

Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής
ΠΜΣ Προηγμένα Συστήματα Πληροφορικής

Μεταπτυχιακή Διατριβή

Το Κατανεμημένο Σύστημα Αρχείων EuterpeFS

από

Άγγελος - Χρήστος Αναδιώτης

Επιβλέπων: Καθηγητής Χρήστος Δουληγέρης

Πειραιάς, Ιανουάριος 2011

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΛΗ

Τριμελής Εξεταστική Επιτροπή

Χρήστος Δουλιγέρης
Καθηγητής

Δημήτριος Βέργαδος
Λέκτορας

Παναγιώτης Κοτζανικολάου
Λέκτορας

ΓΑΛΕΡΙΟ ΤΕΡΡΑ

Στους δικούς μου ανθρώπους

Περίληψη

Η παρούσα ματαπτυχιακή διατριβή επικεντρώνεται στο σχεδιασμό και την υλοποίηση ενός καταναμημένου συστήματος αρχείων, του EuterpeFS. Το EuterpeFS έχει δύο βασικούς στόχους: Την αποδοτική οργάνωση των αρχείων και την ανεξαρτησία ως προς την τοποθεσία τους για την πρόσβαση σε αυτά.

Τα καταναμημένα συστήματα αρχείων χρησιμοποιούνται επιτυχώς εδώ και πολλά χρόνια και θεωρούνται αναπόσπαστο κομμάτι ενός καταναμημένου συστήματος. Οι χρήστες τους επιθυμούν να έχουν πρόσβαση στα δεδομένα τους οπουδήποτε κι αν βρίσκονται εύκολα, γρήγορα και με ασφάλεια.

Η συνεχής αύξηση του όγκου της πληροφορίας που διακινείται στο διαδίκτυο υποδεικνύει την απαίτηση για καλύτερη οργάνωσή της, ώστε να μπορεί να ακαλυφθεί και να προσπελαθεί ευκολότερα από τους χρήστες. Ειδικά η οργάνωση πολυμεσικού περιεχομένου, το οποίο είναι δύσκολο να οριστεί μονοσήμαντα, αποτελεί μια πρόκληση, στην οποία η κοινότητα του MPEG έχει προτείνει το πρότυπο MPEG-21.

Το EuterpeFS, επιχειρώντας να δώσει μια απάντηση σε αυτές τις απαιτήσεις οργάνωσης των δεδομένων, υιοθετεί και επεκτείνει το πρότυπο MPEG-21, ενώ τα παρέχει διαφανώς πάνω από ένα καταναμημένο σύστημα αρχείων.

Abstract

This master thesis focuses on the design and development of the EuterpeFS distributed file system. EuterpeFS has two main goals: The efficient files organization and the support of remote access to them, independently of their location.

Distributed file systems have been successfully used for years and they are considered to be a fundamental part of a distributed system. The motivation towards their design and implementation has been the users' need for easy, efficient and secure access to their files, despite their location.

The continuous growth of information being circulated over the Internet suggests the need for a better organization, so that the users may easily discover and access data. Especially the organization of multimedia content, which is hard to define unambiguously, has become a challenge, to which the MPEG community has suggested the MPEG-21 standard.

EuterpeFS tries to address the data organization requirements by enhancing and extending the MPEG-21 standard, while it sets them easily and transparently accessible through a distributed file system.

Περιεχόμενα

Περιεχόμενα	v
1 Εισαγωγή	3
2 Συστήματα Αρχείων στο LINUX	5
2.1 Εισαγωγή	5
2.2 Βασικά Χαρακτηριστικά Συστημάτων Αρχείων	6
2.3 Τύποι Αρχείων	6
2.4 Διαμέριση του Σκληρού Δίσκου	7
2.5 Προσάρτηση Διαμερίσματος	8
2.6 Βασικά Γνωρίσματα Αρχείων	8
2.7 Το Εικονικό Σύστημα Αρχείων	9
2.8 FUSE: Σύστημα Αρχείων στο Χώρο Χρηστών	10
3 Κατανεμημένα Συστήματα Αρχείων	13
3.1 Γενικά	13
3.2 Απαιτήσεις Κατανεμημένων Συστημάτων Αρχείων	13
3.2.1 Διαφάνεια	14
3.2.2 Υποστήριξη ταυτόχρονων ενημερώσεων των αρχείων	15
3.2.3 Αντίγραφα Αρχείων	15
3.2.4 Υποστήριξη ετερογένειας	16
3.2.5 Ανοχή Βλαβών	16
3.2.6 Συνέπεια	16
3.2.7 Ασφάλεια	16
3.2.8 Αποδοτικότητα	16
3.3 Αρχιτεκτονική Κατανεμημένων Συστημάτων Αρχείων	17
3.3.1 Υπηρεσία Αρχείων	17
3.3.2 Υπηρεσία Καταλόγου	17

3.3.3	Εφαρμογή Πελάτη	17
3.4	Μελέτες Περίπτωσης	18
3.4.1	Sun Network File System	18
3.4.2	Andrew File System	19
4	Το Πρότυπο MPEG-21	23
4.1	Εισαγωγή	23
4.2	Βασικά Συστατικά	25
4.2.1	Container	25
4.2.2	Item	25
4.2.3	Component	27
4.2.4	Anchor	28
4.2.5	Descriptor	28
4.2.6	Condition	29
4.2.7	Choice	29
4.2.8	Selection	30
4.2.9	Annotation	31
4.2.10	Assertion	32
4.2.11	Resource	32
4.2.12	Fragment	33
4.2.13	Statement	33
4.2.14	Predicate	33
5	EuterpeFS	35
5.1	Εισαγωγή	35
5.2	Αρχιτεκτονική	35
5.2.1	Επισκόπηση Αρχιτεκτονικής	36
5.2.2	Λειτουργίες	37
5.2.3	Τεχνολογίες	43
5.3	Έλεγχος Σχεδιαστικών Απαιτήσεων	45
5.3.1	Διαφάνεια	46
5.3.2	Υποστήριξη ταυτόχρονων ενημερώσεων των αρχείων	46
5.3.3	Αντίγραφα Αρχείων	47
5.3.4	Υποστήριξη ετερογένειας	47
5.3.5	Ανοχή Βλαβών	47
5.3.6	Συνέπεια	47
5.3.7	Ασφάλεια	47
5.3.8	Αποδοτικότητα	48
6	Συμπεράσματα και Μελλοντικό Έργο	49
	Βιβλιογραφία	51

Α΄ Παράδειγμα Μουσικού Άλμπουμ	55
Πίνακας Συμβόλων και Συντμήσεων	69
Πίνακας Εικόνων	70

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΛ

Ευχαριστίες

Πάνω από όλα, θα ήθελα να ευχαριστήσω την οικογένειά μου για την υποστήριξή της σε όλη την πορεία των σπουδών μου. Την κοπέλα μου για τη στήριξή της σε μια πολύ κουραστική και απαιτητική χρονιά. Τον κύριο Καθηγητή Χρήστο Δουληγέρη που με εμπιστεύτηκε για την εκπόνηση αυτής της Διπλωματικής Διατριβής. Τον κύριο Καθηγητή Ιάκωβο Στ. Βενιέρη και την κυρία Καθηγήτρια Δήμητρα Ι. Κακλαμάνη για την ευκαιρία που μου έδωσαν να ξεκινήσω μια ερευνητική προσπάθεια, ένα πρώτο δείγμα της οποίας φιλοδοξεί να αποτελέσει η παρούσα Εργασία. Το Δρ. Γεώργιο Λιουδάκη, ο οποίος υπήρξε πάντα πολύτιμος συνομιλητής με καίριες παρατηρήσεις και συμβουλές. Όλους τους ανθρώπους με τους οποίους συνεργάστηκα αυτή τη χρονιά για τη στήριξή τους.

Κεφάλαιο 1

Εισαγωγή

Ένας από τους τομείς των κατανεμημένων συστημάτων που αποτελούσε πάντα μια ελκυστική ερευνητική περιοχή είναι αυτός των κατανεμημένων συστημάτων αρχείων. Άλλωστε, ο διαμοιρασμός των αρχείων πάνω από το δίκτυο μεταξύ πολλών υπολογιστών και χρηστών έχει αποτελέσει και μια από τις πλέον ευρέως διαδεδομένες μορφές εφαρμογής των κατανεμημένων συστημάτων στην καθημερινότητά μας από τα πρώτα χρόνια.

Στις μέρες υπάρχουν αρκετά κατανεμημένα συστήματα αρχείων, καθένα από τα οποία συνδυάζει διαφορετικά χαρακτηριστικά και είναι προσανατολισμένο σε διαφορετικές περιπτώσεις. Ενδεικτικά αναφέρονται το Sun Network File System [22], το Andrew File System [20], το Coda [25] και το Google File System [19] ως επιτυχημένες και ευρέως διαδεδομένες περιπτώσεις κατανεμημένων συστημάτων αρχείων.

Το κάθε ένα από τα προαναφερθέντα συστήματα επικεντρώνεται σε μια περιοχή και προσπαθεί να δώσει μια αποδοτική λύση. Έτσι, αν και θα μπορούσε κάποιος να πει ότι το NFS και το AFS ακολουθούν μια διαφορετική προσέγγιση για να κάνουν κάτι αντίστοιχο, το Coda έχει αναπτυχθεί με έμφαση σε κινητές συσκευές με χαμηλή συνδεσιμότητα και το Google File System για να διαχειρίζεται πολύ μεγάλα αρχεία, όπως αυτά που πολλές φορές καλείται να διαχειριστεί η ίδια η Google.

Το κατανεμημένο σύστημα αρχείων EuterpeFS επικεντρώνεται σε μια διαφορετική περιοχή. Αναγνωρίζοντας την ανάγκη για περισσότερα μεταδεδομένα και παρακολουθώντας την εξέλιξη του σημασιολογικού ιστού και την ενσωμάτωση όλο και περισσότερης πληροφορίας στους πόρους που διακινούνται στο διαδίκτυο, οδηγήθηκαμε στην απόφαση να δημιουργήσουμε ένα κατανεμημένο σύστημα αρχείων που

θα λαμβάνει υπόψη του αυτές τις σχεδιαστικές αρχές, ενώ θα επιτρέπει να χρησιμοποιηθεί σε συνεργασία με οποιοδήποτε άλλο καταναμημένο σύστημα αρχείων.

Μέσα σε αυτό το πνεύμα αποφασίστηκε η χρήση του προτύπου του ψηφιακού αντικειμένου από το MPEG-21 [21]. Έχοντας αυτό ως βάση, το EutereFS μπορεί να περιγράψει με έναν προτυποποιημένο τρόπο τους πόρους (αρχεία) που εξυπηρετεί πάνω από το δίκτυο. Φυσικά, για να γίνει αυτό έπρεπε να γίνουν κάποιες μικρές προσθήκες σε αυτά που περιγράφει το πρότυπο, χωρίς όμως, κατά την άποψη του συγγραφέα, να θίγεται κάτι θεμελιώδες.

Στο υπόλοιπο της παρούσας Μεταπτυχιακής Διατριβής, γίνεται μια ανασκόπηση των βασικών αρχών των συστημάτων αρχείων στο LINUX (Κεφάλαιο 2), των αρχών σχεδίασης των καταναμημένων συστημάτων αρχείων (Κεφάλαιο 3 και του δεύτερου μέρους του προτύπου MPEG-21 μαζί με τις όποιες επεκτάσεις κρίθηκαν απαραίτητες για την υποστήριξη του EutereFS (Κεφάλαιο 4). Στη συνέχεια (Κεφάλαιο 5) περιγράφεται το καταναμημένο σύστημα αρχείων EutereFS και ελέγχεται κατά πόσο ανταποκρίνεται στις αρχές σχεδίασης των καταναμημένων συστημάτων αρχείων που παρουσιάστηκαν στα προηγούμενα Κεφάλαια. Στο Κεφάλαιο 6 γίνεται αναφορά στο έργο που πρόκειται να ακολουθήσει αυτήν την Εργασία. Τέλος, στο Παράρτημα δίνεται ένα παράδειγμα ψηφιακού αντικειμένου, το οποίο έχει ληφθεί κυρίως από το [21], έχει προσαρμοστεί στις ανάγκες του EutereFS και χρησιμοποιήθηκε για τους ελέγχους.

Κεφάλαιο 2

Συστήματα Αρχείων στο LINUX

"Σε ένα UNIX σύστημα, τα πάντα είναι αρχεία - αν κάτι δεν είναι αρχείο, τότε είναι διεργασία"

2.1 Εισαγωγή

Το σύστημα αρχείων είναι μια μέθοδος αποθήκευσης και οργάνωσης των αρχείων των ηλεκτρονικών υπολογιστών [3]. Ειδικότερα στα UNIX συστήματα (το LINUX συγκαταλέγεται σε αυτήν την κατηγορία), τα συστήματα αρχείων υποστηρίζουν και ειδικού τύπου αρχεία, τα οποία μπορεί να αναπαριστούν σωλήνες (pipes), στόμια (sockets) και συσκευές. Επίσης, οι κατάλογοι θεωρούνται και αυτοί αρχεία, τα οποία απλά περιέχουν ονόματα και άλλων αρχείων. Η δομή που χρησιμοποιείται για την αναπαράσταση είναι δενδρική, όπου έχουμε κλαδιά να περιέχουν άλλα κλαδιά κ.ο.κ. [5].

Το λειτουργικό σύστημα, απο τη μεριά του, πρέπει να είναι σε θέση να υποστηρίξει οποιοδήποτε σύστημα αρχείων με διαφάνεια προς το χρήστη. Για το λόγο αυτό, καθορίζει κάποιες βασικές δομές και λειτουργίες, τις οποίες θα πρέπει να λάβει υπόψη του ο σχεδιαστής ενός συστήματος προκειμένου να μπορέσει να φτιάξει το δικό του σύστημα αρχείων. Καθώς, μάλιστα, τα τελευταία χρόνια οι ανάγκες για νέα συστήματα αρχείων, προσαρμοσμένα στις σύγχρονες απαιτήσεις, αυξάνονται, έχουν αναπτυχθεί και αντίστοιχες υποδομές που καθιστούν την υλοποίηση νέων συστημάτων αρχείων πιο προσιτή.

2.2 Βασικά Χαρακτηριστικά Συστημάτων Αρχείων

Ένα από τα οφέλη της ύπαρξης του συστήματος αρχείων και του μέσου φυσικής αποθήκευσης (π.χ. σκληρός δίσκος) είναι ότι το πρώτο παρέχει ένα ολοκληρωμένο περιβάλλον προγραμματιστικής διεπαφής, το οποίο αποδεσμεύει τον προγραμματιστή εφαρμογών από επίπονες διαδικασίες που αφορούν το υλικό και τη διαχείριση συσκευών.

Συγκεκριμένα, τα συστήματα αρχείων δίνουν τη δυνατότητα χειρισμού τόσο των δεδομένων όσο και των χαρακτηριστικών των αρχείων. Με τον όρο δεδομένα εννοούμε ένα σύνολο από bit τα οποία μπορούμε να τα διαβάζουμε και να τα γράφουμε στο αποθηκευτικό μέσο. Με τον όρο χαρακτηριστικά, εννοούμε όλες εκείνες τις πληροφορίες που χαρακτηρίζουν το αρχείο και συνήθως έχουν να κάνουν με δικαιώματα πρόσβασης, με τον τύπο των αρχείων και με το χρόνο προσπέλασης και δημιουργίας. Περισσότερες πληροφορίες για τα χαρακτηριστικά των αρχείων περιέχονται στις παρακάτω Ενότητες αυτού του Κεφαλαίου.

Εκτός όμως από τα δεδομένα και τα χαρακτηριστικά, υπάρχει και ένας ακόμη όρος που χρησιμοποιείται όλο και περισσότερο τον τελευταίο καιρό για να περιγράψει την πληροφορία: τα μεταδεδομένα. Τα μεταδεδομένα συχνά θεωρούνται ένα υπερσύνολο των χαρακτηριστικών των αρχείων, περιέχοντας επιπλέον πληροφορία για την καλή διαχείριση των αρχείων. Μάλιστα, οι τελευταίες εκδόσεις διάφορων συστημάτων αρχείων (π.χ. ext4) δίνουν τη δυνατότητα στο χρήστη να ορίσει ο ίδιος και δικά του μεταδεδομένα, που ονομάζονται εκτεταμένα χαρακτηριστικά των αρχείων. Ο ορισμός αυτών των εκτεταμένων χαρακτηριστικών είναι πλήρως δυναμικός και εναπόκειται στο χρήστη να τα χρησιμοποιήσει κατά περίπτωση. Έτσι, χρησιμοποιώντας αυτές τις δυνατότητες, έχουν αναπτυχθεί συστήματα που αξιοποιούν πληροφορία την οποία αποθηκεύουν σε ειδικές περιοχές ονομάτων των εκτεταμένων χαρακτηριστικών των αρχείων για δική τους χρήση (π.χ. Security Enhanced Linux - SELinux).

2.3 Τύποι Αρχείων

Στο LINUX έχουμε 7 διαφορετικούς τύπους αρχείων, καθένας από τους οποίους αντιστοιχεί σε ένα μοναδικό αναγνωριστικό. Έτσι, διακρίνουμε τα αρχεία με βάση τον τύπο τους σε:

- Κανονικά Αρχεία (-).
- Κατάλογοι (d).

- Σύνδεσμοι (l).
- Ειδικά Αρχεία (c).
- Στόμια (s).
- Σωλήνες (p).
- Συσκευές Μπλοκ (b).

Ο χρήστης μπορεί να δει τον τύπο του κάθε αρχείου, χρησιμοποιώντας το όρισμα -l στην εντολή ls του LINUX που χρησιμεύει για να πάρει τα αρχεία του. Έτσι, ένα παράδειγμα εκτέλεσης είναι το εξής:

```
aca@linux-jaix:/dev> ls -l
total 0
crw-rw----+ 1 root audio   14, 12 2010-09-28 08:45 adsp
crw-rw----+ 1 root audio   14,  4 2010-09-28 08:45 audio
drwxr-xr-x  2 root root    320 2010-09-28 08:44 block
prw-----  1 root root      0 2010-09-28 08:45 boot splash
drwxr-xr-x  2 root root     80 2010-09-28 11:44 bsg
drwxr-xr-x  3 root root     60 2010-09-28 11:44 bus
lrwxrwxrwx  1 root root      3 2010-09-28 08:44 cdrom -> sr0
lrwxrwxrwx  1 root root      3 2010-09-28 08:44 cdrw -> sr0
drwxr-xr-x  2 root root   3340 2010-09-28 08:45 char
crw-----  1 root root      5,  1 2010-09-28 08:45 console
```

Εικόνα 2.1: Ενδεικτικό αποτέλεσμα εντολής ls -l.

2.4 Διαμέριση του Σκληρού Δίσκου

Η διαμέριση του σκληρού δίσκου είναι μια τεχνική που έχει πολλαπλά οφέλη. Ενδεικτικά κάποια από αυτά είναι η καλύτερη οργάνωση των αρχείων σε συγκεκριμένες περιοχές του δίσκου και η προστασία από μια εκτεταμένη βλάβη στο σύστημα αρχείων. Έτσι, σε περίπτωση που συμβεί κάποιο ατύχημα στο σύστημα αρχείων ενός διαμερίσματος του δίσκου, τα αρχεία που βρίσκονται στα υπόλοιπα διαμερίσματα είναι πολύ πιθανό να διατηρηθούν ανέπαφα.

