

Τεχνικό εγχειρίδιο
για την εφαρμογή
Ηλεκτρονικής κατάθεσης αιτήσεων

στον

ΟΡΓΑΝΙΣΜΟ ΒΙΟΜΗΧΑΝΙΚΗΣ ΙΔΙΟΚΤΗΣΙΑΣ

Αναθεώρηση Ιούλιος 2019

Εργο	Ηλεκτρονική κατάθεση αιτήσεων στον ΟΒΙ efiling.obι.gr
Φορέας	Οργανισμός Βιομηχανικής Ιδιοκτησίας obi.gr
Φορέας υλοποίησης	ΕΚΕΦΕ ΔΗΜΟΚΡΙΤΟΣ ΕΠΕΤΥΠ Δικτύων - Ινστιτούτο Πληροφορικής & Τηλεπικοινωνιών ycor@iit.demokritos.gr iit.demokritos.gr
Επιστημονικός Υπεύθυνος	Δρ.Ιωάννης Κοροβέσης ΕΚΕΦΕ ΔΗΜΟΚΡΙΤΟΣ ΕΠΕΤΥΠ Δικτύων - Ινστιτούτο Πληροφορικής & Τηλεπικοινωνιών ycor@iit.demokritos.gr
Συντάκτες	Παναγιώτης Σκαρβέλης sl45sms@yahoo.gr Επιμέλεια : Μάρω Φασσέα mfassea@gmail.com

Περιεχόμενα

Εισαγωγή

1.1. Σκοπός

1.2. Αρχιτεκτονική και εργαλεία

1.3. Σύνοψη παρουσίαση των εργαλείων

1.3.1. Σύστημα διεπαφής/ δημιουργίας φορμών

1.3.1.1. Το εργαλείο

1.3.2. Συστήματα Enterprise Content Management – ECM

1.3.2.1. Γενικά

1.3.2.2. Alfresco

1.3.3. Business Process Modeling BPM

1.3.3.1. Το πρότυπο BPMN

1.3.3.2. BONITA με χρήση προτύπου BPMN

1.3.4. Σύνδεσμοι για τεκμηρίωση και εγχειρίδια των εργαλείων

2. Εγκατάσταση

2.1. Γενικά για την εγκατάσταση

2.2. Versioning

3. Προδιαγραφές για την υποστήριξη συντήρηση

3.1. Γενικά

3.2. Χρόνοι ανταπόκρισης και επίλυσης

Τεκμηρίωση του χρόνου ανταπόκρισης

Χρόνοι ανταπόκρισης και επίλυσης

3.3. Διαχειριστικά

3.4. Απαιτούμενες γνώσεις και δεξιότητες του Αναδόχου

3.5. Troubleshooting

3.6. Διάγραμμα bonita

3.7. Υπόδειγμα δελτίου δυσλειτουργίας

1.Εισαγωγή

1.1. Σκοπός

Το παρόν κείμενο αποτελεί το εγχειρίδιο του προγραμματιστή (Developers Guide) του συστήματος «Ηλεκτρονική κατάθεση αιτήσεων στον Οργανισμό Βιομηχανικής Ιδιοκτησίας». Αποσκοπεί στο να ενημερώσει και διευκολύνει τους τεχνικούς που θα κληθούν να υποστηρίξουν το σύστημα.

Η εφαρμογή αποσκοπεί στην εξυπηρέτηση ηλεκτρονικής κατάθεσης αιτήσεων στον Ο.Β.Ι. από ενδιαφερόμενους πολίτες ή/και φορείς.

Περιγραφή της εφαρμογής μπορείτε να βρείτε στο επισυναπτόμενο παράρτημα που περιέχει την ανάλυση καθώς και στο github (<https://github.com/OBI-GRIPO/e-filing>).

Η χρήση του παρόντος και γενικά η υποστήριξη της εφαρμογής προϋποθέτει πολύ καλή γνώση και εμβάθυνση στη χρήση των χρησιμοποιηθέντων εργαλείων.

