχόμενο ή για ιντερνετικά κανάλια.

Στον τομέα του προγράμματος, τα ιδιωτικά δίκτυα ακολουθούν την καθιερωμένη διάκριση μεταξύ ενημέρωσης και ψυχαγωγίας, αν και η αναλογία μεταξύ των τομέων δεν είναι σταθερή, ενώ τα δημόσια δίκτυα δίνουν έμφαση στον πολιτισμό και στην εκπαίδευση. Στην ιστορία της ελληνικής τηλεόρασης παρήχθησαν εκατοντάδες σίριαλ, τηλεταινίες και θεατρικά έργα όπως και πολλά ενημερωτικά και ψυχαγωγικά προγράμματα με σημαντική επίδραση στο κοινό. Πάντως, λόγω της οικονομικής κρίσης, η παραγωγή πρωτότυπου προγράμματος σημειώνει κάμψη και κυριαρχούν οι ξένες σειρές και οι επαναλήψεις.[6]



Εικόνα 2: Ιδιωτική τηλεόραση

Εύκολα, λοιπόν, μπορεί κανείς ότι τα πολυμέσα είναι μία από τις πιο πολυσυζητημένες και ενδιαφέρουσες τεχνολογίες αφού από την εμφάνιση της τηλεόρασης τη δεκαετία του '60, που η επεξεργασία οπτικού και ακουστικού υλικού γινόταν με μη αυτόματους,

υψηλής βέβαια τεχνολογίας για την εποχή, τρόπους ως και σήμερα, αποτελεί στοίχημα η επεξεργασία και διανομή του τηλεοπτικού σήματος.

Το ενδιαφέρον αυτό είναι απόλυτα δικαιολογημένο, αφού τα πολυμέσα αποτελούν το σημείο συνάντησης πέντε μεγάλων βιομηχανιών:

- της πληροφορικής (από την περίοδο 1966 - σήμερα)
- των τηλεπικοινωνιών
- των ηλεκτρονικών εκδόσεων
- της βιομηχανίας audio και video
- της βιομηχανίας της τηλεόρασης και του κινηματογράφου.

Μια ανάλογη αναστάτωση επέφερε και η εμφάνιση της επιστήμης των δικτύων υπολογιστών στην δεκαετία του 70, φέρνοντας πιο κοντά την πληροφορική με τις τηλεπικοινωνίες.

Ενώ αυτή η προσέγγιση οδήγησε σε προϊόντα που στόχευαν κυρίως στην αγορά των επιχειρήσεων. Τα πολυμέσα έκαναν κάτι περισσότερο, διεύρυναν την αγορά των προϊόντων των παραπάνω βιομηχανιών που πλέον στοχεύουν και στους καταναλωτές.

Η πληθώρα και οι ποικιλία των νέων προϊόντων καθώς και η προσπάθεια εκμετάλλευσης του ενδιαφέροντος που επέδειξε το αγοραστικό κοινό, για την τεχνολογία των πολυμέσων, συνετέλεσαν στην σύγχυση που υπάρχει ακόμα και σήμερα όσον αφορά στο τι είναι και τι δεν είναι ένα σύστημα πολυμέσων.

Πλέον, όλο και περισσότερο χρησιμοποιούμε αυτό που ονομάζεται "διαδραστικό περιεχόμενο" και σίγουρα οι περισσότεροι θεωρούμε ως κάτι δεδομένο τους πολυμεσικούς τρόπους διεπαφής, αναμειγνύοντας ήχο, κίνηση, εικόνα, βίντεο και animation. Οι πολυμεσικές εφαρμογές, τίτλοι πολυμέσων ή πολυμέσα αποτελούν σήμερα μια από τις πιο δυναμικά εξελισσόμενες τεχνολογίες στον χώρο της πληροφορικής, με πρακτικές εφαρμογές σε πολλούς διαφορετικούς τομείς δραστηριότητας, όπως εκπαίδευση, επαγγελματική κατάρτιση, αγορά-διαφήμιση, παρουσιάσεις, ψυχαγωγία, ηλεκτρονικά παιχνίδια, περίπτερα παροχής πληροφοριών (kiosks), τηλεδιάσκεψη, Internet κλπ.

Η επέλαση της τεχνολογίας και η αυξημένη ζήτηση, μας έχουν φέρει - εδώ και χρόνια - στο σημείο, να μπορεί ο καθένας να στήσει τον δικό του server περιλαμβάνοντας ό,τι θέλει. Η απλούστευση των διαδικασιών εγκατάστασης ενός server και συγκεκριμένων εφαρμογών, επιτρέπει στους περισσότερους να έχουν έναν ιδιόκτητο server διαμοιρασμού, συγχρονισμού, αρχείων mail, multimedia και ένα σωρό δυνατότητες ακόμα. Άλλωστε, είναι μύθος πως ένας sever χρειάζεται δυνατό ή πολύ σύγχρονης τεχνολογίας υπολογιστή. Όπως - θα το επαναλάβω - μύθος είναι ότι μπορεί να γίνει μόνο από έμπειρους, καθώς λύσεις όπως το Sandstorm, έρχονται για να δώσουν αυτήν την επιλογή ακόμα και σε αρχάριους.

Στην παρούσα διατριβή θα προσπαθήσουμε να κάνουμε ένα τεχνολογικό άλμα στον τρόπο διαχείρισης του τηλεοπτικού προϊόντος αλλά και στην – με ένα διαφορετικό τρόπο – εκμετάλλευση των υφιστάμενων υποδομών ενός τηλεπτικού σταθμού με γνώμονα την πλήρη αυτοματοποίηση της διαδρομής που ακολουθεί ένα video, από

τη στιγμή της λήψης του ως και την προβολή του. Βεβαίως ένα τέτοιο σύστημα είναι ιδιαίτερα εκτενές και η πλήρης ανάπτυξή του ξεφεύγει από τα όρια μιας μεταπτυχιακής διατριβής, παρόλα αυτά έχει αναπτυχθεί και θα παρουσιαστεί ένα καίριο μέρος αυτού που έχει να κάνει με την επανακωδικοποίηση του υλικού, την επεξεργασία του αλλά και την εκμετάλλευση του πρωτοκόλλου MOS για την επικοινωνία με τους Video Servers.

Το πρωτόκολλο MOS (Media Object Server Communications Protocol (MOS) αποτελεί επέκταση της γλώσσας σήμανσης ανοιχτών προτύπων XML, αναπτύχθηκε αρχικώς από το δίκτυο Associated Press και υιοθετήθηκε από την τηλεοπτική και ραδιοφωνική κοινότητα λόγω της ευχρηστίας και ακρίβειάς του. Το Media Object Server Communications Protocol (MOS) αποτελεί πλέον έργο ανοιχτού κώδικα με μια τεράστια κοινότητα προγραμματιστών και τεχνικών να το υποστηρίζουν.

2 ΑΝΤΙΚΕΙΜΕΝΟ ΚΑΙ ΔΟΜΗ ΤΗΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΑΤΡΙΒΗΣ

Το παρόν λογισμικό παραδίδεται στο πλαίσιο της μεταπτυχιακής διατριβής με θέμα «Σχεδιασμός και Ανάπτυξη Τμήματος Πληροφοριακού Συστήματος Αυτοματισμού Διαχείρισης Τηλεοπτικού Υλικού» και αφορά σε μια πλατφόρμα που εξυπηρετεί την ηλεκτρονική διαχείριση επιχειρήσεων παραδοσιακής, καλωδιακής και διαδικτυακής τηλεόρασης.

Στο παρόν κείμενο θα αναλυθούν τα αρχιτεκτονικά στοιχεία του συστήματος, καθώς και οι επιλογές υλοποίησης. Γίνεται επίσης ανάλυση των υποστηρικτικών δομών που είναι απαραίτητες για τη λειτουργία του συστήματος. Ειδικά για τους σκοπούς επίδειξης και διαμόρφωσης των χαρακτηριστικών οθονών της διαδικτυακής εφαρμογής έχουν χρησιμοποιηθεί δοκιμαστικά δεδομένα.

2.1 ΥΠΗΡΕΣΙΕΣ ΔΙΑΔΙΚΤΥΟΥ

Μια υπηρεσία διαδικτύου είναι μια διεπαφή που περιγράφει μια συλλογή υπηρεσιών που είναι προσβάσιμες μέσω του διαδικτύου με καθορισμένα XML μηνύματα. Περιγράφεται από μια XML περιγραφή, που λέγεται περιγραφή της υπηρεσίας. Περιλαμβάνει όλες τις λεπτομέρειες που είναι απαραίτητες για την αλληλεπίδραση με την υπηρεσία, όπως η μορφή των μηνυμάτων και τα πρωτόκολλα μεταφοράς. Η διεπαφή κρύβει τις λεπτομέρειες της εφαρμογής, γεγονός που επιτρέπει να χρησιμοποιηθεί ανεξάρτητα από την πλατφόρμα λογισμικού ή υλικού στην οποία εφαρμόζεται και ανεξάρτητα από τη γλώσσα προγραμματισμού που έχει χρησιμοποιηθεί.

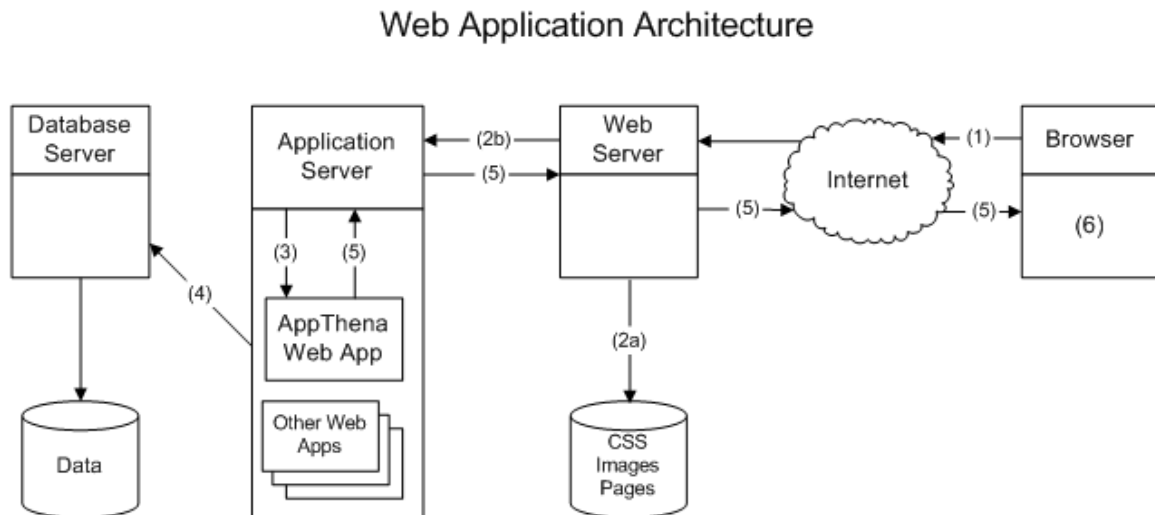
Η αρχιτεκτονική των υπηρεσιών διαδικτύου βασίζεται στην αλληλεπίδραση μεταξύ τριών οντοτήτων με τους εξής ρόλους: παροχή της υπηρεσίας (Service Provider), αποθήκευση της υπηρεσίας (Service Registry) και αίτηση για παροχή της υπηρεσίας (Service Requestor). Οι συνδιαλλαγές περιλαμβάνουν τη δημοσίευση, την εύρεση και τη δέσμευση υπηρεσίας, όπως φαίνεται στην εικόνα που ακολουθεί.

2.2 ΟΙ ΡΟΛΟΙ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΜΙΑΣ ΥΠΗΡΕΣΙΑΣ ΔΙΑΔΙΚΤΥΟΥ

Παροχή υπηρεσίας: Είναι η πλατφόρμα που φιλοξενεί την πρόσβαση στην υπηρεσία.

Αίτηση για παροχή υπηρεσίας: Είναι η εφαρμογή που αναζητά την υπηρεσία και αρχικοποιεί την συνδιαλλαγή με την υπηρεσία. Τον ρόλο αυτό μπορεί να τον έχει μια άλλη υπηρεσία (υπηρεσία διαδικτύου).

Αποθήκευση της υπηρεσίας: Είναι ένας χώρος αποθήκευσης, όπου οι παροχείς υπηρεσιών αποθηκεύουν τις περιγραφές των υπηρεσιών τους. Οι αιτήσεις για παροχή υπηρεσίας απευθύνονται στη συγκεκριμένη οντότητα για να πάρουν πληροφορίες για την περιγραφή της υπηρεσίας.



Εικόνα 3: Βασική Αρχιτεκτονική εφαρμογών διαδικτύου

2.3 ΛΕΙΤΟΥΡΓΙΕΣ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΩΝ ΥΠΗΡΕΣΙΩΝ ΔΙΑΔΙΚΤΥΟΥ

Για να επωφεληθεί μια εφαρμογή από μια υπηρεσία διαδικτύου πρέπει να συμβούν τρεις συμπεριφορές:

- δημοσίευση της συμπεριφοράς της υπηρεσίας,
- αναζήτηση ή εύρεση της περιγραφής της υπηρεσίας και
- δέσμευση ή επίκληση των υπηρεσιών που βασίζονται στην περιγραφή της υπηρεσίας.

Αυτές οι συμπεριφορές μπορούν να συμβούν κάθε μια απλά ή επαναληπτικά. Πιο συγκεκριμένα οι λειτουργίες είναι οι εξής:

Δημοσίευση: Για να είναι προσβάσιμη, η περιγραφή μιας υπηρεσίας θα πρέπει να δημοσιευτεί, έτσι ώστε ο αιτών της υπηρεσίας να μπορεί να την βρει.

Εύρεση: Στην λειτουργία αναζήτησης ο αιτών της υπηρεσίας ανακτά την περιγραφή της απευθείας ή ρωτά την καταχώρηση της υπηρεσίας για τον τύπο της υπηρεσίας που χρειάζεται. Η λειτουργία της αναζήτησης μπορεί να εμπλακεί σε δυο διαφορετικές φάσεις του κύκλου ζωής για τον αιτών της υπηρεσίας: Στο χρόνο σχεδιασμού για να ανακτηθεί η διεπαφή της περιγραφής υπηρεσίας για την ανάπτυξη του προγράμματος και στον χρόνο εκτέλεσης για να ανακτηθεί η δέσμευση και η τοποθεσία της υπηρεσίας για επίκληση.

Δέσμευση: Τελικά μια υπηρεσία απαιτείται να εμπλακεί. Στην λειτουργία της δέσμευσης ο αιτών της υπηρεσίας εμπλέκει ή αρχικοποιεί μια αλληλεπίδραση με την υπηρεσία στο χρόνο εκτέλεσης χρησιμοποιώντας τις λεπτομέρειες της «δέσμευσης» για να εντοπίσει, να επικοινωνήσει και να επικαλεστεί την υπηρεσία.

2.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΜΙΑΣ ΥΠΗΡΕΣΙΑΣ ΔΙΑΔΙΚΤΥΟΥ

Υπηρεσία. Μια υπηρεσία διαδικτύου είναι μια διεπαφή που περιγράφεται από μια περιγραφή της υπηρεσίας και η υλοποίησή της είναι η υπηρεσία. Μια υπηρεσία είναι μια ενότητα λογισμικού, που εφαρμόζεται σε πλατφόρμες προσβάσιμες μέσω δικτύου και παρέχεται από τον παροχέα υπηρεσιών. Μπορεί να κληθεί ή να αλληλεπιδράσει με τον αντίστοιχο καλούντα της υπηρεσίας. Μπορεί να λειτουργήσει και ως καλών χρησιμοποιώντας άλλες υπηρεσίες διαδικτύου στην υλοποίηση της.

Περιγραφή της υπηρεσίας. Η περιγραφή της υπηρεσίας περιέχει τις λεπτομέρειες της διεπαφής και της υλοποίησης της υπηρεσίας. Αυτό περιλαμβάνει τους τύπους δεδομένων, τις λειτουργίες, τις πληροφορίες για τη δέσμευση της υπηρεσίας και την τοποθεσία του δικτύου. Μπορεί επίσης, να περιλαμβάνει κατηγοριοποίηση και άλλα μεταδεδομένα για να φιλοξενηθεί, ανακαλυφθεί και να χρησιμοποιηθεί από τους καλούντες της υπηρεσίας. Η περιγραφή της υπηρεσίας μπορεί να δημοσιευθεί σε έναν καλούντα της υπηρεσίας ή σε έναν καταχωρητή υπηρεσιών.

Η αρχιτεκτονική των υπηρεσιών διαδικτύου εξηγεί πως μπορεί να γίνει η αρχικοποίηση των στοιχείων και να εφαρμοστούν οι λειτουργίες με διαλειτουργικό τρόπο.

2.5 ΣΥΝΟΠΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η πλατφόρμα διαχείρισης πολυμεσικού υλικού εκμεταλεύεται πλήρως τη λογική των εφαρμογών διαδικτύου και έχει αναπτυχθεί με αρθρωτή λογική και αποτελείται από επιμέρους λογισμικά τα οποία επικοινωνούν μεταξύ τους με τη χρήση Restful Web Services.

Το MediaStack αποτελεί μια ολοκληρωμένη λύση συστήματος διαχείρισης πολυμέσων. Αποτελεί ένα πλήρως αυτοματοποιημένο σύστημα που χρησιμοποιεί ως input κάποια πηγή βίντεο (κάμερες, video servers, transcoders, video streams κ.α.) σε οποιοδήποτε format (raw, mp4, avi, mov, flv κ.α.) και εξάγει κωδικοποιημένα βίντεο καθώς και ένα σύνολο πλήρως ενημερωμένων μετα-δεδομένων με στόχο την αυτόματη προώθησή τους στα μέσα προβολής, όπως ψηφιακή τηλεόραση, Smart TV, Internet Websites, Youtube, φορητές συσκευές, CDN Providers κτλ, είτε ως live streaming, είτε ως video on demand.

Το πληροφοριακό σύστημα, έχει σχεδιαστεί ώστε να είναι δυνατή η επεξεργασία εκατοντάδων video ταυτόχρονα σε ποιότητα ως και Full HD ακόμα και αν είναι πολύ μεγαλύτερα σε όγκο. Τα εξαγόμενα παράγονται σε πολλαπλές ποιότητες με την ποιότητα του video να προσαρμόζεται αυτόματα ανάλογα με την ταχύτητα σύνδεσης του χρήστη.

Για τον σωστό συντονισμό της ροής του προγράμματος ενός τηλεοπτικού σταθμού, πρέπει το σύστημα να μπορεί να διαχειριστεί όχι μόνο την πληθώρα των video αλλά και των συνοδευτικών τους πληροφοριών. Ένα video έχει τίτλο, περιγραφή, επιτρεπόμενες ημερομηνίες προβολής, συνδεδεμένη φωτογραφία, πολλαπλές ποιότητες, έχει “δημοσιευτεί” σε διάφορα σημεία, έχει λέξεις-κλειδιά (tags) για αναζήτηση κλπ. Όλες αυτές τις πληροφορίες τις ονομάζουμε metadata και τις διαχειριζόμαστε μαζί με τα video που αυτές αφορούν, μέσω της εφαρμογής MediaStack.

2.6 ΕΜΠΛΕΚΟΜΕΝΟΙ ΦΟΡΕΙΣ

Για την ορθή αλλά και ολοκληρωμένη εκτέλεση του λογισμικού είναι απαραίτητη η συμβολή τριών παραγόντων:

1. **Τηλεοπτικός Σταθμός.** Στο κανάλι βρίσκονται τα original videos και τα metadata τους και καταχωρούνται στο MediaStack. Υπάρχουν πολλοί εμπλεκόμενοι φορείς όπως αναλύονται παρακάτω.

2. **CDN.** Η χρήση CDN και NetStorage κρίνεται ως βέλτιστη πρακτική αλλά όχι απαραίτητη. Η αιτία είναι ο τεράστιος όγκος πολυμεσικού υλικού που απαιτεί όχι μόνο μεγάλες δυνατότητες αποθήκευσης αλλά και ισχυρή σύνδεση στο διαδίκτυο για την απρόσκοπτη μετάδοση περιεχομένου. Όλα αυτά μπορεί να τα παρέχει κάποιο Content Delivery Network όπως η Akamai, η Amazon κ.α. Όλα τα video για να παίξουν πρέπει να σταλούν στην υπηρεσία CDN ή σε κάποιο πάροχο, ώστε να εξασφαλίζει το απαραίτητο bandwidth (ταχύτητες) ώστε παντού στον κόσμο να διανέμεται το περιεχόμενό μας χωρίς προβλήματα και καθυστερήσεις. Επίσης αποτελεί και μια ασπίδα προστασίας ώστε το μέσο μετάδοσης να είναι απρόσβλητο από τις χιλιάδες επιθέσεις που δέχεται.

3. **Κανάλια Προβολής.** Ο κάθε στόχος προβολής αντλεί όλες τις πληροφορίες σχετικά με τα video (metadata) από το MediaStack και «σχεδιάζει» αυτόματα τον τρόπο προβολής και τις αλληλεπιδράσεις του περιεχομένου τους. Τα μέσα προβολής «παίζουν» τα video μέσω λιστών (παρόμοιων με τις playlists των mp3 players) που τις ονομάζουμε Collections.

2.7 ΔΙΑΔΙΚΑΣΙΑ ΚΑΙ ΠΗΓΕΣ ΠΑΡΑΓΩΓΗΣ VIDEO

Αρχικά παράγεται ένα video με κάποιο από τους παρακάτω τρόπους.

- Από Video Library. Χρησιμοποιείται κυρίως για σειρές.
- Από το σύστημα καταγραφής του αέρα, απ' όπου γίνεται η εξαγωγή του video που μας ενδιαφέρει.
- Καταγραφή του αέρα σε **real-time από PC με ειδικές κάρτες video capture**. Χρήση κυρίως για ομιλίες και live events.
- **UGC (User Generated Content)** δηλαδή video που έρχεται μέσω τρίτων – πχ από κινητά τηλέφωνα ή θεατές
- Περιεχόμενο που προκύπτει από montage των original όπως π.χ. η δημιουργία αποσπασμάτων trailers, preview κτλ.



Εικόνα 4: Πηγές λήψης τηλεοπτικού προγράμματος

2.8 ΜΗΧΑΝΙΣΜΟΙ ΚΑΙ ΔΙΑΔΙΚΑΣΙΕΣ

- Οι χρήστες παράγουν ή/και εισάγουν τα video στο MediaStack μαζί με τα metadata τους και τα τοποθετούν στα κατάλληλα Collections (=playlists) ρυθμίζοντας και την σειρά με την οποία παίζουν.
- Μόλις ο χρήστης δημοσιεύσει ένα video, ένας αυτόματος μηχανισμός του MediaStack κωδικοποιεί το video σε κατάλληλη ποιότητα και το στέλνει στην υπηρεσία CDN, στο Youtube κλπ. Ο μηχανισμός αυτός παρακολουθεί την επιτυχή του αποστολή και ενημερώνει ανάλογα το MediaStack με τη χρήση Agents που έχουν ειδικά γραφεί σε Python και οι οποίοι ελέγχονται από ένα μηχανισμό messaging όπως το RabbitMQ. Με αυτόματο επίσης μηχανισμό του MediaStack, τα metadata δημοσιεύονται (σε μορφή API) σε κατάλληλο σημείο, προσπελάσιμο από τους στόχους προβολής. Έτσι δημοσιεύεται το πρόγραμμα του σταθμού, τα Collections και τα περιεχόμενά τους, οι περιλήψεις και οι περιγραφές, οι ημερομηνίες ισχύος, τα tags κλπ. Αντίστοιχα αν η αποστολή γίνεται στο Youtube, παράγεται αντίστοιχο API σύμφωνα με τις προδιαγραφές του YouTube.

2.9 Η ΚΑΙΝΟΤΟΜΙΑ ΤΟΥ MEDIASTACK – ΥΠΟΔΟΜΗ

Το MediaStack αναπτύχθηκε αξιοποιώντας με εντελώς νέο τρόπο μια υφιστάμενη πλατφόρμα συνδυάζοντας ένα πλήθος από τεχνολογίες.

Βασικό σημείο στη δομή του είναι η open source version ενός από τα πιο διαδεδομένα Asset Management Systems (AMS), του **Alfresco**.

Η δόμηση είναι ως εξής:

Το περιεχόμενο (Content) είναι το βασικό στοιχείο πληροφορίας. Το Content αφορά ένα video. Στην πραγματικότητα δεν είναι το ίδιο το video αλλά metadata αυτού. Το content συνδέεται και με άλλα contents όπως πχ είναι η φωτογραφία που χρησιμοποιείται σαν thumbnail. Το content μπορεί να έχει επιπλέον χαρακτηριστικά (aspects) ανάλογα με την φύση του. Για παράδειγμα μπορεί να έχει το Genre aspect το οποίο κατηγοριοποιεί το content με πληροφορία χρήσιμη για το Youtube (αν το Genre Aspect είναι “Ειδήσεις”, τότε θα πάει στο Youtube Channel Ειδήσεις του τηλεοπτικού σταθμού). Στο Alfresco σώζονται τα metadata του content και οι συσχετίσεις μεταξύ τους. Τα video με τις ποιότητές τους και το που έχουν σταλεί (CDN, YouTube κλπ) είναι πληροφορίες του content. Το σχήμα είναι επεκτάσιμο σε πολύ μεγάλο βαθμό. Υπότιτλοι, σενάρια, ηχητικά, κείμενα, κλπ είναι content που μπορούν να προστεθούν στο Alfresco και να συσχετιστούν μεταξύ τους. Το σύστημα υποστηρίζει ταχύτατη αναζήτηση καθώς κάθε content γίνεται index μόλις εισάγεται στο Alfresco. Πολύ σημαντικό είναι ότι υπάρχουν δικαιώματα πρόσβασης που συνδέονται με το κεντρικό σύστημα διαχείρισης χρηστών και ομάδων (LDAP Directory) και ο χρήστης βλέπει / αλλάζει / ρυθμίζει ότι επιτρέπεται.

Το Alfresco ελέγχεται από **την βασική εφαρμογή του MediaStack** που τρέχει σε Red Hat JBoss και είναι γραμμένη στο περιβάλλον Spring 4 με εργαλείο ανάπτυξης το Eclipse κατάλληλα διαμορφωμένο. Για να αποφευχθούν διπλές υλοποιήσεις, για πιο εύκολη διασύνδεση μεταξύ των συστημάτων και για υπάρχει ξεκάθαρη λειτουργία, όλη η εφαρμογή βασίζεται σε API (Application Programming Interface). Το MediaStack έχει δικό του API και μιλά στο API του Alfresco. Το MediaStack «μιλά» με API στο ERP του τηλεοπτικού σταθμού που είναι το βασικό σύστημα του σταθμού που χειρίζεται το πρόγραμμα, τα δικαιώματά του, τις διαφημίσεις, την ροή του σταθμού, την προώθησή του στην τιμολόγηση και γενικά την κάθετη εφαρμογή για το κανάλι μας.

Βασικό υποσύστημα της επεξεργασίας και της δημοσίευσης της πληροφορίας αποτελεί ένα υποσύστημα του MediaStack που ονομάζουμε **Program Manager (PM)** που υλοποιεί τους συνδυασμούς του περιεχομένου στα σημεία δημοσίευσης με τις κατάλληλες ποιότητες. Επίσης δίνει τις κατάλληλες εντολές στον Controller για επεξεργασία και αποστολή video (δείτε παρακάτω).

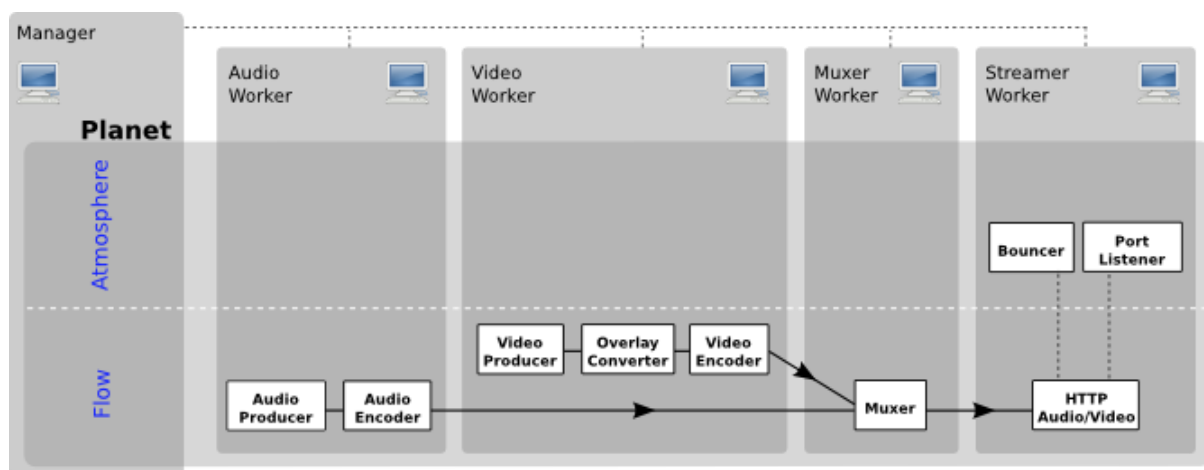
Το MediaStack χρησιμοποιεί **RDMBS (PostgreSQL Server)** για να αποθηκεύσει τα collections αλλά και διάφορα στάδια της δημοσίευσης της πληροφορίας. Κάθε content μπορεί να ανήκει σε ένα ή περισσότερα Collections τα οποία δημοσιεύονται σε ένα ή περισσότερα publishing points όπως η εταιρία CDN, το YouTube, η DMZ του σταθμού (metadata για

εξωτερικούς συνεργάτες). Αυτόματες διαδικασίες υλοποιούν τους συσχετισμούς και το publishing.