Στο LINUX υπάρχουν δύο βασικά είδη διαμερισμάτων:

- Το διαμέρισμα δεδομένων, που χρησιμοποιείται για την αποθήκευση των κανονικών μόνιμων δεδομένων των χρηστών και του συστήματος.
- Το διαμέρισμα swap, που χρησιμοποιείται από το σύστημα ως επιπλέον χώρος φυσικής μνήμης.

2.5 Προσάρτηση Διαμερίσματος

Για να χρησιμοποιηθεί ένα διαμέρισμα δεδομένων τόσο από τους χρήστες όσο και από το ίδιο το σύστημα, θα πρέπει να προσαρτηθεί σε κάποιο σημείο προσάρτησης. Αυτό το σημείο προσάρτησης αποτελεί στην ουσία το σημείο εισόδου για αυτό το διαμέρισμα. Όλα τα διαμερίσματα διασυνδέονται στο ανώτερο επίπεδο μέσω του διαμερίσματος ρίζα (root partition(/)).

Η προσάρτηση ενός διαμερίσματος γίνεται με χρήση της εντολής mount. Εκτός από τα διαμερίσματα που προσαρτώνται χειροκίνητα από το χρήστη, υπάρχουν και κάποια που μπορεί να φορτώνονται αυτόματα από το σύστημα κατά την εκκίνηση. Όλα αυτά είναι δηλωμένα στο αρχείο /etc/fstab. Επίσης, στο αρχείο /etc/mtab καταγράφονται όλα τα διαμερίσματα που είναι προσαρτημένα ανά πάσα στιγμή.

2.6 Βασικά Γνωρίσματα Αρχείων

Τα βασικά γνωρίσματα ενός αρχείου στο LINUX, τα οποία θα πρέπει να γνωρίζει το λειτουργικό σύστημα βρίσκονται στη δομή stat. Τα στοιχεία αυτής της δομής είναι τα εξής [15]:

- Το αναγνωριστικό της συσκευής όπου βρίσκεται το αρχείο.
- Ο αριθμός ευρετηρίου του αρχείου στο δίσκο (i-node).
- Ο τρόπος πρόσβασης στο αρχείο (mode). Πρόκειται για έναν αριθμό, ο οποίος: (α) δηλώνει τα προνόμια του ιδιοκτήτη του αρχείου, της ομάδας του ιδιοκτήτη και όλων των υπολοίπων ως προς την ανάγνωση, την εγγραφή και την εκτέλεση του αρχείου και (β) ορίζει τον τύπο του αρχείου (κανονικό αρχείο, κατάλογος, συσκευή χαρακτήρων, συσκευή μπλοκ, σωλήνας FIFO, συμβολικός σύνδεσμος, στόμιο).
- Ο αριθμός των σκληρών συνδέσμων (hard links) του αρχείου. Ο σκληρός σύνδεσμος χρησιμοποιείται όταν θέλουμε να χρησιμοποιήσουμε τον ίδιο αριθμό ευρετηρίου για δύο αρχεία.
- Ο μοναδικός αριθμός που αντιστοιχεί στο χρήστη-ιδιοκτήτη του αρχείου.
- Ο μοναδικός αριθμός που αντιστοιχεί στην ομάδα του χρήστη-ιδιοκτήτη του αρχείου.
- Ο μοναδικός αριθμός που αντιστοιχεί στη συσκευή που αναπαριστά αυτό το αρχείο (αν πρόκειται για τέτοιου είδους αρχείο).

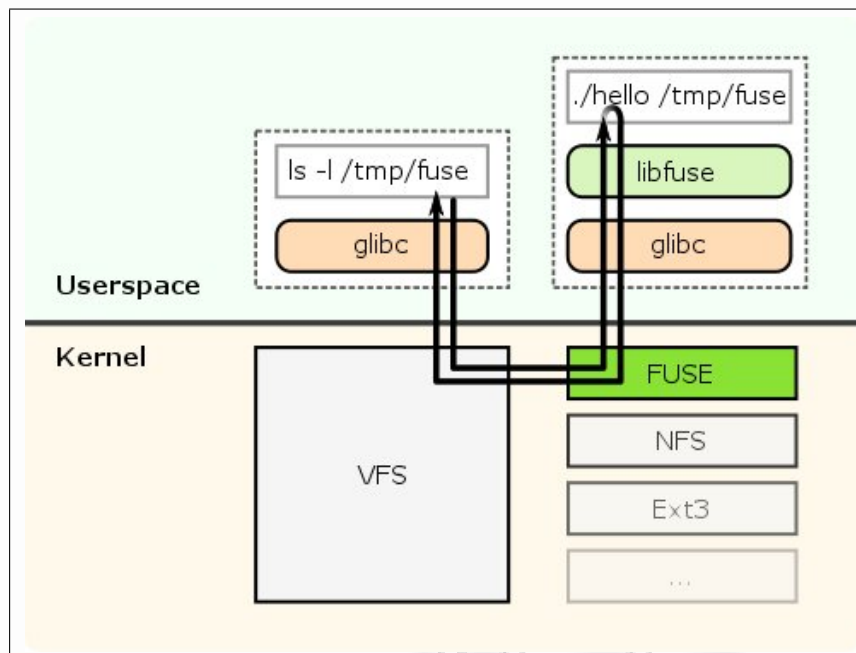
- Το μέγεθος του αρχείου σε bytes.
- Το μέγεθος μπλοκ του συστήματος αρχείων.
- Ο αριθμός μπλοκ των 512B που έχουν καταληφθεί για αυτό το αρχείο.
- Η χρονική στιγμή της τελευταίας προσπέλασης του αρχείου.
- Η χρονική στιγμή της τελευταίας αλλαγής του αρχείου.
- Η χρονική στιγμή της τελευταίας αλλαγής κατάστασης του αρχείου. Ουσιαστικά αναφέρεται σε αλλαγή κάποιου από τα προαναφερθέντα στοιχεία αυτής της δομής.

2.7 Το Εικονικό Σύστημα Αρχείων

Το Εικονικό Σύστημα Αρχείων (Virtual File System - VFS) αποτελεί ένα επίπεδο αφαίρεσης των συστημάτων αρχείων στο LINUX. Στόχος του είναι η διαχείριση όλων των (πιθανώς διαφορετικών) συστημάτων αρχείων που είναι προσαρτημένα ανά πάσα στιγμή. Για το λόγο αυτό, διατηρεί ειδικές δομές δεδομένων που περιγράφουν τόσο το συνολικό σύστημα αρχείων (εικονικό) όσο και κάθε ένα από τα επιμέρους (πραγματικά) συστήματα αρχείων [16].

Για να μπορέσει να υποστηρίξει αυτές τις λειτουργίες, το VFS χρησιμοποιεί δύο βασικές δομές: το υπερμπλοκ (superblock) που αφορά τα συστήματα αρχείων και το ευρετήριο κόμβου (i-node) που αφορά το κάθε αρχείο. Το superblock περιλαμβάνει πληροφορίες σχετικές με τη συσκευή όπου περιέχεται το εκάστοτε σύστημα αρχείων, ένα δείκτη στο πρώτο i-node του αντίστοιχου συστήματος αρχείων, το μέγεθος του μπλοκ, δείκτες στις λειτουργίες superblock που υποστηρίζονται από το σύστημα αρχείων, τον τύπο του συστήματος αρχείων (διατηρείται σχετικός πίνακας με τους υποστηριζόμενους τύπους συστημάτων αρχείων) και άλλες πληροφορίες συγκεκριμένες για κάθε τύπο συστήματος αρχείων. Το i-node, μεταξύ άλλων, περιλαμβάνει πληροφορίες σχετικά με τη συσκευή όπου διατηρείται το κάθε αρχείο, τον αριθμό ευρετηρίου του αρχείου, τον τρόπο πρόσβασης, τον ιδιοκτήτη του αρχείου, τις λειτουργίες που υποστηρίζονται από το σύστημα αρχείων για αυτό το αρχείο.

Για παράδειγμα, αν θέλουμε να προσαρτήσουμε ένα σύστημα αρχείων, θα ακολουθούσαμε την παρακάτω διαδικασία. Ας θεωρήσουμε ότι ο διαχειριστής του συστήματος (η εντολή προσάρτησης mount απαιτεί προνόμια διαχειριστή) επιθυμεί να προσαρτήσει ένα flash disk με σύστημα αρχείων ntfs στον κατάλογο /media/usbdisk με προνόμια μόνο για ανάγνωση. Η εντολή προσάρτησης θα διαμορφωθεί ως εξής:
\$ mount -t ntfs -o ro /dev/sda1 /media/usbdisk



Εικόνα 2.2: Δομή του FUSE [4].

Το πρώτο πράγμα που πρέπει να κάνει το VFS είναι να εντοπίσει το σύστημα αρχείων που έχει δοθεί. Αν δεν είναι ανάμεσα στα γνωστά του συστήματα, τότε περνάει τον έλεγχο στον πυρήνα, ο οποίος μπορεί είτε να απορρίψει απευθείας το αίτημα της προσάρτησης με τη δικαιολογία ότι δεν αναγνωρίζεται το σύστημα αρχείων είτε (κάποιοι πυρήνες το υποστηρίζουν) να ψάξει ανάμεσα στα kernel modules του για να βρει το δοθέν σύστημα αρχείων.

Στη συνέχεια, το VFS αναλαμβάνει να εντοπίσει το i-node του σημείου προσάρτησης και να εξετάσει αν υπάρχει ήδη κάποιο άλλο σύστημα αρχείων προσαρτημένο εκεί.

Τέλος, το superblock του VFS αναλαμβάνει να ενημερωθεί με τα στοιχεία των αντίστοιχων δομών δεδομένων που περιέχονται στο superblock του προς προσάρτηση συστήματος αρχείων.

2.8 FUSE: Σύστημα Αρχείων στο Χώρο Χρηστών

Το FUSE παρέχεται ως μια ανεξάρτητη μονάδα λογισμικού που μπορεί να φορτωθεί δυναμικά στον πυρήνα του λειτουργικού συστήματος (kernel module) και επιτρέπει στους χρήστες του συστήματος να γράψουν τα δικά τους συστήματα αρχείων χωρίς να χρειάζεται να επέμβουν στον κώδικα του πυρήνα. Στην ουσία το FUSE παρέχει τη διεπαφή που ενώνει τον κώδικα του χρήστη με αυτόν του πυρήνα [4]. Ενδεικτικό

σχήμα για τη δομή του FUSE δίνεται στην Εικόνα 2.2.

Χωρίς να είναι απαραίτητο να παρέχει ουσιαστικές ρουτίνες αποθήκευσης και ανάκτησης από το σκληρό δίσκο, όπως όλα τα παραδοσιακά συστήματα αρχείων, το FUSE μπορεί να λειτουργήσει σε ένα πιο υψηλό επίπεδο, στηριζόμενο στο υποκείμενο σύστημα αρχείων για τις πρωταρχικές λειτουργίες.

Η προγραμματιστική διεπαφή του FUSE παρέχει ένα σύνολο συναρτήσεων που καλείται να υλοποιήσει ο χρήστης και οι οποίες υπερβαίνουν τις κανονικές ρουτίνες του συστήματος, χωρίς αυτό όμως να σημαίνει ότι δεν μπορούν να χρησιμοποιηθούν και αυτές. Για παράδειγμα, για το άνοιγμα ενός αρχείου, θα μπορούσαμε απλά να καλούμε την κανονική κλήση συστήματος για το δοθέν αρχείο, χωρίς να κάνουμε τίποτα περισσότερο.

Το FUSE είναι αρκετά διαδεδομένο λόγω της απλότητας σε συνδυασμό με τη σταθερότητα που παρέχει. Έτσι, υπάρχει ένας μεγάλος αριθμός συστημάτων αρχείων που έχουν στηριχθεί σε αυτό. Σχετική λίστα διατηρείται στην επίσημη ιστοσελίδα του FUSE [13].

Κεφάλαιο 3

Κατανεμημένα Συστήματα Αρχείων

3.1 Γενικά

Τα κατανεμημένα συστήματα αρχείων επιτρέπουν το διαμοιρασμό αρχείων πάνω από κάποιο δίκτυο. Κύριο χαρακτηριστικό τους είναι ότι παρέχουν εγγυήσεις απόδοσης και αξιοπιστίας ανάλογες (σε μερικές περιπτώσεις καλύτερες) με ένα τοπικό σύστημα αρχείων. Ένα κατανεμημένο σύστημα αρχείων επιτρέπει σε απομακρυσμένες διεργασίες οποιουδήποτε υπολογιστή του δικτύου να έχουν πρόσβαση στα αρχεία με διαφάνεια [18].

3.2 Απαιτήσεις Κατανεμημένων Συστημάτων Αρχείων

Σε αυτήν την Ενότητα, θα παρουσιαστούν οι βασικές απαιτήσεις για τη σχεδίαση ενός κατανεμημένου συστήματος αρχείων. Οι απαιτήσεις αυτές έχουν προκύψει κυρίως μέσα από την εμπειρία χρόνων στο χώρο των κατανεμημένων συστημάτων αρχείων. Στην παρούσα εργασία, θα παρουσιαστούν οι απαιτήσεις όπως έχουν δοθεί από τους Coulouris et al. στο [18].

3.2.1 Διαφάνεια

Η διαφάνεια (σε όλες τις μορφές της) αποτελεί μια σύτωση ή άλλως απαραίτητη ιδιότητα ενός καλά οργανωμένου και αποδοτικού κατανεμημένου συστήματος. Ειδικά όταν μιλάμε για ένα κατανεμημένο σύστημα αρχείων, το οποίο καλείται να εξυπηρετήσει κρίσιμες εφαρμογές και μεγάλο όγκο κίνησης, χωρίς να υπάρχει κάποια διαφοροποίηση στις εφαρμογές του χρήστη, η διαφάνεια γίνεται προϋπόθεση. Συγκεκριμένα, ο σχεδιαστής ενός κατανεμημένου συστήματος αρχείων καλείται να ικανοποιεί τις εξής κατηγορίες διαφάνειας:

Διαφάνεια Πρόσβασης

Οι εφαρμογές του πελάτη δε θα πρέπει να κάνουν διάκριση κατά την Είσοδο/Έξοδο για το αν έχουν να κάνουν με τοπικό ή κατανεμημένο σύστημα αρχείων. Οι κλήσεις συστήματος που υποστηρίζονται από το τοπικό σύστημα αρχείων θα πρέπει να υποστηρίζονται παρέχοντας αντίστοιχη λειτουργικότητα και από ένα κατανεμημένο σύστημα αρχείων.

Διαφάνεια Τοποθεσίας

Οι εφαρμογές του πελάτη δε θα πρέπει να κάνουν διάκριση αν το αρχείο στο οποίο πρόκειται να αποκτήσουν πρόσβαση ανήκει σε τοπικό ή κατανεμημένο σύστημα αρχείων. Έτσι, ο χώρος ονομάτων είναι κοινός τόσο για το τοπικό όσο και για το κατανεμημένο σύστημα αρχείων. Για παράδειγμα, θα μπορούσαν σε ένα σύστημα να υπάρχουν:

- ένας κατάλογος `/home/foo/local` που ανήκει στο τοπικό σύστημα αρχείων
- ένας κατάλογος `/home/foo/remote` ανήκει στο κατανεμημένο σύστημα αρχείων

και οι εφαρμογές του συστήματος να προσπελούν αρχεία που βρίσκονται και στους δύο χώρους ονομάτων χωρίς καμία διαφοροποίηση.

Διαφάνεια Μετακίνησης

Ακόμα και στην περίπτωση που κάποιο αρχείο αλλάξει θέση και παρέχεται από κάποιον άλλο εξυπηρετητή, οι εφαρμογές του πελάτη δε θα πρέπει να επηρεάζονται. Με άλλα λόγια, θα πρέπει να διατηρείται η διαφάνεια τοποθεσίας ανεξάρτητα από τη φυσική θέση του αρχείου.

Διαφάνεια Απόδοσης

Γενικά οι απαιτήσεις σε απόδοση ενός κατανεμημένου συστήματος αρχείων μπορεί να ποικίλουν μέσα σε κάποιο εύρος. Το εύρος αυτό αποτελεί σχεδιαστική παράμετρο του συστήματος αρχείων. Σε κάθε περίπτωση, για φόρτο που κυμαίνεται μέσα σε αυτό το εύρος, το κατανεμημένο σύστημα αρχείων θα πρέπει να έχει σταθερή -σχετικά- απόδοση.

Διαφάνεια Κλιμάκωσης

Το κατανεμημένο σύστημα αρχείων θα πρέπει να διατηρεί σταθερή την απόδοσή του, ακόμα και αν προστεθούν πολλοί πελάτες, προκαλώντας αύξηση στο φόρτο των δεδομένων που διακινούνται πάνω από το δίκτυο αλλά ακόμα και στο ίδιο το δίκτυο, εισάγοντας τις σχετικές καθυστερήσεις και την πολυπλοκότητα.

3.2.2 Υποστήριξη ταυτόχρονων ενημερώσεων των αρχείων

Η υποστήριξη ταυτόχρονων ενημερώσεων ενός αρχείου από πολλούς χρήστες θεωρείται κάτι το αυτονόητο για ένα κατανεμημένο σύστημα αρχείων. Γενικά, η ταυτόχρονη πρόσβαση σε πόρους έχει αποτελέσει θέμα μακροχρόνιας έρευνας, η οποία έχει κατασταλάξει σε σύγχρονους μηχανισμούς που έχουν ενσωματωθεί στο UNIX και αφορούν το κλείδωμα συγκεκριμένων αρχείων ή εγγραφών.

3.2.3 Αντίγραφα Αρχείων

Η τεχνική των αντιγράφων στα κατανεμημένα συστήματα είναι ευρέως διαδεδομένη και, φυσικά, δε θα μπορούσε να λείπει από ένα κατανεμημένο σύστημα αρχείων, καθώς θα μπορούσε να προσφέρει διπλό όφελος. Αφενός καταφέρνουμε εξισορρόπηση φόρτου, καθώς υπάρχουν πολλοί εξυπηρετητές από τους οποίους παρέχεται το κάθε αρχείο (κάτι τέτοιο μπορεί να έχει και σοβαρή επίδραση στην απόδοση του συστήματος) και αφετέρου, χρησιμοποιώντας τεχνικές κρυφής μνήμης, καταφέρνουμε αύξηση της ταχύτητας εξυπηρέτησης, χρησιμοποιώντας λιγότερους πόρους, σε σχέση με την πρώτη περίπτωση.

3.2.4 Υποστήριξη ετερογένειας

Εδώ, με τον όρο ετερογένεια εννοούμε την υποστήριξη του συστήματος αρχείων για διαφορετικές πλατφόρμες υλικού και λειτουργικών συστημάτων.