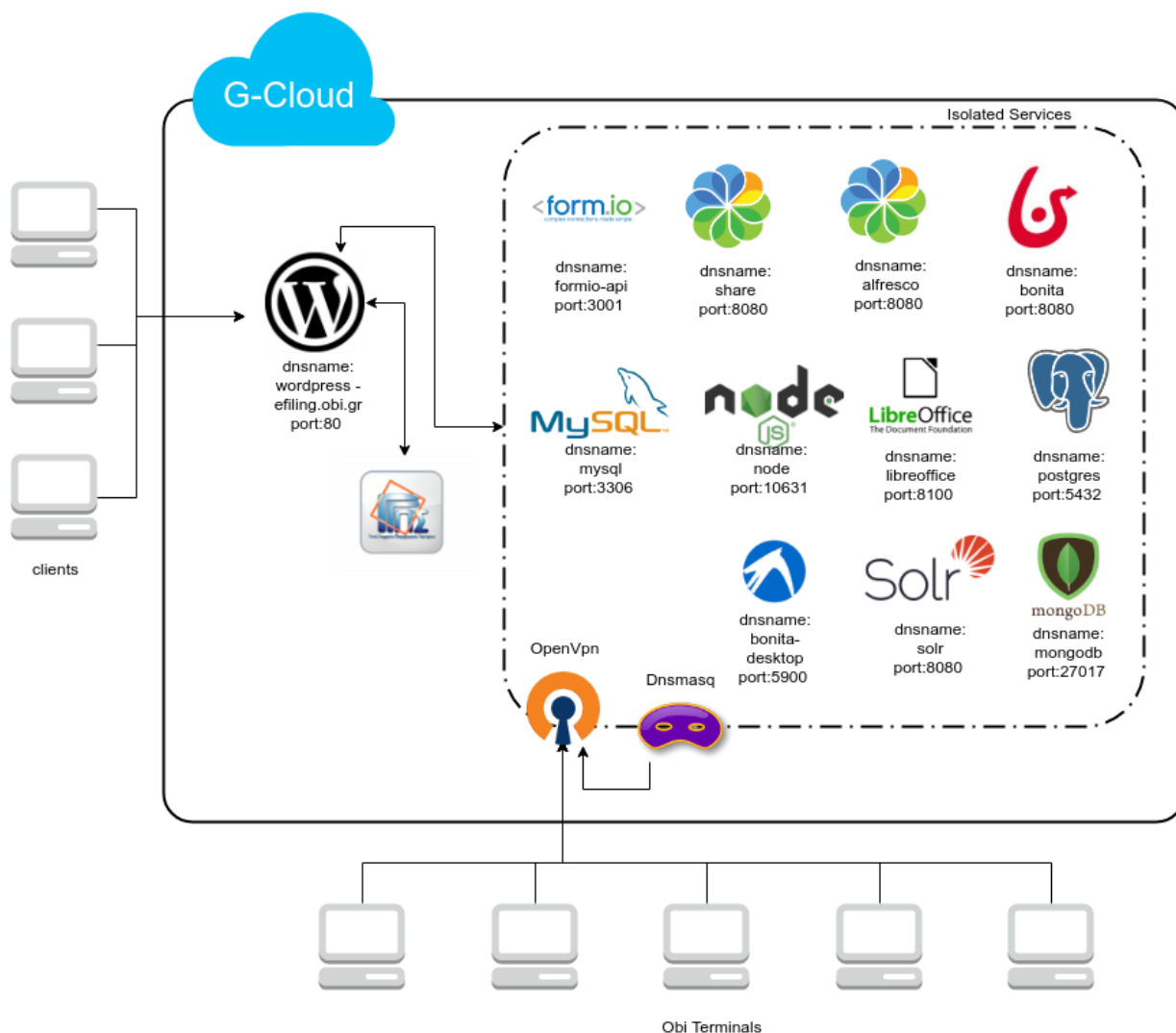
1.2. Αρχιτεκτονική και εργαλεία

Το σύστημα είναι μία εφαρμογή για την απομακρυσμένη κατάθεση αιτήσεων από πλευράς των ενδιαφερομένων για την κτήση τίτλων ευρεσιτεχνίας και άλλων συναφών ζητημάτων. Συνδυάζει έναν αριθμό εργαλείων Ελεύθερου Λογισμικού/ Λογισμικού Ανοικτού Κώδικα (ΕΛ/ΑΚ), παρέχοντας ένα ενιαίο περιβάλλον εργασίας με σκοπό την εκτέλεση των παρακάτω λειτουργιών:

- Εγγραφή χρηστών και ταυτοποίηση μέσω συστημάτων ΓΓΠΣ ή ψηφιακής υπογραφής
- Κατάθεση αιτήσεων και αναγκαίων εγγράφων
- Πληρωμή μέσω τραπεζής

- Παραλαβή αιτήσεων και συνακολούθων εγγράφων από τα εμπλεκόμενα τμήματα
- Αποστολή αιτήσεων στο σύστημα Πρωτοκόλλου

Τα κύρια τεχνικά μέρη του συστήματος είναι (Σχήμα 1):



Σχήμα 1

- Το Περιβάλλον Διεπαφής είναι ο ιστότοπος efilong.obi.gr που έχει υλοποιηθεί σε Wordpress και προσφέρει στο χρήστη ένα πλήρες περιβάλλον για την εκτέλεση των απαιτούμενων λειτουργιών. Οι οθόνες/φόρμες που υποδέχονται τις αιτήσεις έχουν υλοποιηθεί με το εργαλείο formio.
- Για τη διαχείριση των εγγράφων χρησιμοποιείται το εργαλείο alfresco.

Για τη διαχείριση των διαδικασιών χρησιμοποιείται το εργαλείο bonita.

Ακόμη

- oauth 2 provider : Ο πάροχος διαδικασίας ταυτοποιήσεων χρηστών, εν προκειμένω η Γενική Γραμματεία Πληροφοριακών συστημάτων. Όπως φαίνεται και στην ανάλυση η ταυτοποίηση γίνεται είτε μέσω ΓΓΠΣ είτε με ψηφιακή υπογραφή.
- Payment gateway : πάροχος διαδικασίας πληρωμών, εν προκειμένω η ALPHA BANK
- Το form.io εργαλείο φορμών της εφαρμογής
- Τα δεδομένα των αιτήσεων αποθηκεύονται σε βάση δεδομένων MongoDB.
- Το Bonita για τη διαχείριση των διαδικασιών
- Το alfresco χρησιμοποιείται για τη διαχείριση εγγράφων

Στο διάγραμμα διακρίνονται και οι διεπαφές μεταξύ των επιμέρους συστημάτων.

Στο κάτω μέρος του σχήματος παρουσιάζεται μια εξαπλουστευμένη, χάριν διευκόλυνσης της κατανόησης, διαδικασία όπου :

- ο αιτών επιλέγει τύπο αίτησης τη συμπληρώνει και την υποβάλλει
- το σύστημα αποθηκεύει τα συνοδευτικά έγγραφα στο alfresco
- στο σύστημα τα έγγραφα μετατρέπονται σε pdf και σημαίνονται
- ο εντεταλμένος υπάλληλος ελέγχει τα έγγραφα και στέλνει τη σχετική απόδειξη
- ο αιτών πληρώνει μέσω του παρόχου πληρωμών (Τράπεζα)
- το σύστημα αποδίδει αριθμό κατάθεσης
- το σύστημα αποστέλλει έγκριση

Από το σχήμα απουσιάζει το wordpress το οποίο χρησιμοποιήθηκε για την κατασκευή του ιστοτόπου.

1.3. Σύντομη παρουσίαση των εργαλείων

1.3.1. Σύστημα διεπαφής/ δημιουργίας φορμών

1.3.1.1. Το εργαλείο

Για τη δημιουργία φορμών υιοθετείται το ανοιχτό εργαλείο <form.io>.