Ένας ολόκληρος μηχανισμός που τον ονομάζουμε για συντομία **Controller** αναλαμβάνει να μοιράσει και να αυτοματοποιήσει τις εργασίες στα κατάλληλα μηχανήματα. Η επανακωδικοποίηση (transcoding) η αποστολή του video FTP, ο έλεγχος της σωστής αποστολής, η ενημέρωση του Alfresco για την επιτυχή ή μη παραγωγή/αποστολή της νέας ποιότητας και πλήθος άλλων διεργασιών είναι δουλειά του Controller. Ο Controller πρέπει να μιλά μέσω API.

Οι πληροφορίες που ανταλλάσσουν μεταξύ τους τα υποσυστήματα είναι χιλιάδες κάθε ημέρα. Για τη διασφάλιση της πληροφορίας μεταξύ των συστημάτων, έχει χρησιμοποιηθεί **Message Queuing software, το Rabbit MQ** ώστε να μην «χάνεται» κανένα μήνυμα ακόμα και σε περίπτωση failure κάποιου από αυτά. **Η διαδικασία λοιπόν επικοινωνίας των υποσυστημάτων είναι εντελώς ασύγχρονη.**

Για το playout του video έχει χρησιμοποιηθεί ένας ανοιχτού κώδικα streaming server σε Linux, βασισμένος στο FluMotion.



Εικόνα 5: Streaming Video Server Architecture

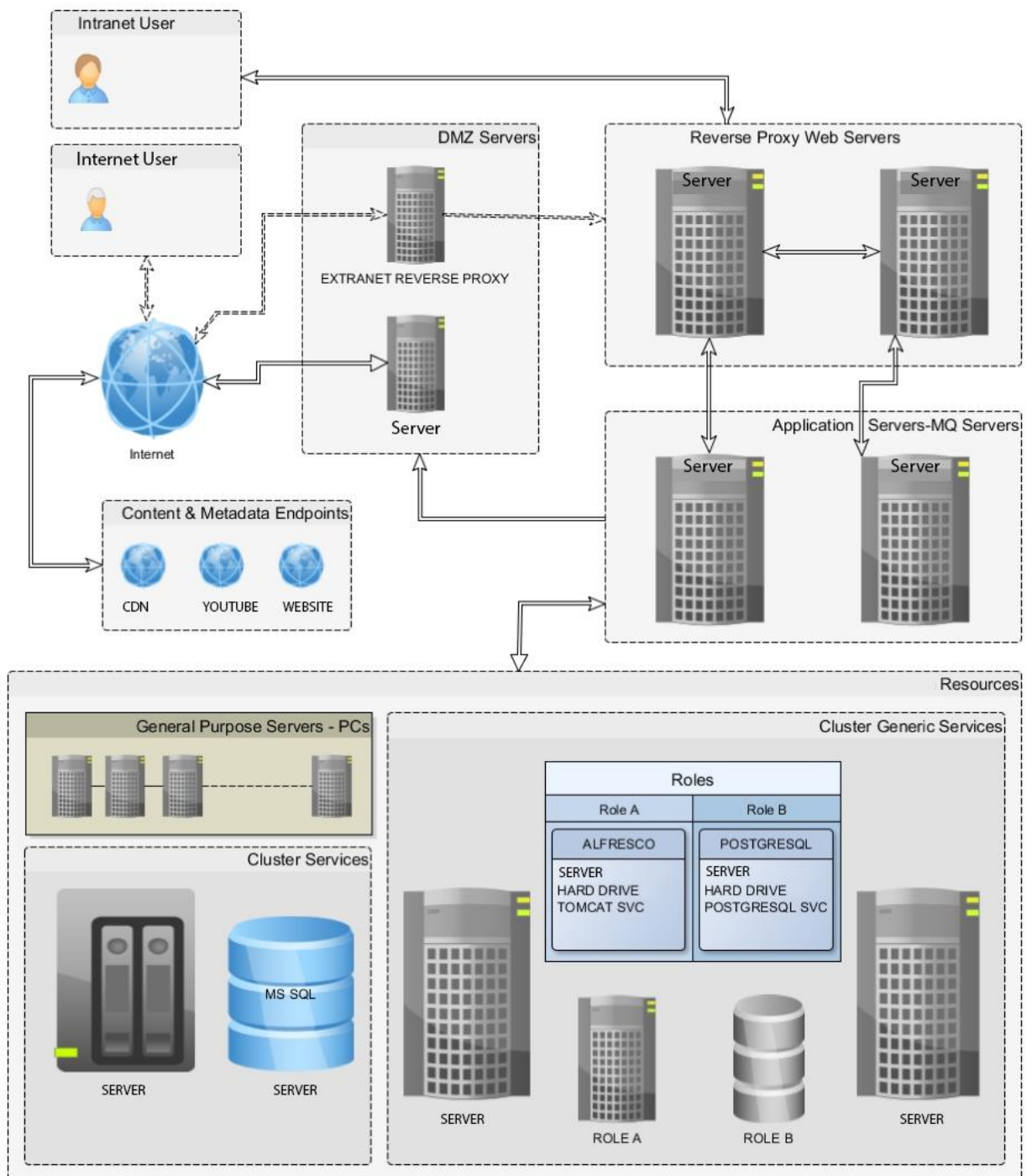
2.10 ΥΛΟΠΟΙΗΣΗ ΒΑΣΙΚΗΣ ΥΠΗΡΕΣΙΑΣ

Η υλοποίηση της υπηρεσίας περιλαμβάνει δύο στάδια εφαρμογής. Το λειτουργικό κομμάτι και το κομμάτι παρουσίασης.

Αυτά τα δύο στάδια αναπτύχτηκαν ως διαφορετικές οντότητες και συγχωνεύτηκαν ώστε να δημιουργηθεί το τελικό αποτέλεσμα. Στο τμήμα της παρουσίασης επιλέχθηκε η χρήση του Bootstrap / AngularJS framework, με στόχο την δημιουργία μιας κοινής εικόνας της εφαρμογής (responsiveness), σε οποιαδήποτε πλατφόρμα επιλέγει ο χρήστης (pc/κινητά/tablet). Πέρα από τις πολλές διευκολύνσεις που παρουσιάζει το σύστημα αυτό (grid, icons, buttons, modals), πολύ σημαντικό ρόλο στην επιλογή της διαδραμάτισε και η πολύ καλή υποστήριξη και τεκμηρίωση από την ομάδα ανάπτυξης.






Το λειτουργικό κομμάτι υλοποιήθηκε βασισμένο σε ένα από τα πλέον διαδεδομένα frameworks το “Spring Framework” το οποίο είναι ένα λογισμικό ανοικτού κώδικα (ελεύθερο λογισμικό) που σκοπό έχει να διευκολύνει την ανάπτυξη J2EE λογισμικού σε μεγάλη έκταση.

2.11 SERVER STACK DIAGRAM




Εικόνα 6: Media Stack Architecture

2.12 MediaStack Servers Implementation





	<p>Το λειτουργικό σύστημα των application & database servers</p>
	<p>Ο Red Hat JBoss επιλέχθηκε ως application server</p>
	<p>Στα πλαίσια της επικοινωνίας των υποσυστημάτων επιλέχθηκαν API του Apache Foundation</p>
	<p>Database Server</p>
	<p>Ο streaming server που επιλέχθηκε για τη μεταπτυχιακή διατριβή</p>

2.13 MediaStack Software Implementation

	<p>Ως τεχνολογία ανάπτυξης επιλέχθηκε η J2EE</p>
	<p>Ο Asset Manager του Media Stack</p>
	<p>Ο Hazelcast παρέχει μια distributed second level cache για τα Hibernate entities, collections και queries.</p>
	<p>Αποτελεί ένα πλούσιο multi-protocol messaging broker που υποστηρίζει AMQP 0-9-1, STOMP 1.0 through 1.2, MQTT 3.1.1, AMQP 1.0</p>

	Το AngularJS επιλέχθηκε ως το Frontend Framework
---	--

2.14 Media Stack Development Tools

	Το eclipse επιλέχθηκε ως το βέλτιστο περιβάλλον ανάπτυξης
	Ο repository manager του project
	Το Media Stack έχει αναπτυχθεί με το Spring Framework 4
	FrontEnd Development Tools

2.15 ΕΝΣΩΜΑΤΩΣΗ ΔΟΚΙΜΑΣΤΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΕΛΕΓΧΟΣ ΛΕΙΤΟΥΡΓΙΑΣ

Για τον έλεγχο του συστήματος καταχωρήθηκαν δοκιμαστικά δεδομένα προκειμένου να μπορεί να γίνει ο έλεγχος λειτουργίας των εκάστοτε υποσυστημάτων. Βάση των αρχικών ρυθμίσεων είχε επιλεγεί, κατά τον χρόνο υλοποίησης, η βάση να διαγράφεται και να αναδομείται οδηγούμενη από μια διαδικασία αρχικοποίησης των δεδομένων που απαιτούνταν κατά την επανεκκίνηση του Spring Container.

Αυτό ήταν χρήσιμο ώστε να υπάρχουν πάντα τα δεδομένα που απαιτούνται με ένα σταθερό τρόπο χωρίς να πρέπει σε κάθε αλλαγή και επανεκκίνηση του εξυπηρετητή να εισάγονται εκ' νέου, αλλά ούτε και να χρειάζεται να τα διαγράφουμε. Το σύστημα μέσω κατάλληλης αλλαγής σε συγκεκριμένο αρχείο ρυθμίσεων δύναται να αλλάξει αυτή του την συμπεριφορά και να διατηρεί την υπάρχουσα βάση χωρίς να την καταστρέφει και να την αναδομεί εκ' νέου, όπως και θα ήταν το επιθυμητό σε συστήματα παραγωγής.

Όπως ήδη αναφερθήκαμε, η επιθυμία για την μελλοντική δυνατότητα χρήσης διαφορετικών τεχνολογιών δομών δεδομένων (βάσεων δεδομένων) μας οδήγησε στην επιλογή της χρήσης ORM μέσω JPA annotation και hibernate implementation, δίνοντας έτσι ένα επίπεδο αφαιρετικότητας, το οποίο είναι επιθυμητό για την σωστή συντήρηση και την επέκταση της εφαρμογής σε βάθος χρόνου. Ότι αφορά στις υποδομές για την βάση δεδομένων

χρησιμοποιήθηκε η εγγενής δυνατότητα επικοινωνίας με PostgreSQL βάση δεδομένων, ως η πιο ευέλικτες και χωρίς μεγάλες ανάγκες δέσμευσης πόρων του συστήματος.

Η PostgreSQL επιλέχθηκε μεταξύ άλλων λόγω της έλλειψης περιορισμών που υπάρχουν σε άλλα RDBMS όπως η MySQL, ενώ αποτελεί έργο ανοιχτού κώδικα και δεν συνοδεύεται από εμπορική άδεια χρήσης.

Τέλος σε ότι αφορά στις διάφορες κλήσεις που απαιτήθηκαν μεταξύ client και server κυρίως κατά την εμφάνιση στα ημερολόγια του διαχειριστικού που λειτουργούν με ασύγχρονο τρόπο (ajax), έγινε χρήση τεχνολογιών REST με JSON, ενώ για την μετατροπή από αντικείμενα σε Json format και το ανάποδο έγινε χρήση της βιβλιοθήκης Spring Services.

3 Τεχνολογίες που χρησιμοποιήθηκαν

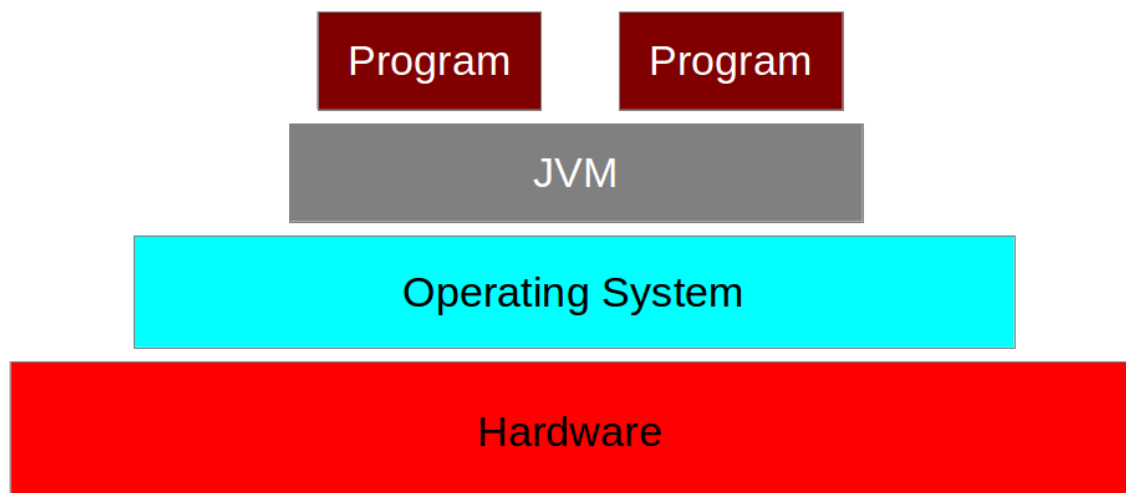
3.1 Java

Η Java είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems. Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας.

Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (σύντομα θα τρέχουν και σε Playstation καθώς και σε άλλες κονσόλες παιχνιδιών) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα.

Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (assembly) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM).

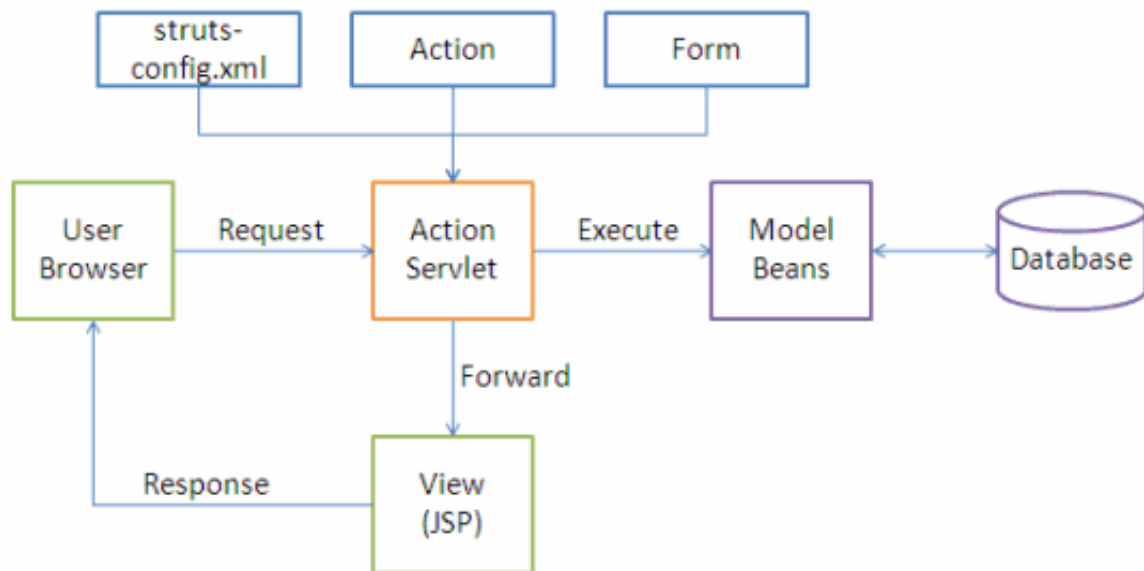
Για να γράψει κάποιος κώδικα Java δε χρειάζεται τίποτα άλλο παρά έναν επεξεργαστή κειμένου, όπως το Σημειωματάριο (Notepad) των Windows ή ο vi (γνωστός στο χώρο του Unix). Παρ'όλ'αυτά, ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) βοηθάει πολύ, ιδιαίτερα στον εντοπισμό σφαλμάτων (debugging). Υπάρχουν αρκετά διαθέσιμα, ενώ πολλά από αυτά έρχονται δωρεάν.



Εικόνα 7: Σχηματική παράσταση του Java Virtual Machine και του τρόπου εκτέλεσης των J2EE εφαρμογών

3.2 Spring MVC

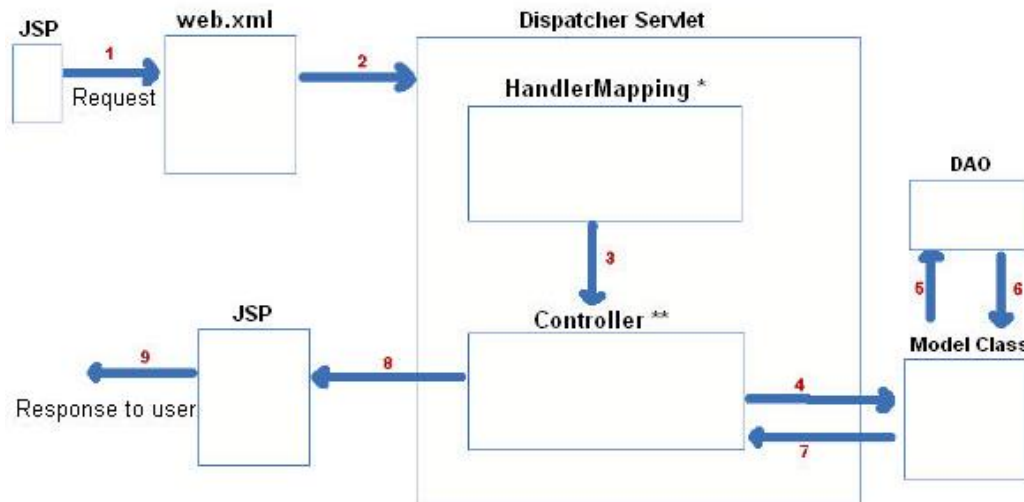
Το Spring MVC framework παρέχει την αρχιτεκτονική Model-View-Controller καθώς και κάποια έτοιμα συστατικά τα οποία μπορούν να χρησιμοποιηθούν για την ανάπτυξη ευέλικτων και χαλαρά συνδεδεμένων web εφαρμογών.



Εικόνα 8: Αρχιτεκτονική MVC

Το πρότυπο MVC οδηγεί σε διαχωρισμό των διαφόρων πτυχών της εφαρμογής (λογική είσοδο, επιχειρηματική λογική, και λογική UI), παρέχοντας παράλληλα μια χαλαρή σύνδεση μεταξύ αυτών των στοιχείων.

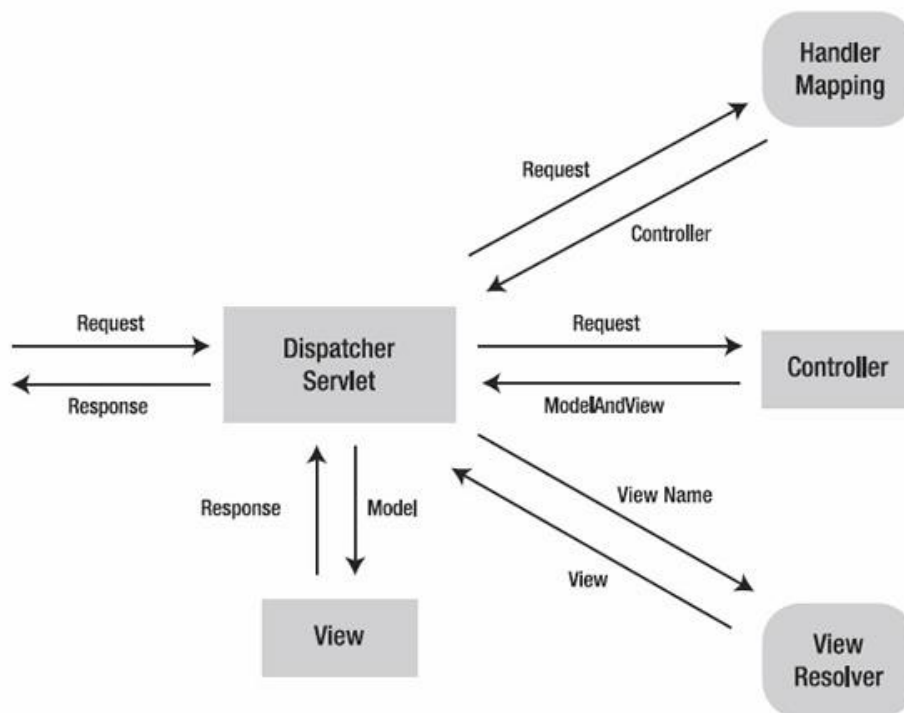
- Το Model ενσωματώνει τα στοιχεία της αίτησης και σε γενικές γραμμές αποτελείται από POJO (Plain Old Java Objects).
- Το View είναι υπεύθυνο για την παροχή των δεδομένων του μοντέλου και γενικότερα δημιουργεί εξόδους HTML που ο browser του πελάτη μπορεί να ερμηνεύσει.
- Ο Controller είναι υπεύθυνος για την επεξεργασία των αιτήσεων των χρηστών και την οικοδόμηση του κατάλληλου μοντέλου.



Εικόνα 9: Spring MVC Application Flow

3.3 DispatcherServlet

Το Spring Web model-view-controller (MVC) framework έχει σχεδιαστεί γύρω από ένα DispatcherServlet το οποίο χειρίζεται όλα τα HTTP requests και responses. Η ροή επεξεργασίας του Spring Web MVC DispatcherServlet απεικονίζεται στο επόμενο διάγραμμα.



Εικόνα 10: Spring Web Mvc Dispatcherservlet

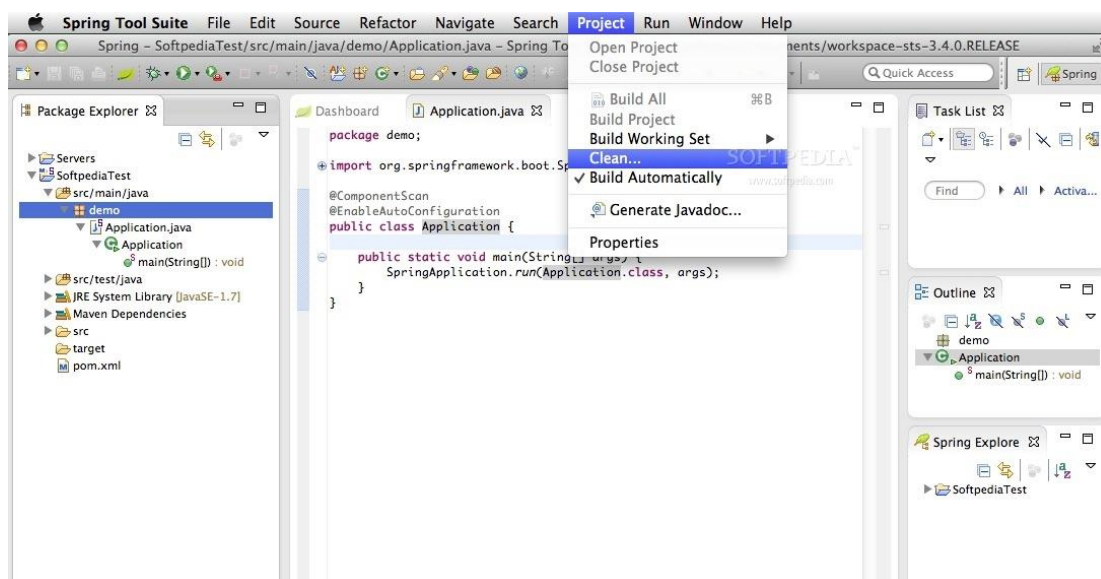
Στην συνέχεια παρουσιάζεται η σειρά των γεγονότων που αντιστοιχούν σε μια εισερχόμενη αίτηση HTTP (HTTP request) στον DispatcherServlet:

1. Αφού λάβει ένα αίτημα HTTP, το DispatcherServlet θα συμβουλευθεί το HandlerMapping, ώστε να καλέσει τον κατάλληλο Controller.
2. Ο Controller θα παραλάβει το αίτημα, και θα καλέσει την κατάλληλη μέθοδο, ανάλογα με τη μέθοδο HTTP βάσει της οποίας εστάλη το αίτημα (GET, POST κτλ). Η κληθείσα μέθοδος θα καθορίσει τα δεδομένα μοντέλου, βάσει της λογικής της εφαρμογής, και θα επιστρέψει το όνομα της εμφάνισης στο DispatcherServlet.
3. Το DispatcherServlet θα συμβουλευθεί τον ViewResolver, ώστε να επιλέξει την κατάλληλη εμφάνιση για το αίτημα.
4. Όταν η εμφάνιση οριστικοποιηθεί, το DispatcherServlet περνάει τα δεδομένα μοντέλου στην εμφάνιση, η οποία, εν τέλει, θα εμφανιστεί στον περιηγητή

3.4 Spring Tool Suite

Το Spring Tool Suite είναι ένα Eclipse-based περιβάλλον ανάπτυξης που είναι προσαρμοσμένο για την ανάπτυξη Spring εφαρμογών. Παρέχει ένα έτοιμο προς χρήση περιβάλλον για την εφαρμογή, την διόρθωση, το τρέξιμο, και την ανάπτυξη εφαρμογών, συμπεριλαμβανομένων των ενσωματώσεων για Pivotal Tc Server, Pivotal Cloud Foundry, Git, Maven, AspectJ, και βρίσκεται στην κορυφή των τελευταίων κυκλοφοριών του Eclipse.

Στο Spring Tool Suite περιλαμβάνεται και η έκδοση προγραμματιστών του Pivotal tc Server, τον αντικατάσταση του Apache Tomcat που είναι βελτιστοποιημένος για το Spring. Με την Spring Insight κονσόλα, η έκδοση tc Developer Server παρέχει μια γραφική προβολή σε πραγματικό χρόνο των μετρήσεων της απόδοσης της εφαρμογής που επιτρέπει στους προγραμματιστές να εντοπίζουν και να μπορούν να διαγνώσουν τα προβλήματα από την οθόνη του υπολογιστή τους. Το Spring Tool Suite υποστηρίζει την στόχευση της εφαρμογής σε τοπικούς, εικονικούς και cloud-based servers. Είναι ελεύθερα διαθέσιμο για την ανάπτυξη και χρήση εσωτερικών επιχειρηματικών δραστηριοτήτων χωρίς χρονικό περιορισμό, πλήρως open-source και σύμφωνα με τους όρους του Eclipse Public License.



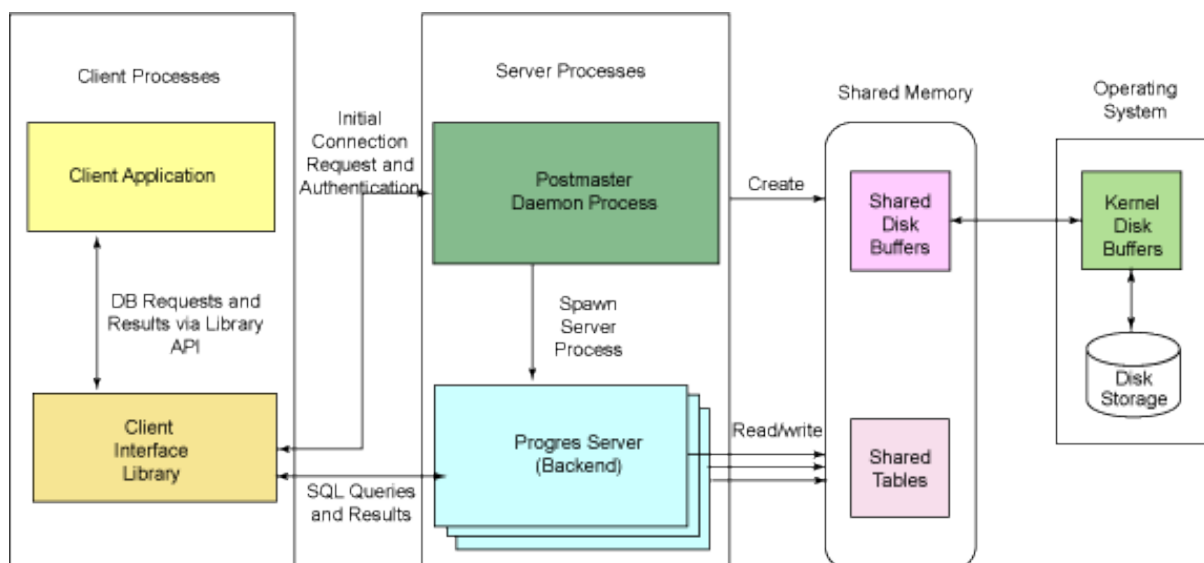
Εικόνα 11: Pivotal STS

3.5 PostgreSQL

Η PostgreSQL είναι μια σχεσιακή βάση δεδομένων ανοικτού κώδικα με πολλές δυνατότητες. Η ανάπτυξη της διαρκεί ήδη πάνω από δύο δεκαετίες και βασίζεται σε μια αποδεδειγμένα καλή αρχιτεκτονική η οποία έχει δημιουργήσει μια ισχυρή αντίληψη των χρηστών της γύρω από την αξιοπιστία, την ακεραιότητα δεδομένων και την ορθή λειτουργία.