3.2.5 Ανοχή Βλαβών

Η ανοχή βλαβών είναι μία ακόμα βασική αρχή των κατανεμημένων συστημάτων και, μάλιστα, πολλές φορές δύσκολα υλοποιήσιμη. Ευτυχώς τα κατανεμημένα συστήματα αρχείων δεν ανήκουν σε αυτές τις περιπτώσεις, καθώς χρησιμοποιώντας σημειολογία at-most-once (ή at-least-once με φρονιδα για τις πολλαπλές ενημερώσεις που μπορεί να λάβουν χώρα σε αυτήν την περίπτωση), μη κρατώντας κατάσταση (stateless) και κάνοντας χρήση αντιγράφων, καταφέρνουν να παρουσιάζουν μεγάλη ανοχή σε βλάβες.

3.2.6 Συνέπεια

Η συνέπεια αποτελεί σημαντικό αλλά και αρκετά πολύπλοκο παράγοντα στη σχεδίαση ενός κατανεμημένου συστήματος αρχείων. Με τον όρο συνέπεια εννοούμε το να βλέπουν το ίδιο περιεχόμενο διαφορετικές διεργασίες που έχουν ανοικτό το ίδιο αρχείο (το οποίο βέβαια μπορεί να είναι διασκορπισμένο σε αντίγραφα στο δίκτυο με κάθε διεργασία να βλέπει διαφορετικό αντίγραφο του αρχείου).

3.2.7 Ασφάλεια

Η ασφάλεια ενός τυπικού συστήματος αρχείων έγκειται στην αναγνώριση του χρήστη που επιχειρεί πρόσβαση και στον έλεγχο των προνομίων του. Σε ένα κατανεμημένο σύστημα αρχείων χρειαζόμαστε έναν επιπλέον μηχανισμό αφενός για την απομακρυσμένη αυθεντικοποίηση του χρήστη και αφετέρου για την ασφαλή μετάδοση των δεδομένων (π.χ. κρυπτογράφηση).

3.2.8 Αποδοτικότητα

Η αποδοτικότητα είναι ένας παράγοντας που μετράται με διαφορετικές μονάδες, ανάλογα με την περιοχή εφαρμογής του κατανεμημένου συστήματος αρχείων. Έτσι, για παράδειγμα, για άλλες περιπτώσεις μπορεί να είναι ο χρόνος προσπέλασης, ενώ για άλλες οι περιορισμένες ανάγκες για τοπικό αποθηκευτικό χώρο. Σε κάθε περι-

πτωση, θα πρέπει να παρέχεται ένα σύστημα αρχείων που να μπορεί να χρησιμοποιηθεί από τις εφαρμογές κάθε πλατφόρμας χωρίς να θέτει περιορισμούς και να περιορίζει με ανώμαλο τρόπο την απόδοση του συστήματος.

3.3 Αρχιτεκτονική Κατανεμημένων Συστημάτων Αρχείων

Η σχεδίαση ενός κατανεμημένου συστήματος αρχείου, όπως άλλωστε και κάθε συστήματος, εναπόκειται στο χρήστη και δεν είναι απαραίτητο να ακολουθηθούν κάποια συγκεκριμένα πρότυπα, αρκεί να ικανοποιούνται οι απαιτήσεις (ή τουλάχιστον οι περισσότερες από αυτές) όπως παρουσιάστηκαν στην προηγούμενη Ενότητα. Σε κάθε περίπτωση, όμως, στη βιβλιογραφία [18, 27] συναντάμε κάποιες βασικές αρχές ως προς τα θεμελιώδη συστατικά ενός κατανεμημένου συστήματος αρχείων, οι οποίες θα εκτεθούν σε αυτήν την Ενότητα.

3.3.1 Υπηρεσία Αρχείων

Η υπηρεσία αρχείων (flat file service) ουσιαστικά ορίζει την πρόσβαση στα αρχεία μέσω των μεθόδων που ορίζει η διεπαφή του συστήματος αρχείων. Για να μπορέσει αυτή η υπηρεσία να διαχειριστεί σωστά τα αρχεία χρησιμοποιεί μοναδικά, για όλο το κατανεμημένο σύστημα αρχείων, αναγνωριστικά αρχείων (UFID) τα οποία ανατίθενται στα αρχεία κατά τη δημιουργία τους [18].

3.3.2 Υπηρεσία Καταλόγου

Η υπηρεσία καταλόγου ορίζει την αντιστοίχιση μεταξύ του UFID και του πραγματικού ονόματος του αρχείου. Έτσι, όταν η εφαρμογή του πελάτη ζητήσει ένα αρχείο με το όνομά του, η υπηρεσία καταλόγου θα βρει το αντίστοιχο UFID για το δεδομένο αρχείο και θα το επιστρέψει στον πελάτη. Επιπλέον, η υπηρεσία καταλόγου παρέχει την απαραίτητη λειτουργικότητα για διαχείριση των καταλόγων του συστήματος αρχείων, την προσθήκη νέων καταλόγων, αρχείων κ.α.

3.3.3 Εφαρμογή Πελάτη

Η εφαρμογή πελάτη τρέχει σε κάθε υπολογιστή χρήστη του κατανεμημένου συστήματος αρχείων. Ο ρόλος της είναι να παρέχει ένα περιβάλλον διεπαφής διαφανές προς το χρήστη υλοποιώντας τις διαδικασίες πάνω σε αρχεία με τέτοιο τρόπο ώστε ο

χρήστη να έχει την αίσθηση ότι αλληλεπιδρά με το τοπικό σύστημα αρχείων. Τέλος, η εφαρμογή πελάτη είναι αυτή που αναλαμβάνει να φέρει σε πέρας την επικοινωνία με τους εξυπηρετητές του κατανεμημένου συστήματος αρχείων.

3.4 Μελέτες Περίπτωσης

3.4.1 Sun Network File System

Το κατανεμημένο σύστημα αρχείων Sun NFS [24, 22] (ή απλά NFS όπως έχει καθιερωθεί) υλοποιεί το πρωτόκολλο NFS [23] που περιλαμβάνει ένα σύνολο απομακρυσμένες κλήσεις διαδικασίας (RPC) στον εξυπηρετητή του συστήματος αρχείων. Αν και η αρχική του υλοποίηση ήταν προορισμένη για UNIX συστήματα, το NFS πλέον υποστηρίζει ένα μεγάλο εύρος διαδεδομένων λειτουργικών συστημάτων, από το MS-DOS μέχρι το VMS [26].

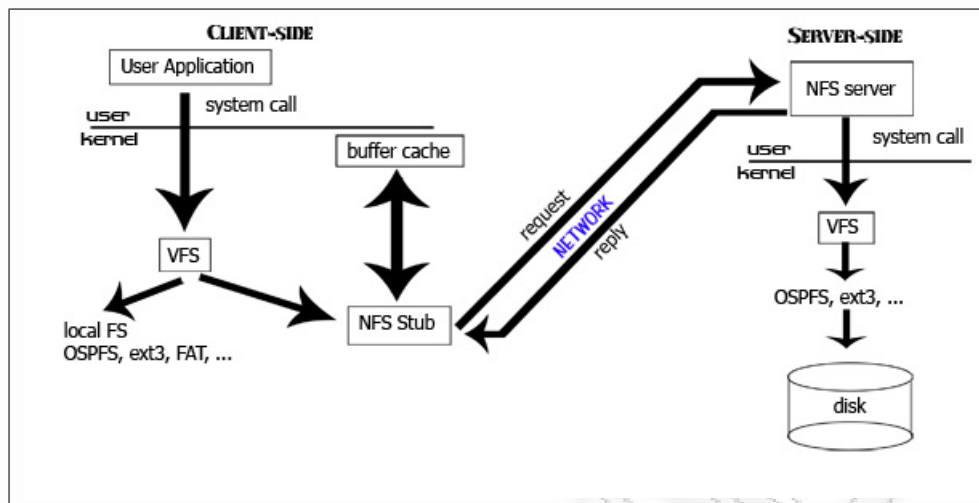
Στο σχήμα επικοινωνίας του NFS, κάθε υπολογιστής που διαμοιράζει κάποια αρχεία του, θα πρέπει να τρέχει τον εξυπηρετητή NFS. Ο πελάτης, για κάθε αίτηση που φτάνει από το χρήστη, εξακριβώνει σε ποιον εξυπηρετητή πρέπει να απευθυνθεί για να κάνει τη ζητούμενη λειτουργία και στη συνέχεια πραγματοποιεί την αντίστοιχη απομακρυσμένη κλήση διαδικασίας.

Το πρωτόκολλο επικοινωνίας που χρησιμοποιείται μπορεί να είναι είτε TCP είτε UDP, ενώ το πρωτόκολλο επικοινωνίας είναι έτσι σχεδιασμένο ώστε ο κάθε εξυπηρετητής να μην κρατά κατάσταση για κάθε πελάτη και έτσι να είναι πιο εύκολη η ανάκαμψη από βλάβη.

Η ενσωμάτωση του απομακρυσμένου συστήματος αρχείων στον εκάστοτε υπολογιστή γίνεται με χρήση του εικονικού συστήματος αρχείων (VFS). Συγκεκριμένα, το NFS παρέχει σύνδεση με το VFS μέσω ενός προγράμματος που αναλαμβάνει να χειρίζεται τα αναγνωριστικά των απομακρυσμένων αρχείων με διαφάνεια ως προς το τοπικό σύστημα αρχείων. Ένα ενδεικτικό σχήμα της λειτουργίας του NFS δίνεται στην Εικόνα 3.1.

Καθώς μιλάμε για ένα κατανεμημένο σύστημα αρχείων, όπου στον υπολογιστή του πελάτη μπορεί να βρίσκονται προσαρτημένοι κατάλογοι από διάφορες υπολογιστές όπου τρέχουν στιγμιότυπα του εξυπηρετητή, δεν μπορεί να χρησιμοποιείται ο αριθμός ευρετηρίου που δίνεται από τον αντίστοιχο υπολογιστή για κάθε αρχείο για την προσπέλασή του, καθώς κάτι τέτοιο θα μπορούσε να οδηγήσει σε σύγχυση.

Για το λόγο αυτό, το NFS διατηρεί εσωτερικά ξεχωριστά αναγνωριστικά για τα αρχεία που χειρίζεται. Τα αναγνωριστικά αυτά προκύπτουν συνδυάζοντας τον πραγ-



Εικόνα 3.1: Δομή του NFS [17].

ματικό αριθμό ευρετηρίου του κάθε αρχείου με δύο ακόμα πεδία: το αναγνωριστικό του συστήματος αρχείων και έναν αριθμό που παράγεται από το NFS εν είδει i-node.

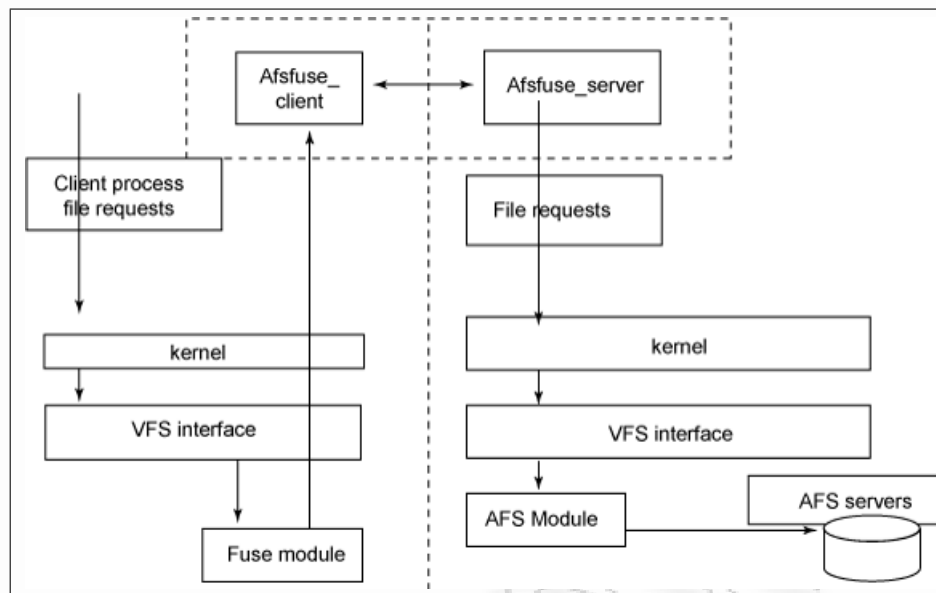
Αν και ο πελάτης του NFS χρησιμοποιεί κρυφή μνήμη, σε περίπτωση που θεωρήσουμε ότι δεν υπάρχει, η σημειολογία του πρωτοκόλλου επιβάλλει για κάθε πράξη πάνω σε αρχεία του συστήματος να υπάρχει επικοινωνία με τον αντίστοιχο εξυπηρετητή. Ειδικά στις πράξεις εγγραφής, η απόφαση αν θα είναι write through ή write commit εξαρτάται από την υλοποίηση του πρωτοκόλλου RPC που χρησιμοποιείται για την επικοινωνία.

Το NFS αποτελεί ένα σχετικά γρήγορο κατανεμημένο σύστημα αρχείων. Ένα από τα μειονεκτήματά του (συνέπεια και της φροντίδας των δημιουργών του για ταχύτητα) είναι η μη διαχείριση πολλών αντιγράφων. Έτσι, αν αλλάζει ένα αρχείο από κάποιον πελάτη, αυτή η αλλαγή δεν είναι εμφανής στους υπόλοιπους παρά μόνο όταν ξαναγίνει ανάγνωση του αρχείου.

3.4.2 Andrew File System

Το Andrew File System (AFS) [20] ήρθε κατά βάση για να καλύψει την εν γένει αδυναμία του NFS για συνέπεια όταν απαιτείται υποστήριξη για πολλαπλά αντίγραφα ενός αρχείου σε πολλούς πελάτες. Απόρροια αυτής της λύσης ήταν και η υποστήριξη για αργά δίκτυα που προσέφερε το AFS, με έμφαση στα κινητά δίκτυα που έφερε ο απόγονός του, το Coda.

Στο AFS, η διεργασία του εξυπηρετητή ονομάζεται *Vice* ενώ η διεργασία πελάτη ονομάζεται *Venus*. Μία διαφορά με το NFS είναι ότι οι διεργασίες αυτές εκτελούνται



Εικόνα 3.2: Δομή του OpenAFS [6].

στο χώρο χρηστών και στις δύο πλευρές (πελάτης και εξυπηρετητής), με μια από τις διαδεδομένες υλοποιήσεις του, το OpenAFS [11] να χρησιμοποιεί το FUSE, όπως φαίνεται και στην Εικόνα 3.2.

Επίσης, το AFS διαφέρει από το NFS στον τρόπο με τον οποίο προσπελούνται τα αρχεία στον πελάτη. Έτσι, στον πελάτη AFS έχουμε μια μόνιμη κρυφή μνήμη, όπου αντιγράφεται το κάθε αρχείο όταν ζητείται από το σύστημα αρχείων και διατηρείται εκεί ακόμα και μετά από επανεκκινήσεις του υπολογιστή όπου τρέχει ο πελάτης. Οι όποιες αλλαγές πραγματοποιούνται πάνω στο τοπικό αντίγραφο του αρχείου - το αρχικό αρχείο στον εξυπηρετητή ενημερώνεται μόνο αφού κλείσει σε κάποιον πελάτη.

Το AFS για τη βελτίωση της απόδοσής του κάνει κάποιες υποθέσεις σχετικά με το μέγεθος των αρχείων και το πλήθος των αναφορών σε αυτά, οι σημαντικότερες των οποίων συνοψίζονται από τους Coulouris και λοιποί στο [18] ως εξής:

- Τα αρχεία είναι μικρά με τα περισσότερα να έχουν μέγεθος μικρότερο από 10 KB.
- Οι πράξεις ανάγνωσης είναι περίπου έξι φορές περισσότερες από τις πράξεις εγγραφής πάνω σε κάποιο αρχείο.
- Τα αρχεία προσπελούνται κατά βάση σειριακά και σπάνια τυχαία.
- Τα αρχεία τροποποιούνται μόνο από ένα χρήστη. Ακόμα και για περιπτώσεις

διαμοιραζόμενων αρχείων, θεωρούμε ότι ένας μόνο χρήστης μεταβάλλει το αρχείο.

- Τα αρχεία ζητούνται κατά ριπές, με κάθε αρχείο που ζητήθηκε πρόσφατα να είναι πολύ πιθανό να ξαναζητηθεί σύντομα.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

Κεφάλαιο 4

Το Πρότυπο MPEG-21

4.1 Εισαγωγή

Η ολοένα αυξανόμενη ζήτηση για πολυμεσικό περιεχόμενο έχει αυξήσει τις απαιτήσεις των εφαρμογών που το παρέχουν. Καλύτερη διαχείριση του περιεχομένου, διανομή του περιεχομένου με βάση τις δυνατότητες και τις απαιτήσεις των πελατών, προστασία της πνευματικής ιδιοκτησίας, διαχείριση δικαιωμάτων πρόσβασης στο περιεχόμενο είναι μόνο κάποιες από τις πτυχές που εισάγει το πρότυπο MPEG-21 μέσα από ένα σύνολο 20 τεχνικών κειμένων, από τα οποία άλλα έχουν προτυποποιηθεί και άλλα βρίσκονται υπό ανάπτυξη [9, 8].

Για την παρούσα εργασία χρησιμοποιήθηκε, με μικρές τροποποιήσεις για καλύτερη υποστήριξη του κατακευματισμένου συστήματος αρχείων EuterpeFS, το δεύτερο μέρος του προτύπου MPEG-21, Digital Item Declaration (DID), όπως αυτό ορίστηκε στο [21]. Σε αυτό το Κεφάλαιο θα γίνει μια παρουσίαση των βασικών σημείων του δεύτερου μέρους του προτύπου, που βασίζεται αποκλειστικά στο αντίστοιχο ISO standard.

Γενικά, το πρότυπο MPEG-21 περιλαμβάνει τα εξής μέρη [7]:

- Information technology -- Multimedia framework (MPEG-21) -- Part 1: Vision, Technologies and Strategy
- Information technology -- Multimedia framework (MPEG-21) -- Part 2: Digital Item Declaration
- Information technology -- Multimedia framework (MPEG-21) -- Part 3: Digital Item

Identification

- Information technology -- Multimedia framework (MPEG-21) -- Part 4: Intellectual Property Management and Protection Components
- Information technology -- Multimedia framework (MPEG-21) -- Part 5: Rights Expression Language
- Information technology -- Multimedia framework (MPEG-21) -- Part 6: Rights Data Dictionary
- Information technology -- Multimedia framework (MPEG-21) -- Part 7: Digital Item Adaptation
- Information technology -- Multimedia framework (MPEG-21) -- Part 8: Reference software
- Information technology -- Multimedia framework (MPEG-21) -- Part 9: File Format
- Information technology -- Multimedia framework (MPEG-21) -- Part 10: Digital Item Processing
- Information technology -- Multimedia framework (MPEG-21) -- Part 11: Evaluation Tools for Persistent Association Technologies
- Information technology -- Multimedia framework (MPEG-21) -- Part 12: Test Bed for MPEG-21 Resource Delivery
- Information technology -- Multimedia framework (MPEG-21) -- Part 14: Conformance Testing
- Information technology -- Multimedia framework (MPEG-21) -- Part 15: Event Reporting
- Information technology -- Multimedia framework (MPEG-21) -- Part 16: Binary Format
- Information technology -- Multimedia framework (MPEG-21) -- Part 17: Fragment Identification of MPEG Resources
- Information technology -- Multimedia framework (MPEG-21) -- Part 18: Digital Item Streaming
- Information technology -- Multimedia framework (MPEG-21) -- Part 19: Media Value Chain Ontology

4.2 Βασικά Συστατικά

Σε αυτήν την Ενότητα παρουσιάζονται τα βασικά χαρακτηριστικά του DID μέρους του προτύπου MPEG-21. Το περιεχόμενο της Ενότητας είναι παρμένο από το [21] ενώ αναφέρεται ξεχωριστά σε κάθε σημείο οι αλλαγές που ενδεχομένως έχουν γίνει προκειμένου να υποστηριχθεί το EutereFS.