<https://github.com/formio/ngFormBuilder>

<https://help.form.io/developer/api/>

<https://form.io/>

Πρόκειται για μια συνδυαστική πλατφόρμα διαχείρισης φορμών και δεδομένων για διαδικτυακές εφαρμογές. Επιτρέπει σε εκείνους που το χρησιμοποιούν να δημιουργούν ταχύτατα φόρμες και πόρους με τη χρήση ενός απλού form builder που λειτουργεί με τη λογική drag-and-drop, δηλαδή σέρνοντας απλά με το ποντίκι.

Η δημιουργία των φορμών παράγει αυτόματα ένα JSON schema που χρησιμοποιείται από κοινού

- για τη δυναμική εμφάνιση των φορμών, καθώς επίσης και
- για την αυτόματη παραγωγή του αντίστοιχου API για τη διαχείριση των δεδομένων κατά την υποβολή της φόρμας.

Το εργαλείο έχει δοκιμαστεί ήδη σε έναν μακρύ κατάλογο διαφόρων ειδών εφαρμογών με πολυποίκιλες απαιτήσεις.

Προτυποποιημένες οθόνες διευκολύνουν τη σχεδίαση και τη σύνδεση με το πληροφοριακό σύστημα. Ακόμη, ως αποτέλεσμα ο απαιτούμενος παραδοσιακά χρόνος συνεννοήσεων γραπτών και προφορικών, αναθεωρήσεων και εγκρίσεων μειώνεται δραστικά. Εκτός από το αμιγώς λειτουργικό μέρος, οι φόρμες είναι ευγανάγνωστες και με μελετημένη οργάνωση έτσι ώστε να είναι πιο φιλικές στο χειρισμό τους και πιο “εύπεπτες” στην κατανόηση.

Στη χαρακτηριστική μπλε γραμμή στο άνω μέρος αποτυπώνονται οι επιμέρους σελίδες, οι καρτέλες (tabs), και από εκεί ο χρήστης μεταφέρεται στις αντίστοιχες ενότητες.

Με τη χρήση του Form Builder κατά τη σχεδίαση της φόρμας δημιουργείται ταυτόχρονα και αυτόματα πλατφόρμα API (swagger, openAPI) για χρήση CRUD από τρίτους. Με απλά λόγια, αυτό επιτρέπει τη δημιουργία συστημάτων διεπαφής εφαρμογών από τρίτους κατασκευαστές.

1.3.2. Συστήματα Enterprise Content Management – ECM

1.3.2.1. Γενικά

Το ECM (διαχείριση επιχειρηματικού περιεχομένου) είναι ένας όρος - ομπρέλα που καλύπτει:

- τη διαχείριση εγγράφων,
- τη διαχείριση περιεχομένου ιστού,
- την αναζήτηση,
- τη συνεργασία,
- τη διαχείριση αρχείων,
- τη διαχείριση ψηφιακών στοιχείων (DAM),
- τη διαχείριση ροής εργασίας,
- τη λήψη και τη σάρωση.

Το βασικό χαρακτηριστικό ενός ECM συστήματος που το διακρίνει και το χαρακτηρίζει ως ευφυές σε σχέση με την «απλή» διαχείριση περιεχομένου είναι το γεγονός ότι το ECM διασφαλίζει τη σύνδεση μεταξύ της αδόμητης έντυπης και ψηφιακής πληροφορίας με τις διαδικασίες, τις εφαρμογές και τους χρήστες. Είναι, δηλαδή, ενήμερο για τις διαδικασίες της επιχείρησης για την οποία δημιουργείται και ως εκ τούτου είναι προσαρμοσμένο στις ανάγκες της.

Οι πλατφόρμες αυτές μπορούν κατά κανόνα να ενσωματωθούν στο υπάρχον περιβάλλον (Email, ERP, CRM, e-shop, κ.λ.π.) και να αντιμετωπίσουν εκσυγχρονιστικές προκλήσεις, όπως επέκταση σε επιταλάμιες κινητές συσκευές (π.χ. κινητά τηλέφωνα, ταμπλέτες κ.λ.π.) και προσαρμογή στην τεχνολογία νέφους.

Η οργάνωση των εγγράφων – και η διατήρησή τους με αυτόν τον τρόπο – αποτελεί νευραλγικό παράγοντα σωστής οργάνωσης στις περιπτώσεις όπου απαιτείται

- διαφανής πρόσβαση,
- πρόσβαση από πολλά τμήματα ταυτόχρονα,
- εξασφαλισμένη ταχύτητα στην πρόσβαση σε όλες τις σχετικές πληροφορίες
- πρόσβαση μέσα από διαφορετικές εφαρμογές.
- δυνατότητα υποστήριξης όπου είναι απαραίτητο να υποστηρίζεται ολόκληρου του κύκλου ζωής των κρίσιμων εγγράφων .

Για το υλοποιηθέν σύστημα, ως πλατφόρμα ECM χρησιμοποιείται ο συνδυασμός του Alfresco και του Bonita.

.1.3.2.2. Alfresco

Η Alfresco παρέχει μια σύγχρονη, στρατηγική πλατφόρμα υβριδικής διαχείρισης περιεχομένου επιχειρήσεων (ECM) που έχει δυνατότητες να καλύπτει τις πολύπλοκες -και συχνά αντικρουόμενες- απαιτήσεις, τόσο των τμημάτων πληροφορικής όσο και των χρηστών. Έχει σχεδιαστεί για επιχειρήσεις και οργανισμούς όπου απαιτούνται αρχιτεκτονικές σπονδυλωτής δομής (modularity) και επεκτασιμότητας.

Πρόκειται για ισχυρή πλατφόρμα, η οποία μέσω του API της συνεργάζεται με άλλες πλατφόρμες. Μπορεί να αναπτυχθεί σε τοπικές εγκαταστάσεις διακομιστών ή/ και στο νέφος (cloud).