Η PostgreSQL τρέχει σε όλα τα βασικά λειτουργικά συστήματα, στα οποία περιλαμβάνονται το Linux, το UNIX (AIX, BSD, HP-UX, SGI, IRIX, MAC OS X, Solaris, Tru64) και τα Windows. Είναι συμβατή με ACID, και συμπεριλαμβάνει τους περισσότερους SQL92 και SQL99 τύπους δεδομένων συμπεριλαμβανομένων INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL και TIMESTAMP. Επίσης υποστηρίζει αποθήκευση μεγάλων δυαδικών αντικειμένων (binary), όπως εικόνες, ήχοι ή βίντεο.

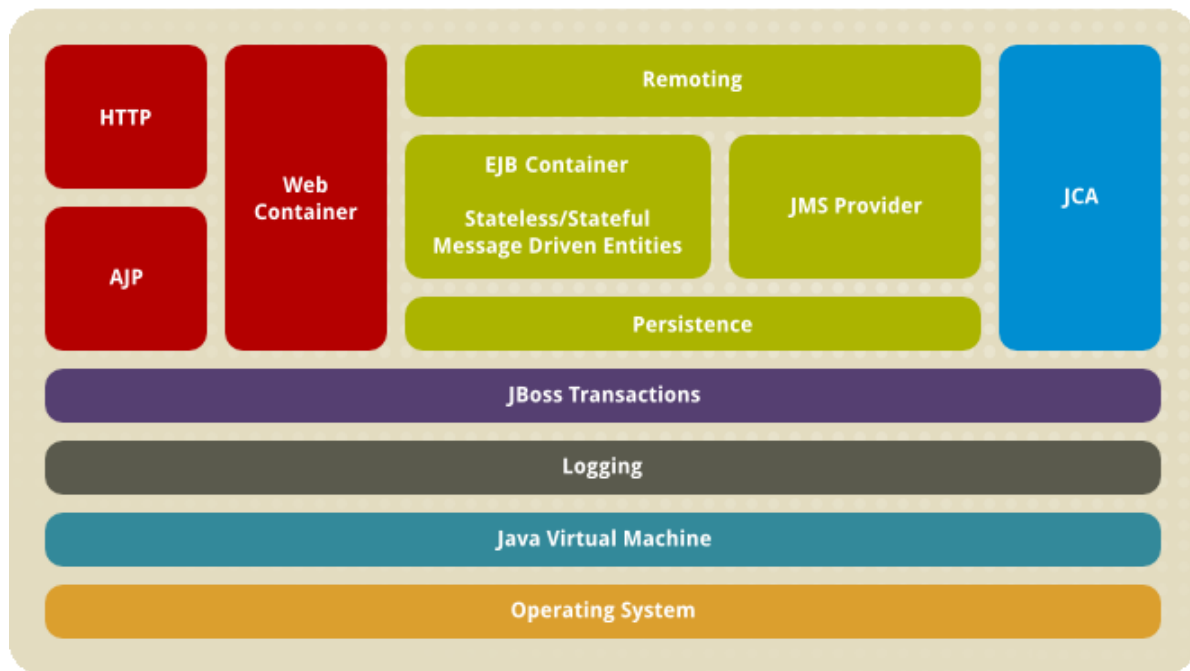
Διαθέτει επίσης περιβάλλοντα προγραμματισμού για τις γλώσσες προγραμματισμού C, C++, Java, Perl, Python, Ruby, Tcl, και υποστήριξη για την πλατφόρμα .NET και το πρότυπο ODBC, ενώ περιλαμβάνει και εξαιρετικό εγχειρίδιο χρήσης.



Εικόνα 12: PostgreSQL Diagram

3.6 JBOSS

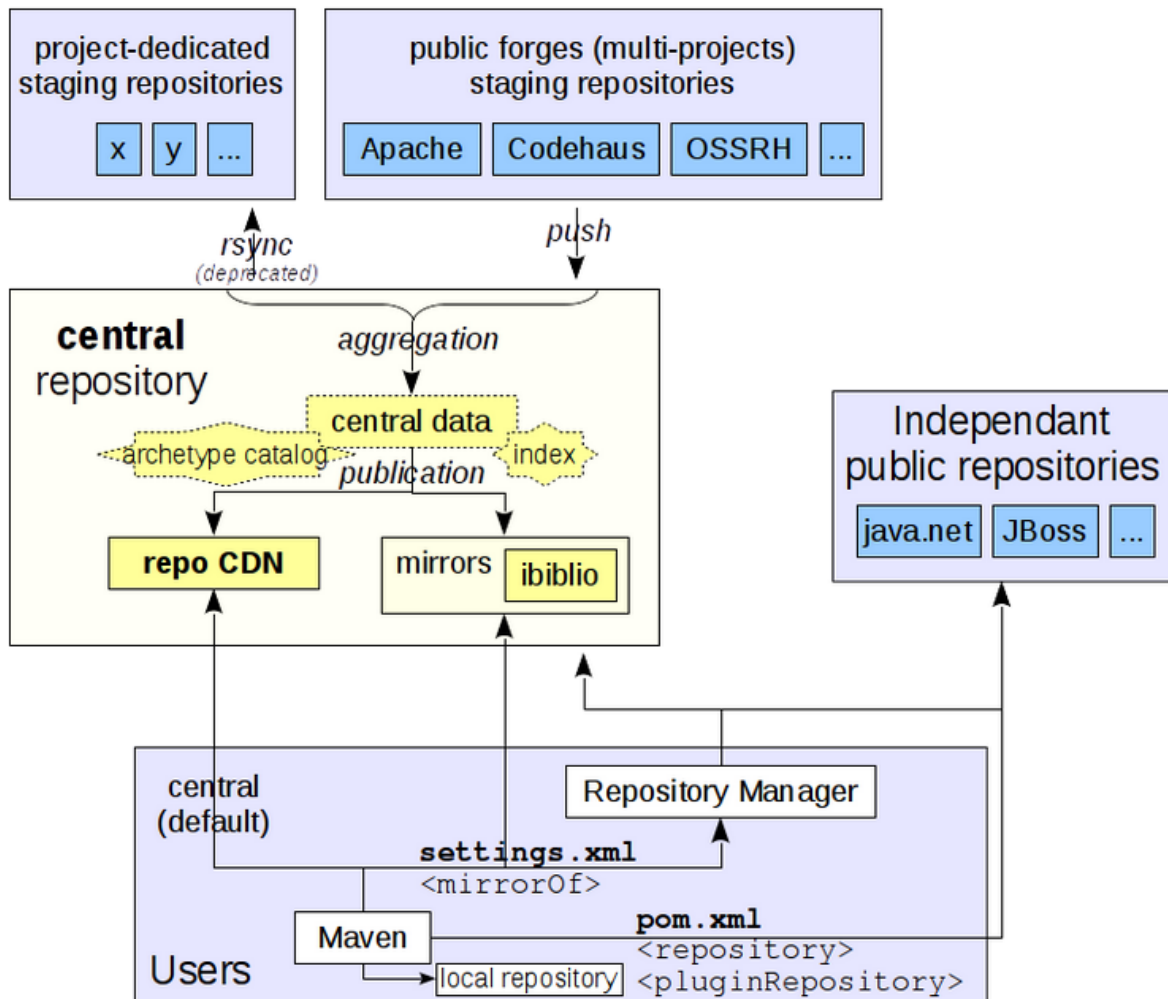
Ο JBoss Application Server είναι μια εφαρμογή λογισμικού που έχει αναπτυχθεί από την εταιρία Red Hat και υλοποιεί τις προδιαγραφές για Java Servlet, JavaServer Pages, Java Expression Language και Java WebSocket τεχνολογιών. Οι προδιαγραφές για Java Servlet, JavaServlet Pages, Java Expression Language και Java WebSocket αναπτύσσονται στο πλαίσιο της Java Community Process.



Εικόνα 13: Διαγραμματική απεικόνιση αρχιτεκτονικής JBoss Application Server

3.7 Maven

Το maven είναι ένα εργαλείο διαχείρισης έργων λογισμικού. Βασισμένο στην έννοια του μοντέλου αντικειμένου του έργου (POM), το Maven μπορεί να διαχειριστεί την κατασκευή (build) ενός έργου, την έκθεση και την τεκμηρίωση απο ένα κεντρικό κομμάτι πληροφοριών.



Εικόνα 14: Διαγραμματική απεικόνιση αξιοποίησης Maven repository

3.8 Javascript

Η JavaScript (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme.

Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side). Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript.

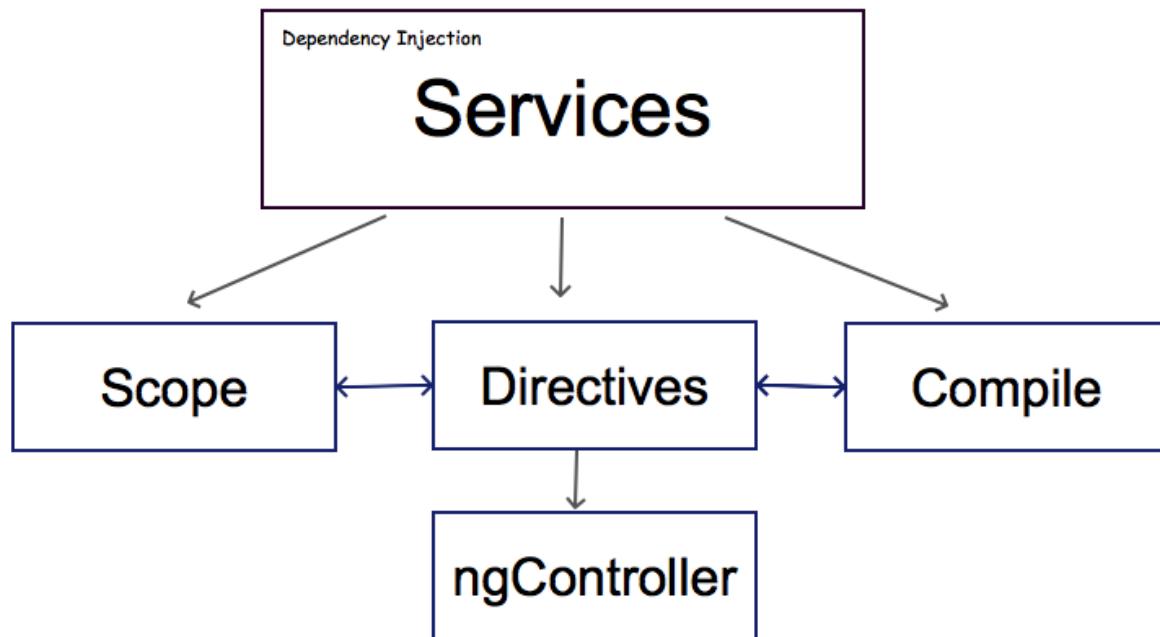
3.9 AngularJS

Το Angular JS είναι ένα open source web application framework που συντηρείται από την Google και από μια κοινότητα προγραμματιστών και επιχειρήσεων για την αντιμετώπιση των προκλήσεων που ανέκυψαν κατά την ανάπτυξη single-page εφαρμογών.

Στόχος είναι να απλοποιήσει την ανάπτυξη και την δοκιμή αυτών των εφαρμογών, παρέχοντας ένα framework για client-side model-view-controller(MVC) και model-view-view - model(MVVM) αρχιτεκτονικών, μαζί με τα συστατικά που χρησιμοποιούνται συνήθως σε πλούσιες εφαρμογές Διαδικτύου.

Η βιβλιοθήκη Angular JS λειτουργεί διαβάζοντας πρώτα την HTML σελίδα, στην οποία έχουν ενσωματωθεί επιπλέον προσαρμοσμένες ετικέτες (tag attributes). Το Angular ερμηνεύει αυτά τα χαρακτηριστικά ως οδηγίες για να δεσμεύσει τμήματα εισόδου ή εξόδου της σελίδας σε ένα μοντέλο που αντιπροσωπεύεται από πρότυπες μεταβλητές JavaScript. Οι τιμές αυτών των μεταβλητών JavaScript μπορεί είτε να ρυθμιστούν με το χέρι εντός του κώδικα, είτε να προέλθουν από στατικούς ή δυναμικούς πόρους JSON.

Σύμφωνα με υπηρεσία analytics JavaScript Libscore, το AngularJS χρησιμοποιείται στις ιστοσελίδες του NBC, Walgreens, Intel, Sprint, ABC News, και περίπου 8.400 άλλους δικτυακούς τόπους τον Ιούλιο του 2015.



Εικόνα 15: AngularJS Framework Diagram

3.10 HTML

Η HTML (ακρωνύμιο του αγγλικού HyperText Markup Language) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας.

Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

3.11 Cascading Style Sheets

Η CSS (Cascading Style Sheets) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης ενός ιστοτόπου.

Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html.

Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

3.12 Bootstrap

Για να δημιουργήσουμε την εφαρμογή μας έτσι ώστε να εμφανίζεται κατάλληλα σε όλα τα μεγέθη και αναλύσεις οθονών, χρησιμοποιήσαμε το Bootstrap. Το Bootstrap είναι το πιο δημοφιλές HTML, CSS και JS Framework για την ανάπτυξη responsive, mobile first εφαρμογών στο διαδίκτυο.

Το Bootstrap, που αρχικά ονομαζόταν Twitter Blueprint, αναπτύχθηκε από τους Mark Otto και ο Jacob Thornton στο Twitter στα μέσα του 2010, ως ένα framework για την ενθάρρυνση της συνέπειας μεταξύ των εσωτερικών εργαλείων της εταιρείας για πάνω από ένα χρόνο πριν από την δημόσια κυκλοφορία του. Πριν από το Bootstrap, διάφορες βιβλιοθήκες χρησιμοποιούνταν για την ανάπτυξη διεπαφής, η οποία οδήγησε σε ασυνέπειες και σε υψηλή επιβάρυνση συντήρησης.

Το Bootstrap είναι συμβατό με τις τελευταίες εκδόσεις του Google Chrome, Mozilla Firefox, Internet Explorer, Opera και Safari. Από την έκδοση 2.0 υποστηρίζει το responsive web design. Αυτό σημαίνει ότι η διάταξη των ιστοσελίδων προσαρμόζεται δυναμικά, λαμβάνοντας υπόψη τα χαρακτηριστικά της συσκευής που χρησιμοποιείται (υπολογιστής, tablet, κινητό τηλέφωνο). Με την έκδοση 3.0 το Bootstrap υιοθέτησε μια mobile first φιλοσοφία σχεδίασης με έμφαση στο responsive design ως προεπιλογή.

4 ΤΕΧΝΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ

4.1 RESPONSIVE DESIGN

Κατά τα τελευταία χρόνια, υπάρχει μια αύξηση πάνω από 45% από τη χρήση των κινητών τηλεφώνων για πρόσβαση στο διαδίκτυο. Σύμφωνα με την πρόβλεψη της εταιρείας Ericsson, (Vanguardngr, 2014), ο αριθμός των χρηστών κινητής τηλεφωνίας θα φθάσει τα 9 δισεκατομμύρια το 2018.

Λόγω της εξέλιξης των τεχνικών και των συσκευών επικοινωνίας, ο καθένας μπορεί να περιηγηθεί στο Internet χρησιμοποιώντας έναν υπολογιστή, ένα κινητό τηλέφωνο, ένα tablet, μια τηλεόραση, μια κονσόλα παιχνιδιού κ.α. Όλες αυτές οι συσκευές έχουν την δυνατότητα της σύνδεσης στο διαδίκτυο, έχουν τις δικές τους διαστάσεις και χρησιμοποιούν διαφορετικές αναλύσεις οθόνης. Έτσι οι σχεδιαστές ιστοσελίδων πρέπει να λάβουν υπόψη τους ότι το περιεχόμενο της ιστοσελίδας τους είναι αναγνώσιμο και λειτουργικό σε όλες αυτές τις αναλύσεις. Αυτό ήταν το κίνητρο για την ανάπτυξη του Responsive Web Design.

Η κύρια ιδέα για τις βασικές αρχές του Responsive Web Design είναι “Web for all” και “Web on Everything”. Η ουσία αυτής της ιδέας είναι να επιτρέπει την πρόσβαση στο περιεχόμενο των ιστοσελίδων σε όλα τα υπάρχοντα μέσα.

Ουσιαστικά το Responsive Web Design είναι μια προσέγγιση κατά την οποία ο ιστότοπος σχεδιάζεται έτσι ώστε να παρέχει την βέλτιστη εμπειρία απεικόνισης σε ένα ευρύ φάσμα συσκευών από τους σταθερούς υπολογιστές μέχρι τα κινητά τηλέφωνα. Η βέλτιστη εμπειρία απεικόνισης περιλαμβάνει την εύκολη ανάγνωση και πλοήγηση με την ελάχιστη αλλαγή μεγέθους, μετακίνησης ή κύλισης. Με άλλα λόγια, είναι μια προσέγγιση που επιτρέπει την ανάπτυξη και τον σχεδιασμό της ιστοσελίδας έτσι ώστε να ανταποκριθεί στη συμπεριφορά του χρήστη και το περιβάλλον, με βάση το μέγεθος της οθόνης, την πλατφόρμα και τον προσανατολισμό.

Η παρακάτω εικόνα εμφανίζει πως το ίδιο το site εμφανίζεται σε διαφορετικές συσκευές και αναλύσεις οθόνης.



Εικόνα 16: Responsive Design

4.2 Model – View - Controller

Σύμφωνα με την αρχιτεκτονική MVC (Model, View, Controller), στο επίπεδο Model τοποθετούνται οι λειτουργίες που σχετίζονται με την πρόσβαση στην βάση δεδομένων δηλαδή είναι κάποιες συναρτήσεις με τις οποίες εκτελούμε διάφορες λειτουργίες διαχείρισης των δεδομένων που λαμβάνουμε από την βάση. Στο επίπεδο View υπάρχει η HTML της σελίδας της εφαρμογής μας, είναι αυτό που βλέπουμε και τέλος το επίπεδο Controller είναι ο μεσίτης μεταξύ Model και View. Ελέγχει το πώς “τρέχει” η εφαρμογή, μιλάει με το Model, παίρνει τα δεδομένα που ζητά και αφού τα επεξεργαστεί τα στέλνει πίσω στο View επίπεδο για απεικόνιση. Η διαμοιρασμένη αυτή αρχιτεκτονική επιτρέπει στους προγραμματιστές να πραγματοποιούν αλλαγές σε ένα μέρος του κώδικα χωρίς να επηρεάζουν άλλα.

4.3 Spring MVC + Angular

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε η Spring MVC αρχιτεκτονική σε συνδυασμό με την Angular. Η κύρια ιδέα σε σύγκριση με πιο server-side αρχιτεκτονικές είναι να χτίσουμε το server ως ένα σύνολο επαναχρησιμοποιημένων REST υπηρεσιών, και από την MVC προοπτική να μεταφέρουμε τον controller από το backend και να το μετακινήσουμε στο πρόγραμμα περιήγησης.

Όσον αφορά το frontend, έχει οικοδομηθεί γύρω από ένα μοντέλο βασισμένο στην εμφάνιση και χειρίζεται μόνο τη λογική παρουσίαση, και καμία επιχειρηματική λογική. Αυτά είναι τα τρία στρώματα του frontend:

4.4 View Layer

Το View Layer αποτελείται από τα πρότυπα Html, τα CSS και κάθε άλλο Angular directive που εκπροσωπούν τα διάφορα στοιχεία του UI. Παράδειγμα μιας απλής εμφάνισης για μια απεικόνιση φόρμας είναι η αποστολή μηνύματος στον διαχειριστή (contact.jsp).

4.5 Controller Layer

Ο controller layer αποτελείται από Angular controllers οι οποίοι ενώνουν τα δεδομένα που ανακτώνται από το backend και το view. Ο controller προετοιμάζει το view μοντέλο και ορίζει πως πρέπει να αντιδράσει το view για να μοντελοποιήσει τις αλλαγές και το αντίστροφο.

Μια από τις κύριες αρμοδιότητες του controller είναι να εκτελέσει και επικυρώσεις του frontend. Όλες οι επικυρώσεις που γίνονται στο frontend είναι μόνο για την διευκόλυνση του χρήστη – για παράδειγμα είναι χρήσιμο να ενημερώνεται ο χρήστης για τα πεδία απαιτούνται κατά την συμπλήρωση/αποστολή μιας φόρμας.

4.6 Frontend Services Layer

Εδώ βρίσκεται το σύνολο των services της Angular που αλληλεπιδρούν με το backend.

4.7 Backend

Το backend κομμάτι της εφαρμογής έχει αναπτυχθεί χρησιμοποιώντας τα συνήθη επίπεδα του backend (Router Layer, Service Layer, Persistence Layer).

4.8 Router Layer

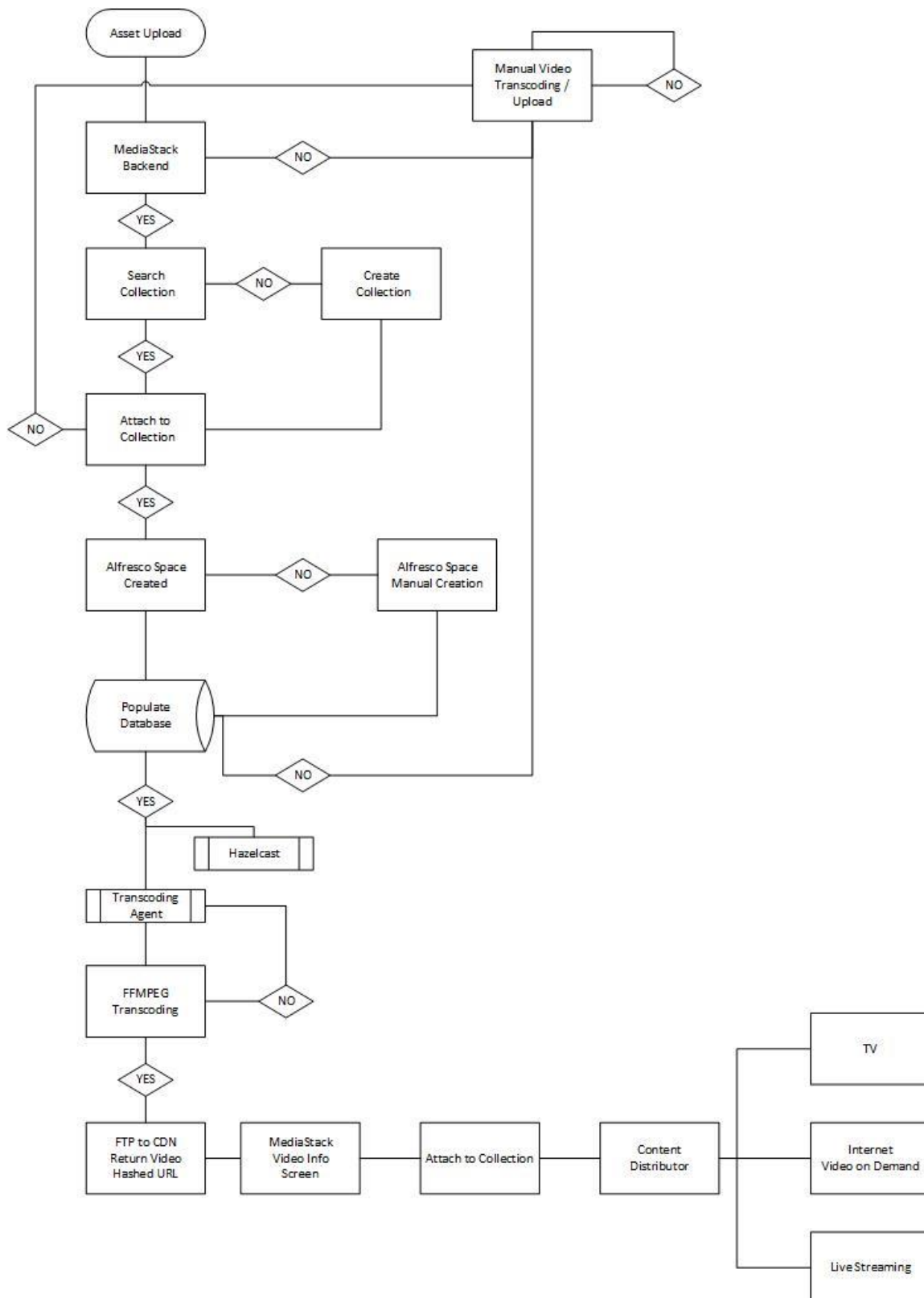
Ορίζει ποια υπηρεσία εισόδου ανταποκρίνεται σε μια δεδομένη HTTP url, και πώς οι παράμετροι πρέπει να διαβάζονται από ένα HTTP request.

Service Layer: περιέχει την επιχειρηματική λογική, όπως επικυρώσεις και καθορίζει το πεδίο των επιχειρηματικών συναλλαγών.

Persistence Layer: χαρτογραφεί τη βάση δεδομένων προς / από τη μνήμη σε αντικείμενα (objects).

5 ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΟΣ

5.1 ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΑΣ – ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ



Εικόνα 17: Διάγραμμα Ροής MediaStack



Εικόνα 18: Σχηματική Απεικόνιση MediaStack

5.2 ΒΑΣΙΚΕΣ ΟΝΤΟΤΗΤΕΣ

5.2.1 Στόχος του συστήματος

Το σύστημα MediaStack έχει ως στόχο τη συνολική υποστήριξη των διαδικασιών αρχειοθέτησης, επεξεργασίας και διάθεσης του περιεχομένου εικόνας, ήχου και video που διαχειρίζεται ένας σταθμός.

5.2.2 Domain Model

Η οντότητα που αποτελεί το βασικό αντικείμενο διαχείρισης για όλες σχεδόν τις διαδικασίες που υποστηρίζει το σύστημα είναι το <video-content>. Κάθε <video-content> αντιστοιχεί σε διαφορετικό περιεχόμενο, ανεξάρτητα από παραμέτρους υλοποίησης (π.χ. διαφορετικές αναλύσεις, κλπ). Ένα <video-content> μπορεί να υπάρχει χωρίς να διαθέτουμε αναγκαστικά κάποια υλοποίησή του (δηλ αρχείο ή ακόμη και σύνδεσμο).

Ένα <video-content> μπορεί να ανήκει σε μία από τις παρακάτω κατηγορίες:

Εκπομπή	Περιεχόμενο που αντιστοιχεί σε επεισόδιο σειράς, απόσπασμα ή υποστηρικτικό video
---------	--

Σειρά	Περιεχόμενο που αντιστοιχεί σε εκπομπή, απόσπασμα ή υποστηρικτικό για την εκπομπή video (π.χ. αμοντάριστο ρεπορτάζ)
Δελτίο Ειδήσεων	Δελτίο ειδήσεων ή απόσπασμα, Αρχείο ειδήσεων
Πρωθητικό	Trailers, previews, κλπ
Διαφημιστικό	Διαφημιστικό Spot
Αρχειακό	Αρχείο ειδήσεων και εκπομπών
Εξωτερική πηγή	Περιεχόμενο από εξωτερική πηγή
Live Streaming	Αναμετάδοση άλλων σταθμών

Πίνακας 1: κατηγορίες <video-content>

Τα <video-content>s χαρακτηρίζονται μέσω metadata και συσχετίζονται μεταξύ τους αλλά και με άλλους τύπους περιεχομένου σε ένα ενιαίο repository.

5.2.3 Data Model

Το κεντρικό αντικείμενο του μοντέλου δεδομένων για το σύστημα αντιστοιχεί στη βασική οντότητα <video-content>.