4.2.1 Container

Το στοιχείο container αποτελεί το υψηλότερο επίπεδο οργάνωσης των περιεχομένων του ψηφιακού αντικειμένου (DI). Όλα τα μέρη του DI που έπονται πρέπει να βρίσκονται μέσα σε κάποιο container. Έτσι, ένα container περιέχει:

- κανένα ή περισσότερα στοιχεία container
- κανένα ή περισσότερα στοιχεία item
- κανένα ή περισσότερα στοιχεία descriptor

Το αντίστοιχο XML σχήμα για ένα στοιχείο container είναι το ακόλουθο:

```
<xsd:element name="CONTAINER">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="DESCRIPTOR" minOccurs="0" maxOccurs="unbounded"
        />
      <xsd:choice>
<xsd:element ref="REFERENCE" />
<xsd:sequence>
  <xsd:element ref="CONTAINER" minOccurs="0" maxOccurs="unbounded" />
  <xsd:element ref="ITEM" minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
  </xsd:choice>
</xsd:sequence>
  <xsd:attributeGroup ref="ID_ATTRS" />
</xsd:complexType>
</xsd:element>
```

4.2.2 Item

Το στοιχείο item ομαδοποιεί ένα σύνολο από αντικείμενα (άλλα στοιχεία item) και στοιχεία component, με τα τελευταία να αποτελούν ουσιαστικό συστατικό τόσο του

προτύπου του MPEG-21 όσο και του κατανεμημένου συστήματος αρχείων EutergeFS.

Ένα item περιέχει:

- ένα ή περισσότερα στοιχεία item

είτε

- ένα ή περισσότερα στοιχεία component

και, προαιρετικά:

- κανένα ή περισσότερα στοιχεία choice
- κανένα ή περισσότερα στοιχεία descriptor
- κανένα ή περισσότερα στοιχεία condition
- κανένα ή περισσότερα στοιχεία annotation

Το αντίστοιχο XML σχήμα για ένα στοιχεία item είναι το ακόλουθο:

```
<xsd:element name="ITEM">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="CONDITION" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element ref="DESCRIPTOR" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:choice>
      <xsd:element ref="REFERENCE" />
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="CHOICE" />
        <xsd:element ref="ITEM" />
        <xsd:element ref="COMPONENT" />
      </xsd:choice>
    </xsd:choice>
    <xsd:element ref="ANNOTATION" minOccurs="0" maxOccurs="unbounded" />
  </xsd:complexType>
  <xsd:attributeGroup ref="ID_ATTRS" />
</xsd:element>
```


4.2.3 Component

Το στοιχείο component ουσιαστικά δένει ένα πόρο (στοιχείο resource) με αντίστοιχες περιγραφές του (στοιχεία descriptor). Οι περιγραφές αυτές, κατά το πρότυπο, δε θα πρέπει να έχουν να κάνουν με το περιεχόμενο του ίδιου του πόρου, αλλά με μετα-δεδομένα που σχετίζονται με δομικά στοιχεία του πόρου (όπως για παράδειγμα το ρυθμό μετάδοσης ή το σύνολο χαρακτήρων). Για το καταναμημένο σύστημα αρχείων EuterpeFS χρειάστηκε να γίνει μια επέκταση στο στοιχείο component προσθέτοντας μια ιδιότητα NAME σε αυτό. Η ιδιότητα αυτή αποτελεί στην ουσία το όνομα του αρχείου όπως θα φαίνεται στον τοπικό δίσκο του πελάτη. Καθώς μέσα στο component βρίσκεται όλη η πληροφορία που αφορά το αρχείο αυτό καθεαυτό, θεωρήθηκε σκόπιμο να τοποθετηθεί το όνομα εδώ. Έτσι, ένα component περιέχει:

- ακριβώς μια ιδιότητα name
- ακριβώς ένα στοιχείο resource
- κανένα ή περισσότερα στοιχεία descriptor
- κανένα ή περισσότερα στοιχεία anchor
- κανένα ή περισσότερα στοιχεία condition

Το αντίστοιχο XML σχήμα για ένα στοιχείο component είναι το ακόλουθο:

```
<xsd:element name="COMPONENT">
  <xsd:complexType>
    <xsd:attribute name="NAME" type="xsd:string" />
    <xsd:sequence>
      <xsd:element ref="CONDITION" minOccurs="0" maxOccurs="unbounded" />
    >
      <xsd:element ref="DESCRIPTOR" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:choice>
      <xsd:element ref="REFERENCE" />
      <xsd:element ref="RESOURCE" minOccurs="1" maxOccurs="unbounded" />
    </xsd:choice>
    <xsd:element ref="ANCHOR" minOccurs="0" maxOccurs="unbounded" />
  </xsd:complexType>
  <xsd:attributeGroup ref="ID_ATTRS" />
</xsd:element>
```

4.2.4 Anchor

Το στοιχείο anchor χρησιμοποιείται για να ενώσει στοιχεία descriptor με στοιχεία fragment, με τα τελευταία να παρουσιάζονται σε επόμενη παράγραφο. Ένα στοιχείο anchor περιέχει:

- ακριβώς ένα στοιχείο fragment
- κανένα ή περισσότερα στοιχεία descriptor
- κανένα ή περισσότερα στοιχεία condition

Το αντίστοιχο XML σχήμα για ένα στοιχείο anchor είναι το ακόλουθο:

```
<xsd:element name="ANCHOR">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="CONDITION" minOccurs="0" maxOccurs="unbounded" />
    >
      <xsd:element ref="DESCRIPTOR" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="PRECEDENCE" type="xsd:unsignedInt" use="default" value="0" />
    <xsd:attribute name="FRAGMENT" type="xsd:uriReference" />
    <xsd:attribute name="NAME" type="xsd:ID" />
  </xsd:complexType>
</xsd:element>
```

4.2.5 Descriptor

Το στοιχείο descriptor περικλείει πληροφορίες για το στοιχείο που το περιέχει. Η πληροφορία αυτή μπορεί να παρουσιάζεται σαν απλό κείμενο είτε να δίνεται μέσω κάποιου στοιχείου component, οπότε στην τελευταία περίπτωση μπορεί να πάρει την οποιαδήποτε μορφή. Ένα στοιχείο descriptor περιέχει:

- κανένα ή περισσότερα στοιχεία descriptor
- ακριβώς ένα:
 - στοιχείο component, ή
 - στοιχείο statement
- κανένα ή περισσότερα στοιχεία condition

Το αντίστοιχο XML σχήμα για ένα στοιχείο descriptor είναι το ακόλουθο:

```
<xsd:element name="DESCRIPTOR">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="CONDITION" minOccurs="0" maxOccurs="unbounded" />
    >
      <xsd:element ref="DESCRIPTOR" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:choice>
      <xsd:element ref="REFERENCE" />
      <xsd:element ref="COMPONENT" />
      <xsd:element ref="STATEMENT" />
    </xsd:choice>
  </xsd:complexType>
  <xsd:attributeGroup ref="ID_ATTRS" />
</xsd:element>
```

4.2.6 Condition

Το στοιχείο condition χρησιμοποιείται για να σημειώσει ότι το στοιχείο που το περιέχει είναι προαιρετικό, με την έννοια ότι εξαρτάται από κάποιο άλλο στοιχείο, το οποίο τίθεται από τα στοιχεία selection που θα παρουσιαστούν παρακάτω. Εδώ μπορεί να έχουμε συνδυασμό condition τόσο με σύνδεση AND (τελεστές μέσα στο ίδιο στοιχείο condition) όσο και με σύνδεση OR (πολλά condition). Ένα στοιχείο condition περιέχει:

- ένα ή περισσότερα στοιχεία predicate

Το αντίστοιχο XML σχήμα για ένα στοιχείο condition είναι το ακόλουθο:

```
<xsd:element name="CONDITION">
  <xsd:complexType>
    <xsd:attribute name="REQUIRE" type="xsd:IDREFS" />
    <xsd:attribute name="EXCEPT" type="xsd:IDREFS" />
  </xsd:complexType>
</xsd:element>
```

4.2.7 Choice

Το στοιχείο choice ουσιαστικά ομαδοποιεί ένα σύνολο από στοιχεία selection που αποτελούν και τους περιορισμούς στην πρόσβαση σε πόρους ενός στοιχείου item.

Στο EuterpeFS το στοιχείο choice είναι ο περιέχων κατάλογος που περιέχει τους υποκαταλόγους που ορίζονται από τα στοιχεία selection που ορίζονται στην επόμενη Παράγραφο. Ένα στοιχείο choice περιέχει:

- ένα ή περισσότερα στοιχεία selection
- κανένα ή περισσότερα στοιχεία descriptor
- κανένα ή περισσότερα στοιχεία condition

Το αντίστοιχο XML σχήμα για ένα στοιχείο choice είναι το ακόλουθο:

```
<xsd:element name="CHOICE">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="CONDITION" minOccurs="0" maxOccurs="unbounded" />
    >
      <xsd:element ref="DESCRIPTOR" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="MIN_SELECTIONS" type="xsd:nonNegativeInteger" />
    <xsd:attribute name="MAX_SELECTIONS" type="xsd:nonNegativeInteger" />
    <xsd:attribute name="DEFAULT" type="xsd:IDREFS" />
    <xsd:attribute name="CHOICE_ID" type="xsd:ID" />
  </xsd:complexType>
</xsd:element>
```

4.2.8 Selection

Το στοιχείο selection είναι άμεσα συνδεδεμένο με τα στοιχεία choice και condition. Στην ουσία, εδώ είναι που ορίζονται οι συνθήκες, οι οποίες κατόπιν ελέγχονται σε διάφορα σημεία με βάση τα στοιχεία condition. Για το EuterpeFS, το στοιχείο selection αποτελεί τον κατάλογο γονιό του πόρου που έχει αυτό το στοιχείο ως condition. Ένα στοιχείο selection περιέχει:

- ακριβώς ένα στοιχείο predicate. Ουσιαστικά το predicate δεν έχει κάποια φυσική υπόσταση. Εδώ εννοούμε ότι το selection παίρνει μια τιμή αληθής/ψευδής και η οποία απεικονίζεται αφαιρετικά από το predicate.
- κανένα ή περισσότερα στοιχεία descriptor

- κανένα ή περισσότερα στοιχεία condition

Το αντίστοιχο XML σχήμα για ένα στοιχείο selection είναι το ακόλουθο:

```
<xsd:element name="SELECTION">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="CONDITION" minOccurs="0" maxOccurs="unbounded" />
    >
      <xsd:element ref="DESCRIPTOR" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
      <xsd:element ref="OVERRIDE" minOccurs="0" maxOccurs="unbounded" />
    </xsd:complexType>
      <xsd:attribute name="SELECT_ID" type="xsd:ID" use="required" />
    </xsd:element>
```

4.2.9 Annotation

Τα στοιχεία annotation χρησιμοποιούνται για να παράσχουν πληροφορίες πάνω σε οποιοδήποτε στοιχείο του ψηφιακού αντικειμένου, χωρίς να έχουν κάποια επίδραση στο περιεχόμενο. Ένα στοιχείο annotation περιέχει:

- κανένα ή περισσότερα στοιχεία assertion
- κανένα ή περισσότερα στοιχεία descriptor
- κανένα ή περισσότερα στοιχεία anchor

Το αντίστοιχο XML σχήμα για ένα στοιχείο annotation είναι το ακόλουθο:

```
<xsd:element name="ANNOTATION">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="REFERENCE" />
    <xsd:sequence>
      <xsd:element ref="ASSERTION" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element ref="DESCRIPTOR" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element ref="ANCHOR" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    </xsd:choice>
      <xsd:attribute name="TARGET" type="xsd:IDREF" use="required" />
      <xsd:attributeGroup ref="ID_ATTRS" />
    </xsd:complexType>
  </xsd:element>
```

4.2.10 Assertion

Το στοιχείο `assertion` χρησιμοποιείται για να καθορίσει μια κατάσταση που μπορεί να βρίσκεται κάποιο στοιχείο `choice`. Αυτό επιτυγχάνεται δίνοντας τιμές στο λογικό στοιχείο `predicate` που είναι συνδεδεμένο με κάθε `selection` του στοιχείου `choice`. Ένα στοιχείο `assertion` περιέχει:

- κανένα ή περισσότερα στοιχεία `predicate`

Το αντίστοιχο XML σχήμα για ένα στοιχείο `assertion` είναι το ακόλουθο:

```
<xsd:element name="ASSERTION">
  <xsd:complexType>
    <xsd:attribute name="TARGET" type="xsd:IDREF" use="required" />
    <xsd:attribute name="TRUE" type="xsd:IDREFS" use="optional" />
    <xsd:attribute name="FALSE" type="xsd:IDREFS" use="optional" />
  </xsd:complexType>
</xsd:element>
```

4.2.11 Resource

Το στοιχείο `resource` είναι ίσως το πιο ουσιώδες του μοντέλου, καθώς χρησιμοποιείται για να καθορίσει τον ίδιο τον πόρο, παρέχοντας ταυτόχρονα μια μοναδική διεύθυνση όπου αυτός ο πόρος μπορεί να βρίσκεται. Στο `EuterpeFS`, η διεύθυνση αυτή αντιστοιχεί στο URL που βρίσκεται το διαμοιραζόμενο αρχείο.

Για την υποστήριξη του `EuterpeFS`, το στοιχείο `Resource` επεκτάθηκε ώστε να περιλαμβάνει και ένα γνώρισμα `SIZE`, το οποίο διατηρεί ο εξυπηρετητής και αντιστοιχεί στο μέγεθος του αρχείου που αντιπροσωπεύεται με αυτό το `Resource`. Σημειώνεται ότι αν και απαραίτητο ως προς την ύπαρξή του, το στοιχείο `Resource` δε χρειάζεται να έχει κάποια αρχική τιμή, καθώς ενημερώνεται από τον εξυπηρετητή του `EuterpeFS`.

Το αντίστοιχο XML σχήμα για ένα στοιχείο `resource` είναι το ακόλουθο:

```
<xsd:element name="RESOURCE">
  <xsd:complexType mixed="true">
    <xsd:attribute name="TYPE" type="xsd:uriReference" />
    <xsd:attribute name="REF" type="xsd:uriReference" />
    <xsd:attribute name="LOCALPATH" type="xsd:uriReference" />
    <xsd:attribute name="SIZE" type="xsd:unsignedInt" />
  </xsd:complexType>
</xsd:element>
```

4.2.12 Fragment

Το στοιχείο fragment υποδεικνύει ένα συγκεκριμένο σημείο ή ένα εύρος μέσα στο στοιχείο resource και μπορεί να διαφοροποιείται ανάλογα με τον τύπο του στοιχείου resource. Έτσι, για παράδειγμα, το fragment θα μπορούσε να ορίζει μόνο κάποια κεφάλαια ενός βιβλίου που μπορεί να δει ενδεικτικά ο χρήστης που δεν το έχει αγοράσει.

Το αντίστοιχο XML σχήμα για ένα στοιχείο fragment είναι το ακόλουθο:

```
<element name="Fragment" type="didmodel:FragmentType" abstract="true" />
<complexType name="FragmentType" abstract="true">
  <complexContent>
    <extension base="didmodel:DIDBaseType" />
  </complexContent>
</complexType>
```

4.2.13 Statement

Το στοιχείο statement ορίζει ένα κείμενο που χρησιμοποιείται για να παρέχει πληροφορίες για το στοιχείο που το περιέχει.

Το αντίστοιχο XML σχήμα για ένα στοιχείο statement είναι το ακόλουθο:

```
<xsd:element name="STATEMENT">
  <xsd:complexType mixed="true">
    <xsd:sequence>
      <xsd:any namespace="##any" processContents="skip" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="TYPE" type="xsd:uriReference" />
  </xsd:complexType>
</xsd:element>
```

4.2.14 Predicate

Το στοιχείο predicate αντιπροσωπεύει ουσιαστικά μια λογική μεταβλητή που λαμβάνει τιμές αληθής / ψευδής / μη-ορισμένη.

Δεν υπάρχει κάποιο συγκεκριμένο XML σχήμα για το στοιχείο predicate, καθώς πρόκειται για μια αφηρημένη έννοια. Έτσι, για παράδειγμα, predicate θεωρείται το γνώρισμα REQUIRE ή το γνώρισμα EXCEPT στο στοιχείο condition.

Κεφάλαιο 5

EuterpeFS

5.1 Εισαγωγή

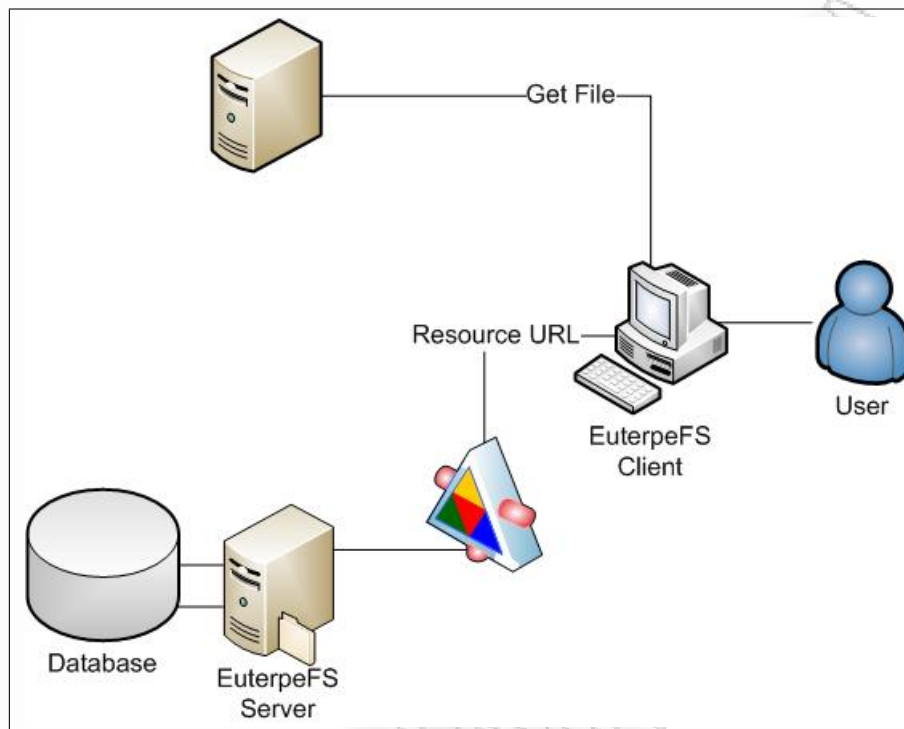
Το EuterpeFS είναι ένα κατακευματισμένο σύστημα αρχείων που βασίζεται στο πρότυπο MPEG-21 και έχει ως στόχο να παράσχει απομακρυσμένη πρόσβαση σε δεδομένα. Εκμεταλλεύομενη τις δυνατότητες που παρέχει ο ορισμός ενός ψηφιακού αντικειμένου από το MPEG-21, η υπηρεσία αρχείων και καταλόγων του EuterpeFS δε χρειάζεται να έχει την ίδια φυσική τοποθεσία με αυτή των αρχείων. Έτσι, το EuterpeFS μπορεί να λειτουργήσει σαν ένα υπερδίκτυο (overlay) που να χρησιμοποιεί υποκείμενα πρωτόκολλα για τη μεταφορά αρχείων.