Η πλατφόρμα της Alfresco αποτελείται από τρία βασικά προϊόντα:

- Alfresco Content Services (ACS): Διαχείριση περιεχομένου επιχειρήσεων (ECM), που αποτελεί και το βασικό κορμό του Alfresco. Περιλαμβάνει κεντρικό αποθετήριο περιεχομένου και μεταδεδομένων, διεπαφή ιστού με το όνομα Share, δυνατότητα ορισμού αυτοματοποιημένων επιχειρηματικών κανόνων και ευρετηρίαση πλήρους κειμένου που παρέχεται με τη χρήση του Apache Solr.
- Alfresco Process Services (APS): Διαχείριση επιχειρηματικών διαδικασιών (BPM) που απορρέουν από το έργο Activiti.
- Alfresco Governance Services (AGS): Γνωστό ως Alfresco Records Management, το AGS είναι ένα πρόσθετο λογισμικό που παρέχει λειτουργίες διαχείρισης αρχείων.

Όπως προαναφέρθηκε, ο συνδυασμός Bonita BPM και Alfresco ACS αποτελεί ένα πλήρες, επεκτάσιμο και χρηστικό ECM σύστημα.

1.3.3. Business Process Modeling BPM

Με τον όρο Επιχειρηματική Μοντελοποίηση Διαδικασιών (Business Process Modeling - BPM), στον τομέα διαχείρισης επιχειρησιακών διαδικασιών (business process management) και στη

μηχανική πληροφοριακών συστημάτων (systems engineering), περιγράφεται η δραστηριότητα της αποτύπωσης και παρουσίασης των διαδικασιών μιας επιχείρησης με τρόπο ώστε η τρέχουσα διαδικασία να είναι δυνατόν να αναλυθεί, βελτιωθεί και να αυτοματοποιηθεί.

1.3.3.1. Το πρότυπο BPMN

Η τεχνική της Μοντελοποίησης των Διαδικασιών (Process Modeling) βασίζεται στη λογική των γραφικών μεθόδων απεικόνισης. Τα γραφικά αποτυπώνουν και αναδεικνύουν την απαραίτητη λεπτομέρεια και είναι πολύ πιο κατάλληλα για παρουσίαση, έλεγχο και επισκόπηση της ροής των δραστηριοτήτων και των παράλληλων εργασιών απ' ό,τι τα κείμενα.

Το πρότυπο της BPMN είναι ένα ανοιχτό πρότυπο για μοντελοποίηση, εφαρμογή και αναπαράσταση επιχειρησιακών διαδικασιών. Με την προτυποποίηση της BPMN δίνεται στο χρήστη η δυνατότητα να κατασκευάσει Διαγράμματα Επιχειρησιακών Διαδικασιών (Business Process Diagrams), τα οποία με τη σειρά τους απευθύνονται σε χρήστες που σχεδιάζουν και διαχειρίζονται διαφορετικές επιχειρησιακές διαδικασίες.

Παράλληλα, η BPMN εξασφαλίζει μια έτοιμη έγκυρη χαρτογράφηση προς μια εκτελέσιμη γλώσσα. Η χρήση "χαρτογράφησης" γεφυρώνει το χάσμα που δημιουργείται από τη μία από τα αρχικά διαγράμματα ροής, τα οποία συνήθως σχεδιάζουν οι αναλυτές στην προσπάθειά τους να περιγράψουν τις διαδικασίες υποβολής, και από την άλλη από εργασίες ανάπτυξης.

Για παράδειγμα, με τη χρήση BPMN, οι ροές των διαδικασιών μπορούν να αναπαρασταθούν σε διαγράμματα τα οποία αντικατοπτρίζουν κάθε δραστηριότητα, σε φυσικό επίπεδο δραστηριοτήτων και όχι σε γλώσσα προγραμματισμού ή αρχείο περιγραφής, όπως για παράδειγμα είναι ένα xml. Αυτό έχει ως συνέπεια τη δυνατότητα ανάγνωσης των σταδίων της ροής διαδικασίας από όλους και όχι μόνο τους προγραμματιστές, με ό,τι αυτό σημαίνει για τις δυνατότητες διαχείρισης, βελτίωσης και αναθεώρησης.

1.3.3.2. BONITA με χρήση προτύπου BPMN

Στο υλοποιηθέν σύστημα έχει χρησιμοποιηθεί το πρόγραμμα Bonita ως ολοκληρωμένη σουίτα εφαρμοσμένου BPM.

Η σουίτα Bonita BPM διαχειρίζεται επιχειρηματικές διαδικασίες και ροές εργασιών, με τη χρήση ανοικτού κώδικα που έχει δημιουργηθεί από το 2001. Δίνει τη δυνατότητα δημιουργίας διαχειριστικών φορμών και διαθέτει υποδομές διασύνδεσης με όλες σχεδόν τις υπηρεσίες που μπορεί να χρειαστεί ένα σύστημα, όπως:

- βάσεις δεδομένων,
- web services τρίτων,
- mail servers,
- συστήματα αποθήκευσης αρχείων,
- συστήματα οργάνωσης κ.λ.π.

Το Bonita BPM έχει τρία βασικά δομικά στοιχεία:

- Bonita Studio:

Το Bonita Studio δίνει τη δυνατότητα στο χρήστη να δημιουργεί γραφικά επιχειρηματικών διαδικασιών σύμφωνα με το πρότυπο BPMN.

Ο χρήστης μπορεί επίσης να συνδέσει διαδικασίες με άλλα τμήματα του πληροφοριακού συστήματος όπως για παράδειγμα

- μηνύματα,
- προγραμματισμός επιχειρηματικών πόρων,
- διαχείριση εταιρικού περιεχομένου και
- βάσεις δεδομένων

έτσι ώστε να δημιουργηθεί μια αυτόνομη επιχειρηματική εφαρμογή προσβάσιμη ως φόρμα ιστού.

Το Bonita Studio έχει και αυτό δυνατότητα σχεδιασμού φορμών με γραφικό τρόπο για χρήση του τελικού χρήστη. Χρήση αυτής της δυνατότητας μπορεί*****

- Bonita BPM Engine:

Ο μηχανισμός BPM είναι μια εφαρμογή Java που εκτελεί τις ροές διαδικασιών που δημιουργούνται με το Bonita Studio.