5.2.4 Video content

Τα χαρακτηριστικά αυτής τη οντότητας παρατίθενται στον παρακάτω πίνακα:

Χαρακτηριστικό	Εξήγηση
Τίτλος	Τίτλος που χαρακτηρίζει το περιεχόμενο του video – να μπορεί να χρησιμοποιηθεί για υποστήριξη της ταυτοποίησής του. (π.χ. όνομα σειράς και αριθμός ή τίτλος επεισοδίου)
Μικρή περιγραφή - λεζάντα	Σύντομη περιγραφή του περιεχομένου, κατάλληλη για χρήση σε λίστες – καταλόγους (web video-galleries)
Είναι απόσπασμα; αναφορά στο πλήρες video	Χαρακτηρισμός του video ως αποσπάσματος ή πλήρους video. Σε περίπτωση που είναι απόσπασμα, ο χαρακτηρισμός θα συνοδεύεται από αναφορά στο πλήρες video (εάν είναι καταχωρημένο στο σύστημα)
Διάρκεια	Η διάρκεια του περιεχομένου. Θεωρητικά δε πρέπει να υπάρχει απόκλιση από αυτή την τιμή στα αρχεία

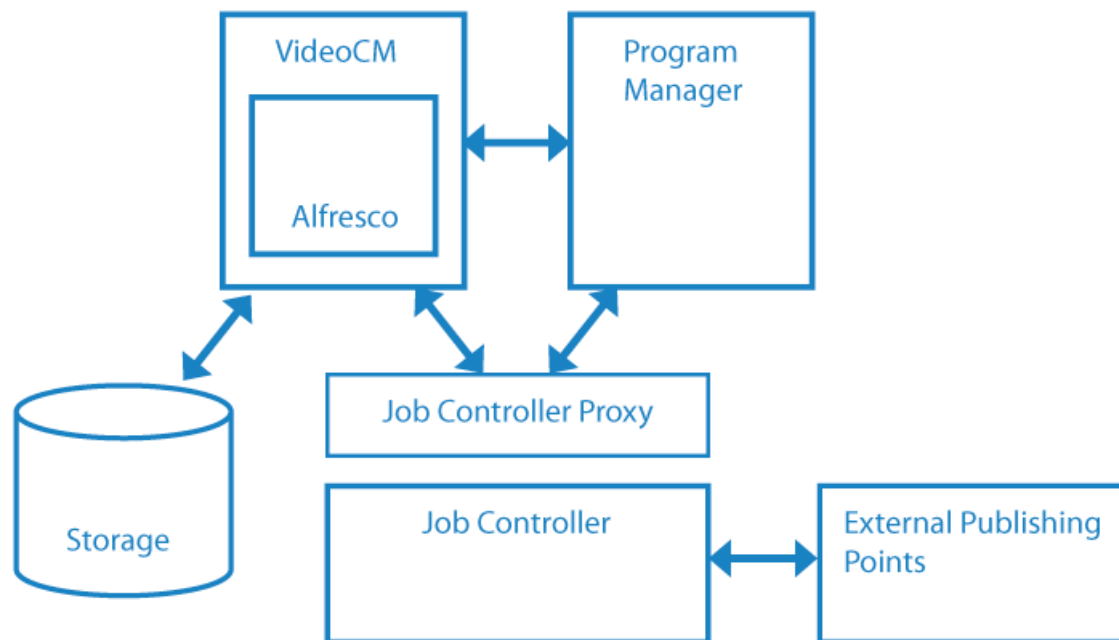
	που αντιστοιχούν στην «υλοποίηση» του video, αλλά μικρές αποκλίσεις μπορούν να εμφανιστούν από διαδικασίες αυτόματης εξαγωγής μεταδεδομένων, κλπ.
Image	Αντιπροσωπευτική εικόνα για το video
Υπότιτλοι	Αρχείο υποτίτλων
Περίληψη	Σύνδεσμος σε περιεχόμενο περίληψης
Transcript	Σύνδεσμος με περιεχόμενο κειμένου που αντιστοιχεί στην «απομαγνητοφώνηση» και πλήρη περιγραφή της δράσης στο περιεχόμενο
Tags	Μετα-δεδομένα
Καταλληλότητα	Κωδικός καταλληλότητας
δικαιώματα	Τα δικαιώματα μεταβολής ή δημιουργίας ενός <video-content> καθορίζονται από την κατηγορία στην οποία ανήκει καθώς και την ομάδα στην οποία ανήκει ο χρήστης.
Κατάσταση έγκρισης	Κατάσταση που αναπαριστά το στάδιο στη βασική ροή εργασίας με βήματα draft / approved / cancelled.
Αρχεία	Τα αρχεία με τα οποία έχουμε συνδέσει το <video-content>. Κάθε αρχείο αποτελεί μια άλλη κωδικοποίηση, αποθήκευση, κλπ του ίδιου περιεχομένου

Πίνακας 2: κατηγορίες <video-content>

5.3 Αρχιτεκτονική

Η αρχιτεκτονική του συστήματος βασίζεται στην απομόνωση της λειτουργικότητας διαχείρισης περιεχομένου (video content) σε ένα υποσύστημα, και των λειτουργιών δημοσίευσης των video στα διαφορετικά εξωτερικά σημεία δημοσίευσης σε ένα άλλο διακριτό υποσύστημα (ακολουθεί τεκμηρίωση).

Στο επόμενο σχήμα παρουσιάζεται σε υψηλό επίπεδο η συνολική αρχιτεκτονική.



Εικόνα 19: Αρχιτεκτονική Media Stack

Το υποσύστημα VideoCM αφορά στη διαχείριση του περιεχομένου τύπου video και των σχετιζόμενων αρχείων (<video-content>s και related content) καθώς και υποστηρικτικών οντοτήτων και εργασιών όπως τα <collection>s και τα <workflow>s.

Το υποσύστημα Program Manager έχει ως βασικό στόχο την εφαρμογή των επιχειρησιακών κανόνων για τη δημοσίευση των video (<collection>s) στα σημεία δημοσίευσης και τον συντονισμό των σχετικών εργασιών.

Το υποσύστημα Job Controller παρέχει οριζόντια υπηρεσία εκτέλεσης εργασιών όπως η μεταφορά αρχείων, video trans-coding κλπ.

5.4 Διαδικασίες διαχείρισης του περιεχομένου Video

Η διαχείριση του περιεχομένου video είναι καταναμημένη στις επιχειρησιακές οντότητες του τηλεοπτικού σταθμού και για αυτό μπορεί να διαφέρουν από σταθμό σε σταθμό.

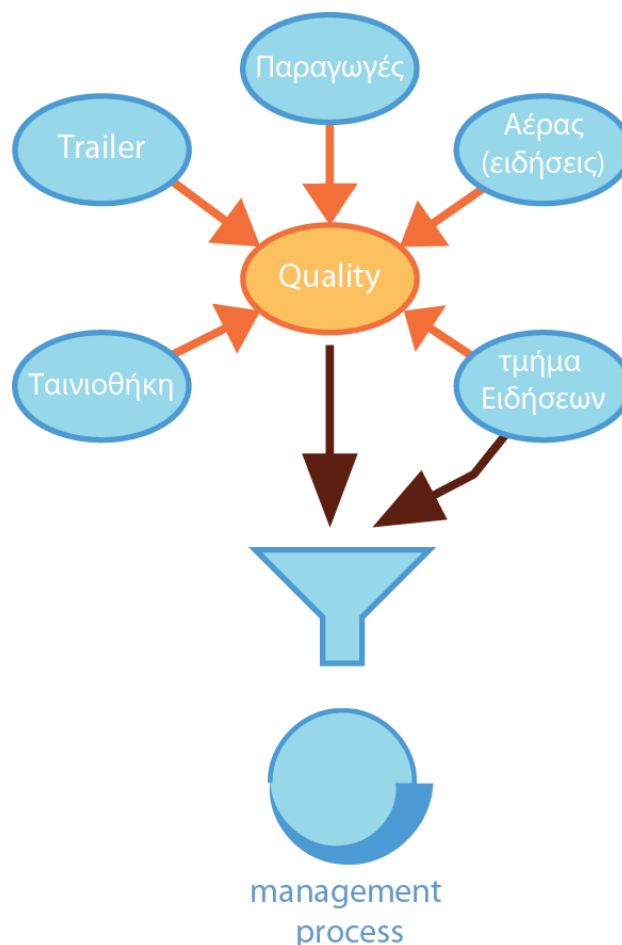
Αρχικά διακρίνεται μια «οριζόντια διαίρεση» σε δύο επίπεδα. Στο πρώτο επίπεδο πραγματοποιείται η παραγωγή των αρχείων video (πρωτογενή αρχεία για το VMEDIA2), και στο δεύτερο επίπεδο γίνεται η επεξεργασία των μεταδεδομένων και η διαχείριση των διαδικασιών που έχουν αντικείμενο ένα <video-content>.

5.5 Παραγωγή πρωτογενών αρχείων video

Το πρώτο επίπεδο είναι αρμοδιότητα, σχεδόν για όλες τις περιπτώσεις του τμήματος Quality. Στο συγκεκριμένο τμήμα γίνεται η παραγωγή των αρχείων όπου αυτά δρομολογούνται προς το σύστημα διαχείρισης για αυτόματη δημιουργία εγγραφών <video-content>.

Το ρόλο της παραγωγής αρχείων video έχει (για μικρό πλήθος αρχείων) και το τμήμα

Ειδήσεων, όπως όμως οι απαιτήσεις για την ενημέρωση του συστήματος είναι ίδιες.



Εικόνα 20: Αρχιτεκτονική Ροής Προγράμματος Media Stack

Για την υλοποίηση του πρώτου επιπέδου η τεχνική που θα εφαρμοστεί είναι ίδια με του υφιστάμενου συστήματος όπου συγκεκριμένα directories υποδέχονται πηγαία αρχεία με κωδικοποίηση του filename τέτοια ώστε να προκύπτουν από αυτή τιμές για metadata του νέου <video-content>.

Πέρα από την αυτόματη δημιουργία του <video-content> και τη συμπλήρωση των

metadata εκτελούνται και άλλες εργασίες όπως π.χ. η προσθήκη του <video-content> σε <collection>s.

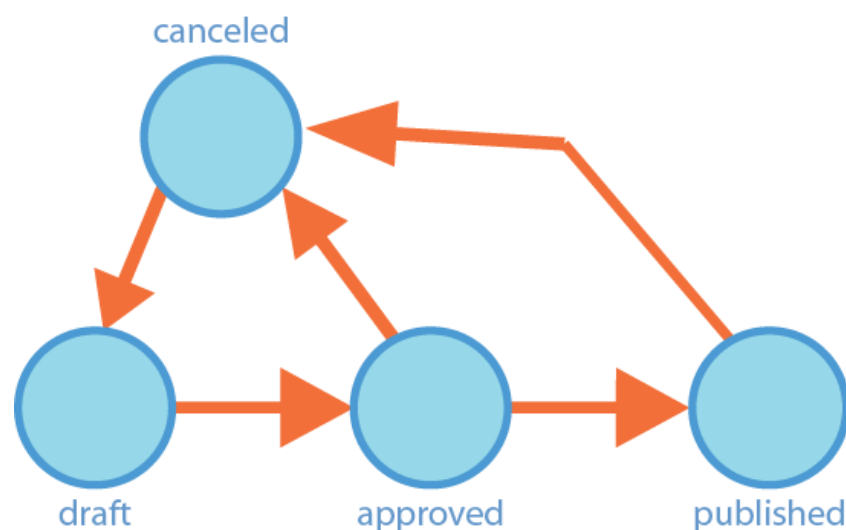
5.6 Διαχείριση μεταδεδομένων και ροών εργασίας

Στο επόμενο επίπεδο μπορεί να λειτουργήσει οποιαδήποτε επιχειρησιακή οντότητα του MEGA. Οι βασικές εργασίες που αφορούν αυτό το επίπεδο είναι η συμπλήρωση μεταδεδομένων, ο έλεγχος της συμμόρφωσης του περιεχομένου με συγκεκριμένους κανόνες⁷, και η διαχείριση της εκτέλεσης ροών εργασίας που αφορούν το κάθε <video-content>.

Η αυτόματη δημιουργία ενός <video-content> μετά από την παραγωγή του σχετικού αρχείου (στη γενική περίπτωση από το τμήμα Quality), στις περισσότερες περιπτώσεις ακολουθείται από την ενεργοποίηση μιας ροής εργασίας η οποία έχει ως αντικείμενο αυτό το <video-content>.

Πρώτο βήμα αυτή της διαδικασίας είναι ο χαρακτηρισμός, η συμπλήρωση των μεταδεδομένων του περιεχομένου και ο έλεγχος συμμόρφωσης με τους κανόνες.

Επόμενα βήματα μπορούν να περιλαμβάνουν την επιβεβαίωση και την δρομολόγηση προς δημοσίευση. Παράδειγμα μιας τέτοιας διαδικασίας περιγράφεται με το επόμενο state diagram.



Εικόνα 21: Διαχείριση μεταδεδομένων και ροών εργασίας

Ένα σημαντικό τμήμα της επεξεργασίας ενός <video-content> αποτελούν οι εργασίες που αφορούν τη δημοσίευσή του μετά την ολοκλήρωση της επεξεργασίας των μεταδεδομένων και την «έγκριση» για τη δημοσίευσή του.

Η διαχείριση των εργασιών αυτών αποτελεί αντικείμενο διακριτού υποσυστήματος που ολοκληρώνει το σύνολο της λειτουργικότητας που απαιτείται για την υλοποίηση της διαχείρισης «Προγράμματος Δημοσιεύσεων».

5.7 Διαχείριση προγράμματος δημοσιεύσεων - «Program Manager»

Η διαχείριση των δημοσιεύσεων των <video-content> αποτελεί το τελευταίο στάδιο της επεξεργασίας για μεγάλο ποσοστό των εγγράφων λόγω κριτηρίων όπως:

- υψηλή κρισιμότητα των διαδικασιών
- χρήση των <video-content> μετά την ολοκλήρωση της πρωτογενούς επεξεργασίας
- ισχυρή εξάρτηση από το εξωτερικό περιβάλλον (σημεία δημοσίευσης)
- μεταβλητότητα κανόνων ανά περίπτωση
- πληθώρα πλεονεκτημάτων λόγω απόζευξης από το σύστημα διαχείρισης εγγράφων

5.7.1 Διαχείριση προγράμματος δημοσιεύσεων / «Program Manager».

Οι βασικές οντότητες του υποσυστήματος είναι:

COLLECTION: το <collection> είναι οντότητα η οποία αντιστοιχεί σε διατεταγμένη λίστα από <video-content>. Το <collection> αποτελεί αντικείμενο του υποσυστήματος διαχείρισης εγγράφων ταυτόχρονα όμως βασικό στοιχείο επικοινωνίας με τον <program-manager>. Όσον αφορά στον <program-manager> τα <collection>s αποτελούν λίστες περιεχομένου προς δημοσίευση.

CHANNEL: <channel> είναι η έννοια που αντιπροσωπεύει το τελικό σημείο δημοσίευσης. Η ακριβής αντιστοίχιση του <channel> με web-site ή εσωτερική δομή κάποιου provider (π.χ. web site, playlist στο youtube) εξαρτάται από τη συγκεκριμένη περίπτωση.

PROGRAM: το <program> είναι η εφαρμογή κανόνων δημοσίευσης ενός <collection> σε ένα <channel>.

Οι κανόνες αυτοί έχουν δύο ειδών αποτελέσματα:

- τροποποίηση μεταδεδομένων που αφορούν στη δημοσίευση του <video-content> όπως π.χ. το χρονικό διάστημα κατά το οποίο θα παραμένει δημοσιευμένο ή ο τιμοκατάλογος.
- εκτέλεση εργασιών που σχετίζονται με την «προετοιμασία» του <video-content> και την αποτίμηση εκ νέου των κανόνων.

REGISTRATION: Το <registration> είναι βοηθητική οντότητα με στόχο να επιτρέπει την παρακολούθηση από το υποσύστημα διαχείρισης εγγράφων τη χρήση <video-content> σε ενεργές λίστες δημοσίευσης και να διευκολύνει έτσι την

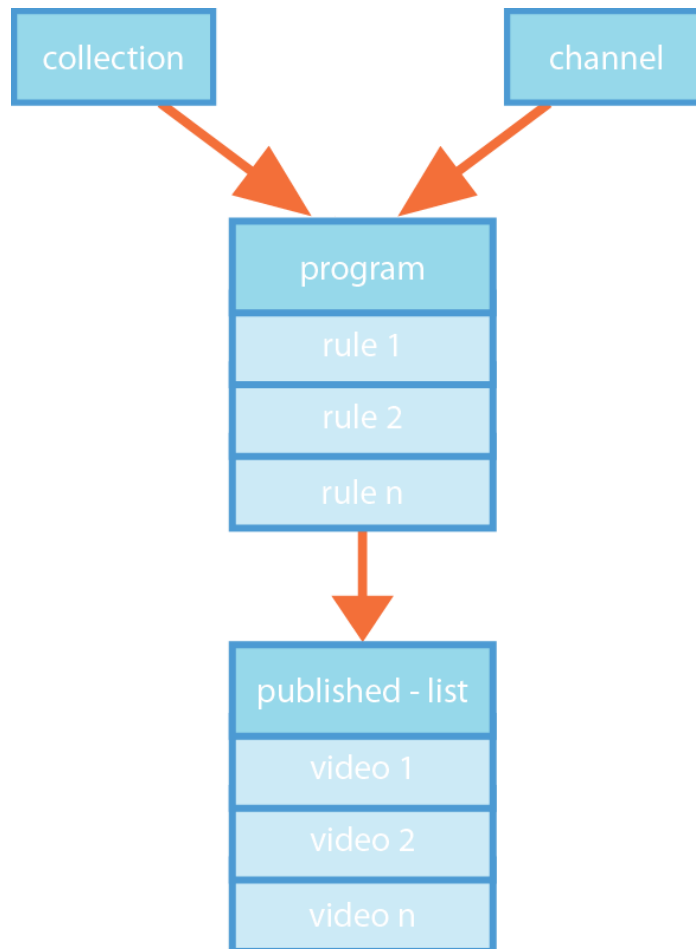
5.7.2 Επικοινωνία των δύο υποσυστημάτων στις περιπτώσεις μεταβολών.

PUBLISHED-LIST: Η εφαρμογή των κανόνων πάνω σε ένα <collection> παράγει ένα <published-list>. Είναι μια δυναμική δομή η οποία περιλαμβάνει για κάθε <video-content> του <collection> ένα σύνολο (αναφορών) φυσικών αρχείων (στη γενική περίπτωση είναι ένα αλλά υπάρχουν και περιπτώσεις όπου απαιτείται το ίδιο <video-content> σε διαφορετικές κωδικοποιήσεις) και τιμές για τα μεταδεδομένα που αφορούν τη δημοσίευση.

RULE: Ένας κανόνας (<rule>) που εφαρμόζεται πάνω σε ένα <video-content> ενός <collection> είναι στην πραγματικότητα μια «ελεύθερη» εκτέλεση βημάτων επεξεργασίας που λαμβάνει ως είσοδο την τρέχουσα μορφή της αντίστοιχης εγγραφής στο <published-list> και παράγει τη νέα μορφή του χρησιμοποιώντας τόσο υπηρεσίες του υποσυστήματος διαχείρισης εγγράφων, όσο και του Job Controller για την ικανοποίηση συνθηκών (π.χ. την παραγωγή μιας κωδικοποίησης που δεν υπάρχει) που αυτή τη στιγμή δεν ισχύουν.

Η λειτουργία του Program Manager περιλαμβάνει τη συνεχή αποτίμηση των <published-list>s με την εφαρμογή των <rule>s. Σε κάθε επανάληψη της διαδικασίας όπου έχουμε την επιτυχημένη ολοκλήρωση αποτίμησης του συνόλου των <rule>s σε ένα στοιχείο ενός του εκάστοτε <published-list>, εκτελείται το τελικό στάδιο επεξεργασίας που αφορά στην υλοποίηση της επικοινωνίας με το τελικό σημείο δημοσίευσης.

Η διαδικασία αυτή έχει διαφορετικές υλοποιήσεις που αντιστοιχούν στους διαφορετικούς τρόπου επικοινωνίας με του φορείς δημοσίευσης.



Εικόνα 22: Ροή πολυμεσικού περιεχομένου

Κάθε στοιχείο ενός <published-list> μπορεί να περιλαμβάνει:

- λίστα με αρχεία video σε συγκεκριμένες κωδικοποιήσεις (codec – container και ανάλυση)
- υποστηρικτικά αρχεία (π.χ. φωτογραφίες)
- μεταδεδομένα (π.χ. περίληψη)
- μεταδεδομένα δημοσίευσης:
- δημοσίευση από – έως
- geo-blocking rules
- τιμοκατάλογος, κ.α.

5.8 Επικοινωνία Program Manager με MediaStack

Ο Program Manager πρέπει να επαναλαμβάνει τον κύκλο της αποτίμησης των κανόνων σε ένα <published-list> κάθε φορά που λαμβάνει χώρα μια μεταβολή σε ένα registered <video-content>.

Για το λόγο αυτό υπάρχει μηχανισμός ασύγχρονης επικοινωνίας μεταξύ των δύο υποσυστημάτων, του program manager και της διαχείρισης εγγράφων.

Κατά την αποτίμηση ενός κανόνα, υπάρχει πολλές φορές ανάγκη να πραγματοποιηθούν μεταβολές που αφορούν τη διαχείριση εγγράφων (μεταβολές σε <video-content>). Παράδειγμα τέτοιας μεταβολής είναι η παραγωγή ενός νέου αρχείου με διαφορετική κωδικοποίηση.

Αυτού του είδους η επικοινωνία εξαρτάται από την υλοποίηση του κάθε <rule> και για την κάλυψη των αναγκών αυτών παρέχεται API ώστε να πραγματοποιούνται οι σχετικές μεταβολές από τις εξωτερικές στη διαχείριση εγγράφων διαδικασίες.

Κάθε τέτοια μεταβολή με τη σειρά της θα προκαλέσει την εκ νέου αποτίμηση των κανόνων και οι κύκλοι εκτέλεσης θα επαναλαμβάνονται μέχρι να μην υπάρχουν πλέον εγγραφές στα <published-list> σε εκκρεμότητα.

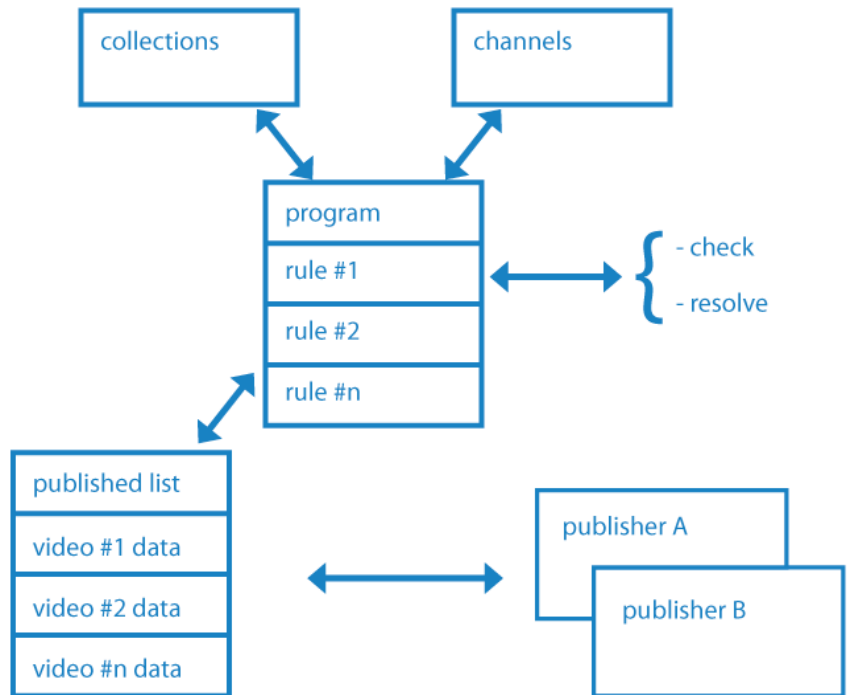
5.8.1 Υλοποίηση κανόνων

Η υλοποίηση των κανόνων αποτελεί λειτουργικότητα καίριας σημασίας. Παραδείγματα κανόνων μπορούμε να δούμε παρακάτω:

- has title / has description / has thumbnail / has mp4 @ akamai /
- plays after broadcast / allowed for GR /

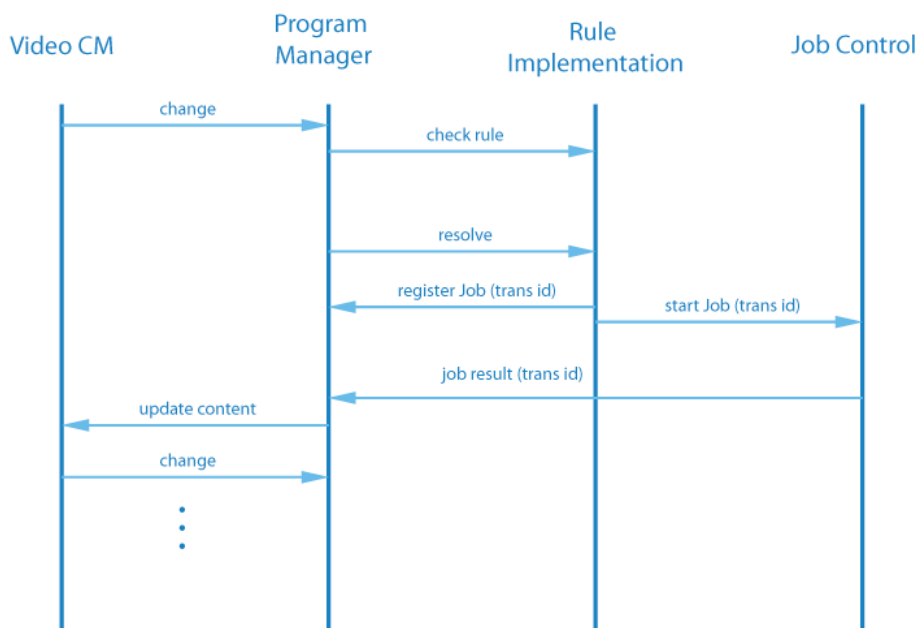
κάθε κανόνας πρέπει να περιλαμβάνει τα παρακάτω:

1. «σώμα» υλοποίησης όπου ελέγχει το σχετικό στοιχείο εισόδου (από το <published-list>)
2. «resolver» που είναι υπεύθυνος για την πραγματοποίηση των ενεργειών που απαιτούνται για την ικανοποίηση του κανόνα. Τέτοιες ενέργειες μπορεί να είναι:
 - a. η επικοινωνία με εξωτερικό σύστημα ή η χρήση των μεταδεδομένων του <video-content> για την «συμπλήρωση» του <published-list>
 - b. η ενεργοποίηση ενός workflow που σαν αποτέλεσμα θα έχει την μεταβολή του <video-content>
 - c. την αρχικοποίηση κάποιας λίστας εργασιών για τον Job Controller.



Εικόνα 23: Η υλοποίηση των κανόνων αποτελεί λειτουργικότητα καίριας σημασίας.

Για όσες περιπτώσεις απαιτείται η ασύγχρονη εκτέλεση μιας σειράς εργασιών (για την ικανοποίηση ενός rule), παράγεται ένα transaction ID το οποίο μεταφέρεται διαφανώς κατά την εκτέλεση των εργασιών και επιστρέφεται μαζί με τα αποτελέσματα (ή την επιβεβαίωση της εκτέλεσης) ώστε να γνωρίζει ο Program Manager (ή το VideoCM) αν θα ενεργοποιήσει ένα νέο κύκλο αποτίμησης κανόνων.



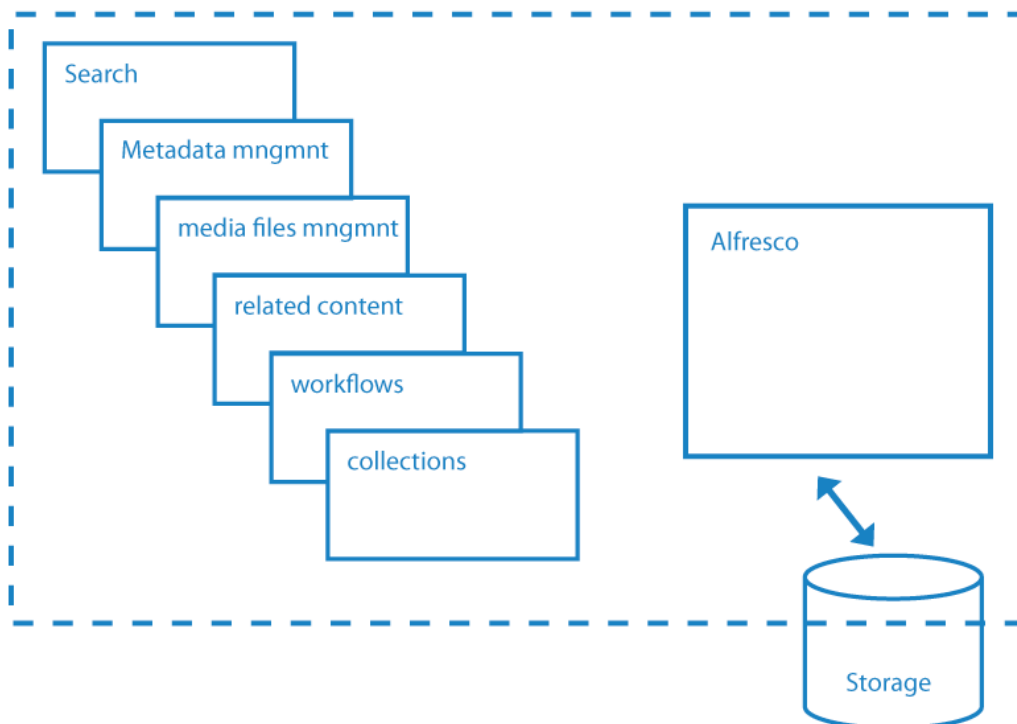
Εικόνα 24: Αρχιτεκτονική MediaStackRules.