Η καινοτομία που επιχειρεί να εισάγει το EuterpeFS στα κατακευματισμένα συστήματα αρχείων είναι, αφενός η καλύτερη οργάνωση των δεδομένων κάνοντας χρήση του προτύπου MPEG-21 και, αφετέρου, η ανεξαρτησία από τον τρόπο με τον οποίο ο τελικός χρήστης θα φτάσει στην πληροφορία καθώς το σύστημα αρχείων δεν είναι υπεύθυνο για τη μεταφορά του ίδιου του αρχείου, αλλά της πληροφορίας, η οποία είναι απαραίτητη για να φτάσει ο χρήστης στο αρχείο.

5.2 Αρχιτεκτονική

Το EuterpeFS έχει πολλαπλούς στόχους. Σε αυτήν την Ενότητα, θα παρουσιαστούν αυτοί οι στόχοι μαζί με τις τεχνολογίες που χρησιμοποιήθηκαν για να πραγματοποιηθούν αυτοί οι στόχοι και τις λειτουργίες που τελικά παρέχει το σύστημα αρχείων τη στιγμή που γράφεται αυτό κείμενο. Η Εικόνα 5.1 δείχνει μια αφαιρετική προσέγγιση

της αρχιτεκτονικής του EuterpeFS.



Εικόνα 5.1: Αρχιτεκτονική EuterpeFS.

5.2.1 Επισκόπηση Αρχιτεκτονικής

Η αρχιτεκτονική του EuterpeFS διαφέρει σε φιλοσοφία από άλλες αρχιτεκτονικές κατανεμημένων συστημάτων αρχείων που απαντώνται στη βιβλιογραφία, όπως αυτά που παρουσιάστηκαν στο Κεφάλαιο 1. Βασικός στόχος της αρχιτεκτονικής του είναι να καλύψει προβλήματα οργάνωσης των αρχείων, τα οποία όσο μεγαλώνει ο αριθμός των αρχείων γίνονται όλο και περισσότερο έντονα και να δώσει τη δυνατότητα σε κατόχους υπολογιστών περιορισμένων δυνατοτήτων να μπορούν να έχουν απομακρυσμένη πρόσβαση στα αρχεία τους, πολλές φορές χωρίς αυτά να είναι καν αποθηκευμένα σε δικό τους υπολογιστή.

Το EuterpeFS επιτρέπει στο χρήστη να βλέπει οποιοδήποτε αρχείο έχει κάποιο Uniform Resource Locator (URL). Έτσι, ένα αρχείο του EuterpeFS θα μπορούσε να είναι ένα αρχείο σε κάποιο δίσκο στο οποίο δίνεται πρόσβαση μέσω κάποιου άλλου κατανεμημένου συστήματος αρχείων (π.χ. Sun NFS), μια ιστοσελίδα, κάποιο αρχείο σε μια ηλεκτρονική βιβλιοθήκη (π.χ. ένα επιστημονικό άρθρο σε μια διαδικτυακή ψηφιακή βιβλιοθήκη) και οτιδήποτε άλλο μπορεί να είναι προσβάσιμο μέσα από ένα παρεχόμενο URL.

Η υπηρεσία αρχείων του συστήματος παρέχει διαφάνεια πρόσβασης ώστε ο χρήστης να έχει την αίσθηση ότι προσπελαύνει ένα τοπικό αρχείο, ενώ στην πραγματικότητα η μόνη πληροφορία που κρατείται τοπικά είναι η τοποθεσία του αρχείου καθώς και κάποιες επιπλέον πληροφορίες που έχουν να κάνουν με τον τοπική απεικόνιση του αρχείου (απαραίτητες από το λειτουργικό σύστημα). Για να μπορέσει τελικά ο χρήστης να φτάσει στο αρχείο μέσω του URL, το EuterpeFS κρατά δομές με αντιστοιχίες πρωτοκόλλων με δυναμικά φορτωνόμενα προγράμματα (plugins), τα οποία στην πραγματικότητα είναι αυτά που οδηγούν τελικά στα αρχεία.

Ο εξυπηρετητής αρχείων του EuterpeFS αρκεί να διατηρεί δομές με τις απαραίτητες πληροφορίες που χρειάζεται ο πελάτης (τοποθεσία αρχείου, μέγεθος, τύπος) καθώς και τα απαραίτητα μεταδεδομένα. Τα τελευταία χωρίζονται σε δύο κατηγορίες: εκείνα τα οποία χρησιμοποιούνται από την υπηρεσία καταλόγων και εκείνα που προσδιορίζουν σημασιολογικά τα αρχεία.

Τα μεταδεδομένα που χρησιμοποιούνται από την υπηρεσία καταλόγου είναι εκείνα που ορίζονται από το ISO/IEC MPEG-21 Digital Item Definition Language (DIDL) [21] και με τον τρόπο που αξιοποιούνται από το EuterpeFS δίνουν τη δυνατότητα για διαφορετικά μονοπάτια πρόσβασης στο ίδιο αρχείο. Για να το καταφέρει αυτό, το EuterpeFS αντιλαμβάνεται τους καταλόγους ως συνθήκες που πρέπει να ισχύουν ώστε να φτάσει ο χρήστης στο κάθε αρχείο. Το MPEG-21 DIDL παρέχει έναν επίσημο, καθιερωμένο και προτυποποιημένο τρόπο για την αναπαράσταση αυτών των συνθηκών και αυτός ήταν και ο πρωταρχικός λόγος που επελέγει.

Η δεύτερη κατηγορία μεταδεδομένων χρησιμοποιείται για τη σημασιολογική περιγραφή των αρχείων στον εξυπηρετητή. Αυτή η λειτουργικότητα δεν αξιοποιείται ακόμα, αλλά αποτελεί μια από τις πρώτες και σημαντικότερες μελλοντικές επεκτάσεις του συστήματος και αφορά τη σημασιολογική αναζήτηση των αρχείων. Εκμεταλλευόμενο αυτά τα μεταδεδομένα, το EuterpeFS θα μπορεί να πραγματοποιήσει γρήγορες σημασιολογικές αναζητήσεις, αφού, σε σχέση με τα παραδοσιακά συστήματα αρχείων, όλα τα μεταδεδομένα είναι συγκεντρωμένα και αναμένεται να υπάρξει σημαντική βελτίωση στην απόδοση.

5.2.2 Λειτουργίες

Σε αυτήν την Ενότητα θα παρουσιαστούν οι λειτουργίες του κατανεμημένου συστήματος αρχείων EuterpeFS, ακολουθώντας σε γενικές γραμμές τη σημειολογία που χρησιμοποιείται για την περιγραφή της αρχιτεκτονικής των συστημάτων αρχείων, όπως έγινε και στην Ενότητα 3.3.

Υπηρεσία Καταλόγων

Όπως αναφέρθηκε και σε προηγούμενη Ενότητα, η υπηρεσία καταλόγων βασίζεται στο πρότυπο MPEG-21 και στον τρόπο με τον οποίο οργανώνονται οι πόροι που αντιπροσωπεύει ένα ψηφιακό αντικείμενο. Συγκεκριμένα, το EuterpeFS βασίζεται στην παραδοχή ότι ο χρήστης τοποθετεί τα αρχεία μέσα σε καταλόγους με βάση τις προϋποθέσεις που απαιτούνται προκειμένου να χρησιμοποιήσει το κάθε αρχείο. Εδώ να σημειωθεί ότι οι προϋποθέσεις αυτές δεν έχουν να κάνουν με την ασφάλεια, αλλά περισσότερο με το περιεχόμενο των αρχείων.

Το EuterpeFS θεωρεί ότι ο κόμβος που προσαρτάται αντιστοιχεί σε ένα ψηφιακό αντικείμενο με ό,τι αυτό περιέχει. Τα στοιχεία choice και selection αποτελούν την ιεραρχική δομή καταλόγων του συστήματος αρχείων, ενώ τα παιδιά του κάθε καταλόγου υπολογίζονται με βάση το στοιχείο condition. Έτσι, για παράδειγμα, ας θεωρήσουμε το ακόλουθο κομμάτι από ένα ψηφιακό αντικείμενο:

```
<COMPONENT NAME=" Heart_And_Soul ">
  <CONDITION REQUIRE=" HIGH_BITRATE " />
  <RESOURCE REF=" http://www.youtube.com/watch?v=dZfFgQDy38I" TYPE="
    audio/mp3" SIZE="" />
</COMPONENT>
```

Σε αυτήν την περίπτωση, το αρχείο με όνομα Heart_And_Soul θα βρίσκεται μέσα στον κατάλογο HIGH_BITRATE. Το HIGH_BITRATE πρόκειται ουσιαστικά για κάποιο στοιχείο selection, το οποίο με τη σειρά του ανήκει σε κάποιο στοιχείο choice, με το τελευταίο να αποτελεί και το γονιό του καταλόγου HIGH_BITRATE.

Έτσι, όταν ζητηθούν από το EuterpeFS τα περιεχόμενα του καταλόγου ρίζα (root directory), γίνεται αναζήτηση μέσα στο ψηφιακό αντικείμενο για να βρεθούν όλα τα στοιχεία που δεν έχουν εμφωλευμένο κάποιο στοιχείο condition. Τα στοιχεία selection δε θα παρουσιαστούν σαν ξεχωριστοί κατάλογοι, καθώς αποτελούν υποκαταλόγους του στοιχείου choice στο οποίο είναι εμφωλευμένα.

Όταν ζητηθεί οποιοσδήποτε άλλος κατάλογος, το EuterpeFS, εφόσον το όνομα που ζητείται είναι κάποιο στοιχείο choice, επιστρέφει όλα τα στοιχεία selection που ομαδοποιούνται υπό αυτό το choice ως καταλόγους. Αν το όνομα που ζητάτε είναι κάποιο selection, τότε γίνεται αναζήτηση μέσα στο ψηφιακό αντικείμενο για στοιχεία choice, selection, item, component που μπορεί να έχουν αυτό το selection ως condition. Σε αυτήν την περίπτωση, τα choice, selection επιστρέφονται ως κατάλογοι, ενώ το component ως το αρχείο.

Εδώ πρέπει να γίνει μια παρατήρηση σχετικά με τα στοιχεία item: Είναι πολύ συχνό το φαινόμενο ένα στοιχείο item να περιέχει πολλά components. Ένα τέτοιο παράδειγμα θα μπορούσε να είναι ένα τραγούδι (item) που διατίθεται με διάφορους

ρυθμούς bit (bit rate) ή ένα κείμενο που διατίθεται σε διάφορες διαμορφώσεις (MS Word, Adobe PDF, TXT, κ.α.). Σε αυτήν την περίπτωση το EutereFS θεωρεί ότι το κάθε component κληρονομεί όλες τις ιδιότητες και τους περιορισμούς του item. Συνεπώς, αν υπάρχει κάποιο condition για το item το ίδιο condition θα ισχύει και για όλα τα περιεχόμενα στοιχεία component.

Υπηρεσία Αρχείων

Η υπηρεσία αρχείων του EutereFS, όπως παρουσιάστηκε και στην Ενότητα 3.3, αφορά τις λειτουργίες πάνω στα αρχεία αυτά καθαυτά. Σημειώνεται ότι η ανακάλυψη των αρχείων δεν είναι αρμοδιότητα αυτής της υπηρεσίας, αλλά της υπηρεσίας καταλόγου, όπως παρουσιάστηκε στην προηγούμενη Ενότητα.

Μια θεμελιώδης διαφορά της υπηρεσίας αρχείων του EutereFS σε σχέση με τη γενική μεθοδολογία που ακολουθείται είναι η μη ανάθεση UFID (βλ. Ενότητα 3.3) στα αρχεία που χειρίζεται η υπηρεσία. Αυτό συμβαίνει διότι δε θεωρείται απαραίτητο να ακολουθηθεί αυτή η διαδικασία καθώς το μοναδικό αναγνωριστικό του κάθε αρχείου είναι η ίδια η πραγματική τοποθεσία του που δίνεται από το URL του. Αν υπάρχει κι άλλο αρχείο με αυτό το URL δε θα θέλαμε να έχει διαφορετικό αναγνωριστικό καθώς κάτι τέτοιο θα ερχόταν σε αντίθεση με τη φιλοσοφία του EutereFS ως προς την οργάνωση των αρχείων.

Συγκεκριμένα, σε περίπτωση που είχαμε διαφορετικό UFID για κάθε αρχείο, όπου αρχείο εννοούμε ολόκληρη τη διαδρομή (μαζί με τους καταλόγους) προς αυτό, τότε ανατρέπεται όλη η φιλοσοφία του EutereFS να οδηγεί στο ίδιο αρχείο από εναλλακτικές διαδρομές.

Καθώς το EutereFS δεν είναι ένα τυπικό καταναμημένο σύστημα αρχείων, με την έννοια ότι λειτουργεί σαν ένα overlay για οποιοδήποτε σύστημα αρχείων, η υπηρεσία αρχείων μπορεί να έχει ποικίλες υλοποιήσεις και να μην υποστηρίζει, απαραίτητα, όλες τις λειτουργίες που αναφέρθηκαν στην Ενότητα 3.3. Η μόνη λειτουργικότητα που υποστηρίζεται σίγουρα είναι αυτή της λήψης των χαρακτηριστικών (attributes) και της φυσικής τοποθεσίας ενός αρχείου.

Κάποια στοιχεία που απαιτούνται για την υποστήριξη αυτών των λειτουργιών διατηρούνται μέσα στο ψηφιακό αντικείμενο (όπως το μέγεθος του αρχείου) και είναι ευθύνη της υπηρεσίας αρχείων το να υπάρχουν εκεί. Αυτό επιτυγχάνεται μέσω της επέκτασης του προτύπου MPEG-21 που έχει γίνει για την υποστήριξη ύπαρξης πληροφορίας για το μέγεθος του αρχείου (κάτι που είναι απαραίτητο για την πλευρά του πελάτη) και παρουσιάστηκε στην Ενότητα 4.2.11. Από την άλλη στοιχεία που προκύπτουν έμμεσα (όπως το αν μεταφέρεται αρχείο ή κατάλογος) επισυνάπτονται στα μηνύματα που ανταλλάσσονται μεταξύ εφαρμογής πελάτη και υπηρεσίας αρχείων

και δε διατηρούνται στο ψηφιακό αντικείμενο.

Το EuterpeFS υποστηρίζει την κατ' απαίτηση μεταφορά των δεδομένων. Έτσι, όταν διαβάζεται ένας κατάλογος, δε μεταφέρονται και τα δεδομένα των αρχείων. Αρκεί μόνο να φανεί ότι υπάρχουν (να δημιουργηθεί δηλαδή η δομή stat του λειτουργικού συστήματος, όπως παρουσιάστηκε και στην Ενότητα 2.6). Τα δεδομένα θα μεταφερθούν, μόνο εφόσον ζητηθεί από το χρήστη και η μεταφορά θα γίνει από τη φυσική τοποθεσία του αρχείου, η οποία παρέχεται μέσω του URL του. Ευθύνη, δηλαδή, της υπηρεσίας αρχείων είναι να ενημερώσει για το πού βρίσκεται το αρχείο και όχι να το παρέχει απευθείας στην εφαρμογή πελάτη.

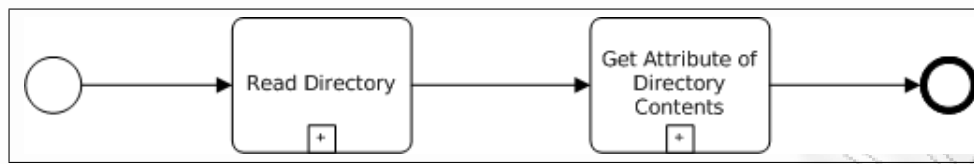
Το σχήμα αυτό προτιμήθηκε από μια υπηρεσία αρχείων που θα παρέχει απευθείας τα αρχεία στο χρήστη για λόγους απόδοσης. Σε αντίθετη περίπτωση, θα έπρεπε η υπηρεσία αρχείων να διατηρεί ενημερωμένα αντίγραφα όλων των αρχείων που εξυπηρετεί και να παράσχει απευθείας τα αρχεία στο χρήστη. Αυτή η προσέγγιση έχει δύο εγγενείς αδυναμίες: Αφενός μεγάλο φόρτο του εξυπηρετητή που θα πρέπει να διατηρεί τα ενημερωμένα αντίγραφα των αρχείων και, αφετέρου, μεγάλες απαιτήσεις σε εύρος ζώνης μεταξύ πελάτη και εξυπηρετητή, αφού ο εξυπηρετητής θα πρέπει να είναι σε θέση να στέλνει μεγάλο όγκο δεδομένων παράλληλα σε πολλούς πελάτες.

Όσον αφορά το είδος πρόσβασης σε κάθε αρχείο (mode), αυτή τη στιγμή το EuterpeFS υποστηρίζει μόνο ανάγνωση αρχείων και, συνεπώς, τα αρχεία διατίθενται μόνο για ανάγνωση στο χρήστη που προσάρτησε το σύστημα αρχείων στον τοπικό του κατάλογο. Η εγγραφή και η διαγραφή αρχείων και καταλόγων βρίσκονται ακόμα σε στάδιο δοκιμών. Με την ενσωμάτωσή τους, αναμένεται να γίνει και προσθήκη αντίστοιχων εγγραφών στο στοιχείο component (βλ. Ενότητα 4.2.3) για να ενημερώνεται η εφαρμογή του πελάτη και για άλλα χαρακτηριστικά του αρχείου.

Εφαρμογή Πελάτη

Η εφαρμογή πελάτη είναι επιφορτισμένη με το να παράσχει διαφανώς τις υπηρεσίες καταλόγου και αρχείων στον πελάτη. Ο τελικός χρήστης του συστήματος δε θα πρέπει να έχει επίγνωση του ότι λειτουργεί πάνω από ένα κατανεμημένο σύστημα αρχείων.

Στην Εικόνα 5.2 δίνεται ένα διάγραμμα BPMN που απεικονίζει αφαιρετικά την ενορχήστρωση της εφαρμογής πελάτη για την ανάκτηση ενός απομακρυσμένου καταλόγου. Όπως φαίνεται και από το διάγραμμα, αρχικά πραγματοποιείται μια υποδιαδικασία για την ανάγνωση του καταλόγου και στη συνέχεια μια υποδιαδικασία για την ανάγνωση των βασικών γνωρισμάτων των περιεχομένων του καταλόγου - γνωρίσματα απαραίτητα για το λειτουργικό σύστημα.