Το API της μηχανής επιτρέπει στο χρήστη να αλληλεπιδρά προγραμματιστικά με τις διαδικασίες της.

- Bonita Portal:

Πρόκειται για μια πύλη που δίνει τη δυνατότητα σε κάθε τελικό χρήστη να διαχειρίζεται τις υποθέσεις του και όλες τις εργασίες στις οποίες εμπλέκεται, καθώς και να έχει οργανωμένες διεπαφές όπως το webmail. Η πύλη επιτρέπει επίσης στον υπεύθυνο μιας διαδικασίας να

διαχειρίζεται και να λαμβάνει αναφορές σχετικά με αυτήν.

1.3.4. Σύνδεσμοι για τεκμηρίωση και εγχειρίδια των εργαλείων

Form-IO

[Users' guide](#)

Λοιπά

<https://github.com/formio/formio>

<https://gist.github.com/travist/ad065eccc474ce33177cecbcb02b299>

Bonita

[Documentation](#)

και

<https://www.vicon.com/downloads/documentation/bonita-quick-start-guide>

<https://github.com/bonitasoft>

Alfresco

<https://docs.alfresco.com/>

Και

<https://www.alfresco.com/blogs/ask-and-you-shall-receive-improved-alfresco-online-documentation/>

Wordpress

<https://wordpress.org/news/category/documentation/>

2. Εγκατάσταση

Σημείωση : η κατανόηση και η αξιοποίηση των παρακάτω απαιτεί εξοικείωση με την ορολογία και τις δυνατότητες του docker (<https://www.docker.com/>).

2.1. Γενικά για την εγκατάσταση

Το σύστημα έχει υλοποιηθεί σε docker swarm, το οποίο έχει αναπτυχθεί σε 5 νη που έχουν διατεθεί προρυθμισμένα με λειτουργικό Linux-Ubuntu LTS στο g-cloud από την ΚτΠ. Η διανομή των nodes του docker swarm έχει επιμεριστεί με βάση το φόρτο.

Ένα εκ των νη ρυθμίστηκε ως NFS Server για την αποθήκευση των δεδομένων που παράγονται από τα υποσυστήματα. Το ίδιο νη αποτελεί και node του docker swarm εξυπηρετώντας τις απαιτούμενες βάσεις δεδομένων MongoDB και MySQL.

Ενα δεύτερο νη ρυθμίστηκε ως docker registry για τη φιλοξενία των images και το versioning των συστημάτων. Το συγκεκριμένο registry νη αποτελεί και το node που εξυπηρετεί το υποσύστημα bonita.

Ένα εκ των νη ρυθμίστηκε ως docker swarm master και ταυτόχρονα εξυπηρετεί και την λειτουργία των υποσυστημάτων wordpress, formio.

Τα άλλα δύο έχουν μοιραστεί για τις λειτουργίες του alfresco.

Το σύνολο των αναφερομένων λειτουργιών περιγράφονται αναλυτικά στο αρχείο docker-composer.yml ως κάτωθι :

```
version: '3.6'
```

```
volumes:
```

```
registry-volume:
```

```
driver: local
```

```
driver_opts:
```

```
type: nfs
```

```
o: "addr=xxx.xxx.xxx.xxx,hard,sec=sys,tcp,rw"
```

```
device: "[:/media/hdd2/obi_registry-volume"
```

```
assets:
```

```
driver: local
```

```
driver_opts:
```

```
type: nfs
```

```
o: "addr=xxx.xxx.xxx.xxx,hard,sec=sys,tcp,rw"
```

```
device: "[:/media/hdd2/obi_assets"
```

```
formio-volume:
```

```
driver: local
```

```
driver_opts:
  type: nfs
  o: "addr=xxx.xxx.xxx.xxx,hard,sec=sys,tcp,rw"
  device: "/media/hdd2/obi_formio-volume"

bonita-volume:
  driver: local
  driver_opts:
    type: nfs
    o: "addr=xxx.xxx.xxx.xxx,nolock,hard,sec=sys,tcp,rw"
    device: "/media/hdd2/obi_bonita-volume"

alfresco-data-volume:
  driver: local
  driver_opts:
    type: nfs
    o: "addr=xxx.xxx.xxx.xxx,hard,sec=sys,tcp,rw"
    device: "/media/hdd2/obi_alfresco-data-volume"

alfresco-tomcat-logs:
  driver: local
  driver_opts:
    type: nfs
    o: "addr=xxx.xxx.xxx.xxx,soft,nolock,sec=sys,tcp,rw"
    device: "/media/hdd2/obi_alfresco-tomcat-logs"

alfresco-share-logs:
  driver: local
  driver_opts:
    type: nfs
    o: "addr=10.9.134.215,soft,nolock,sec=sys,tcp,rw"
    device: "/media/hdd2/obi_alfresco-share-logs"

alfresco-sorl-logs:
  driver: local
  driver_opts:
    type: nfs
    o: "addr=xxx.xxx.xxx.xxx,soft,nolock,sec=sys,tcp,rw"
    device: "/media/hdd2/obi_alfresco-sorl-logs"

alfresco-sorl-data:
  driver: local
  driver_opts:
    type: nfs
    o: "addr=xxx.xxx.xxx.xxx,hard,sec=sys,tcp,rw"
    device: "/media/hdd2/obi_alfresco-sorl-data"

ssl-certs:
  driver: local
  driver_opts:
    type: nfs
    o: "addr=xxx.xxx.xxx.xxx,soft,nolock,sec=sys,tcp,rw"
    device: "/media/hdd2/obi_ssl-certs"

wp-content:
  driver: local
  driver_opts:
```

```

type: nfs
o: "addr=xxx.xxx.xxx.xxx,soft,nolock,sec=sys,tcp,rw"
device: "/media/hdd2/obi_wp-content"

networks:
obi_net:
  external: true

services:

obiregistry:
  image: registry:2
  volumes:
    - registry-volume:/var/lib/registry
    - /mnt/registry/certs:/certs
  ports:
    - 5001:5001
  networks:
    - obi_net
  environment:
    REGISTRY_HTTP_ADDR: "0.0.0.0:5001"
    REGISTRY_HTTP_TLS_CERTIFICATE: "/certs/domain.crt"
    REGISTRY_HTTP_TLS_KEY: "/certs/domain.key"
  deploy:
    replicas: 1
    restart_policy:
      delay: 25s
      window: 120s
    placement:
      constraints: [node.hostname == 050-OBI-DOCKER-REGISTRY]

postgres:
  image: postgres:9.4
  networks:
    - obi_net
  volumes:
    - /media/hdd2/obi_postgres-volume:/var/lib/postgresql/data
    - /media/hdd2/obi_assets/docker-entrypoint-initdb.d:/docker-entrypoint-initdb.d
  environment:
    - POSTGRES_PASSWORD={--postgres_password--}
    - ALFRESCO_DB=somedb
    - ALFRESCO_DB_USER={--alfresco_db_user--}
    - ALFRESCO_DB_USER_PASSWORD={--alfresco_db_password--}
    - BONITA_DB_NAME=somedb
    - BONITA_DB_USER=someuser
    - BONITA_DB_PASS={--bonita_db_pass--}
    - BONITA_BIZ_DB_NAME=somedbdb
    - BONITA_BIZ_DB_USER=someuser
    - BONITA_BIZ_DB_PASS={--bonita_db_biz_pass--}
  command:
    - -c
    - max_prepared_transactions=100
  deploy:
    replicas: 1
    restart_policy:
      delay: 0s
      window: 120s