5.9 Video CM - User interfaces

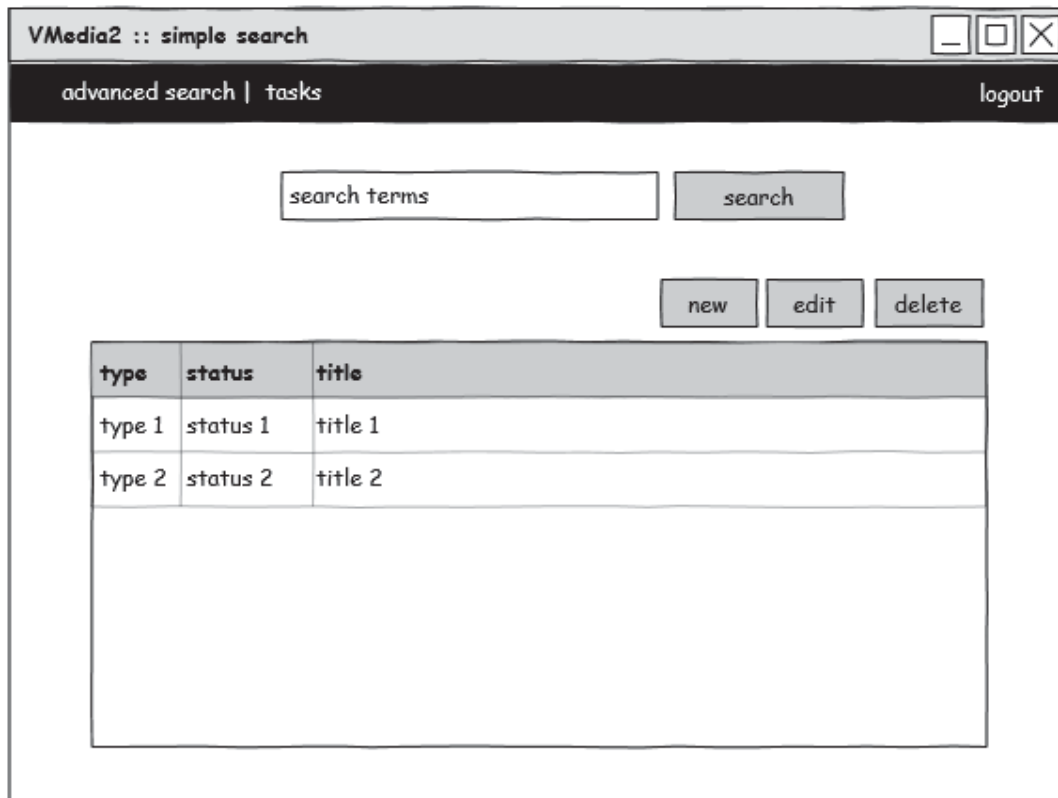
Τα κύρια User Interfaces που εντοπίζονται να αντιστοιχούν με τις απαιτήσεις χρήσης είναι:

Αναζήτηση	Η αναζήτηση των <video-content> με λιτή δομή για full-text-search και advanced mode για αναζήτηση με αναλυτικά κριτήρια
Επεξεργασία	Η «καρτέλα» του <video-content> για τη διαχείριση μεταδεδομένων, σχετικών εγγραφών στο document management περιβάλλον, διαχείριση εκτέλεσης εργασιών, κλπ
Εργασίες	Οι εργασίες παρουσιάζουν σε συνοπτική μορφή λίστας τις εκκρεμότητες (assigned tasks) του χρήστη με ευκολίες για ταξινόμηση και μεταφορά στο σχετικό περιεχόμενο για ολοκλήρωση της σχετικής εργασίας.

Πίνακας 3: κύρια user interfaces



Εικόνα 25: Ποή MediaStack User Interface.



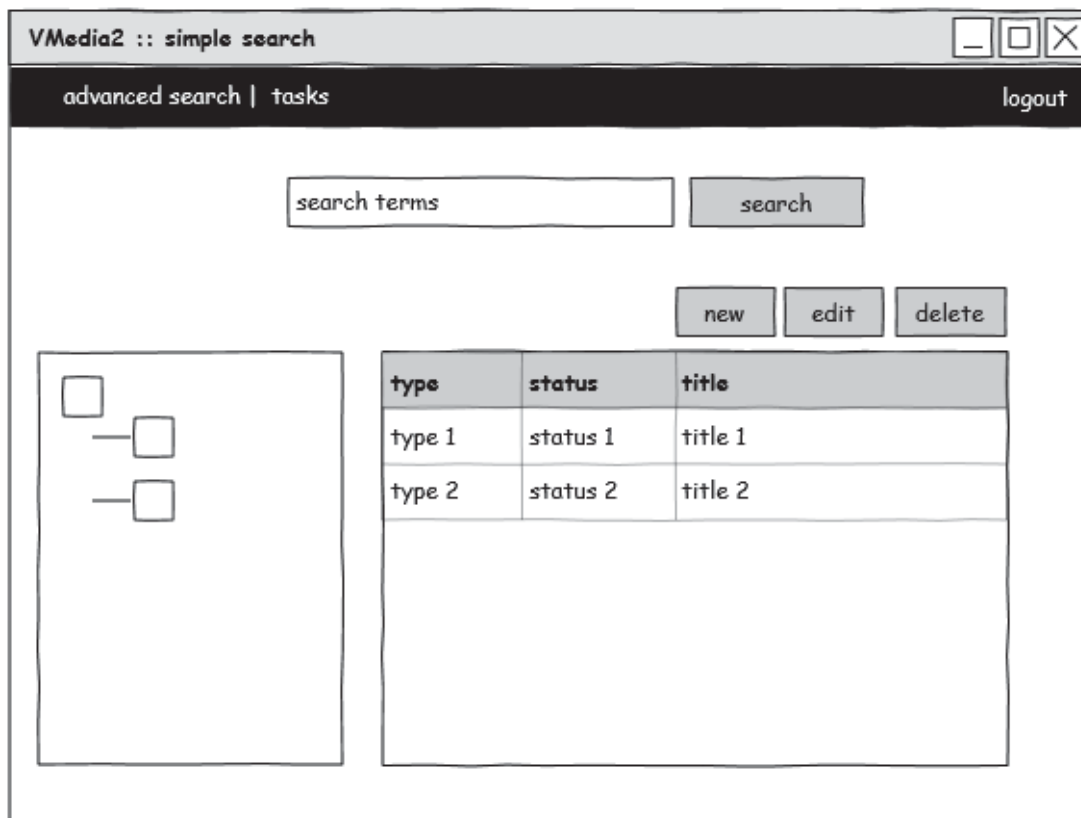
Εικόνα 25: Mockup αναζήτησης περιεχομένου με βάση τον τίτλο της εκπομπής

Η αναζήτηση έχει δύο μορφές: την απλή αναζήτηση και την σύνθετη αναζήτηση. Στην απλή αναζήτηση ο χρήστης συμπληρώνει όρους οι οποίοι χρησιμοποιούνται για full-text search στις εγγραφές που έχει δικαίωμα να δει ο χρήστης.

Τα αποτελέσματα της αναζήτησης εμφανίζονται στην ίδια σελίδα με σελιδοποίηση.

Στο interface της αναζήτησης θα μπορούσε να προστεθεί control με την ιεραρχική οργάνωση του περιεχομένου έτσι ώστε ο χρήστης να έχει τη δυνατότητα να περιορίσει τα αποτελέσματα της αναζήτησης στο περιεχόμενο που βρίσκεται από τον επιλεγμένο κόμβο του δέντρου και κάτω.

5.9.1 Αναζήτηση



Εικόνα 26: Mockup αναζήτησης περιεχομένου με βάση meta – δεδομένα από το Alfresco

The image shows a mockup of an advanced search interface for a system named VMedia2. The window title is "VMedia2 :: advanced search". At the top, there are navigation links for "simple search" and "tasks", and a "logout" link. The search form includes the following fields and options:

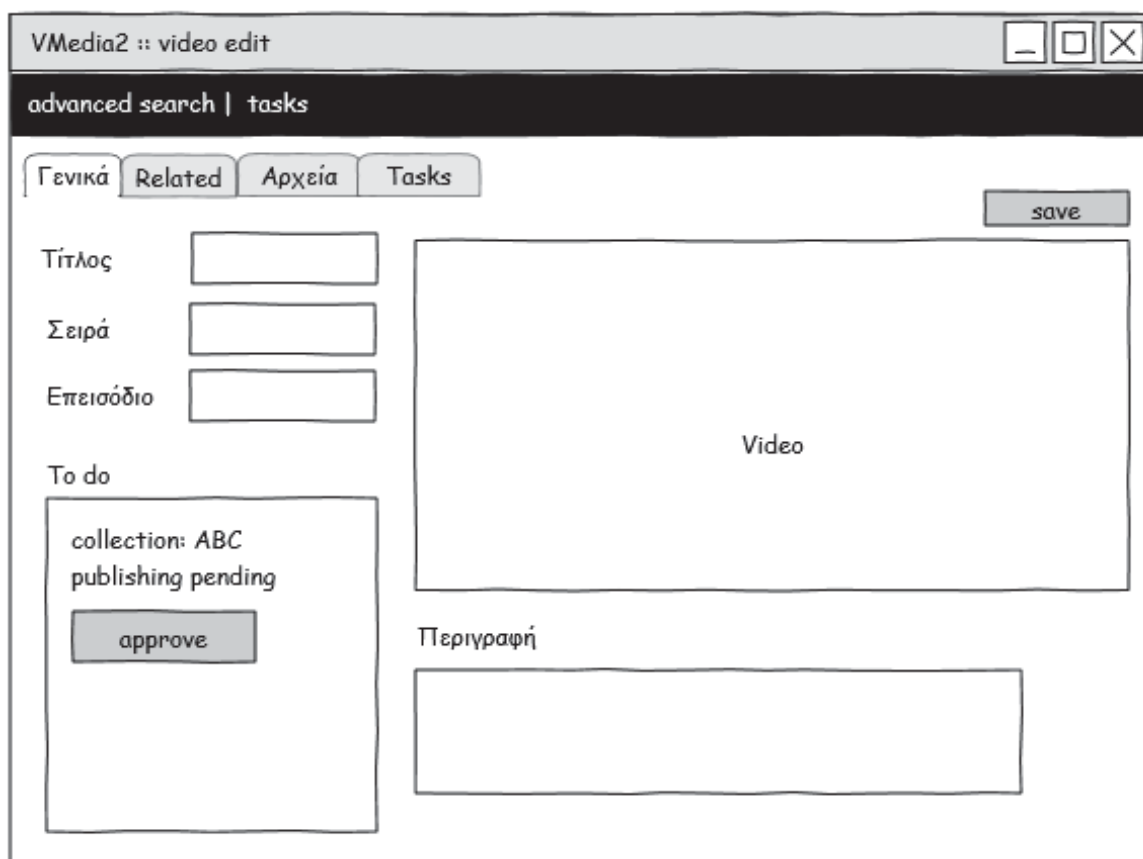
- search terms:** A text input field followed by a "search" button.
- category:** A dropdown menu.
- series code:** A text input field.
- episode code:** A text input field.
- status:** Three checkboxes labeled "published", "approved", and "canceled".
- title:** A text input field.
- tags:** A text input field.
- filetype:** Three checkboxes labeled "mp4", "flv", and "youtube".

Εικόνα 27: Mockup σύνθετης αναζήτησης περιεχομένου.

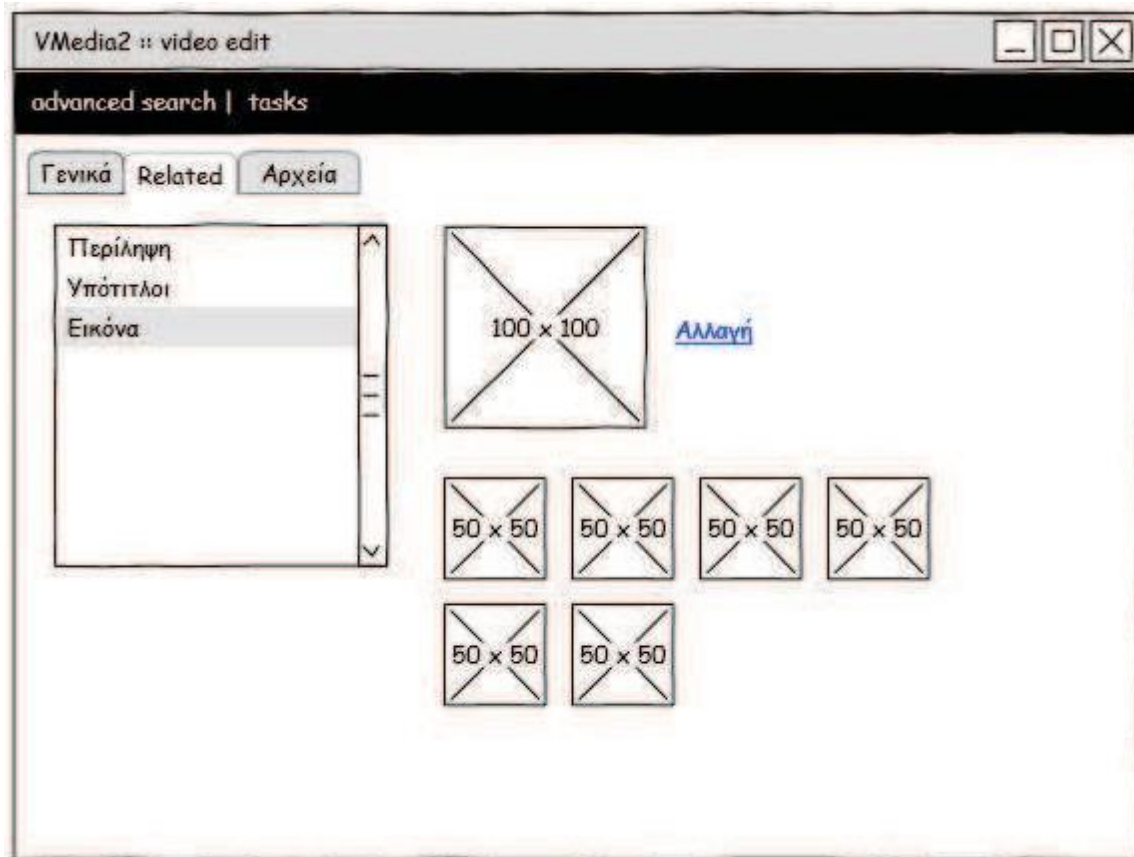
Για την αναζήτηση εγγραφών του αρχείου ειδήσεων όπου απαιτούνται άλλα εξειδικευμένα metadata που αφορούν μόνο τις εγγραφές αυτής της κατηγορίας, μπορούν να ακολουθηθούν εναλλακτικές προσεγγίσεις, όπως:

1. Διακριτή φόρμα αναζήτησης για τις εγγραφές του αρχείου ειδήσεων
2. Χρήση της αναλυτικής αναζήτησης, όπου στην περίπτωση που επιλέγεται η συγκεκριμένη κατηγορία στο σχετικό κριτήριο, θα εμφανίζονται επιπλέον κριτήρια που αφορούν την αναζήτηση σε αυτές τις εγγραφές.

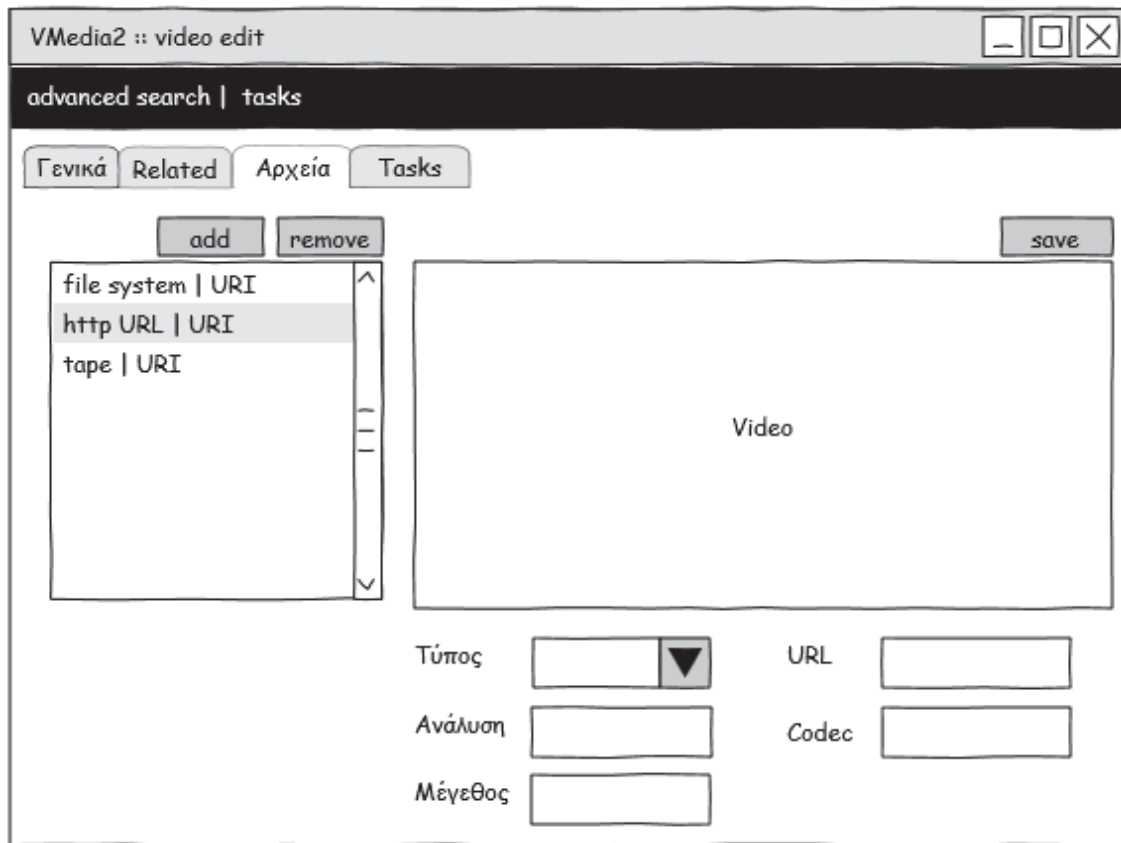
5.9.2 Επεξεργασία



Εικόνα 28: Mockup επεξεργασίας περιεχομένου.

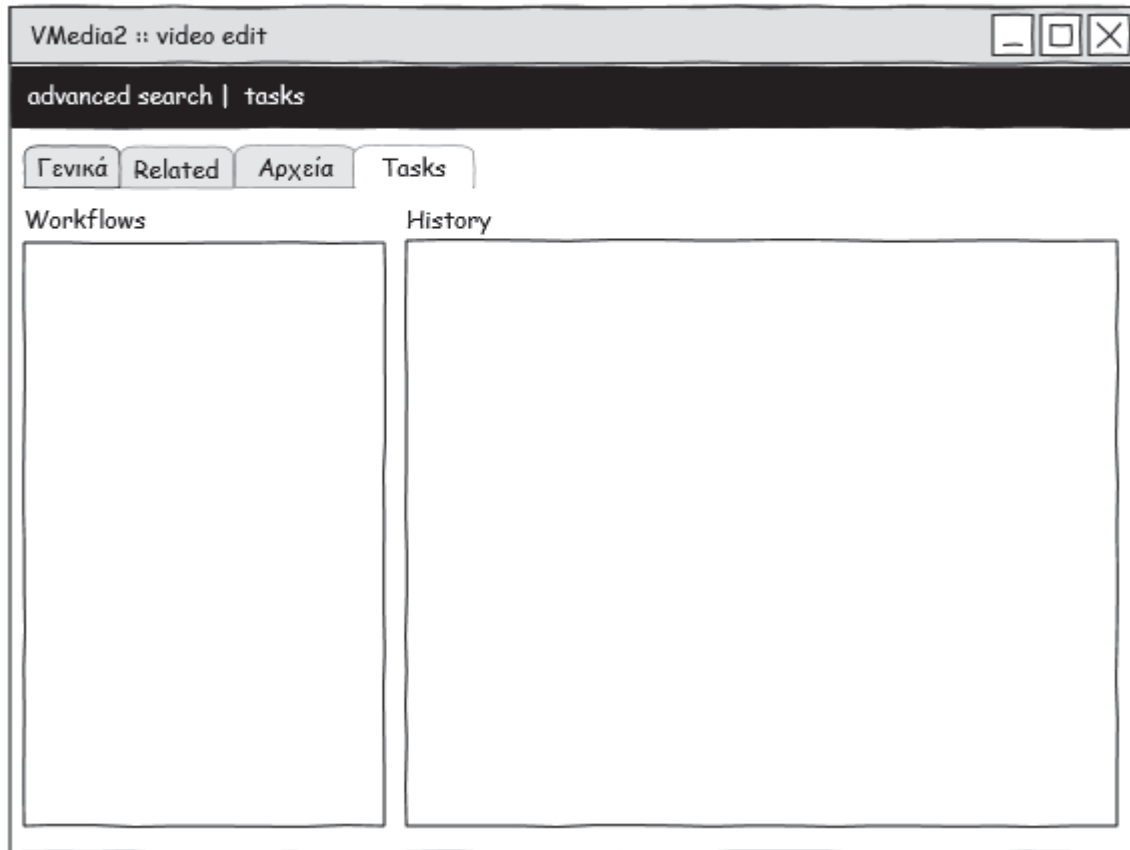


Εικόνα 29: Mockup επεξεργασίας λεπτομερειών περιεχομένου.



Εικόνα 30: Mockup επεξεργασίας δεδομένων Alfresco

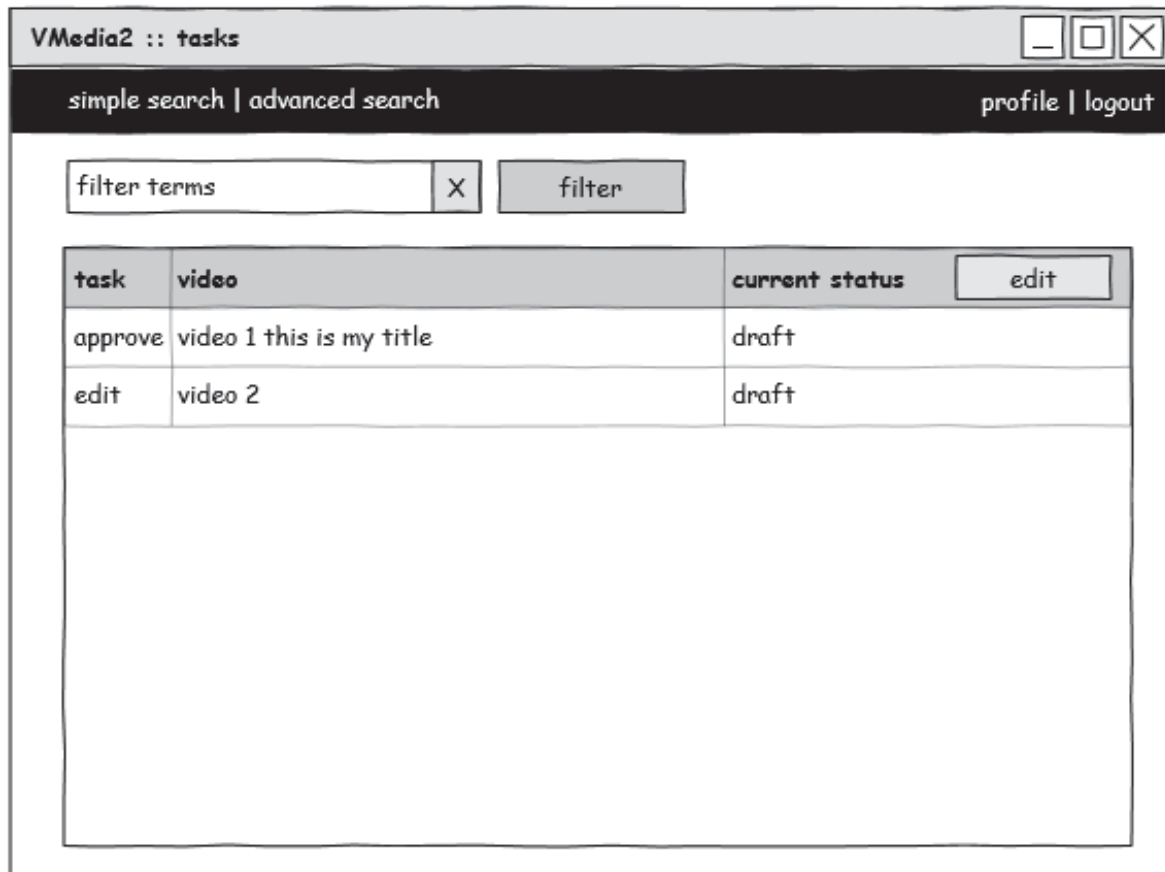
Για τη σύνδεση video content με image θα υπάρχει η δυνατότητα δημιουργίας εικόνων/thumbnails από το video, φόρτωση εικόνας ή επιλογή από το αρχείο images. Θα υπάρχει preselected image για κάποιους τύπους video-content.



Εικόνα 31: Mockup σύνδεση video content με image.

5.9.3 Εργασίες

Το user interface των εργασιών θα παρέχει μια λίστα των εκκρεμών εργασιών που αφορούν το χρήστη.



Εικόνα 32: Automation Flow

5.10 Θέματα υλοποίησης του data model

Ένα βασικό ζήτημα που αφορά την υλοποίηση του data model είναι ποιες οντότητες και χαρακτηριστικά θα αποτελούν στοιχεία διαχείρισης από το document management system (alfresco) και ποια από αυτά θα υλοποιηθούν ως σχεσιακά δεδομένα (database tables).

5.11 Κανόνες και διαδικασίες διαχείρισης των δεδομένων

5.11.1 Ρόλοι Και Δικαιώματα

Τα δικαιώματα διαχείρισης του <video-content> είναι απλά. Ο κάθε χρήστης διαθέτει πλήρη δικαιώματα δημιουργίας, τροποποίησης και διαγραφής ενός <video-content> ανάλογα με την ομάδα που ανήκει.

Τα <video-content> ανήκουν σε κατηγορίες (οι οποίες αποτελούν αντικείμενο επεξεργασίας για διαφορετικά business-units) και υπάρχει αντιστοίχιση με ομάδες χρηστών όσον αφορά την αρμοδιότητα διαχείρισής τους.

Εάν ένας χρήστης ανήκει σε κάποια ομάδα, αυτόματα αποκτά δικαιώματα διαχείρισης των <video-content> που ανήκουν στις κατηγορίες που αυτή η ομάδα μπορεί να διαχειρίζεται.

5.12 Εξωτερικά υποσυστήματα

5.12.1 Δανεισμός Κασετών

Η μεταφορά των <tape-content> στο νέο σύστημα διαχείρισης θα προσφέρει ένα νέο εργαλείο στους χρήστες του παλαιού συστήματος, με την εξαίρεση της λειτουργικότητας του δανεισμού των κασετών.

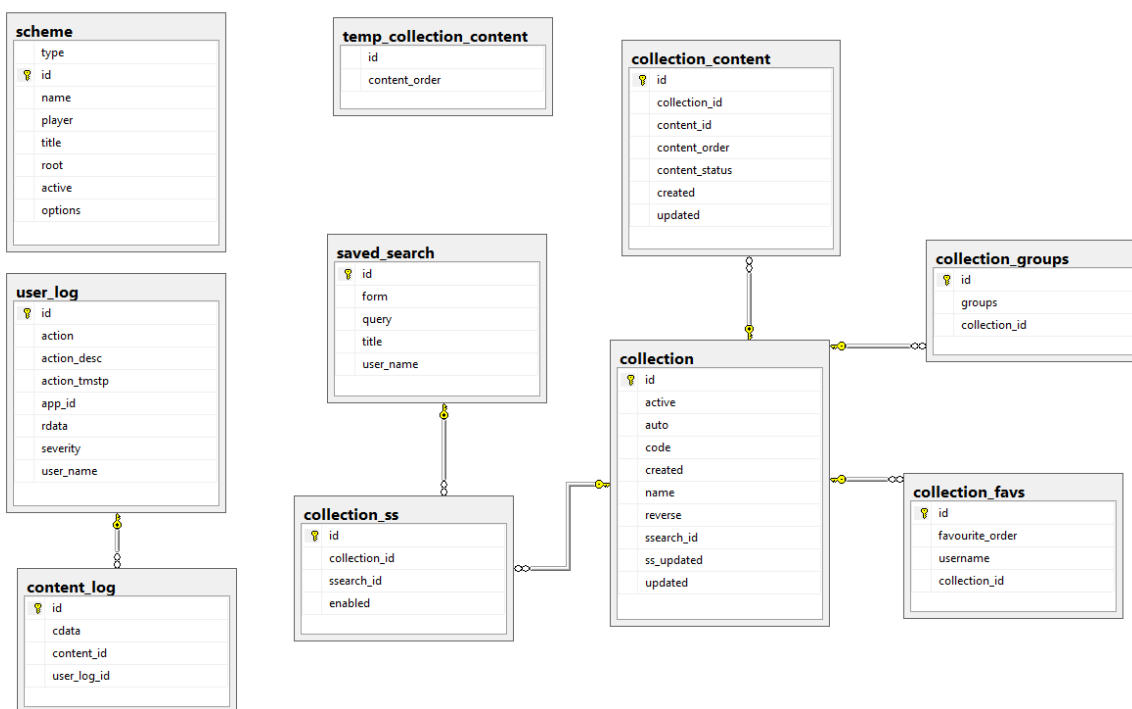
Η συγκεκριμένη διαδικασία πρέπει να υποστηριχθεί με την υλοποίηση αυτόνομης εξωτερικής εφαρμογής.