Εικόνα 5.2: Αφαιρετική Ενορχήστρωση Ανάκτησης Απομακρυσμένου Καταλόγου.

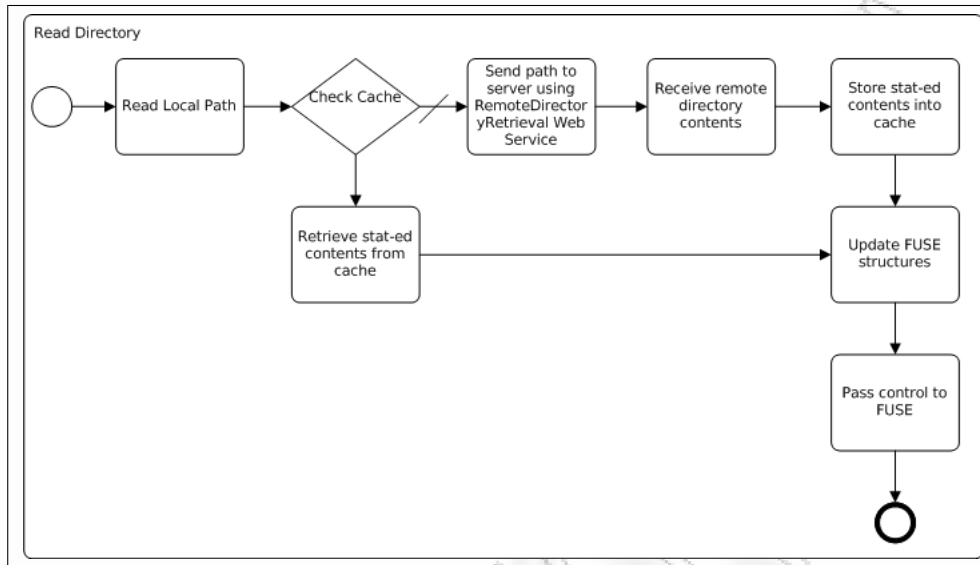
Η υποδιαδικασία για την ανάγνωση του καταλόγου φαίνεται στο BPMN διάγραμμα της Εικόνας 5.3. Αρχικά, η υπηρεσία πελάτη διαβάζει τη διαδρομή του τρέχοντος καταλόγου και αναζητά αντίστοιχη εγγραφή στην κρυφή μνήμη. Εδώ να σημειωθεί ότι η κρυφή μνήμη διατηρείται από ανεξάρτητη διεργασία, η οποία και αναλαμβάνει να την κρατά ενημερη. Επίσης, στην κρυφή μνήμη διατηρούνται τα περιεχόμενα των καταλόγων καθώς και τα μεταδεδομένα που απαιτεί το λειτουργικό σύστημα (στοιχεία της δομής stat). Σε περίπτωση εύστοχου χτυπήματος στην κρυφή μνήμη, η υπηρεσία πελάτη ανακτά την απαραίτητη πληροφορία τοπικά και ανημερώνει τις δομές του FUSE.

Η κρυφή μνήμη είναι ένας δυναμικός πίνακας κατακερματισμού, στον οποίο χρησιμοποιείται ως κλειδί το όνομα του κάθε αρχείου και καταλόγου (μαζί με τη διαδρομή των καταλόγων ανώτερης ιεραρχίας). Κάθε φορά που γίνεται λήψη του ονόματος και των χαρακτηριστικών ενός αρχείου ή καταλόγου, ενημερώνεται πρώτα η κρυφή μνήμη. Τα απαραίτητα δεδομένα κρατούνται μέσα σε ξεχωριστή προγραμματιστική δομή, η οποία αντιστοιχίζεται στο εκάστοτε κλειδί. Για να ανταποκριθεί στις τρέχουσες απαιτήσεις, το EuterpeFS διατηρεί σε αυτήν τη δομή τον τύπο του περιεχομένου (αρχείο ή κατάλογος), το μέγεθός του και, σε περίπτωση που πρόκειται για αρχείο, η διαδρομή (URL) προς την πραγματική θέση του και τα περιεχόμενά του. Τα τελευταία ενημερώνονται μόνο σε περίπτωση που επιχειρηθεί ανάγνωση του αρχείου.

Ο χρόνος που κρατούνται τα δεδομένα στην κρυφή μνήμη είναι παραμετροποιήσιμος και, μέχρι να γίνουν οι κατάλληλες μετρήσεις αφήνεται στην πλευρά του χρήστη. Οι χρόνοι μπορεί να ποικίλουν με βάση τις δυνατότητες του πελάτη -το EuterpeFS μπορεί να υποστηρίξει και φορητές συσκευές που τρέχουν LINUX και έχουν περιορισμένη μνήμη. Έτσι, όταν γίνεται από το χρήστη μια αίτηση να δει τα περιεχόμενα ενός καταλόγου ή να διαβάσει ένα αρχείο, το EuterpeFS πρώτα κοιτά την κρυφή μνήμη του και στη συνέχεια, αν δε βρει έγκυρη εγγραφή εκεί, ζητά τα αντίστοιχα δεδομένα από τον εξυπηρετητή.

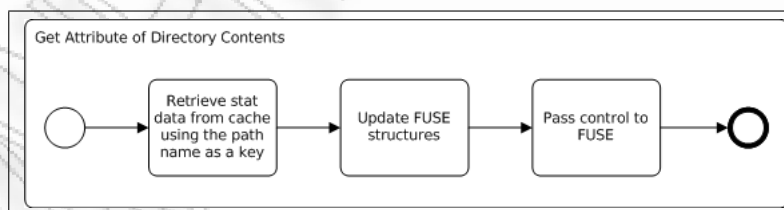
Σε περίπτωση που η πληροφορία δε διατίθεται στην κρυφή μνήμη, είτε επειδή δεν υπήρξαν ποτέ είτε επειδή ακυρώθηκαν λόγω χρόνου, το EuterpeFS αναλαμβάνει να τα ζητήσει από τον εξυπηρετητή. Έτσι, καλεί την αντίστοιχη υπηρεσία εξυπηρετητή, στην οποία δίνει ως όρισμα την τοπική διαδρομή. Η υπηρεσία καταλόγου του εξυπηρετητή επιστρέφει τη ζητούμενη πληροφορία και ο πελάτης την αποθηκεύει αρχικά

στην κρυφή μνήμη και στη συνέχεια ενημερώνει τις δομές του FUSE.



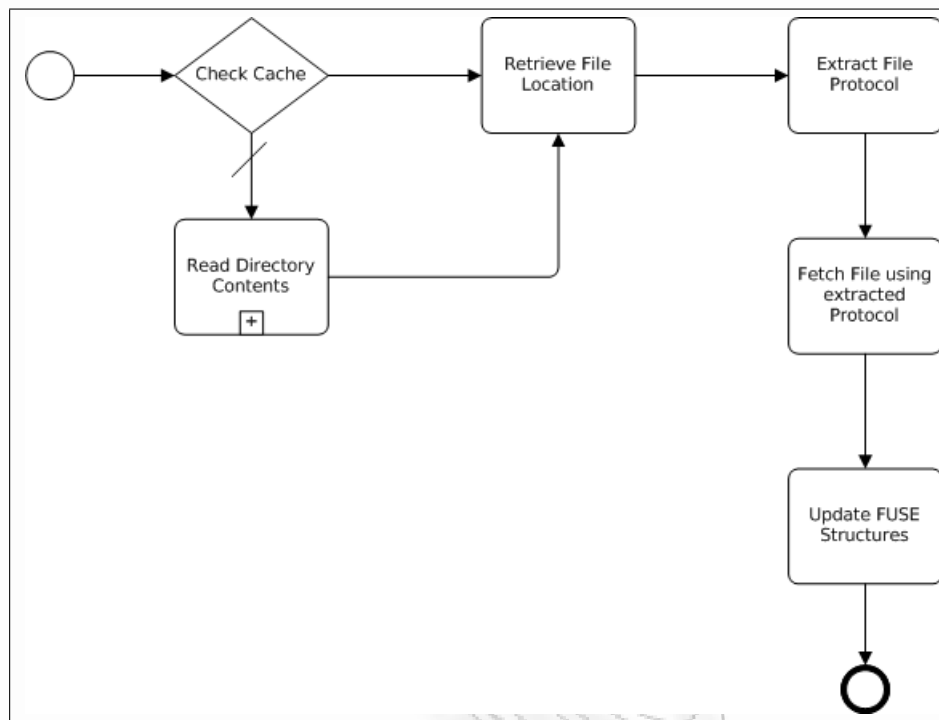
Εικόνα 5.3: Ενορχήστρωση Ανάγνωσης Καταλόγου.

Το λειτουργικό σύστημα διατηρεί μια βασική δομή (stat) με κάποια βασικά χαρακτηριστικά, όπως έχει παρουσιαστεί στο Κεφάλαιο 2.6. Ακολουθώντας αυτήν την απαίτηση, το EuterpeFS ακολουθεί τη διαδικασία που απεικονίζεται στο αντίστοιχο BPMN διάγραμμα της Εικόνας 5.4 για την ανάκτηση αυτών των χαρακτηριστικών. Όπως παρουσιάστηκε και στο προηγούμενο διάγραμμα, τα περιεχόμενα ενός καταλόγου αποθηκεύονται κατά την ανάκτησή τους στην κρυφή μνήμη. Καθώς η διαδικασία της ανάκτησης των χαρακτηριστικών τους αποτελεί κομμάτι της διαδικασίας ανάκτησης του απομακρυσμένου καταλόγου και λαμβάνει χώρα αμέσως μετά την αποθήκευση των δεδομένων στην κρυφή μνήμη, το EuterpeFS ανακτά τα χαρακτηριστικά των αρχείων απευθείας από την κρυφή μνήμη και ενημερώνει τις κατάλληλες δομές του FUSE.



Εικόνα 5.4: Ενορχήστρωση Ανάκτησης Χαρακτηριστικών των Αρχείων.

Όταν ο χρήστης ζητήσει να ανοίξει κάποιο αρχείο, τότε η εφαρμογή πελάτη του EuterpeFS θα αναζητήσει την τοποθεσία του αρχείου και στη συνέχεια θα προσπαθήσει να το κατεβάσει, χρησιμοποιώντας το πρωτόκολλο που έχει καθοριστεί στο URL της τοποθεσίας, όπως φαίνεται και στην Εικόνα 5.5.



Εικόνα 5.5: Ενορχήστρωση Ανάγνωσης Αρχείου.

Αυτή τη στιγμή η εφαρμογή πελάτη υποστηρίζεται πλήρως από το LINUX ενώ δεν έχει δοκιμαστεί ακόμα σε άλλες πλατφόρμες UNIX. Καθώς, όμως, βασίζεται σε βιβλιοθήκες οι οποίες είναι διαθέσιμες και στις υπόλοιπες πλατφόρμες (Apache Axis2, FUSE, Berkeley XML DB) δεν αναμένεται να υπάρξουν προβλήματα συμβατότητας.

Όπως αναφέρθηκε και στην προηγούμενη Ενότητα, όταν γραφόταν η παρούσα διατριβή, το σύστημα αρχείων παρείχε μόνο δυνατότητα ανάγνωσης των αρχείων και των καταλόγων. Ανάλογα με το είδος του αρχείου και του πρωτοκόλλου που απαιτείται για να διαβαστεί, ανοίγει και η αντίστοιχη εφαρμογή. Έτσι, αν πρόκειται για ένα video που μπορεί να το δει ο χρήστης μέσα από κάποιο site τότε όταν επιλέξει το αρχείο, το τελευταίο θα ανοίξει μέσα από το φυλλομετρητή του.

5.2.3 Τεχνολογίες

Το κατακευμασμένο σύστημα αρχείων EuterpeFS χρησιμοποιεί υπηρεσιοστρεφείς αρχιτεκτονικές (SOA) για την επικοινωνία μεταξύ του πελάτη και του εξυπηρετητή. Η βασική υπηρεσία που βρίσκεται πίσω από τον εξυπηρετητή του EuterpeFS είναι ένας Apache Axis2/C HTTP Server [1] ο οποίος παρέχει τις υπηρεσίες ιστού (web services) του εξυπηρετητή του EuterpeFS.

Τόσο η υπηρεσία αρχείων όσο και η υπηρεσία καταλόγων εξυπηρετούνται από την υπηρεσία RemoteDirectoryRetrieval. Μια ενδεικτική αίτηση για ένα κατάλογο θα είναι ένα SOAP μήνυμα που θα έχει την παρακάτω μορφή (έχοντας αφαιρέσει τις επικεφαλίδες του SOAP που αφορούν τη μεταφορά και όχι το πρωτόκολλο του EuterpeFS αυτό καθεαυτό):

```
<directoryName>
  <text>EXTRA.CONTENT.CHOICE</text>
</directoryName>
```

Μια ενδεικτική απάντηση του εξυπηρετητή προς την εφαρμογή του πελάτη θα μπορούσε να έχει τη μορφή (σημειώνεται ότι τα ονόματα των παραδειγμάτων έχουν προκύψει με μικρές τροποποιήσεις του παραδείγματος που δίνεται στο Appendix A του MPEG-21 ISO standard [21]):

```
<contents>
  <fs_resource name="IMAGE_FORMAT_CHOICE" type="0" resource="" size="
    4096" />
  <fs_resource name="IMAGE_SIZE_CHOICE" type="0" resource="" size="4096
    " />
  <fs_resource name="album_cover_art" type="1" resource="http://www.dmu.
    com/always_red/always_red/B000002U0A.1.jpg" size="1635" />
  <fs_resource name="album_video" type="1" resource="http://www.dmu.com
    /always_red/always_red/obliv.mov" size="1635" />
  <fs_resource name="album_review1" type="1" resource="http://www.dmu.
    com/always_red/always_red/review1.txt" size="1635" />
  <fs_resource name="album_review2" type="1" resource="http://www.dmu.
    com/always_red/always_red/review2.txt" size="1635" />
  <fs_resource name="album_review3" type="1" resource="http://www.dmu.
    com/always_red/always_red/review3.txt" size="1635" />
  <fs_resource name="Save_It" type="1" resource="http://www.dmu.com/
    always_red/always_red/Save_It.txt" size="1635" />
  <fs_resource name="I_Haven't_been_Anywhere" type="1" resource="http:
    //www.dmu.com/always_red/always_red/I_Havent_been_Anywhere.txt"
    size="1635" />
  <fs_resource name="Sawdust_and_Sticks" type="1" resource="http://www.
    dmu.com/always_red/always_red/Sawdust_and_Sticks.txt" size="1635"
    />
  <fs_resource name="When_the_Thistle_Blooms" type="1" resource="http:
    //www.dmu.com/always_red/always_red/When_the_Thistle_Blooms.txt"
    size="1635" />
</contents>
```

Εδώ να αναφέρουμε ότι ο τύπος 0 αναφέρεται σε κατάλογο και ο τύπος 1 αναφέρεται σε αρχείο. Το μέγεθος των καταλόγων είναι το καθιερωμένο για το LINUX (4096 bytes) ενώ το μέγεθος των αρχείων ενημερώνεται ασύγχρονα από τον εξυπηρετητή. Τα εν λόγω αρχεία δεν υπάρχουν στην πραγματικότητα και για το λόγο αυτό έχουν το μέγεθος του html κειμένου που αναφέρει ότι δεν βρέθηκαν.

Όπως φαίνεται και από αυτό το παράδειγμα, ο εξυπηρετητής του EuterpeFS δε χρειάζεται να κρατά στην πραγματικότητα κανένα αρχείο. Απλά ενημερώνει για τη θέση των αρχείων και μετά ο χρήστης μπορεί να πάει στην κάθε τοποθεσία και να τα πάρει από εκεί. Η βασική δουλειά του εξυπηρετητή είναι να κρατά τις εγγραφές με τα μεταδεδομένα, τα οποία είναι στη μορφή που δόθηκε στο Κεφάλαιο 4. Για το λόγο αυτό διατηρεί μια βάση δεδομένων όπου μπορεί να αποθηκεύει δεδομένα σε μορφή XML. Στην παρούσα εργασία επιλέχθηκε η Oracle Berkeley DB XML [12]. Κάθε υπηρεσία ιστού πραγματοποιεί κλήσεις συναρτήσεων στη βάση δεδομένων για να ανακαλυφθούν τα αποτελέσματα. Στη συνέχεια, συλλέγει και μορφοποιεί τα αποτελέσματα και τα στέλνει πίσω στον πελάτη. Η χρήση της βάσης δεδομένων μπορεί να μας εξασφαλίσει καλή απόδοση στην αναζήτηση των μεταδεδομένων, αφού κρατά δομές ευρετηρίων με αποδοτικό τρόπο.

Στην πλευρά του πελάτη, το EuterpeFS χρησιμοποιεί το FUSE, το οποίο παρουσιάστηκε στην Ενότητα 2.8. Το FUSE είναι αυτό που τελικά παρέχει τη διαφάνεια του συστήματος αρχείων στο χρήστη. Όταν πραγματοποιείται μια κλήση συστήματος στην περιοχή που έχει γίνει προσάρτηση το EuterpeFS, το FUSE προωθεί αυτήν την κλήση στην εφαρμογή πελάτη του EuterpeFS, η οποία με τη σειρά της χρησιμοποιεί το μοντέλο επικοινωνίας που παρουσιάστηκε παραπάνω για να πάρει από τον εξυπηρετητή τους ζητούμενους πόρους, ενώ χρησιμοποιεί και πάλι το FUSE για την παρουσίαση στο χρήστη.

Όταν έρχονται δεδομένα από τον εξυπηρετητή για κάποιο αρχείο ή κάποιο κατάλογο, η εφαρμογή του πελάτη χρησιμοποιεί έναν πίνακα κατακερματισμού, όπου κρατά προσωρινά τις εγγραφές, εν είδει κρυφής μνήμης. Η τεχνολογία που χρησιμοποιήθηκε για τον πίνακα κατακερματισμού προέκυψε από τη βιβλιοθήκη του System Security Services Daemon (SSSD) libdhash, η οποία επιτρέπει τη δημιουργία δυναμικών πινάκων κατακερματισμού.

Η μεταφορά των δεδομένων γίνεται με τη βοήθεια του εργαλείου cURL και της αντίστοιχης βιβλιοθήκης του libcurl [2]. Έτσι, τα πρωτόκολλα μεταφοράς δεδομένων είναι αυτά που υποστηρίζονται από αυτήν τη βιβλιοθήκη: DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, TELNET και TFTP. Επίσης υποστηρίζονται πιστοποιητικά SSL και λειτουργίες HTTP POST, HTTP PUT, FTP upload, αυθεντικοποίηση χρήστη με όνομα και κωδικό πρόσβασης.

5.3 Έλεγχος Σχεδιαστικών Απαιτήσεων

Στην Ενότητα 3.2 παρουσιάστηκαν οι σχεδιαστικές απαιτήσεις ενός καταναμιμένου συστήματος αρχείων. Σε αυτήν την Ενότητα θα αξιολογήσουμε κατά πόσο το

EuterpeFS ικανοποιεί αυτές τις απαιτήσεις και τι μέριμνα έχει ληφθεί για να τις καλύψει ενδεχομένως στο μέλλον.