```

placement:
constraints: [node.hostname == 050-OBI-FILESERVER]

mongodb:
image: mongo
networks:
- obi_net
volumes:
- /media/hdd2/obi_mongo-volume:/data/db
deploy:
replicas: 1
restart_policy:
delay: 0s
window: 120s
placement:
constraints: [node.hostname == 050-OBI-FILESERVER]

formio-api:
image: 050-OBI-DOCKER-REGISTRY:5001/formio-api:1.2
build:
context: \${PWD}/vendors/dfio
networks:
- obi_net
extra_hosts:
- "{--formio_host--}: {--formio_host_ip--}"
environment:
- VIRTUAL_PROTOCOL=http
- VIRTUAL_HOST=formio-api
- VIRTUAL_PORT=3001
- DB_HOST=mongodb
- DB_NAME=formio
- ADMIN_EMAIL={--formio_admin_email--}
- ADMIN_PASS={--formio_admin_password--}
- JWT_SECRET={--formio_jwt_secret--}
depends_on:
- mongodb
volumes:
- formio-volume:/app
- /mnt/waitmongodb.sh:/waitmongodb.sh
deploy:
replicas: 1
restart_policy:
delay: 25s
window: 120s
placement:
constraints: [node.hostname == 050-OBI-DOCKER02]
entrypoint:
- bash
- -c
- |
set -e
echo 'Waiting for MongoDB to be available'
aa=\$(/waitmongodb.sh &>/dev/null || true)
exec /start.sh

fio2pdf:
image: 050-OBI-DOCKER-REGISTRY:5001/fio2pdf:1.4


```

build:
  context: ${PWD}/vendors/fio2pdf
networks:
  - obi_net
extra_hosts:
  - "${--formio_host--}: ${--formio_host_ip--}"
environment:
  - FORMIO_ADMIN_EMAIL=${--formio_admin_email--}
  - FORMIO_ADMIN_PASSWORD=${--formio_admin_password--}
depends_on:
  - formio-api
deploy:
  replicas: 1
  restart_policy:
    delay: 25s
    window: 120s
  placement:
    constraints: [node.hostname == 050-OBI-DOCKER04]

payum:
image: 050-OBI-DOCKER-REGISTRY:5001/payumserver:1.2
build:
  context: ${PWD}/vendors/payum
networks:
  - obi_net
extra_hosts:
  - "${--formio_host--}: ${--formio_host_ip--}"
environment:
  - PAYUM_MONGO_URI=mongodb://mongodb:27017/payum_server
  - PAYUM_DEBUG=0
depends_on:
  - mongodb
volumes:
  - ${PWD}/assets/payum/public:/payum/public
  - ${PWD}/assets/payum/.env:/payum/.env
deploy:
  replicas: 1
  restart_policy:
    delay: 25s
    window: 120s
  placement:
    constraints: [node.hostname == 050-OBI-DOCKER01]
entrypoint:
  - bash
  - -c
  - |
    set -e
    echo 'Waiting for MongoDB to be available'
    aa=$(/waitmongodb.sh &>/dev/null || true)
    exec /entrypoint.sh

payumui:
image: payum/server-ui
networks:
  - obi_net
depends_on:
  - payum

```

```

deploy:
  replicas: 1
  restart_policy:
    delay: 25s
    window: 120s
  placement:
    constraints: [node.hostname == 050-OBI-DOCKER02]

bonita:
  image: bonita
  networks:
    - obi_net
  environment:
    - POSTGRES_ENV_POSTGRES_PASSWORD={--postgres_password--}
    - ENSURE_DB_CHECK_AND_CREATION=false
    - DB_ADMIN_USER=someuser
    - DB_ADMIN_PASS={--postgres_password--}
    - DB_VENDOR=someuser
    - DB_HOST=somehost
    - DB_PORT=5432
    - DB_NAME=somedb
    - DB_USER=someuser
    - DB_PASS={--bonita_db_pass--}
    - BIZ_DB_NAME=somedb
    - BIZ_DB_USER=someuser
    - BIZ_DB_PASS={--bonita_db_biz_pass--}
    - TENANT_LOGIN={--bonita_tenant_login--}
    - TENANT_PASSWORD={--bonita_tenant_password--}
    - PLATFORM_LOGIN={--bonita_platform_login--}
    - PLATFORM_PASSWORD={--bonita_platform_password--}
    - HTTP_API=true
  depends_on:
    - postgres
  volumes:
    - bonita-volume:/opt/bonita
    - /mnt/bonita/certs.crt:/certs.crt
  entrypoint:
    - bash
    - -c
    - |
      set -e
      echo 'set certs'
      keytool -import -alias obismtp -keystore /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/security/cacerts -file /certs.crt -noprompt
-storepass changeit
      echo 'Waiting for Postgres to be available'
      export PGPASSWORD="$${POSTGRES_ENV_POSTGRES_PASSWORD}"
      maxTries=20
      while [ "$${maxTries}" -gt 0 ] && ! psql -h "$${DB_HOST}" -U 'postgres' -c '!'; do
        let maxTries--
        sleep 5
      done
      echo
      if [ "$${maxTries}" -le 0 ]; then
        echo ">&2 'error: unable to contact Postgres after 20 tries'"
        exit 1
      fi
      exec /opt/files/startup.sh