5.13 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Όπως αναφέρθηκε παραπάνω, το σύστημα μπορεί να υποστηρίξει την ταυτόχρονη και ασύγχρονη επεξεργασία πολλαπλών βίντεο ταυτόχρονα.

Για το λόγο αυτό ως βάση δεδομένων επιλέχθηκε η PostgreSQL, η οποία αφενός αποτελεί έργο ανοιχτού κώδικα, αλλά ταυτόχρονα δεν περιλαμβάνει κανένα περιορισμό αναφορικά τόσο με την ποσότητα των database transactions, όσο και με τη χωρητικότητα.

Το σχήμα της βάσης δεδομένων παρουσιάζεται διαγραμματικά παρακάτω:



Εικόνα 33: Media Stack Entity Diagram

5.14 ΑΝΑΛΥΣΗ ΠΙΝΑΚΩΝ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Η βάση δεδομένων του MediaStack περιλαμβάνει τους παρακάτω πίνακες:

Scheme
Type
Id
Name
Player
Title
Root
Active
Options

Ο πίνακας Scheme αποθηκεύει γενικές πληροφορίες εκκίνησης του τηλεοπτικού προγράμματος. Περιλαμβάνει πληροφορίες σχετικά με το ίδιο το πρόγραμμα, τα διαθέσιμα κανάλια προβολής, καθώς και βασικά στοιχεία όπως ο τίτλος και κωδικός προγράμματος.

User log
Id
cdata
Content_id
User_log_id

User log details
Id
Action
Action_desc
Action_tmstp
App_id
Rdata
Severity
User_name

Οι πίνακες User Log, User Log details, αποθηκεύουν πληροφορίες σχετικά με τη δημιουργία ενός προγράμματος ή μιας εκπομπής στους video servers. Περιλαμβάνουν στοιχεία σχετικά με το χρήστη που δημιούργησε το πρόγραμμα, την εγγραφή των meta-δεδομένων στα διάφορα υποσηστήματα της εφαρμογής καθώς και τις προτεραιότητες αναμετάδοσης σε συγκεκριμένη ζώνη ώρας. Η τελευταία αποτελεί πληροφορία που δανείζεται από το σύστημα του σταθμού που είναι υπεύθυνο για τα συμβόλαια προβολής προγράμματος.

Temp collection content
Id
Content_order

Όταν ένα τηλεοπτικό πρόγραμμα “ανοίξει” για πρώτη φορά στο σύστημα και ώσπου να συμπληρωθούν όλες του οι πληροφορίες και να λάβει τα δικαιώματα και σήματα προβολής από τα υπόλοιπα υποσυστήματα του σταθμού, θεωρείται προσωρινό. Κατα συνέπεια αποθηκεύεται στον πίνακα Temp collection content.

Saved search
Id
Form
Query
Title
User_name

Ο πίνακας Saved Search, προσφέρει μια βοηθητική λειτουργία στους χρήστες του λογισμικού. Είναι υπεύθυνος για την δημιουργία custom κριτηρίων αναζήτησης εκπομπών ανά χρήστη.

Collection_ss
Id
Collection_id
Search_id
Enabled

Collection_content
Id
Collection_id
content_id
Content_order
Content_status
Created
Updated

Collection
Id
Active
Auto
Code
Created
Name
Reverse
ssearch_id
Ss_updated
Updated

Collection_groups
Id
Groups
Collection_id

Collection_favs
Id
Favourite_order
Username
Collection_id

Τέλος οι πίνακες με πρόθεμα «Collection» περιλαμβάνουν όλη την πληροφορία προγράμματος.

5.15 ΠΕΡΙΓΡΑΦΗ ΑΥΤΟΜΑΤΙΣΜΩΝ

5.16 FFMPEG

Το Ffmpeg είναι ένα πλήρες, πολυ-πλατφορμικό εργαλείο για την καταγραφή, τη ροή και τη μετατροπή ήχου και βίντεο.

Συνήθως το ffmpeg βρίσκεται πίσω από διάφορα προγράμματα μετατροπής βίντεο από ένα μορμά σε άλλο, καταγραφής βίντεο από συσκευές ή ως μηχανή για αποστολή ροής βίντεο-ήχου στο δίκτυο.

Αποτελεί την ναυαρχίδα των πολυμέσων και είναι σε θέση να αποκωδικοποιήσει, κωδικοποιήσει, να κάνει transcode (άμεση μετατροπή κωδικοποίησης από αναλογική σε αναλογική ή ψηφιακή σε ψηφιακή), σύμπλεξη (mux) ή αποσύμπλεξη (demux) σημάτων, να κάνει streaming, να φιλτράρει και να παίζει ούτε λίγο ούτε πολύ οτιδήποτε οι άνθρωποι και οι μηχανές έχουν δημιουργήσει.

Υποστηρίζει τις πιο παλιές μέχρι τις πλέον σύγχρονες μορφές αρχείων.

Περιέχει τις βιβλιοθήκες libavcodec, libavutil, libavformat, libavfilter, libavdevice, libswscale και libswresample που μπορούν να χρησιμοποιηθούν από τις εφαρμογές.

Επίσης τα: ffmpeg (εργαλείο γραμμής εντολών), ffmpeg (multimedia streaming server για ζωντανές μεταδόσεις), ffplay (απλό πρόγραμμα αναπαραγωγής) και ffprobe (απλός αναλυτής πολυμέσων) που μπορεί να χρησιμοποιηθούν από τους τελικούς χρήστες για μετατροπές ψηφιακών / αναλογικών σημάτων, streaming και παίξιμο.

Προσπαθεί να παρέχει την καλύτερη δυνατή τεχνική λύση για τους προγραμματιστές εφαρμογών αλλά και τους τελικούς χρήστες.

Για να επιτευχθεί αυτό, συνδυάζονται οι καλύτερες διαθέσιμες ελεύθερες επιλογές λογισμικού.

Η ασφάλεια αποτελεί υψηλή προτεραιότητα και η αναθεώρηση του κώδικα γίνεται πάντα με γνώμονα την ασφάλεια.

Στο MediaStack το ffmpeg χρησιμοποιείται από αυτόματους agents, οι οποίοι πυροδοτούνται μόλις το βίντεο λάβει όλες τις απαιτούμενες μετα-πληροφορίες. Στην περίπτωση αυτή δημιουργείται ένα message το οποίο ξεκινά το video transcoding.

5.17 ALFRESCO

Το Alfresco ECM ανήκει σε μια οικογένεια λογισμικών υπό την ονομασία Enterprise Content Management (ECM) και πρόκειται για μια κατηγορία συστημάτων που επιτρέπουν την αποτελεσματικότερη διαχείριση εγγράφων, πολυμέσων και άλλων αρχείων και διευκολύνουν την αρχειοθέτησή τους. Τα συστήματα ECM μπορούν να βελτιώσουν τη συνεργασία στο περιβάλλον εργασίας και την αποδοτικότητα της δουλειάς συγκεντρώνοντας όλη την πληροφορία κάθε έργου σε ένα ψηφιακό χώρο (workspace), παρέχοντας ένα ενιαίο περιβάλλον διαχείρισης.

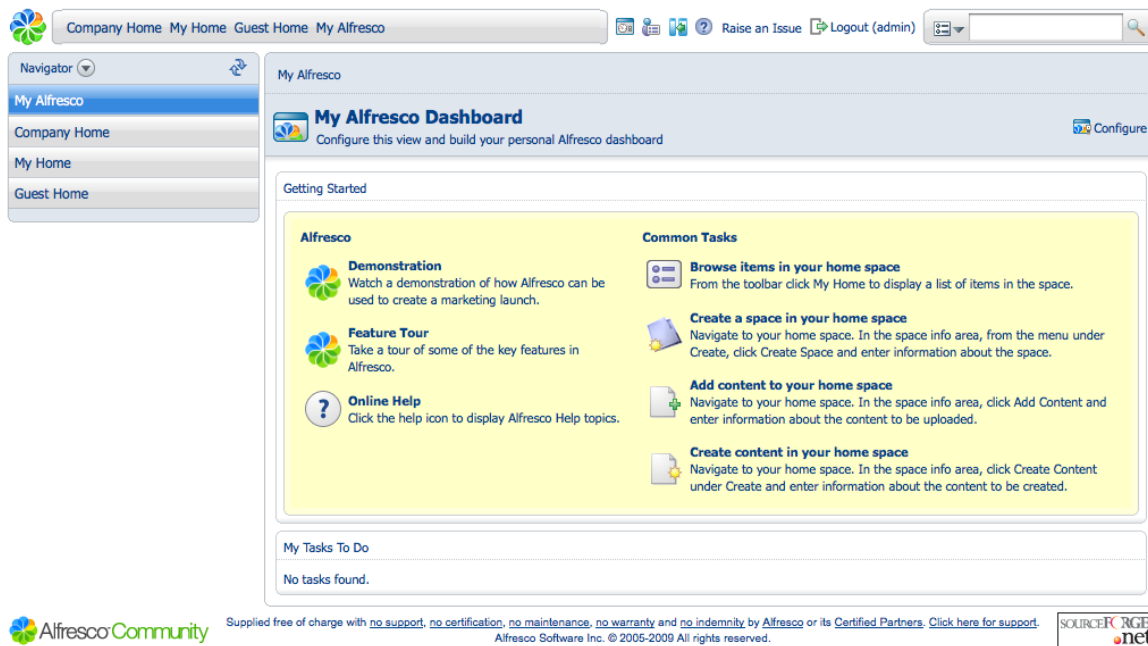


Εικόνα 33: Alfresco Asset Manager Flow

Η πλατφόρμα ανοιχτού κώδικα Alfresco ECM είναι ένα ισχυρό και αποτελεσματικό εργαλείο διαχείρισης περιεχομένου. Βασίζεται σε τεχνολογίες ανοιχτού λογισμικού και σε ανοιχτά πρότυπα εγγράφων επιδεικνύοντας σεβασμό στην “αυτοδιάθεση” του χρήστη σχετικά με το περιεχόμενο που παράγει και διαχειρίζεται.

Οι σημερινές ανάγκες ενός τηλεοπτικού σταθμού χαρακτηρίζονται από μεγαλύτερη κινητικότητα (mobility) του προσωπικού (εξωτερικά ρεπορτάζ, εξωτερικά γυρίσματα κ.τ.λ.), αλληλεπίδραση με κοινωνικά δίκτυα (social features), ανάγκη για απομακρυσμένη πρόσβαση (cloud) στο εταιρικό περιεχόμενο και αναβαθμισμένες απαιτήσεις αναζήτησης του περιεχομένου. Δυνατότητες όπως η κοινή χρήση αρχείων με συγχρονισμό, η χρήση μεταδεδομένων (metadata) για την περιγραφή των αρχείων, η διαθεσιμότητα εφαρμογών στο cloud και μάλιστα με αξιοποίηση ανοιχτών προτύπων γίνονται ολοένα και πιο απαραίτητες όταν ενδιαφέρει η αποδοτικότητα και η αποτελεσματικότητα στο σημερινό εργασιακό περιβάλλον.

Το Alfresco προσφέρει λύση για όλα τα παραπάνω και κατά συνέπεια αποτελεί την ιδανική λύση για τη διανομή και διάθεση τηλεοπτικού υλικού.



Εικόνα 34: Alfresco User Interface

Alfresco mobile: το ενιαίο περιβάλλον εργασίας που παρέχει το Alfresco διατίθεται για smartphones (iphone, android) και για ταμπλέτες (ipad, android) επιτρέποντας στο χρήστη να έχει πρόσβαση στο εταιρικό περιεχόμενο από τη συσκευή που επιθυμεί.

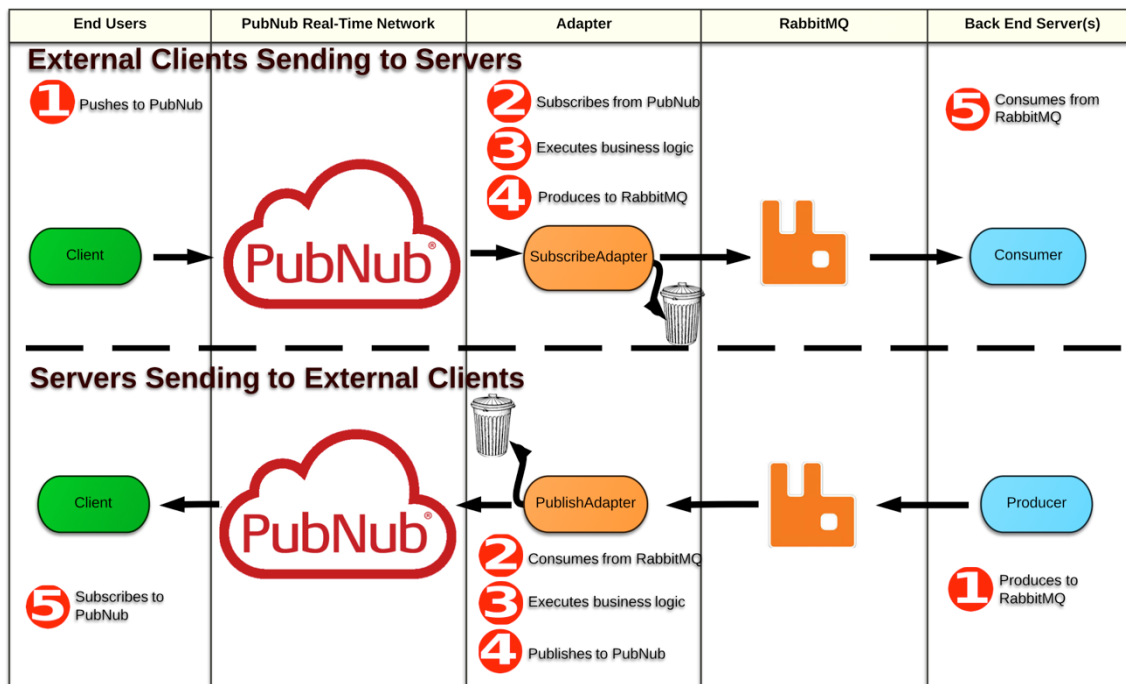
Social networking: Η κοινωνική διάσταση στην ανάπτυξη της επιχειρηματικότητας παίζει όλο και σημαντικότερο ρόλο. Η πλατφόρμα του Alfresco ενσωματώνει εργαλεία επικοινωνίας με κοινωνικά δίκτυα (facebook, twitter) για εύκολη και άμεση κοινοποίηση του επιθυμητού περιεχομένου σε αυτά, με έμφαση στην ασφάλεια του περιεχομένου και τη δυνατότητα επέκτασης της χρήσης τους.

Το Alfresco παρέχει εγγενώς λειτουργία απομακρυσμένης πρόσβασης στα εταιρικά δεδομένα είτε μέσω του server της εταιρίας, ή μέσω cloud. Η δυνατότητα συγχρονισμού του cloud με τον εταιρικό server, ώστε να βγαίνουν προς τα έξω επιλεγμένα δεδομένα, δίνει ακόμα μεγαλύτερες εγγυήσεις ασφάλειας.

Η συνεργατική δουλειά με τη χρήση του Alfresco επιτρέπει στους χρήστες να συγκεντρώνονται στην εργασία τους, αντί να ξοδεύουν ενέργεια στην αναζήτηση αρχείων και σε ατέρμονες συνεννοήσεις. Δυνατότητες όπως η αποθήκευση εκδόσεων των αρχείων (versioning) και η ανιχνευσιμότητα των αλλαγών (audit trail), η αναζήτηση μέσα στο κείμενο των εγγράφων (εκτός από τα μεταδεδομένα και τον τίτλο τους), η συγκέντρωση όλης της πληροφορίας του έργου σε ένα σημείο και ο καθορισμός της προσβασιμότητας των χρηστών στα αρχεία μέσω ανάθεσης ρόλων επιτρέπουν τον έλεγχο των δεδομένων και την παραγωγική τους χρήση.

5.18 RABBIT MQ

Το RabbitMQ αποτελεί ένα “message broker”. Είναι ένα project ανοιχτού κώδικα το οποίο σχεδιάστηκε ως ένα κατανεμημένο σύστημα μηνυμάτων (messaging) για συλλογή και μεταφορά Log αρχείων με ελάχιστη καθυστέρηση (low latency) και high-throughput σε πραγματικό χρόνο.



Εικόνα 35: RabbitMQ and JMS Flow

Το RabbitMQ βρίσκει χρήση στο messaging καθώς χαρακτηριστικά όπως high throughput, τεμάχιση αρχείων (built-in partitioning), διατήρηση αντιγράφων (replication) και η ανοχή σε σφάλματα (fault-tolerance) το κάνουν ιδανική λύση σε εφαρμογές επεξεργασίας μηνυμάτων (message processing applications) μεγάλης κλίμακας. Συχνά χρησιμοποιείται για παρακολούθηση (monitoring) δεδομένων συγκεντρώνοντας στατιστικά και δεδομένα χρήσης από κατανεμημένες εφαρμογές.

Μέσω του RabbitMQ είναι δυνατό οι λεπτομέρειες που αφορούν τα αρχεία των log να αφαιρεθούν και να αποσταλεί μια πιο “καθαρή” έκδοση των log ή των event data ως ρεύμα (stream) μηνυμάτων. Άλλες εφαρμογές του RabbitMQ είναι η επεξεργασία ρευμάτων (stream processing), event sourcing, παρακολούθηση δραστηριότητας web εφαρμογών κτλ.

5.19 HAZELCAST

Το λογισμικό Hazelcast είναι ένα δημοφιλές σύστημα distributed caching το οποίο συνεργάζεται με το Spring Framework και αποτελεί αναπόσπαστο μέρος του MediaStack, αφού συμβάλει με διπλό τρόπο. Αρχικά διατηρεί την ακεραιότητα του συστήματος ακόμα και κατά τη διάρκεια της επεξεργασίας του πολυμεσικού υλικού και επιπρόσθετα ελαχιστοποιεί τις αιτήσεις προς τη βάση δεδομένων.

Η υλοποίηση του μηχανισμού του caching στο MediaStack, βασίζεται σε συγκεκριμένα βήματα:

Το Hazelcast, αρχικά, αποφασίζει αν το μήνυμα επιδέχεται caching και στη συνέχεια αναγνωρίζει αν υπάρχει διαθέσιμος χώρος για την αποθήκευση του. Στην περίπτωση που δεν υπάρχει χώρος, να αποφασίσει ποιά cached αντικείμενα θα αντικατασταθούν.

Η cache, με την λήψη κάποιας αίτησης θα πρέπει να αποφασίσει αν είναι σε θέση να την εξυπηρετήσει. Εάν είναι όντως σε θέση να διεκπεραιώσει την αίτηση, θα πρέπει να επιστρέψει τον πόρο στον αιτούμενο client και να ενημερώσει το MediaStack, το οποίο μέσω του hazelcast, φροντίζει να έχει μία πολιτική συνέπειας (συνάφειας) για να διατηρεί την κατάσταση επικαιρότητας (freshness information) του πόρου.

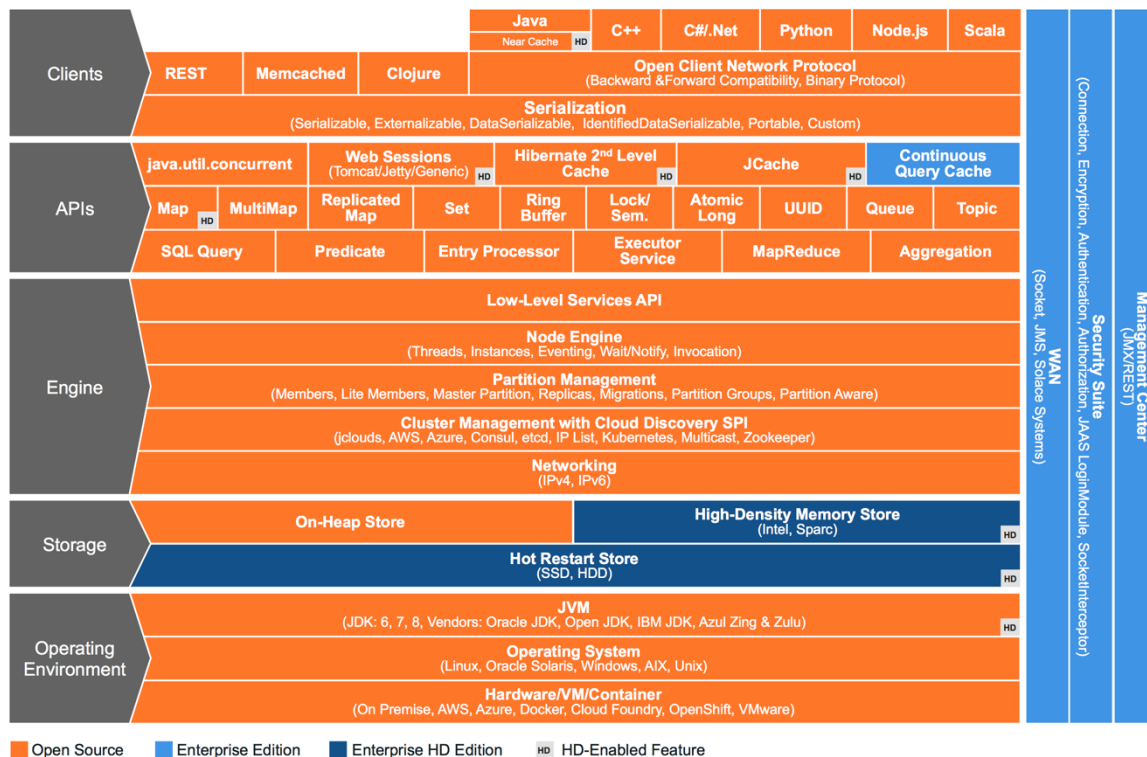
Μετά την απόφαση για αποθήκευση ενός πόρου, το hazelcast ελέγχει αν μπορεί να προχωρήσει στην αποθήκευση χωρίς να διαγράψει κάποια από τα αντικείμενα που ήδη βρίσκονται στην cache. Εάν όχι, ενεργοποιείται ο μηχανισμός cache replacement (αντικατάστασης cache). Ο μηχανισμός αυτός εισάγει κάποια επιβάρυνση, ειδικά αν μικρότερα αντικείμενα που είναι ήδη cached πρέπει να διαγραφούν. Πρόσθετη επιβάρυνση προκαλείται όταν λαμβάνονται, μελλοντικά, αιτήσεις για αντικείμενα τα οποία έχουν διαγραφεί. Σε αυτήν την περίπτωση πρέπει να εγκατασταθούν νέες συνδέσεις για την άντληση τους από τους origin servers. Συχνά, πόροι οι οποίοι θεωρούνται ότι δεν είναι επίκαιροι (stale) διαγράφονται από την cache ακόμη και στην περίπτωση που αυτή δεν είναι πλήρης. Έτσι περιορίζεται η ανάγκη για ενεργοποίηση του μηχανισμού cache replacement την στιγμή διεκπεραίωσης μίας αίτησης (μείωση της υστέρησης που εκλαμβάνεται ο χρήστης).

Μόλις προκύπτει ελεύθερος χώρος, η cache εξάγει πληροφορία για το μήνυμα όπως το χρόνο τελευταίας μεταβολής καθώς και πληροφορία για την λήξη του πόρου. Επικεφαλίδες όπως οι Expire και Cache-control: max-stale μεταφέρουν πληροφορία σχετική με την λήξη (expiration) του πόρου. Αυτά τα πεδία συντελούν στο να είναι η cache συμβατή με τους περιορισμούς του πρωτοκόλλου HTTP για το βάθος χρόνου στο οποίο η απάντηση μπορεί να επιστραφεί ως σημασιολογικά έγκυρη. Μία cache η οποία είναι συμβατή με το πρωτόκολλο είναι υποχρεωμένη να εξασφαλίζει ότι οι απαντήσεις που επιστρέφει θεωρούνται από τον origin server ως επίκαιρες (fresh). Αν απουσιάζουν πληροφορίες λήξης του πόρου, η cache προσδιορίζει ευρεστικά ένα χρόνο λήξεως για να αποφασίσει πότε καθίσταται μη-εγκυρος (stale). Ο αλγόριθμος μπορεί να βασίζεται στη τιμή του πεδίου Last-Modified που είναι συνηθισμένη με τον πόρο. Για παράδειγμα, μία cache μπορεί να προσθέσει ένα

καθορισμένο χρονικό διάστημα π.χ. 10' στην τιμή Last-modified και να χρησιμοποιήσει την νέα τιμή σαν διάστημα ελέγχου επικαιρότητας (freshness interval). Επίσης, παράγεται ένα κλειδί για χρήση σε μελλοντικές αναφορές (lookups). Το κλειδί αυτό είναι μία τιμή κατακερματισμού (hash value) η οποία βασίζεται στο URL του πόρου.

Εάν ένας πόρος που αντιπροσωπεύει κάποιο κλειδί που αναζητείται στην cache εντοπιστεί, θεωρείται ότι συνέβει ένα cache hit. Τότε, ανάλογα με την πολιτική της cache και ενδεχόμενους περιορισμούς που επιβάλλονται από πεδία επικεφαλίδας, ένας επανέλεγχος ορθότητας μπορεί να εκτελεστεί για να διαπιστωθεί εάν ο πόρος είναι επίκαιρος.

Εάν ο έλεγχος αυτός αποβεί θετικός, η αίτηση ικανοποιείται από την cache. Διαφορετικά, η cache ανακτά ένα νέο αντίγραφο του πόρου και εφαρμόζει την πολιτική της για να αποφανθεί αν ο πόρος θα πρέπει να αποθηκευτεί παράλληλα με την προώθηση του στον αιτούμενο client. Εάν ο πόρος δεν εντοπιστεί στην cache, (περίπτωση cache miss) η αίτηση προωθείται.



Εικόνα 36: Σχηματική αναπαράσταση Hazelcast Caching

5.20 MEDIA OBJECT SERVER COMMUNICATION PROTOCOL

Το MOS Project είναι μία πρωτοβουλία πολλών ηγετών της βιομηχανίας broadcasting που συνεργάστηκαν προκειμένου να παρέχουν ένα κοινό τρόπο σύνδεσης των συστημάτων διαχείρισης ειδήσεων και των media servers.

Σκοπός του πρωτοκόλλου MOS είναι να παρέχει τη δυνατότητα επιλογής του υπολογιστικού συστήματος διαχείρισης ειδήσεων σε συνδυασμό με το μη-γραμμικό σύστημα αποθήκευσης και επεξεργασίας video που ταιριάζει καλύτερα στις ανάγκες του κάθε τηλεοπτικού σταθμού.

Το πρωτόκολλο επικοινωνίας MOS συνδέει τα προαναφερθέντα συστήματα με έναν ανοικτό, μη αποκλειστικό τρόπο. Παρέχει έναν σύγχρονο, διαδραστικό σύνδεσμο μεταξύ οποιουδήποτε Media Object Server (οποιοδήποτε υπολογιστικό σύστημα ή διακομιστής που αποθηκεύει video, ήχο, ακίνητες φωτογραφίες ή υλικό που έχει δημιουργηθεί από γεννήτριες χαρακτήρων) και των συστημάτων διαχείρισης ειδήσεων. Είναι ένα σημαντικό βήμα ώστε η ενσωμάτωση - ολοκλήρωση των συστημάτων επιτραπέζιας επεξεργασίας, αποθήκευσης και διαχείρισης να καταστεί θέμα ρουτίνας.

Το MOS Project θα μας πάει πολύ πιο μακριά από τη σημερινή κατάσταση όσον αφορά τη δυνατότητα των συστημάτων διαχείρισης ειδήσεων να ελέγχουν τη σειρά μιας playlist, καθώς θα είναι δυνατό τα συστήματα διαχείρισης ειδήσεων να «γνωρίζουν» το περιεχόμενο του media server.

Για παράδειγμα, μόλις ένας τεχνικός ολοκληρώσει ένα clip σε ένα μη γραμμικό επεξεργαστή αποστέλλεται ένα δικτυακό σήμα στον υπολογιστή του master control. Το σήμα μεταφέρει πληροφορία για το clip που εισάχθηκε από τον τεχνικό, που περιλαμβάνει μεταξύ άλλων το μήκος, το θέμα και το όνομα. Αυτή η πληροφορία που μεταφέρθηκε στο σύστημα διαχείρισης ειδήσεων από το MOS επιτρέπει τώρα στο χρήστη να αναζητήσει το clip και να δει μια λίστα από όχι μόνο wire stories και σενάρια αλλά επίσης clips που περιέχονται στο video server, να τα αναζητήσει δυναμικά με τη χρήση μετα-δεδομένων καθώς και να τα επεξεργαστεί σε πραγματικό χρόνο.