5.3.1 Διαφάνεια

Διαφάνεια Πρόσβασης

Κάνοντας χρήση του FUSE το EuterpeFS επιτυγχάνει το στόχο της διαφάνειας πρόσβασης. Ο χρήστης έχει την ψευδαίσθηση ότι αλληλεπιδρά με κάποιο τοπικό αρχείο, ενώ στην πραγματικότητα, κάνοντας ίδιες κλήσεις συστήματος, αλληλεπιδρά με το κατανεμημένο.

Διαφάνεια Τοποθεσίας

Και πάλι μέσω του FUSE, η προσάρτηση του EuterpeFS γίνεται σε κάποιον κατάλογο του τοπικού συστήματος αρχείων, διατηρώντας έτσι την ίδια περιοχή ονομάτων των αρχείων.

Διαφάνεια Μετακίνησης

Αν ένα αρχείο αλλάξει θέση, τότε θα πρέπει να ενημερωθεί ο εξυπηρετητής του EuterpeFS, ώστε να υπάρχει συνέπεια και στις τοποθεσίες που κρατούν οι πελάτες στην κρυφή μνήμη τους.

Διαφάνεια Απόδοσης και Κλιμάκωσης

Το EuterpeFS κάνει χρήση state-of-the-art εργαλείων (Oracle Berkeley DB, Apache Axis2/C Server) που θεωρούνται αξιόπιστα στον τομέα τους, παρέχοντας εγγυήσεις απόδοσης και κλιμάκωσης.

5.3.2 Υποστήριξη ταυτόχρονης ενημερώσεων των αρχείων

Αυτή τη στιγμή το EuterpeFS δεν υποστηρίζει εγγραφή αρχείου.

5.3.3 Αντίγραφα Αρχείων

Αυτή τη στιγμή το EutereFS δεν υποστηρίζει αντίγραφα αρχείων. Άλλωστε, η διαθεσιμότητα των ίδιων των αρχείων δεν είναι αρμοδιότητα του EutereFS αλλά των υπηρεσιών που τα παρέχουν κάθε φορά. Όσον αφορά την τήρηση της δομής του συστήματος αρχείων, στο μέλλον εξετάζεται να προστεθεί υποστήριξη για κατανομή της βάσης δεδομένων σε πολλούς εξυπηρετητές.

5.3.4 Υποστήριξη ετερογένειας

Οι υπηρεσίες ιστού που χρησιμοποιούνται για την ανταλλαγή των δεδομένων μας εξασφαλίζουν ένα βαθμό ετερογένειας, ενώ και το FUSE παρέχει υποστήριξη για διάφορες πλατφόρμες (Linux, Windows, NetBSD, FreeBSD, Mac OS X, OpenSolaris, GNU/Hurd) [14].

5.3.5 Ανοχή Βλαβών

Και σε αυτήν την περίπτωση, ο εξυπηρετητής βασίζεται σε μια βάση δεδομένων που υποστηρίζει επανάκτηση προηγούμενης κατάστασης. Για τον Apache Axis2/C Server δεν υπάρχει κάποιο δεδομένο σχετικά με τη συμπεριφορά του σε βλάβες. Σε κάθε περίπτωση σε όλη τη διάρκεια των δοκιμών και της ανάπτυξης δεν παρουσίασε κάποια ανώμαλη συμπεριφορά. Από την πλευρά του πελάτη, ακόμα και αν πάθει βλάβη, μπορεί να ξανασηκωθεί χωρίς κανένα πρόβλημα.

5.3.6 Συνέπεια

Η συνέπεια είναι ένας παράγοντας που πρόκειται να προστεθεί στο άμεσο μέλλον στο EutereFS. Προβλέπεται ο εξυπηρετητής να κρατά αρχείο με τους πελάτες που διατηρούν ανοικτά αντίγραφα των αρχείων και να τους ειδοποιεί για τυχόν αλλαγές. Επίσης προβλέπεται να γίνει κατάλληλη χρήση των στοιχείων fragment (βλ. Ενότητα 4.2.12) για να μπορεί να διατηρεί ο κάθε πελάτης μόνο το κομμάτι του αρχείου στο οποίο θα κάνει αλλαγές.

5.3.7 Ασφάλεια

Με δεδομένο ότι το EutereFS βασίζεται κατεξοχήν σε υπηρεσίες ιστού για τη μετάδοση των απαραίτητων πληροφοριών, η ασφάλεια θα πρέπει να επικεντρωθεί σε

αυτήν την περιοχή. Η ασφάλεια των υπηρεσιών ιστού (web services) αποτελεί εδώ και κάποια χρόνια αντικείμενο εκτεταμένης έρευνας με διάφορα μοντέλα να έχουν προταθεί στη βιβλιογραφία. Ενδεικτικά, ο οργανισμός Organization for the Advancement of Structured Information Standards (OASIS) έχει προτείνει διάφορες προσεγγίσεις [10].

5.3.8 Αποδοτικότητα

Καθώς βρίσκεται ακόμα υπό ανάπτυξη, το EuterpeFS δεν έχει δοκιμαστεί για την απόδοσή του. Σε κάθε περίπτωση, με δεδομένα τα εργαλεία που χρησιμοποιεί από την πλευρά του εξυπηρετητή, προβλέπεται να έχει καλή απόδοση. Επίσης, εδώ θα πρέπει να σημειωθεί ότι η απόδοση του EuterpeFS θα είναι και ποιοτική, αφού παρέχει κάτι περισσότερο από διαμοιρασμό αρχείων -παρέχει και έναν προτυποποιημένο τρόπο οργάνωσής τους.

Κεφάλαιο 6

Συμπεράσματα και Μελλοντικό Έργο

Τη στιγμή που γράφεται αυτή η εργασία, το καταναμημένο σύστημα αρχείων EuterpeFS παρέχει υποστήριξη για ανάγνωση πόρων, οι οποίοι μπορεί να βρίσκονται αποθηκευμένοι σε οποιοδήποτε χώρο. Ο εξυπηρετητής το μόνο που χρειάζεται να κάνει είναι να δώσει στον πελάτη τη διεύθυνση όπου μπορεί να βρει το δεδομένο πόρο. Όπως είδαμε και σε προηγούμενες Ενότητες, δεν υποστηρίζονται ακόμα όλα τα πιθανά πρωτόκολλα επικοινωνίας.

Πρώτο μέλημα μελλοντικής επέκτασης που θα γίνει σε αυτήν την εργασία είναι η υποστήριξη και άλλων πρωτοκόλλων μεταφοράς αρχείων, πέρα από αυτά που υποστηρίζονται από τη βιβλιοθήκη libcurl. Όταν θα έχει πια ολοκληρωθεί αυτή η απαίτηση, θα υπάρχει αφενός ένα σχήμα όπου θα φορτώνονται plugin με βιβλιοθήκες που θα υποστηρίζουν νέα πρωτόκολλα και αφετέρου θα μπορούν να υποστηριχθούν και άλλα ήδη υπάρχοντα καταναμημένα συστήματα αρχείων κάτω από το πέπλο του EuterpeFS.

Παράλληλα, θα ξεκινήσει και η ανάπτυξη του μέρους του EuterpeFS που έχει να κάνει με τη δημιουργία νέων αρχείων αλλά και την τροποποίηση των ήδη υπαρχόντων. Όπως είδαμε κάτι τέτοιο απαιτεί ιδιαίτερη προσοχή στη συνέπεια του συστήματος αρχείων και για το λόγο αυτό θα γίνει προσπάθεια να εκμεταλλευτούμε όσο γίνεται τα εργαλεία που μας παρέχει η ίδια η δομή του EuterpeFS (που έχει ως βάση το πρότυπο MPEG-21 DIDL) καθώς και το γεγονός ότι ούτως ή άλλως χρησιμοποιούμε βάσεις δεδομένων για να διατηρούμε τα μεταδεδομένα, οπότε μπορούμε, χωρίς σημαντική αύξηση της πολυπλοκότητας και βλάβη της απόδοσης, να χρησιμοποιήσουμε

αυτές τις ήδη υπάρχουσες δομές.

Τέλος, θα δοθεί βαρύτητα στον τομέα της ασφάλειας. Με δεδομένη την έρευνα που έχει γίνει στο χώρο των υπηρεσιών ιστού και την ίδια την προτυποποιημένη οργάνωση του EutereFS, θεωρούμε ότι θα μπορέσουμε να καταλήξουμε σε εκείνα τα εργαλεία που θα μπορέσουν να προσαρμοστούν σωστά, χωρίς να βλάπτεται η απόδοση του συστήματος αρχείων μας.

Βιβλιογραφία

- [1] Apache Axis2/C - TheWebServicesEngine, <http://ws.apache.org/axis2/c/>. [cited at p. 43]
- [2] cURL and libcurl, <http://curl.haxx.se/>. [cited at p. 45]
- [3] File System - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/File_system. [cited at p. 5]
- [4] Filesystem in Userspace - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Filesystem_in_Userspace. [cited at p. 10, 70]
- [5] General overview of the Linux file system, http://tldp.org/LDP/intro-linux/html/sect_03_01.html. [cited at p. 5]
- [6] IBM - Develop your own filesystem with FUSE, <http://www.ibm.com/developerworks/linux/library/l-fuse/>. [cited at p. 20, 70]
- [7] ISO/IEC TR 21000. http://www.iso.org/iso/catalogue_detail.htm?csnumber=40611. [cited at p. 23]
- [8] MPEG - Working Documents, http://mpeg.chiariglione.org/working_documents.htm#MPEG-21. [cited at p. 23]
- [9] MPEG-21, <http://mpeg.chiariglione.org/standards/mpeg-21/mpeg-21.htm>. [cited at p. 23]
- [10] OASIS Web Services Security (WSS) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss. [cited at p. 48]
- [11] OpenAFS, <http://www.openafs.org/>. [cited at p. 20]
- [12] Oracle Berkeley DB, <http://www.oracle.com/technetwork/database/berkeleydb/overview/index.html>. [cited at p. 45]

- [13] Sourceforge.net: FileSystems - fuse,
<http://sourceforge.net/apps/mediawiki/fuse/index.php?title=FileSystems>. [cited at p. 11]
- [14] Sourceforge.net: OperatingSystems - fuse,
<http://sourceforge.net/apps/mediawiki/fuse/index.php?title=OperatingSystems>. [cited at p. 47]
- [15] stat(2): file status - Linux man page,
<http://linux.die.net/man/2/stat>. [cited at p. 8]
- [16] The Linux Kernel - Chapter 9 The File System,
<http://tldp.org/LDP/tlk/fs/filesystem.html>. [cited at p. 9]
- [17] UCLA - CS111 Lecture 17: Distributed Systems II, <http://read.cs.ucla.edu/111/2007spring/notes/lec17>. [cited at p. 19, 70]
- [18] George F. Coulouris, Jean Dollimore, and Tim Kindberg. *Distributed systems: concepts and design*. Pearson Education Ltd, Essex, England, 2001. [cited at p. 13, 17, 20]
- [19] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. *SIGOPS Oper. Syst. Rev.*, 37(5):29--43, 2003. [cited at p. 3]
- [20] John H. Howard, Michael L. Kazar, Sherri G. Menees, David A. Nichols, M. Satyanarayanan, Robert N. Sidebotham, and Michael J. West. Scale and performance in a distributed file system. *ACM Trans. Comput. Syst.*, 6(1):51--81, 1988. [cited at p. 3, 19]
- [21] ISO. ISO/IEC 21000-2:2001: Information technology -- Multimedia framework (MPEG-21) -- Part 2: Digital Item Declaration. Technical report. Approved. [cited at p. 4, 23, 25, 37, 44, 55]
- [22] Network Working Group, Sun Microsystems Inc. NFS: Network File System Protocol Specification. Technical report, March 1989. Request for Comments: 1094. [cited at p. 3, 18]
- [23] S. Shepler, B. Callaghan, D. Robinson, R. Thurlow, C. Beame, M. Eisler, D. Noveck. Network File System (NFS) version 4 Protocol. RFC 3530, April 2003. [cited at p. 18]
- [24] R. Sandberg, D. Golgberg, S. Kleiman, D. Walsh, and B. Lyon. Innovations in internetworking. chapter Design and implementation of the Sun network filesystem, pages 379--390. Artech House, Inc., Norwood, MA, USA, 1988. [cited at p. 18]
- [25] M. Satyanarayanan. The evolution of coda. *ACM Trans. Comput. Syst.*, 20(2):85--124, 2002. [cited at p. 3]
- [26] Sun Microsystems Inc. *NFS Administration Guide*, August 1997. [cited at p. 18]
- [27] A.S. Tanenbaum. *Σύγχρονα Λειτουργικά Συστήματα - Τόμος Β' (Κατανεμημένα Συστήματα)*. Prentice Hall International Inc., Α. Παπασωτηρίου και ΣΙΑ Ο.Ε., 1994. [cited at p. 17]

Παράρτημα

Παράρτημα Α΄

Παράδειγμα Μουσικού Άλμπουμ

Σε αυτό το Παράρτημα δίνεται ένα ενδεικτικό παράδειγμα μουσικού άλμπουμ. Το παράδειγμα προέρχεται από το [21] και είναι ελαφρώς τροποποιημένο για να υποστηρίξει το EuterpeFS.

```
<?xml version="1.0" encoding="UTF-8"?>
<DIDL xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:cr="http://www.mpeg.
    org/mpeg7/Content-Ratings-Scheme"
  xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance" xmlns:profile
    ="http://www.mpeg.org/mpeg21/Profile-Specs"
  xsi:noNamespaceSchemaLocation="http://www.mpeg.org/mpeg21/schemas/didl
    .xsd">
<DECLARATIONS>
  <DESCRIPTOR ID="ALBUM.RATING">
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
        Content ratings provided by Parents of
        Teens, Inc.
      </STATEMENT>
    </DESCRIPTOR>
    <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/xml">
      <cr:violence>None</cr:violence>
      <cr:explicit-language>None</cr:explicit-language>
      <cr:sex>None</cr:sex>
    </STATEMENT>
  </DESCRIPTOR>
</DECLARATIONS>
<CONTAINER>
  <DESCRIPTOR>
    <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
```

```

This is a package for John Q. Consumer.
This package was provided
by Digital Music Unlimited.
  </STATEMENT>
</DESCRIPTOR>
</ITEM>
<CHOICE CHOICE_ID="PLATFORM_CHOICE" TYPE="single">
  <SELECTION SELECT_ID="PLATFORM_WINDOWS">
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/xml">
        <profile:operating-system>Win32</profile:operating-system>
      </STATEMENT>
    </DESCRIPTOR>
  </SELECTION>
  <SELECTION SELECT_ID="PLATFORM_LINUX">
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/xml">
        <profile:operating-system>Linux</profile:operating-system>
      </STATEMENT>
    </DESCRIPTOR>
  </SELECTION>
  <SELECTION SELECT_ID="PLATFORM_MAC">
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/xml">
        <profile:operating-system>MacOS</profile:operating-system>
      </STATEMENT>
    </DESCRIPTOR>
  </SELECTION>
</CHOICE>
<CHOICE CHOICE_ID="ALL_SONGS" TYPE="multiple">
  <SELECTION SELECT_ID="PICK_SONGS">
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
        I want to choose among the individual
        songs.
          </STATEMENT>
        </DESCRIPTOR>
      </SELECTION>
    </CHOICE>
    <CHOICE CHOICE_ID="SONG_PICKER" DEFAULT="SONG1 SONG2 SONG3 SONG4"
      TYPE="multiple">
      <CONDITION REQUIRE="PICK_SONGS" />
      <DESCRIPTOR>
        <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
          Choose the songs you would like:
        </STATEMENT>
      </DESCRIPTOR>
      <SELECTION SELECT_ID="SONG1">
        <DESCRIPTOR>
          <REFERENCE URI="#SONG1_TITLE" />
        </DESCRIPTOR>

```

```

</SELECTION>
<SELECTION SELECT_ID="SONG2">
  <DESCRIPTOR>
    <REFERENCE URI="#SONG2.TITLE" />
  </DESCRIPTOR>
</SELECTION>
<SELECTION SELECT_ID="SONG3">
  <DESCRIPTOR>
    <REFERENCE URI="#SONG3.TITLE" />
  </DESCRIPTOR>
</SELECTION>
<SELECTION SELECT_ID="SONG4">
  <DESCRIPTOR>
    <REFERENCE URI="#SONG4.TITLE" />
  </DESCRIPTOR>
</SELECTION>
</CHOICE>
<CHOICE CHOICE_ID="BITRATE_CHOICE" DEFAULT="LOW_BITRATE" TYPE="
  single">
  <DESCRIPTOR>
    <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
      Please select the fidelity you would
      prefer.
    </STATEMENT>
  </DESCRIPTOR>
  <SELECTION SELECT_ID="LOW_BITRATE">
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
        Low (128 Kbits/sec , $10.00).
      </STATEMENT>
    </DESCRIPTOR>
  </SELECTION>
  <SELECTION SELECT_ID="HIGH_BITRATE">
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
        High (192 Kbits/sec , $15.00).
      </STATEMENT>
    </DESCRIPTOR>
  </SELECTION>
</CHOICE>
<CHOICE CHOICE_ID="EXTRA_CONTENT_CHOICE" TYPE="single">
  <DESCRIPTOR>
    <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
      Would you like to get supplemental
      content for this album?
    </STATEMENT>
  </DESCRIPTOR>
  <SELECTION SELECT_ID="WANT_EXTRA_CONTENT">
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
        Yes, get all of the supplemental

```

```

    content .
        </STATEMENT>
    </DESCRIPTOR>
</SELECTION>
<SELECTION SELECT_ID="ASK_EXTRA_CONTENT">
    <DESCRIPTOR>
        <STATEMENT TYPE="urn:mpeg:mpeg21:did / statement -types / text / plain">
            Yes, but let me choose which items to
            get .
        </STATEMENT>
    </DESCRIPTOR>
</SELECTION>
<SELECTION SELECT_ID="DONT_ASK_ABOUT_CONTENT">
    <DESCRIPTOR>
        <STATEMENT TYPE="urn:mpeg:mpeg21:did / statement -types / text / plain">
            No, I don't want any of the extra stuff .
        </STATEMENT>
    </DESCRIPTOR>
</SELECTION>
</CHOICE>
<CHOICE CHOICE_ID="COVER_ART_CHOICE" TYPE="multiple">
    <CONDITION REQUIRE="ASK_EXTRA_CONTENT" />
    <DESCRIPTOR>
        <STATEMENT TYPE="urn:mpeg:mpeg21:did / statement -types / text / plain">
            Supplemental content:
                </STATEMENT>
        </DESCRIPTOR>
    <SELECTION SELECT_ID="GET_ART">
        <DESCRIPTOR>
            <STATEMENT TYPE="urn:mpeg:mpeg21:did / statement -types / text / plain">
                Include the cover art .
            </STATEMENT>
        </DESCRIPTOR>
    </SELECTION>
    <SELECTION SELECT_ID="GET_LYRICS">
        <DESCRIPTOR>
            <STATEMENT TYPE="urn:mpeg:mpeg21:did / statement -types / text / plain">
                Include the song lyrics .
            </STATEMENT>
        </DESCRIPTOR>
    </SELECTION>
    <SELECTION SELECT_ID="GET_VIDEO">
        <CONDITION REQUIRE="PLATFORM_WINDOWS" />
        <CONDITION REQUIRE="PLATFORM_MAC" />
        <DESCRIPTOR>
            <STATEMENT TYPE="urn:mpeg:mpeg21:did / statement -types / text / plain">
                Include video footage from the latest
                concert .
            </STATEMENT>
        </DESCRIPTOR>
    </SELECTION>