```

```
deploy:
  replicas: 1
  restart_policy:
    delay: 25s
    window: 120s
  placement:
    constraints: [node.hostname == 050-OBI-DOCKER-REGISTRY]
```

```
node:
  image: "node"
  networks:
    - obi_net
  working_dir: /home/node/app
  environment:
    - NODE_ENV=production
  volumes:
    - ${PWD}/obif:/home/node/app
  entrypoint:
    - bash
    - -c
    - |
      groupmod -g 1001 node
      usermod -u 1001 -g 1001 node
      su node -c "npm start"
```

```
command: ""
deploy:
  replicas: 1
  restart_policy:
    delay: 15s
    window: 120s
  placement:
    constraints: [node.role == manager]
```

```
alfresco:
  image: 050-OBI-DOCKER-REGISTRY:5001/alf-platform
  build:
    context: ${PWD}/vendors/alf/alf-platform/
    args:
      - ALFRESCO_DB_USER={--alfresco_db_user--}
      - ALFRESCO_DB_USER_PASSWORD={--alfresco_db_password--}
      - ALFRESCO_ADMIN_PASSWORD={--alfresco_admin_password--}
  environment:
    CATALINA_OPTS: "-Xmx2G -XX:+UseConcMarkSweepGC"
    POSTGRES_ENV_POSTGRES_PASSWORD: "{--postgres_password--}"
    DB_HOST: "somehost"
    ALFRESCO_ADMIN_PASSWORD: "{--alfresco_admin_password--}"
  depends_on:
    - postgres
  networks:
    - obi_net
  extra_hosts:
    - "{--alfresco_host--}:{--alfresco_host_ip--}"
  volumes:
    - alfresco-tomcat-logs:/usr/local/tomcat/logs/
    - alfresco-data-volume:/opt/alf_data
    - /mnt/waitlibreoffice.sh:/mnt/waitlibreoffice.sh
  tmpfs:
```

```

- /tmp
- /usr/local/tomcat/temp/
- /usr/local/tomcat/work/
deploy:
  replicas: 1
  restart_policy:
    delay: 45s
    window: 220s
  placement:
    constraints: [node.hostname == 050-OBI-DOCKER03]
entrypoint:
- bash
- -c
- |
  set -e
  echo 'Waiting for Postgres to be available'
  export PGPASSWORD="${POSTGRES_ENV_POSTGRES_PASSWORD}"
  maxTries=40
  while [ "$maxTries" -gt 0 ] && ! psql -h "$DB_HOST" -U 'postgres' -c '!'; do
    let maxTries--
    sleep 5
  done
  echo
  if [ "$maxTries" -le 0 ]; then
    echo ">&2 'error: unable to contact Postgres after 40 tries'"
    exit 1
  fi
  echo 'Waiting for Libreoffice to be available'
  aa=$(/waitlibreoffice.sh &>/dev/null || true)
  echo "set admin password (just ignore error if alfresco db does not yet exists)"
  psql -U 'postgres' -h "$DB_HOST" alfresco << EOSQL
  UPDATE alf_node_properties
  SET string_value='${ALFRESCO_ADMIN_PASSWORD}'
  WHERE
  node_id= (SELECT anp1.node_id
  FROM alf_node_properties anp1
  INNER JOIN alf_qname aq1 ON aq1.id = anp1.qname_id
  INNER JOIN alf_node_properties anp2 ON anp2.node_id = anp1.node_id
  INNER JOIN alf_qname aq2 ON aq2.id = anp2.qname_id
  WHERE aq1.local_name = 'password'
  AND aq2.local_name = 'username'
  AND anp2.string_value = 'admin')
  AND
  qname_id=(SELECT anp1.qname_id
  FROM alf_node_properties anp1
  INNER JOIN alf_qname aq1 ON aq1.id = anp1.qname_id
  INNER JOIN alf_node_properties anp2 ON anp2.node_id = anp1.node_id
  INNER JOIN alf_qname aq2 ON aq2.id = anp2.qname_id
  WHERE aq1.local_name = 'password'
  AND aq2.local_name = 'username'
  AND anp2.string_value = 'admin');
  EOSQL
  exec /usr/local/tomcat/bin/catalina.sh run

libreoffice:
  image: 050-OBI-DOCKER-REGISTRY:5001/alf-libreoffice
  build:

```

```

    context: ${PWD}/vendors/alf/alf-libreoffice/
networks:
  - obi_net
deploy:
  replicas: 1
  restart_policy:
    delay: 15s
    window: 120s
  placement:
    constraints: [node.hostname == 050-OBI-DOCKER04]

```

solr:

```

image: 050-OBI-DOCKER-REGISTRY:5001/alf-solr
build:
  context: ${PWD}/vendors/alf/alf-solr/
environment:
  CATALINA_OPTS: "-Xms3G -Xmx3G"
depends_on:
  - alfresco
networks:
  - obi_net
volumes:
  - alfresco-solr-logs:/usr/local/tomcat/logs/
  - alfresco-solr-data:/opt/solr
tmpfs:
  - /tmp
  - /usr/local/tomcat/temp/
  - /usr/local/tomcat/work/
deploy:
  replicas: 1
  restart_policy:
    delay: 15s
    window: 120s
  placement:
    constraints: [node.hostname == 050-OBI-DOCKER03]
entrypoint:
  - bash
  - -c
  - |
    echo "Wait alfresco"
    until $(curl --output /dev/null --silent --head --fail http://alfresco:8080/alfresco/); do
      printf '.'
      sleep 5
    done
    exec /usr/local/tomcat/bin/catalina.sh run