Το ίδιο μπορεί να γίνει και για ακίνητες φωτογραφίες και ήχο. Ο υπολογιστής διαχείρισης ειδήσεων δεν αποθηκεύει μόνο του το video ή το clip ήχου, αλλά ξέρει που να το βρει. Στη συνέχεια, μπορεί αυτόματα να μεταφέρει τις αναφορές ή τους «δείκτες» σε συστήματα desktop editing, ώστε να καταστεί δυνατή η αυτόματη απεικόνιση ή ο αυτόματος χειρισμός.

Κατά τη διάρκεια ζωντανής παραγωγής, το πρωτόκολλο MOS επιτρέπει, σε ένα μη γραμμικό video-server, την αλλαγή μιας playlist, σε πραγματικό χρόνο από τον υπολογιστή διαχείρισης ειδήσεων. Ο παραγωγός μπορεί να επαναπρογραμματίσει μια επαναλήψεις, αποσπάσματα ή trailers όταν το show αναπαράγεται μόνο μετακινώντας σκηνές στον υπολογιστή του.

Η πληροφορία μεταφέρεται στο background μεταξύ του υπολογιστή διαχείρισης ειδήσεων και του Media Object Server, ώστε ο παραγωγός να μπορεί αν γνωρίζει ποια clips είναι έτοιμα και ποια όχι.

Όταν η εκπομπή ολοκληρωθεί, το πρωτόκολλο MOS επιτρέπει στους υπολογιστές διαχείρισης ειδήσεων να εντοπίζουν συνεχόμενα οπτικό και ακουστικό υλικό, καθώς μετακινείται από το on-line σύστημα αποθήκευσης στο near-line και στο μακροπρόθεσμο αρχείο. Αυτή η τελευταία ικανότητα είναι ιδιαίτερα σημαντική αφού ξεπερνά κατά πολύ την ικανότητα των σημερινών, σειρικών συστημάτων επικοινωνίας.

Με το MOS είναι επίσης δυνατός ο συνδυασμός servers ήχου και εικόνας, διαφορετικών κατασκευαστών, ώστε να λειτουργούν ταυτόχρονα για το ίδιο rundown (ζωντανή αναμετάδοση). Αυτό το χαρακτηριστικό του MOS θα παρέχει την ευελιξία στα συστήματα διαχείρισης ειδήσεων να ακολουθούν την στην αιχμή της τεχνολογίας, καθώς θα επιτρέπει στους διαχειριστές των συστημάτων να επιλέγουν την καλύτερης γενιάς τεχνολογία από χρόνο σε χρόνο, χωρίς να περιορίζονται από αποκλειστικές συνδέσεις μεταξύ του εξοπλισμού.

MOS Web Page: <http://www.enps.com/mosproto>

MOS Discussion/Collaboration <http://support.enps.com/~mos>

5.21 ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Οι μη λειτουργικές απαιτήσεις περιγράφουν τις ιδιότητες του συστήματος που συνήθως εκφράζονται βάσει χαρακτηριστικών της μορφής απόδοση, χρηστικότητα, ασφάλειας, νομιμότητας και ιδιωτικότητας. Οι μη λειτουργικές απαιτήσεις που πρέπει να πληρεί το σύστημα είναι οι ακόλουθες:

- Ο χρόνο απόκρισης του συστήματος δεν πρέπει να υπερβαίνει τα 3 δευτερόλεπτα. Το σύστημα πρέπει να λειτουργεί αδιάλειπτα (όλο το χρόνο).
- Τα προσωπικά στοιχεία των πελατών πρέπει να προστατεύονται.
- Το σύστημα πρέπει είναι μπορεί να διεκπεραιώσει τουλάχιστον 100 δοσοληψίες ταυτόχρονα.
- Η εκμάθηση του συστήματος δεν πρέπει να απαιτήσει πάνω από 2 ώρες εκπαίδευσης.
- Το σύστημα δεν πρέπει να επιτρέπει την εισαγωγή στοιχείων που δεν ικανοποιούν τον τύπο των αντίστοιχων πεδίων.
- Κάθε χρήστης πρέπει να έχει όνομα εισόδου και κωδικό πρόσβασης.
- Απαίτηση χρήσης: Η αγορά στο σύστημα θα πρέπει να γίνεται μόνο από μέλη (εξουσιοδοτημένοι χρήστες).
- Απαίτηση επικοινωνίας: Το σύστημα θα περιλαμβάνει σύνδεση Βάσεων Δεδομένων με την εφαρμογή μέσω του πρωτοκόλλου JDBC ώστε να υπάρχει συγχρονισμός της λειτουργίας τους.
- Απαιτήσεις υλοποίησης: Θα χρησιμοποιηθεί η γλώσσα JAVA λόγω της δυνατότητας ανάπτυξης δια-πλατφορμικών εφαρμογών με αντικειμενοστραφή προγραμματισμό σε συνδυασμό με χρήση SQL για υποβολή ερωτημάτων προς τη Βάση Δεδομένων.

6 ΡΟΗ ΕΦΑΡΜΟΓΗΣ

Η ορθή λειτουργία του τηλεοπτικού προγράμματος βασίζεται κατά κανόνα στην αρμονική και με σειρά εκτέλεση εργασιών από τους τεχνικούς. Για το λόγο αυτό στην εφαρμογή MediaStack η ορθή εκτέλεση των αυτοματισμών λειτουργίας έχει άμεση σχέση με τους εμπλεκόμενους χρήστες, είτε αυτοί είναι φυσικά πρόσωπα είτε τεχνικός εξοπλισμός.

6.1 ADMINISTRATOR

Πρόκειται για τον διαχειριστή του συστήματος. Ο συγκεκριμένος ρόλος έχει πλήρη δικαιώματα τόσο στις λειτουργίες του λογισμικού όσο και στη διαχείριση των web services που επιτρέπουν την επικοινωνία του λογισμικού με εξωτερικές εφαρμογές και τεχνικό εξοπλισμό. Επίσης ο χρήστης αυτός είναι υπεύθυνος και για τους αυτοματισμούς της εφαρμογής που εκτελούνται μέσω agents.

```
C:\Users\Adrian>ftp -d ftp.stratus.com
Połączony z icebox.stratus.com.
220 Stratus-FTP-server
Użytkownik (icebox.stratus.com:(none)): anonymous
--> USER anonymous
331 Anonymous login ok, send your complete email address as your password.
Hasło:
--> PASS adrian.stolarski@wp.pl
230 Welcome to Stratus Technologies Public FTP Server
This server is for Stratus Technologies customers and partners.
Abusers of this server will be prosecuted to the full extent
of the law.

All transactions on this server are logged.
230 Anonymous access granted, restrictions apply.
ftp> quit
--> QUIT
221 Goodbye.

C:\Users\Adrian>ftp -d ftp.stratus.com
Połączony z icebox.stratus.com.
220 Stratus-FTP-server
Użytkownik (icebox.stratus.com:(none)): anonymous
--> USER anonymous
331 Anonymous login ok, send your complete email address as your password.
Hasło:
--> PASS a@wp.pl
230 Welcome to Stratus Technologies Public FTP Server
This server is for Stratus Technologies customers and partners.
Abusers of this server will be prosecuted to the full extent
of the law.

All transactions on this server are logged.
230 Anonymous access granted, restrictions apply.
ftp> get README.TXT
--> PORT 192,168,0,102,196,80
200 PORT command successful
--> RETR README.TXT
150 Opening ASCII mode data connection for README.TXT (1925 bytes)
226 Transfer complete.
ftp: 1979 bajtów odebranych w 0,00Sekund 1979000,00Kbajtów/sek.
ftp> quit
--> QUIT
221 Goodbye.

C:\Users\Adrian>
```

Εικόνα 37: FTP Agent

```

GRATHA29
10/10/2016 1:15:04 pm
jobid=(19c3404f-9f5b-4e6e-87db-9445a384a688)
actionid=147
profile=vmedia2: Get Youtube status
action=youtube get
file=\\grathf2\vmmediatmp\youtube\replies\7190b287-0d7b-47f5-9e53-2377e5d8f0ad
command=agntwork\youtube_get.bat "\\grathf2\vmmediatmp\youtube\replies\7190b287-0d7b-47f5-9e53-2377e5d8f0ad" *7190b287-0d7b-47f5-9e53-2377e5d8f0ad"
files=.\prog\youtube_get.bat;.\prog\youtube_get.ftp
timeout=36000
console=False
work=\\grathf2\archive\jobs\work\{19c3404f-9f5b-4e6e-87db-9445a384a688}

ERROR executing vmedia2: Get Youtube status > youtube get: The parameter is incorrect (87)

retrying...

GRATHA29
10/10/2016 1:15:17 pm
jobid=(19c3404f-9f5b-4e6e-87db-9445a384a688)
actionid=147
profile=vmedia2: Get Youtube status
action=youtube get
file=\\grathf2\vmmediatmp\youtube\replies\7190b287-0d7b-47f5-9e53-2377e5d8f0ad
command=agntwork\youtube_get.bat "\\grathf2\vmmediatmp\youtube\replies\7190b287-0d7b-47f5-9e53-2377e5d8f0ad" *7190b287-0d7b-47f5-9e53-2377e5d8f0ad"
files=.\prog\youtube_get.bat;.\prog\youtube_get.ftp
timeout=36000
console=False
work=\\grathf2\archive\jobs\work\{19c3404f-9f5b-4e6e-87db-9445a384a688}

ERROR executing vmedia2: Get Youtube status > youtube get: The parameter is incorrect (87)

retrying...

GRATHA29
10/10/2016 1:15:21 pm
jobid=(19c3404f-9f5b-4e6e-87db-9445a384a688)
actionid=147
profile=vmedia2: Get Youtube status
action=youtube get
file=\\grathf2\vmmediatmp\youtube\replies\7190b287-0d7b-47f5-9e53-2377e5d8f0ad
command=agntwork\youtube_get.bat "\\grathf2\vmmediatmp\youtube\replies\7190b287-0d7b-47f5-9e53-2377e5d8f0ad" *7190b287-0d7b-47f5-9e53-2377e5d8f0ad"
files=.\prog\youtube_get.bat;.\prog\youtube_get.ftp
timeout=36000
console=False
work=\\grathf2\archive\jobs\work\{19c3404f-9f5b-4e6e-87db-9445a384a688}

ERROR executing vmedia2: Get Youtube status > youtube get: The parameter is incorrect (87)

retrying...

GRATHA29
10/10/2016 1:15:24 pm

```

Εικόνα 38: Transcoding Agent

```

→ ~ ffmpeg -f video4linux2 -s 320x240 -i /dev/video0 -an -r 10 -c:v libx264 -q 1 -f rtsp -metadata title=test rtsp://54.187.122.130:5554/flvplayback
ffmpeg version git-2016-06-09-e826cee Copyright (c) 2000-2016 the FFmpeg developers
built with gcc 5.2.1 (Ubuntu 5.2.1-22ubuntu2) 20151010
configuration: --enable-libx264 --enable-gpl
libavutil 55. 24.100 / 55. 24.100
libavcodec 57. 45.100 / 57. 45.100
libavformat 57. 37.101 / 57. 37.101
libavdevice 57. 0.101 / 57. 0.101
libavfilter 6. 46.101 / 6. 46.101
libswscale 4. 1.100 / 4. 1.100
libswresample 2. 0.101 / 2. 0.101
libpostproc 54. 0.100 / 54. 0.100
Input #0, video4linux2,v4l2, from '/dev/video0':
Duration: N/A, start: 14202.734024, bitrate: 36864 kb/s
Stream #0:0: Video: rawvideo (YUV2 / 0x32595559), yuyv422, 320x240, 36864 kb/s, 30 fps, 30 tbr, 1000k tbn, 1000k tbc
No pixel format specified, yuv422p for H.264 encoding chosen.
Use -pix_fmt yuv420p for compatibility with outdated media players.
[libx264 @ 0x2e3c940] -qscale is ignored, -crf is recommended.
[libx264 @ 0x2e3c940] using cpu capabilities: MMX2 SSE2Fast SSE3 SSE4.2 AVX FMA3 AVX2 LZCNT BMI2
[libx264 @ 0x2e3c940] profile High 4:2:2, level 1.2, 4:2:2 8-bit
[libx264 @ 0x2e3c940] 264 - core 146 r2555 0c21480 - H.264/MPEG-4 AVC codec - Copyleft 2003-2015 - http://www.videolan.org/x264.html - options: c
abac=1 ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1 psy_rd=1.00:0.00 mixed_ref=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0 d
eadzones=21,11 fast_pskip=1 chroma_qp_offset=-2 threads=6 lookahead_threads=1 sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_compat=0 constr
ained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b_bias=0 direct=1 weightb=1 open_gop=0 weightp=2 keyint=250 keyint_min=10 scenecut=40 intra_refresh
=0 rc_lookahead=40 rc=crf mbtree=1 crf=23.0 qcomp=0.60 qpmin=0 qpmax=69 qpstep=4 ip_ratio=1.40 aq=1:1.00
[rtp @ 0x2e39a00] Using AVStream.codec to pass codec parameters to muxers is deprecated, use AVStream.codecpar instead.
Output #0, rtsp, to 'rtsp://54.187.122.130:5554/flvplayback':
Metadata:
  title           : test
  encoder         : Lavf57.37.101
  Stream #0:0: Video: h264 (libx264), yuv422p, 320x240, q=-1--1, 10 fps, 90k tbn, 10 tbc
Metadata:
  encoder         : Lavc57.45.100 libx264
Side data:
  cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
Stream mapping:
  Stream #0:0 -> #0:0 (rawvideo (native) -> h264 (libx264))
Press [q] to stop, [?] for help
frame= 52 fps= 15 q=25.0 size=N/A time=-00:00:00.09 bitrate=N/A dup=20 drop=44 speed=N/A

```

Εικόνα 39: FFMpeg Agent

Jobs

Status Waiting Paused Ready / Executing Cancelled Complete

Date

File

Profile **Select options**

Submit Query

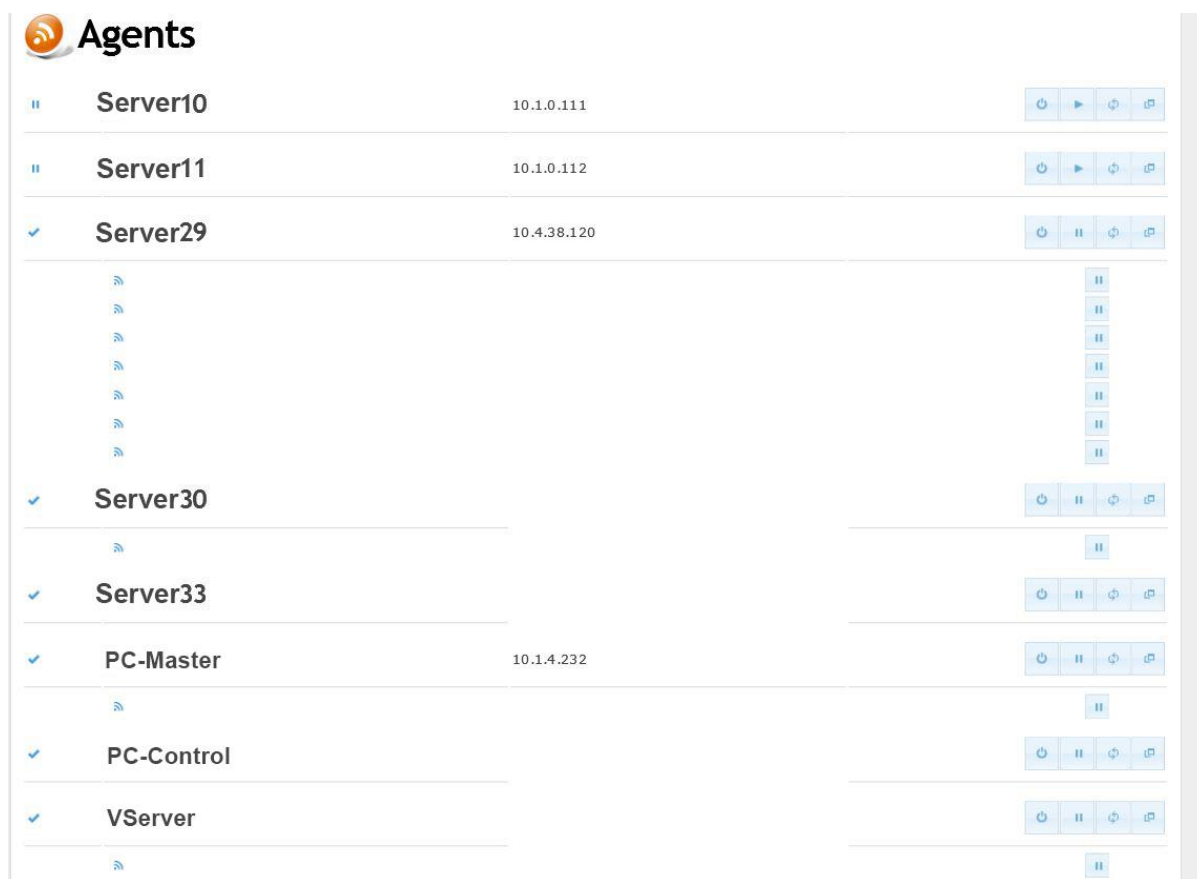
	10/10/2016 1:15:00 μμ Delete Reset Pause Resume Complete
	Pause Resume Complete Console
	10/10/2016 1:15:04 μμ Delete Reset Pause Resume Complete
	Pause Resume Complete Console
	10/10/2016 1:15:07 μμ Delete Reset Pause Resume Complete
	Pause Resume Complete Console
	10/10/2016 1:15:09 μμ Delete Reset Pause Resume Complete
	Pause Resume Complete Console
	10/10/2016 1:15:11 μμ Delete Reset Pause Resume Complete
	Pause Resume Complete Console

Εικόνα 40: Λογισμικό διαχείρισης agents

Live

S1: Main PC+Mobile Streamer	S2: Backup PC+Mobile Streamer	S3: BACKSTAGE Streamer
Start Card Auto Reset	Start Card Auto Reset	Start Card Auto Reset

Εικόνα 41: Λογισμικό διαχείρισης Broadcasting

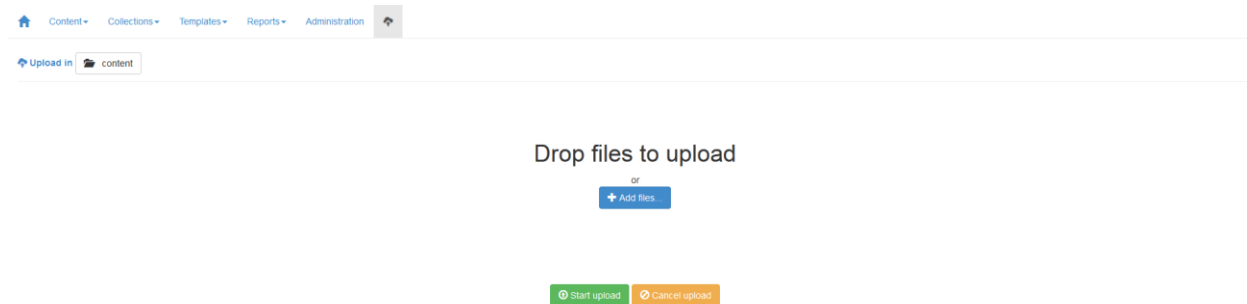


Εικόνα 42: Διαχείριση Agents ανά Server

6.2 ΔΙΑΧΕΙΡΙΣΤΗΣ

Ως διαχειριστής συστήματος ορίζεται κάποιο μέλος της παραγωγής ενός τηλεοπτικού προγράμματος, ο οποίος είναι υπεύθυνος για την ορθή αναμετάδοση στα μέσα προβολής καθώς και τη συμφωνία της εκπομπής με το σήμα αναμετάδοσης, τις ζώνες ώρες κ.α. Ο χρήστης αυτός έχει τα παρακάτω δικαιώματα:

1. Προσθήκη περιεχομένου

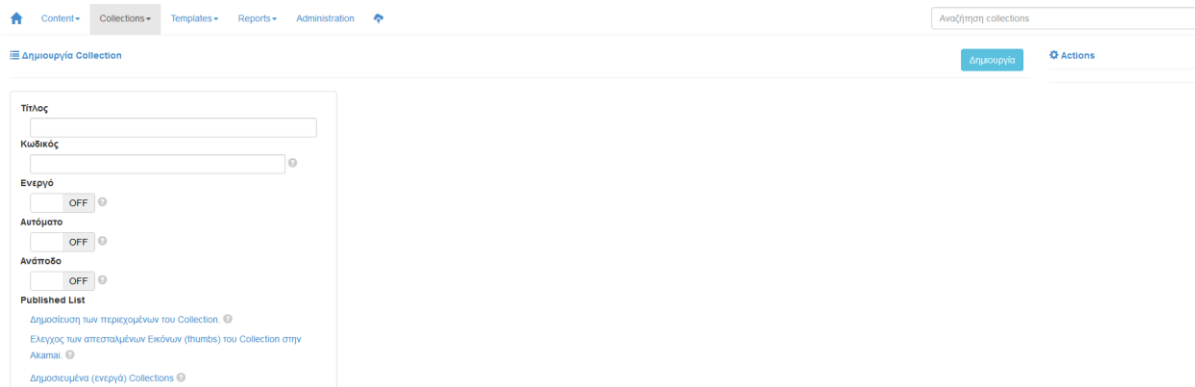


Εικόνα 43: Προσθήκη βίντεο από αρχείο ή ροή

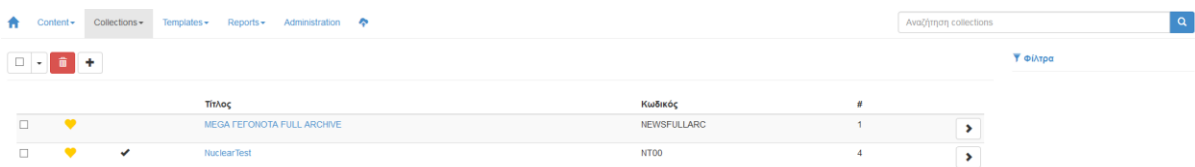
2. Διαχείριση ροής εκπομπών και collections προγράμματος

Space	Τίτλος	Εκπομπή	Επεισόδιο	Κατηγορία
Video Templates	28/08/2014 16:58			full
Video Templates	JOY SHOWBIZ 03/10/2014 12:29			part
Video Templates	JOY FASHION 03/10/2014 12:20			part
Video Templates	JOY HEALTHY LIVING 03/10/2014 12:23			part
Video Templates	JOY HOME 03/10/2014 12:23			part
Video Templates	JOY MAGERIGH 03/10/2014 12:25			part
Video Templates	JOY KIDS 03/10/2014 12:24			part
Video Templates	JOY PEOPLE 03/10/2014 12:28			part
Video Templates	JOY PETS 03/10/2014 12:28			part
Video Templates	JOY BEAUTY 03/10/2014 12:18			part
Video Templates	JOY CULTURE 03/10/2014 12:19			part
Video Templates	Μεσημέρινο Δελτίο 28/08/2014 16:57	MDELTI0		full
Video Templates	Κληρώ Δελτίο 28/08/2014 16:55	KDELTI0		full
Video Templates	MEGA ΓΕΓΟΝΟΤΑ ΑΘΛΗΤΙΚΑ 28/08/2014 17:03			part
Video Templates	MEGA ΓΕΓΟΝΟΤΑ ΔΙΑΦΟΡΑ 28/08/2014 17:05			part
Video Templates	MEGA ΓΕΓΟΝΟΤΑ ΕΛΛΑΔΑ 28/08/2014 17:06			part
Video Templates	MEGA ΓΕΓΟΝΟΤΑ ΚΑΙΡΟΣ 28/08/2014 17:07			part
Video Templates	MEGA ΓΕΓΟΝΟΤΑ ΚΟΣΜΟΣ 28/08/2014 17:10			part
Video Templates	MEGA ΓΕΓΟΝΟΤΑ ΜΟΝΟ ΣΤΟ ΜΕΓΑ 28/08/2014 17:10			part
Video Templates	MEGA ΓΕΓΟΝΟΤΑ ΟΙΚΟΝΟΜΙΑ 28/08/2014 17:11			part
Video Templates	MEGA ΓΕΓΟΝΟΤΑ ΠΟΛΙΤΙΚΗ 28/08/2014 17:12			part
Video Templates	MEGA ΓΕΓΟΝΟΤΑ ΠΟΛΙΤΙΣΜΟΣ 28/08/2014 17:12			part
Video Templates	MEGA ΓΕΓΟΝΟΤΑ ΠΡΟΣΦΑΤΑ 28/08/2014 17:19			part
Video Templates	MEGA ΓΕΓΟΝΟΤΑ ΥΓΕΙΑ 28/08/2014 17:09			part
Video Templates	ΚΛΕΜΜΕΝΑ ΟΝΕΙΡΑ 26/11/2014 12:15			part