```



```

<SELECTION SELECT_ID="GET_REVIEWS">
  <DESCRIPTOR>
    <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
      Include press reviews.
    </STATEMENT>
  </DESCRIPTOR>
</SELECTION>
</CHOICE>
<CHOICE CHOICE_ID="IMAGE_FORMAT_CHOICE" DEFAULT="JPEG_IMAGE">
  <CONDITION REQUIRE="GET_ART" />
  <CONDITION REQUIRE="WANT_EXTRA_CONTENT" />
  <SELECTION SELECT_ID="JPEG_IMAGE">
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/xml">
        <profile:image-type>JPEG</profile:image-type>
      </STATEMENT>
    </DESCRIPTOR>
  </SELECTION>
  <SELECTION SELECT_ID="GIF_IMAGE">
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/xml">
        <profile:image-type>GIF</profile:image-type>
      </STATEMENT>
    </DESCRIPTOR>
  </SELECTION>
  <SELECTION SELECT_ID="BMP_IMAGE">
    <CONDITION REQUIRE="PLATFORM_WINDOWS" />
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/xml">
        <profile:image-type>Windows-BMP</profile:image-type>
      </STATEMENT>
    </DESCRIPTOR>
  </SELECTION>
</CHOICE>
<CHOICE CHOICE_ID="IMAGE_SIZE_CHOICE" DEFAULT="LARGE_IMAGE">
  <CONDITION REQUIRE="GET_ART" EXCEPT="BMP_IMAGE" />
  <CONDITION REQUIRE="WANT_EXTRA_CONTENT" EXCEPT="BMP_IMAGE" />
  <DESCRIPTOR>
    <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
      Please select the image size you would
      like:
    </STATEMENT>
  </DESCRIPTOR>
  <SELECTION SELECT_ID="LARGE_IMAGE">
    <DESCRIPTOR>
      <STATEMENT TYPE="urn:mpeg:mpeg21:did/statement-types/text/plain">
        Large (301x300)
      </STATEMENT>
    </DESCRIPTOR>
  </SELECTION>
  <SELECTION SELECT_ID="SMALL_IMAGE">

```

```

<DESCRIPTOR>
  <STATEMENT TYPE="urn:mpeg:mpeg21:did / statement-types / text / plain">
    Small (130x130)
  </STATEMENT>
</DESCRIPTOR>
</SELECTION>
</CHOICE>
<DESCRIPTOR>
  <STATEMENT TYPE="http://www.dmu.com/content-organizer-hints">
    DMU 9876:: Item Type="Music Album";
  </STATEMENT>
</DESCRIPTOR>
<DESCRIPTOR>
  <REFERENCE URI="#ALBUM.RATING" />
</DESCRIPTOR>
<DESCRIPTOR ID="PERFORMING.GROUP">
  <STATEMENT TYPE="urn:mpeg:mpeg21:did / statement-types / text / xml">
    <rdf:RDF>
      <rdf:Description>
        <dc:creator>Once Blue</dc:creator>
      </rdf:Description>
    </rdf:RDF>
  </STATEMENT>
</DESCRIPTOR>
<DESCRIPTOR>
  <STATEMENT TYPE="urn:mpeg:mpeg21:did / statement-types / text / xml">
    <rdf:RDF>
      <rdf:Description>
        <dc:title>Always Red</dc:title>
        <dc:creator>Jack Jake (vocals)</dc:creator>
        <dc:creator>Jane Juno (vocals, tambourine, finger snaps)
        </dc:creator>
        <dc:contributor>Joe Jump (acoustic & electric guitars, piano
        )
        </dc:contributor>
        <dc:contributor>Jeff Jelly (acoustic bass) </dc:contributor>
        <dc:contributor>Jim Jinks (drums, marimba, shaker, carob pods)
        </dc:contributor>
        <dc:subject>Record Album: Always Red by Always Red </dc:subject>
        <dc:publisher>Acme Records </dc:publisher>
        <dc:identifier>CD-ID: a409c80c</dc:identifier>
        <dc:source>ASIN: B000002U0A</dc:source>
        <dc:date>1995-10-24</dc:date>
      </rdf:Description>
    </rdf:RDF>
  </STATEMENT>
</DESCRIPTOR>
<DESCRIPTOR ID="RIGHTS">
  <STATEMENT TYPE="urn:mpeg:mpeg21:did / statement-types / text / xml">
    <rdf:RDF>
      <rdf:Description>

```

```

    <dc:rights>Copyright 1995, Acme Records, All Rights Reserved.
    Unauthorized duplication is a violation of
    applicable laws.
  </dc:rights>
</rdf:Description>
</rdf:RDF>
</STATEMENT>
</DESCRIPTOR>
<DESCRIPTOR ID="ICON_FILE">
  <DESCRIPTOR>
    <STATEMENT TYPE="http://www.dmu.com/content-organizer-hints">
      DMU 9876:: Descriptor Type="Item Icon";
    </STATEMENT>
  </DESCRIPTOR>
  <COMPONENT NAME="comfortably_numb_icon">
    <RESOURCE
      REF="http://4.bp.blogspot.com/_DtwOfsquo4M/TE5FIHrDXrI/
      AAAAAAAD8/snkPOuxo9Pc/s1600/pink-floyd-the-wall-297x300.jpg
      "
      TYPE="image/jpeg" SIZE="" />
    </COMPONENT>
  </DESCRIPTOR>
  <COMPONENT NAME="my_playlist">
    <CONDITION EXCEPT="PICK_SONGS" />
    <CONDITION REQUIRE="SONG1 SONG2 SONG3 SONG4" />
    <RESOURCE REF="http://www.dmu.com/always_red/always_red/always_red.
      m3u"
      TYPE="audio/x-mpegurl" SIZE="" />
    <ANCHOR PRECEDENCE="1000">
      <DESCRIPTOR>
        <STATEMENT TYPE="text/text">
          Play Entire Album
        </STATEMENT>
      </DESCRIPTOR>
    </ANCHOR>
  </COMPONENT>
</ITEM>
  <CONDITION REQUIRE="SONG1" />
  <DESCRIPTOR>
    <STATEMENT TYPE="http://www.dmu.com/content-organizer-hints">
      DMU 9876:: Item Type="Song";
    </STATEMENT>
  </DESCRIPTOR>
  <DESCRIPTOR ID="SONG1_TITLE">
    <STATEMENT TYPE="text/xml">
      <rdf:RDF>
        <rdf:Description>
          <dc:title>Comfortably_Numb</dc:title>
        </rdf:Description>
      </rdf:RDF>
    </STATEMENT>
  </DESCRIPTOR>

```

```

</STATEMENT>
</DESCRIPTOR>
<DESCRIPTOR>
  <STATEMENT TYPE="text/xml">
    <rdf:RDF>
      <rdf:Description>
        <dc:coverage>233</dc:coverage>
      </rdf:Description>
    </rdf:RDF>
  </STATEMENT>
</DESCRIPTOR>
<DESCRIPTOR>
  <REFERENCE URI="#RIGHTS" />
</DESCRIPTOR>
<DESCRIPTOR>
  <REFERENCE URI="#ALBUM.RATING" />
</DESCRIPTOR>
<COMPONENT NAME="Save_It">
  <CONDITION REQUIRE="LOW_BITRATE" />
  <RESOURCE REF="http://www.youtube.com/watch?v=tfY0A-HLeMo"
    TYPE="audio/mp3" SIZE="" />
  <ANCHOR PRECEDENCE="50">
    <DESCRIPTOR>
      <STATEMENT TYPE="text/text">
        Play Song
      </STATEMENT>
    </DESCRIPTOR>
  </ANCHOR>
</COMPONENT>
<COMPONENT NAME="Comfortably_Numb_192">
  <CONDITION REQUIRE="HIGH_BITRATE" />
  <RESOURCE REF="http://www.youtube.com/watch?v=tfY0A-HLeMo"
    TYPE="audio/mp3" LOCAL_PATH="01_Save_Me.mp3" SIZE="" />
  <ANCHOR PRECEDENCE="100">
    <DESCRIPTOR>
      <STATEMENT TYPE="text/text">
        Play Song
      </STATEMENT>
    </DESCRIPTOR>
  </ANCHOR>
</COMPONENT>
<COMPONENT NAME="Comfortably_Numb">
  <CONDITION REQUIRE="GET_LYRICS" />
  <CONDITION REQUIRE="WANT_EXTRA_CONTENT" />
  <RESOURCE
    REF="http://www.lyricsfreak.com/p/pink+floyd/comfortably+
      numb_20108779.html"
    TYPE="text/text" MODE="text" SIZE="" />
  <ANCHOR PRECEDENCE="25">
    <DESCRIPTOR>
      <STATEMENT TYPE="text/text">

```

```

View Lyrics
                                </STATEMENT>
    </DESCRIPTOR>
  </ANCHOR>
</COMPONENT>
</ITEM>
<ITEM>
  <CONDITION REQUIRE="SONG2" />
  <DESCRIPTOR>
    <STATEMENT TYPE="http://www.dmu.com/content-organizer-hints">
      DMU 9876:: Item Type="Song";
                                </STATEMENT>
    </DESCRIPTOR>
  <DESCRIPTOR ID="SONG2.TITLE">
    <STATEMENT TYPE="text/xml">
      <rdf:RDF>
        <rdf:Description>
          <dc:title>Jet Airliner</dc:title>
        </rdf:Description>
      </rdf:RDF>
    </STATEMENT>
  </DESCRIPTOR>
  <DESCRIPTOR>
    <STATEMENT TYPE="text/xml">
      <rdf:RDF>
        <rdf:Description>
          <dc:coverage>193</dc:coverage>
        </rdf:Description>
      </rdf:RDF>
    </STATEMENT>
  </DESCRIPTOR>
  <DESCRIPTOR>
    <REFERENCE URI="#RIGHTS" />
  </DESCRIPTOR>
  <DESCRIPTOR>
    <REFERENCE URI="#ALBUM.RATING " />
  </DESCRIPTOR>
  <COMPONENT NAME="Jet_Airliner">
    <CONDITION REQUIRE="LOW_BITRATE" />
    <RESOURCE REF="http://www.youtube.com/watch?v=NRJ8tMN2CZ0"
      TYPE="audio/mp3" SIZE="" />
    <ANCHOR PRECEDENCE="50">
      <DESCRIPTOR>
        <STATEMENT TYPE="text/text">
          Play Song
                                </STATEMENT>
      </DESCRIPTOR>
    </ANCHOR>
  </COMPONENT>
<COMPONENT NAME="Jet_Airliner_192">
  <CONDITION REQUIRE="HIGH_BITRATE" />

```

```

<RESOURCE REF=" http://www.youtube.com/watch?v=NRJ8tMN2CZ0"
  TYPE="audio/mp3" SIZE="" />
<ANCHOR PRECEDENCE="100">
  <DESCRIPTOR>
    <STATEMENT TYPE="text/text">
      Play Song
    </STATEMENT>
  </DESCRIPTOR>
</ANCHOR>
</COMPONENT>
<COMPONENT NAME="Jet_Airline">
  <CONDITION REQUIRE="WANT_EXTRA_CONTENT" />
  <CONDITION REQUIRE="GET_LYRICS" />
  <RESOURCE
    REF="http://www.musicsonglyrics.com/M/moderntalkinglyrics/
      moderntalkingjetairlinerlyrics.htm"
    TYPE="text/html" SIZE="" />
  <ANCHOR PRECEDENCE="25">
    <DESCRIPTOR>
      <STATEMENT TYPE="text/text">
        View Lyrics
      </STATEMENT>
    </DESCRIPTOR>
  </ANCHOR>
</COMPONENT>
</ITEM>
<ITEM>
  <CONDITION REQUIRE="SONG3" />
  <DESCRIPTOR>
    <STATEMENT TYPE="http://www.dmu.com/content-organizer-hints">
      DMU 9876:: Item Type="Song";
    </STATEMENT>
  </DESCRIPTOR>
  <DESCRIPTOR ID="SONG3.TITLE">
    <STATEMENT TYPE="text/xml">
      <rdf:RDF>
        <rdf:Description>
          <dc:title>Thorn In My Side</dc:title>
        </rdf:Description>
      </rdf:RDF>
    </STATEMENT>
  </DESCRIPTOR>
  <DESCRIPTOR>
    <STATEMENT TYPE="text/xml">
      <rdf:RDF>
        <rdf:Description>
          <dc:coverage>209</dc:coverage>
        </rdf:Description>
      </rdf:RDF>
    </STATEMENT>
  </DESCRIPTOR>

```

```

<DESCRIPTOR>
  <REFERENCE URI="#ALBUM.RATING" />
</DESCRIPTOR>
<DESCRIPTOR>
  <REFERENCE_URI IDREF="#RIGHTS" />
</DESCRIPTOR>
<COMPONENT NAME="Thorn_In_My_Side">
  <CONDITION REQUIRE="LOW_BITRATE" />
  <RESOURCE REF="http://www.youtube.com/watch?v=_AmkmqYEarw"
    TYPE="audio/mp3" SIZE="" />
  <ANCHOR PRECEDENCE="50">
    <DESCRIPTOR>
      <STATEMENT TYPE="text/text">
        Play Song
      </STATEMENT>
    </DESCRIPTOR>
  </ANCHOR>
</COMPONENT>
<COMPONENT NAME="Thorn_In_My_Side">
  <CONDITION REQUIRE="HIGH_BITRATE" />
  <RESOURCE REF="http://www.youtube.com/watch?v=_AmkmqYEarw"
    TYPE="audio/mp3" SIZE="" />
  <ANCHOR PRECEDENCE="100">
    <DESCRIPTOR>
      <STATEMENT TYPE="text/text">
        Play Song
      </STATEMENT>
    </DESCRIPTOR>
  </ANCHOR>
</COMPONENT>
<COMPONENT NAME="Thorn_In_My_Side">
  <CONDITION REQUIRE="GET_LYRICS" />
  <CONDITION REQUIRE="WANT_EXTRA_CONTENT" />
  <RESOURCE
    REF="http://www.lyricsfreak.com/e/eurythmics/thorn+in+my+
      side_20051758.html"
    TYPE="text/html" SIZE="" />
  <ANCHOR PRECEDENCE="25">
    <DESCRIPTOR>
      <STATEMENT TYPE="text/text">
        View Lyrics
      </STATEMENT>
    </DESCRIPTOR>
  </ANCHOR>
</COMPONENT>
</ITEM>
<ITEM>
  <CONDITION REQUIRE="SONG4" />
  <DESCRIPTOR>
    <STATEMENT TYPE="http://www.dmu.com/content-organizer-hints">
      DMU 9876 :: Item Type="Song";
  </STATEMENT>
  </DESCRIPTOR>
</ITEM>

```

```

        </STATEMENT>
    </DESCRIPTOR>
    <DESCRIPTOR ID="SONG4_TITLE">
        <STATEMENT TYPE="text/xml">
            <rdf:RDF>
                <rdf:Description>
                    <dc:title>Heart And Soul</dc:title>
                </rdf:Description>
            </rdf:RDF>
        </STATEMENT>
    </DESCRIPTOR>
    <DESCRIPTOR>
        <STATEMENT TYPE="text/xml">
            <rdf:RDF>
                <rdf:Description>
                    <dc:coverage>235</dc:coverage>
                </rdf:Description>
            </rdf:RDF>
        </STATEMENT>
    </DESCRIPTOR>
    <DESCRIPTOR>
        <REFERENCE URI="#ALBUM_RATING" />
    </DESCRIPTOR>
    <DESCRIPTOR>
        <REFERENCE URI="#RIGHTS" />
    </DESCRIPTOR>
    <COMPONENT NAME="Heart_And_Soul">
        <CONDITION REQUIRE="LOW_BITRATE" />
        <RESOURCE REF="http://www.youtube.com/watch?v=dZfFgQDy38I"
            TYPE="audio/mp3" SIZE="" />
        <ANCHOR PRECEDENCE="50">
            <DESCRIPTOR>
                <STATEMENT TYPE="text/text">
                    Play Song
                </STATEMENT>
            </DESCRIPTOR>
        </ANCHOR>
    </COMPONENT>
    <COMPONENT NAME="Heart_And_Soul">
        <CONDITION REQUIRE="HIGH_BITRATE" />
        <RESOURCE REF="http://www.youtube.com/watch?v=dZfFgQDy38I"
            TYPE="audio/mp3" SIZE="" />
        <ANCHOR PRECEDENCE="100">
            <DESCRIPTOR>
                <STATEMENT TYPE="text/text">
                    Play Song
                </STATEMENT>
            </DESCRIPTOR>
        </ANCHOR>
    </COMPONENT>
    <COMPONENT NAME="Heart_And_Soul">

```



```
<CONDITION REQUIRE="GET.LYRICS" />
<CONDITION REQUIRE="WANT.EXTRA.CONTENT" />
<RESOURCE
  REF="http://www.elyrics.net/read/t/t-pau-lyrics/heart-and-soul-
    lyrics.html"
  TYPE="text/html" SIZE="" />
<ANCHOR PRECEDENCE="25">
  <DESCRIPTOR>
    <STATEMENT TYPE="text/text">
      View Lyrics
    </STATEMENT>
  </DESCRIPTOR>
</ANCHOR>
</COMPONENT>
</ITEM>
</ITEM>
</CONTAINER>
</DIDL>
```

Πίνακας Συμβόλων και Συντμήσεων

Συντομογραφία	Περιγραφή	Χρήση στο Κείμενο
AFS	Andrew File System	page 3
BPMN	Business Process Model and Notation	page 41
DIDL	Digital Item Declaration Language	page 37
FUSE	File System in Userspace	page 10
MPEG	Moving Picture Experts Group	page 23
NFS	Network File System	page 3
OASIS	Organization for the Advancement of Structured Information Standards	page 48
RPC	Remote Procedure Call	page 18
SSSD	System Security Services Daemon	page 45
SOA	Service Oriented Architecture	page 43
UFID	Unique File Identifier	page 17
URL	Uniform Resource Location	page 36
VFS	Virtual File System	page 9

Πίνακας Εικόνων

2.1	Ενδεικτικό αποτέλεσμα εντολής ls -l.	7
2.2	Δομή του FUSE [4].	10
3.1	Δομή του NFS [17].	19
3.2	Δομή του OpenAFS [6].	20
5.1	Αρχιτεκτονική EutereFS.	36
5.2	Αφαιρετική Ενορχήστρωση Ανάκτησης Απομακρυσμένου Καταλόγου.	41
5.3	Ενορχήστρωση Ανάγνωσης Καταλόγου.	42
5.4	Ενορχήστρωση Ανάκτησης Χαρακτηριστικών των Αρχείων.	42
5.5	Ενορχήστρωση Ανάγνωσης Αρχείου.	43