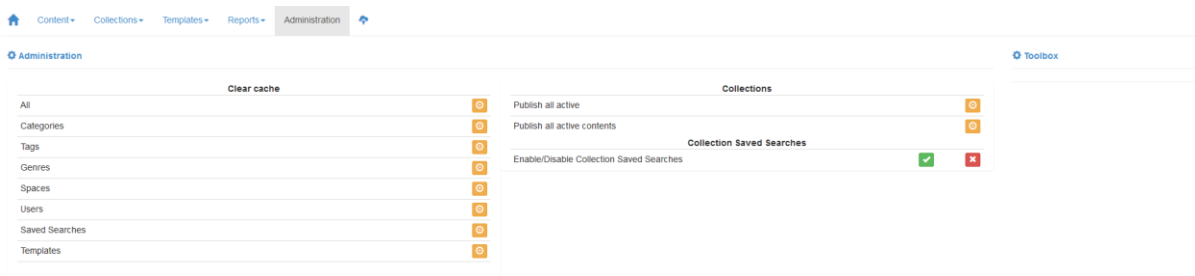
Εικόνα 44: Διαχείριση Collections



Εικόνα 45: Οθόνη εισαγωγής meta-data

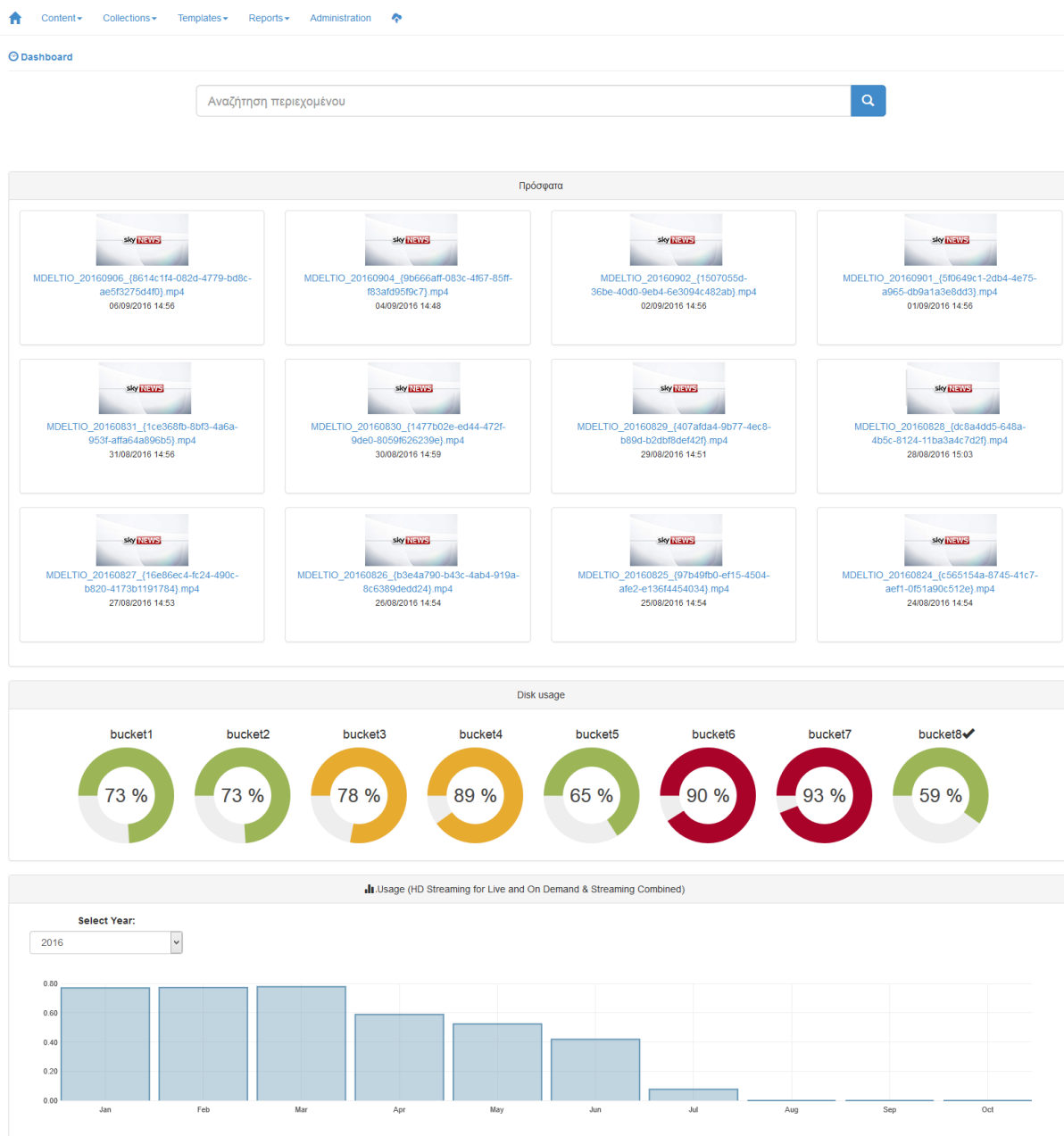


Εικόνα 46: Διαχείριση Playlist



Εικόνα 47: Playlist Streaming Management

3. Προβολή στατιστικών χρήσης και προβολής

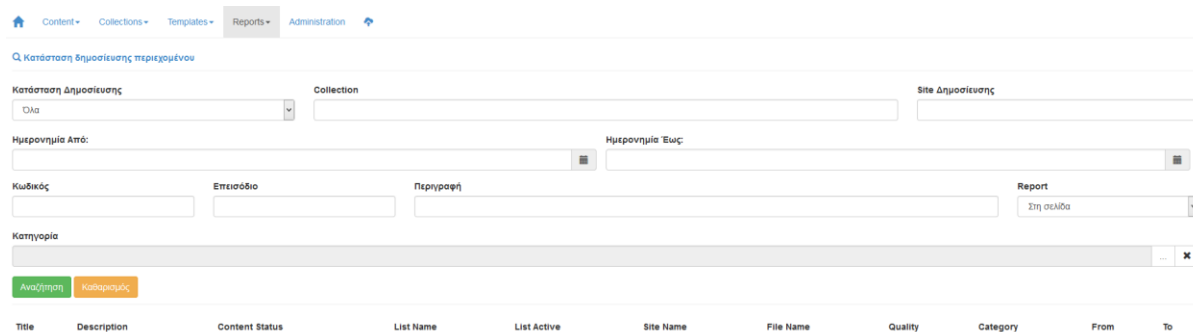


Εικόνα 48: Ενδεικτική οθόνη στατιστικών χρήσης

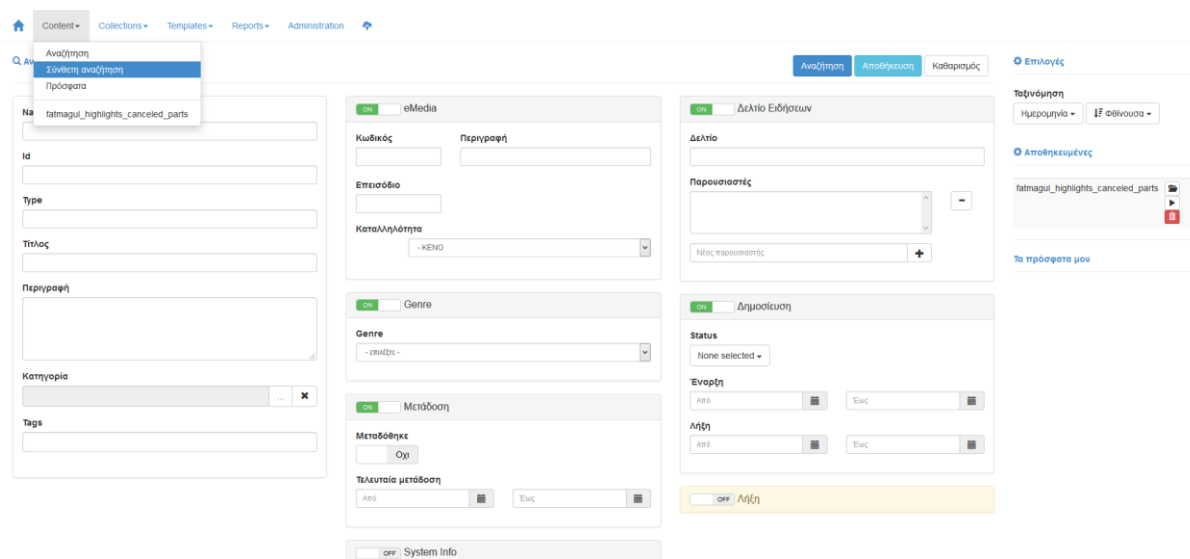
6.3 ΥΠΕΥΘΥΝΟΣ ΡΟΗΣ

Ο υπεύθυνος ροής, αν και έχει λιγότερα δικαιώματα από το διαχειριστή, είναι ο σημαντικότερος χρήστης του λογισμικού. Είναι υπεύθυνος τόσο για το video – on – demand όσο και για τη ζωντανή αναμετάδοση προγράμματος.

Ο ρόλος του περιλαμβάνει τις περισσότερες οθόνες εισαγωγής περιεχομένου, όπως περιγραφές, meta – δεδομένα και διασύνδεση με το εμπορολογιστικό πρόγραμμα του σταθμού και την λήψη των δικαιωμάτων και των ζωνών προβολής.



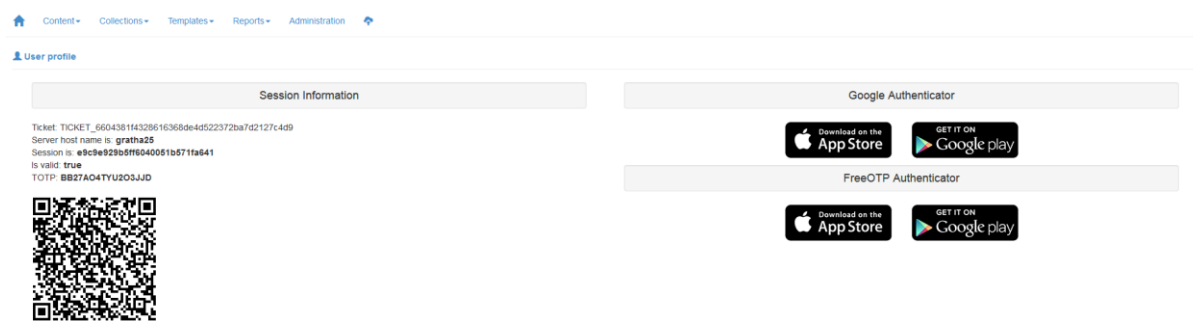
Εικόνα 49: Διαχείριση Collections



Εικόνα 50: Διαχείριση περιεχομένου

6.4 ΧΡΗΣΤΗΣ

Το ρόλο του χρήστη λαμβάνουν όσοι, με απόφαση του υπευθύνου ροής και του διαχειριστή, εισάγουν βίντεο στο σύστημα. Η οθόνη του προφίλ του χρήστη περιλαμβάνει και τη διασύνδεση με εφαρμογές κινητών τηλεφώνων για την άμεση εισαγωγή περιεχομένου απομακρυσμένα.



Εικόνα 51: User Profile

6.5 ΤΕΧΝΙΚΟΣ ΕΞΟΠΛΙΣΜΟΣ

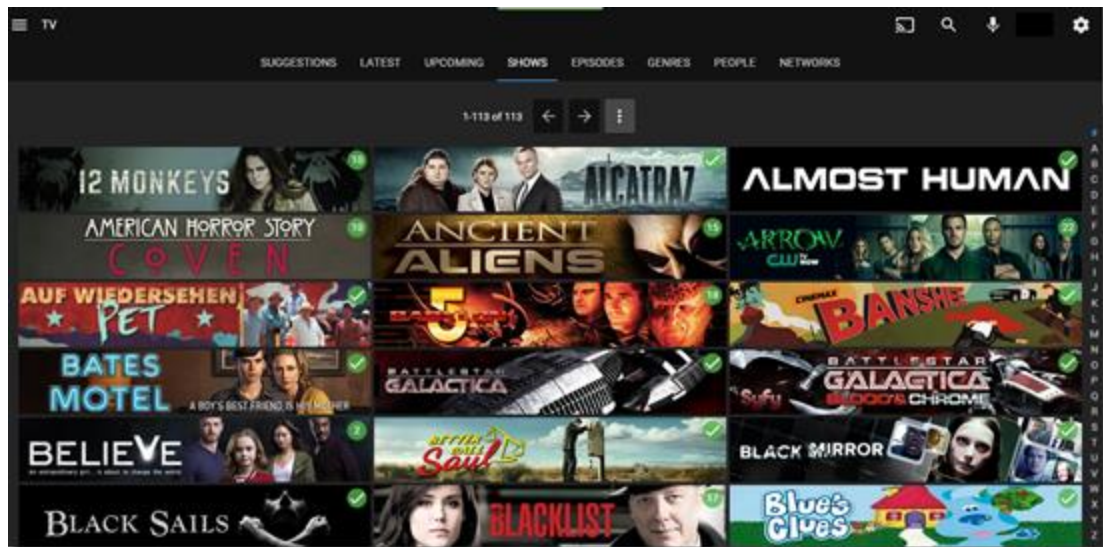
Στο ρόλο αυτό ανατίθενται εικονικοί χρήστες, ο καθένας από τους οποίους αφορά κάποιο μηχάνημα ή λογισμικό του σταθμού. Ενδεικτικός τέτοιος εξοπλισμός είναι οι Video Servers, ο Prompter και ο Telecaster. Οι εικονικοί αυτοί χρήστες διασυνδέουν και επιτρέπουν την επικοινωνία και αναμετάδοση περιεχομένου μέσω του πρωτοκόλλου MOS.



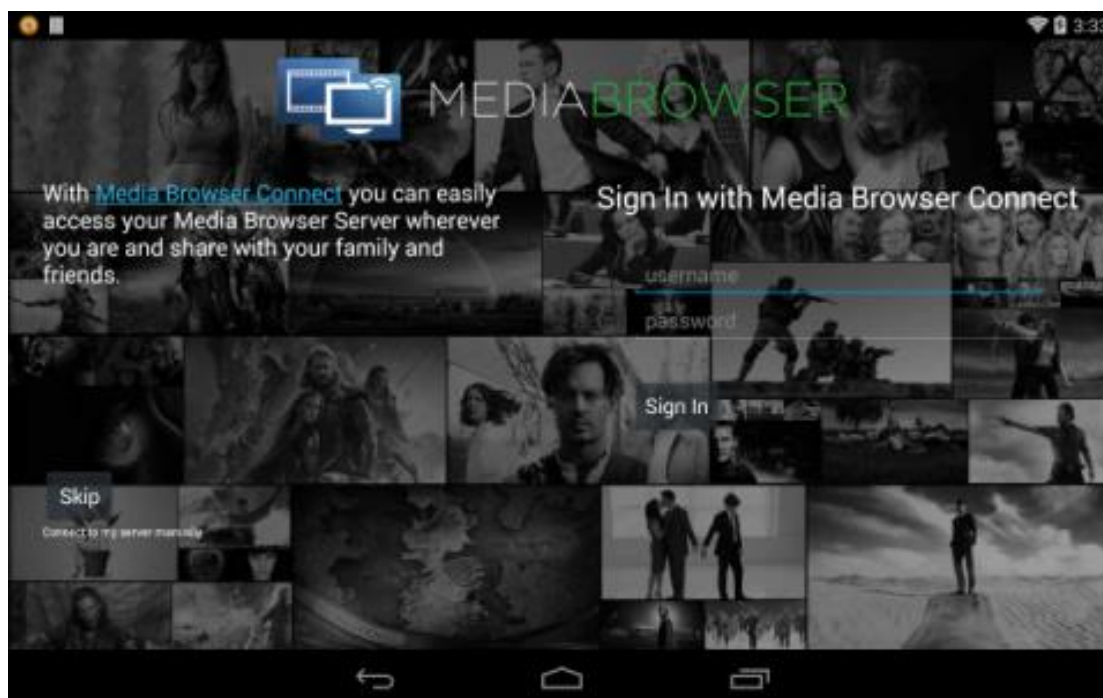
Εικόνα 52: Prompter ειδήσεων

6.6 ΘΕΑΤΗΣ

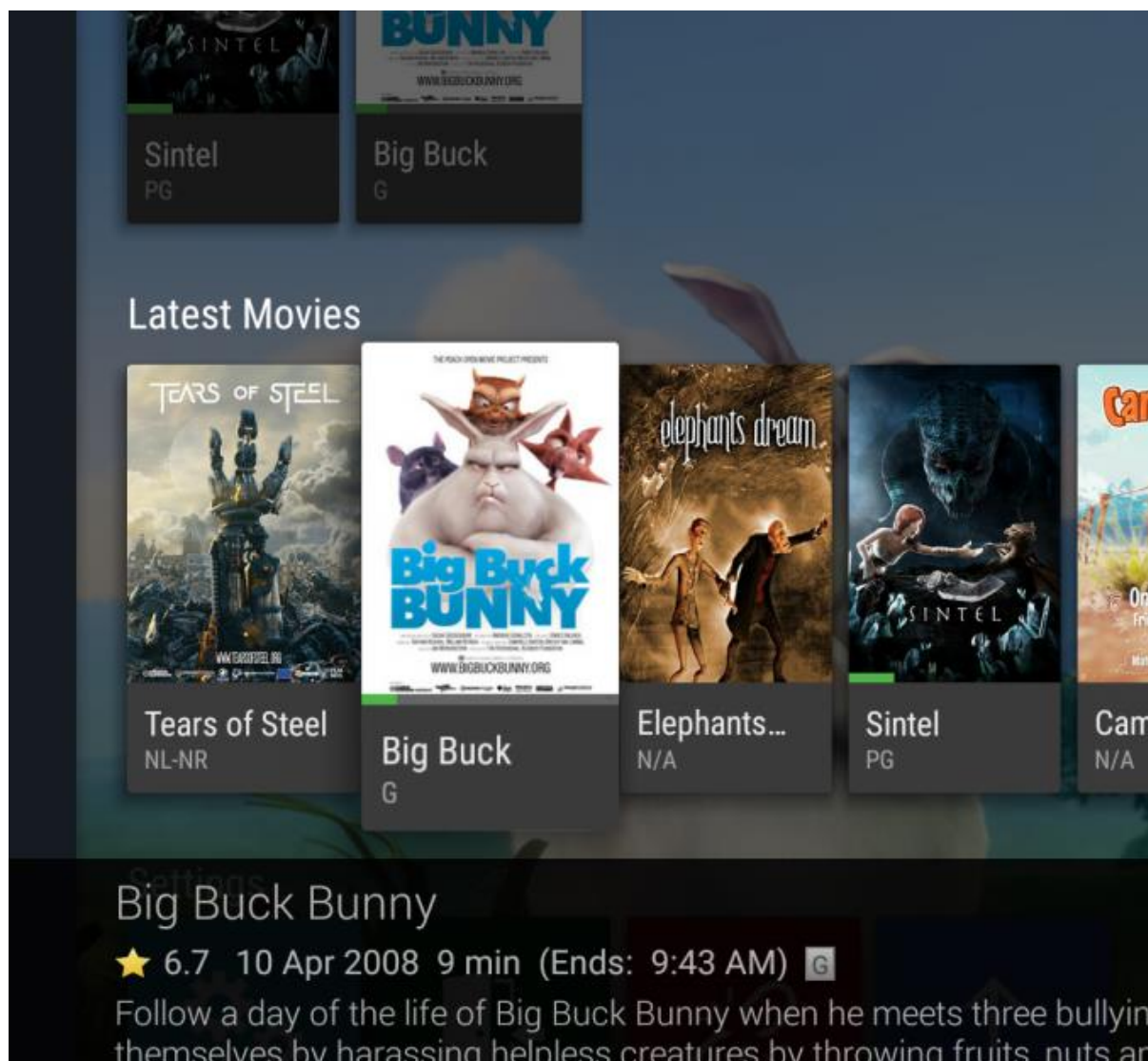
Ο ρόλος του θεατή είναι ο πιο περιορισμένος από όλους και δεν αφορά ευθέως το λογισμικό. Δίνει πρόσβαση στο live streaming καθώς και στο video on demand, αφού η ιστοσελίδα του σταθμού αποτελεί το frontend του συστήματος διαχείρισης περιεχομένου.



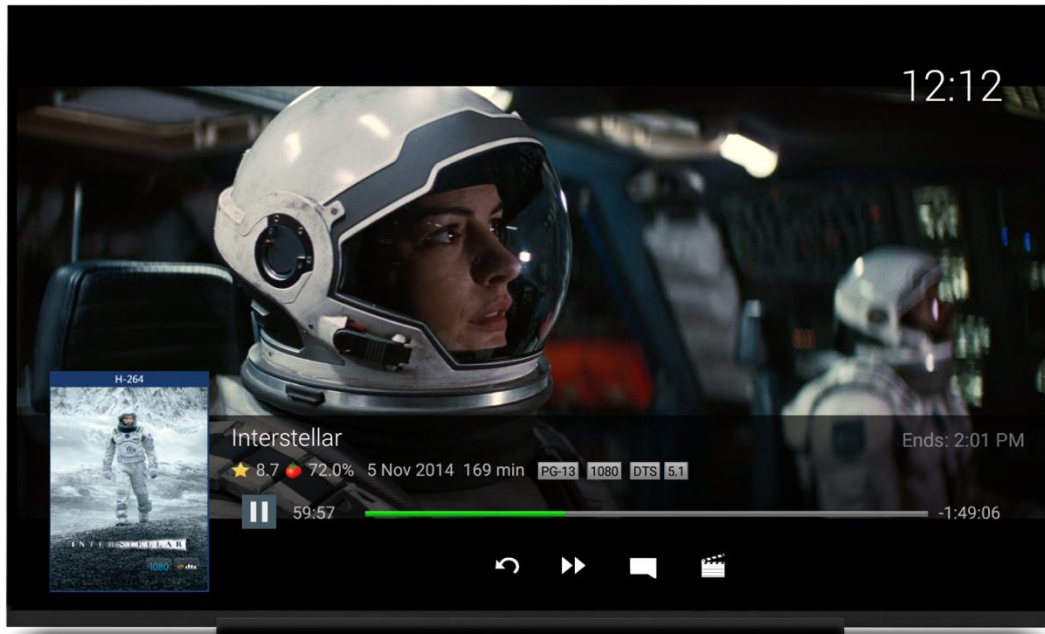
Εικόνα 53: Video on Demand



Εικόνα 54: Video on Demand Mobile Application



Εικόνα 55: Video on Demand Collections



Εικόνα 56: Live Streaming

7 ΣΥΜΠΕΡΑΣΜΑΤΑ

Αν θέλαμε να βγάλουμε κάποια συμπεράσματα για την εφαρμογή Media Stack, αυτά θα περιορίζονταν αναγκαστικά στο πεδίο της ανάπτυξης και λιγότερο στη business λογική της εφαρμογής. Κάτι διαφορετικό από αυτό, θα απαιτούσε πλήθος ανάλυσης αλλά και παραμετροποίησης στα πλαίσια του κάθε τηλεοπτικού σταθμού.

Αν όμως κανείς αποστασιοποιηθεί από την τεχνική όψη της πληροφορικής και την αντιμετωπίσει ως την επιστήμη των αλγορίθμων και της επίλυσης προβλημάτων, τότε μπορεί να αντιληφθεί και το σκοπό του Media Stack CMS.

Το λογισμικό αυτό στοχεύει να αξιοποιήσει υφιστάμενες δημοφιλείς τεχνολογίες όπως η Java και να τις συνδυάσει με τέτοιο τρόπο που να αποτελέσει μια καινοτόμα πρόταση αυτοματισμού του τηλεοπτικού προγράμματος και να καλύψει όλα τα μέσα προβολής της σύγχρονης εποχής.

Ως προς το στόχο αυτό, λοιπόν, τα συμπεράσματα μπορεί να είναι πολύ θετικά, αφού:

1. Επιτρέπει την ταυτόχρονη επεξεργασία εκατοντάδων βίντεο μέσα σε ελάχιστο χρόνο.
2. Μπορεί να αξιοποιεί οποιαδήποτε συσκευή είναι διαθέσιμη
3. Επιτρέπουν την άμεση δημοσίευση περιεχομένου για δημοσιογράφους που βρίσκονται σε ρεπορτάζ εκτός καναλιού
4. Παρέχει άμεση διαθεσιμότητά του πολυμεσικού στο τηλεοπτικό κοινό
5. Μπορεί να χρησιμοποιηθεί ως λογισμικό αξιοποίησης της ταινιοθήκης του σταθμού.

Τέλος, αποτελεί μια ιδιαίτερα σταθερή λύση που δεν απαιτεί την αντικατάσταση κανενός τύπου εξοπλισμού από τον τηλεοπτικό σταθμό.

8 ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Βαλούκος, Στάθης (2008). Ιστορία της ελληνικής τηλεόρασης. Αθήνα: Αιγόκερως. ISBN 978-960-322-347-4

[2] Βαλούκος, Στάθης (2008). Ιστορία της ελληνικής τηλεόρασης. Αθήνα: Αιγόκερως. ISBN 978-960-322-347-4.

Βώβου, Ιωάννα (2010). «Στοιχεία για μια μετα-ιστορία της ελληνικής τηλεόρασης». Στο:

Βώβου, Ιωάννα. Ο κόσμος της τηλεόρασης. Αθήνα: Ηρόδοτος, σελ. 93-140. ISBN 978-960-8256-99-6.

Βώβου, Ιωάννα (2013). «Κρίση και στα ΜΜΕ: Η τηλεοπτική «κουλτούρα της κρίσης»». Στο:

Πλειός, Γιώργος. Η κρίση και τα ΜΜΕ. Αθήνα: Παπαζήσης, σελ. 313-347. ISBN 978-960-02-2783-3.

Δάμπασης, Γιώργος (2002). Την εποχή της τηλεόρασης. Αθήνα: Καστανιώτης. ISBN 960-03-3273-8.

Κάβουρα, Ανδρονίκη (2008). «Ζητήματα εγκυρότητας και αξιοπιστίας στη μεθοδολογία της μέτρησης της τηλεθέασης». Το Βήμα των Κοινωνικών Επιστημών 13 (Καλοκαίρι): 57-80.

Κάρτερ, Γιώργος (2004). Ελληνική ραδιοφωνία τηλεόραση. Αθήνα: Καστανιώτης. ISBN 960-03-3664-4.

Katsoudas, Dimitrios (1985). «Greece: A politically controlled state monopoly broadcasting system». West European Politics 8 (2): 137-151. doi:10.1080/01402388508424530.

Κομνηνού, Μαρία (2001). Από την αγορά στο θέαμα. Αθήνα: Παπαζήσης. ISBN 960-02-1200-7.

Κουκουτσάκη-Monnier, Αγγελική (2010). «Ελληνικά προγράμματα μυθοπλασίας: διαχρονική εξέλιξη και τάσεις παραγωγής». Στο: Βώβου, Ιωάννα. Ο κόσμος της τηλεόρασης. Αθήνα: Ηρόδοτος, σελ. 417-451. ISBN 978-960-8256-99-6.

Λεανδρος, Νίκος (2013). «Τα μέσα στο επίκεντρο της κρίσης: Τα οικονομικά αποτελέσματα οκτώ μεγάλων μέσων». Στο: Πλειός, Γιώργος. Η κρίση και τα ΜΜΕ. Αθήνα: Παπαζήσης, σελ. 31-57. ISBN 978-960-02-2783-3.

Μοσχονάς, Ίων (1996-03-10). «Η προϊστορία της Ελληνικής Τηλεόρασης». Καθημερινή 7 Ημέρες: σελ. 3-5. Αρχειοθετήθηκε από το πρωτότυπο στις 2013-11-03. Ανακτήθηκε στις 2 Μαρτίου 2014.

Μοσχονάς, Ίων (1996-03-10). «Το κανάλι των Ενόπλων Δυνάμεων». Καθημερινή 7 Ημέρες: σελ. 13-15. Αρχειοθετήθηκε από το πρωτότυπο στις 2013-11-03. Ανακτήθηκε στις 2 Μαρτίου 2014.

Παναγιωτοπούλου, Ρόη (2004). Η τηλεόραση εκτός των τειχών. Αθήνα: Καστανιώτης. ISBN 960-03-3726-8.

[3] Παναγιωτοπούλου, Ρόη (2010). «20 χρόνια ελληνικής ιδιωτικής τηλεόρασης (1989-2009): Ένας απολογισμός». Ζητήματα Επικοινωνίας (10): 9-27.

Papathanassopoulos, Stylianos (1990). «Broadcasting, Politics and the State of Socialist Greece». *Media, Culture & Society* 12 (3): 387-397. doi:10.1177/016344390012003007.

Παπαθανασόπουλος, Στέλιος (1993). Απελευθερώνοντας την τηλεόραση (2η έκδοση). Αθήνα: Καστανιώτης. ISBN 960-03-0982-5.

Παπαθανασόπουλος, Στέλιος (1994). Η τηλεόραση στον κόσμο. Αθήνα: Εκδόσεις Παπαζήση. ISBN 960-02-1043-8.

Papathanassopoulos, Stylianos (2007). «The Development of Digital Television in Greece».

Javnost-The Public 14 (1): 93-108. doi:10.1080/13183222.2007.11008938.

Παπαθανασόπουλος, Στέλιος (2008). «Η χρήση και η στάση των Ελλήνων προς τα ΜΜΕ». Στο: Παπαθανασόπουλος, Στέλιος. Ανάλεκτα 2008. Αθήνα: ΙΟΜ-Εκδόσεις Παπαζήση, σελ. 22-44. ISBN 978-960-02-2314-9.

Πασχαλίδης, Γρηγόρης (2005). «Η ελληνική τηλεόραση». Στο: Βερνίκος, Νικόλας.

Δασκαλοπούλου, Σοφία; Μπαντιμαρούδης, Φιλήμων και άλλοι. Πολιτιστικές βιομηχανίες: διαδικασίες, υπηρεσίες και αγαθά. Αθήνα: Κριτική, σελ. 173-200. ISBN 960-218-398-5.

Patrona, Marianna (2009). «'A Mess' and 'Rows': Evaluation in Prime-time TV News Discourse and the Shaping of Public Opinion». *Discourse & Communication* 3 (2): 173-194. doi:10.1177/1750481309102451.

Σμυρναίος, Νίκος (2010a). «Οικονομική συγκέντρωση και θεσμικό πλαίσιο της ιδιωτικής τηλεόρασης». Ο κόσμος της τηλεόρασης. Αθήνα: Ηρόδοτος, σελ. 141-174. ISBN 978-960-8256-99-6.

Σμυρναίος, Νίκος (2010b). «Τα ρευστά όρια των μέσων: Προσφορά και ζήτηση ελληνικών τηλεοπτικών προγραμμάτων στα δίκτυα peer-to-peer». Στο: Βώβου, Ιωάννα. Ο κόσμος της τηλεόρασης. Αθήνα: Ηρόδοτος, σελ. 581-598. ISBN 978-960-8256-99-6.

[2] Vovou, Ioanna (2006). «Histoire politique de la télévision en Grèce». *Le Temps des médias (Printemps)*: 259-269. doi:10.3917/tdm.006.0259.

Yörük, Zafer; Vatikiotis, Pantelis (2013). «Soft Power or Illusion of Hegemony: The Case of the Turkish Soap Opera "Colonialism"». *International Journal of Communication* 7: 2361-2385.

Getting Ready for Responsive Web Design, Διαθέσιμο στο διαδικτυακό τόπο <http://www.tcs.com/SiteCollectionDocuments/White%20Papers/IT-Services-Whitepaper-Responsive-Web-Design-0113-1.pdf>.

Responsive web design – Are we ready for the new age?, Διαθέσιμο στο διαδικτυακό τόπο http://www.iiakm.org/ojakm/articles/2014/volume2_1/OJAKM_Volume2_1pp93-103.pdf.

Βικιπέδια, ελεύθερη εγκυκλοπαίδεια. HTML. Διαθέσιμο στο διαδικτυακό τόπο <https://el.wikipedia.org/wiki/HTML>.

Bootstrap. Διαθέσιμο στο διαδικτυακό τόπο <http://getbootstrap.com/>.

UI Bootstrap, Bootstrap components written in pure AngularJS by the AngularUI. Διαθέσιμο στο διαδικτυακό τόπο <https://angular-ui.github.io/bootstrap/#>.

Spring - MVC Framework Tutorial. Διαθέσιμο στο διαδικτυακό τόπο http://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm.

Web MVC framework. Διαθέσιμο στο διαδικτυακό τόπο <http://docs.spring.io/spring-framework/docs/current/spring-framework-reference/html/mvc.html>.

AngularJS by Google. Διαθέσιμο στο διαδικτυακό τόπο <https://angularjs.org>.

Apache Maven Project. Διαθέσιμο στο διαδικτυακό τόπο <https://maven.apache.org>.

Craig Walls (2011), Spring in action third edition, Manning Shelter Island

Gary Mark, Josh Long and Daniel Rubio (2010), Spring Recipes Second Edition, Apress

Spring Tool Suite, Διαθέσιμο στον διαδικτυακό τόπο <https://spring.io/tools>.