```

share:

```

image: 050-OBI-DOCKER-REGISTRY:5001/alf-share
build:
  context: ${PWD}/vendors/alf/alf-share/
environment:
  CATALINA_OPTS: "-Xmx1G -XX:+UseConcMarkSweepGC"
depends_on:
  - alfresco
networks:
  - obi_net
extra_hosts:

```

```

- "${--alfresco_host--}:${--alfresco_host_ip--}"
volumes:
- alfresco-share-logs:/usr/local/tomcat/logs/
tmpfs:
- /tmp
- /usr/local/tomcat/temp/
- /usr/local/tomcat/work/
deploy:
  replicas: 1
  restart_policy:
    delay: 15s
    window: 120s
  placement:
    constraints: [node.hostname == 050-OBIDOCKER04]
entrypoint:
- bash
- -c
- |
  echo "Wait alfresco"
  until $(curl --output /dev/null --silent --head --fail http://alfresco:8080/alfresco/); do
    printf '.'
    sleep 5
  done
  exec /usr/local/tomcat/bin/catalina.sh run

```

```

https-portal:
  image: steveltn/https-portal
  networks:
  - obi_net
  depends_on:
  - mysql
  ports:
  - 80:80
  volumes:
  - ssl-certs:/var/lib/https-portal
  - /mnt/wp/http_config:/var/lib/nginx-conf/efiling.obi.gr.conf.erb:ro
  - /mnt/wp/https_config:/var/lib/nginx-conf/efiling.obi.gr.ssl.conf.erb:ro
  extra_hosts:
  - "slogin.obi.gr:127.0.0.1"
  environment:
    DOMAINS: 'efiling.obi.gr -> http://wordpress'
    STAGE: 'production'
  deploy:
    replicas: 1
    restart_policy:
      delay: 15s
      window: 120s
    placement:
      constraints: [node.role == manager]

```

```

wordpress:
  image: wordpress
  networks:
  - obi_net
  depends_on:
  - mysql

```

```

volumes:
  - wp-content:/var/www/html/wp-content
  - /mnt/wp/setWpDefines.sh:/usr/local/bin/setWpDefines.sh
  - /mnt/wp/etc/php/conf.d/upload.ini:/usr/local/etc/php/conf.d/upload.ini
environment:
  WORDPRESS_DB_PASSWORD: "{--mysql_password--}"
  WORDPRESS_DB_HOST: "mysql:3306"
deploy:
  replicas: 1
  restart_policy:
    delay: 15s
    window: 120s
  placement:
    constraints: [node.role == manager]

mysql:
  image: mysql
  networks:
    - obi_net
  volumes:
    - /media/hdd2/obi_mysql-data:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: "{--mysql_password--}"
  deploy:
    replicas: 1
    restart_policy:
      delay: 15s
      window: 120s
    placement:
      constraints: [node.hostname == 050-OBI-FILESERVER]

```

2.2. Versioning

Σε περίπτωση αλλαγής ροής το Bonita Studio χρησιμοποιείται για το σχεδιασμό των επιχειρηματικών ροών και στην περίπτωση νέας έκδοσης - σχεδιαγράμματος επιχειρηματικής ροής αυτή ανεβαίνει στον bonita server του συστήματος με αύξοντα αριθμό έκδοσης (1.0, 1.1 κ.ο.κ.) και οι νέες υποθέσεις (cases) ξεκινάνε με αυτή την έκδοση.

Η αρίθμηση της υπόθεσης κρατάει τη συνέχειά της. Δηλαδή, V.vv-xxxxxxx όπου το πρώτο μέρος ο αριθμός της έκδοσης και ακολουθεί η κανονική αρίθμηση.

Είθισται η δοκιμαστική έκδοση να φέρει μονό αριθμό (1.5) ενώ η παραγωγής ζυγό (1.6). Ο κανόνας αυτός παραβιάστηκε στην 1.9 έκδοση για να αντιμετωπισθεί ένα επείγον πρόβλημα.

Σε κάθε αλλαγή των φορμών οφείλει ο τεχνικός να πάρει τον παραγόμενο json κώδικα και να τον αναρτήσει στην αντίστοιχη θέση στο github.

3. Προδιαγραφές για την υποστήριξη συντήρηση

3.1. Γενικά

Το σύνολο της εφαρμογής είναι εγκατεστημένο στο G-Cloud όπου παρέχεται εγγύηση απρόσκοπτης λειτουργίας 24Χ365.

Κατά συνέπεια ο ανάδοχος καλείται να υποστηρίξει

1. το/τα λειτουργικό/α
2. τα Εργαλεία ανάπτυξης
3. την Εφαρμογή
4. τη συνδεσιμότητα με τρίτους φορείς (Τράπεζα , ΓΓΠΣ).
5. Το backup της εφαρμογής και των δεδομένων της

3.2. Χρόνοι ανταπόκρισης και επίλυσης

Τεκμηρίωση του χρόνου ανταπόκρισης

Ο χρόνος ανταπόκρισης πρέπει να είναι ο ελάχιστος δυνατός. Ναι μεν το σύστημα δεν εξυπηρετεί επείγουσες ανάγκες συντρέχουν όμως δύο λόγοι υψηλής αποτίμησης

α) η μη λειτουργία του ιστοτόπου πλήττει το κύρος του Οργανισμού έννοια που δεν αποτιμάται σε χρήματα αλλά είναι υψηλού κόστους για τον Οργανισμό.

β) η μη λειτουργία του ιστοτόπου μπορεί να οδηγήσει τον ενδιαφερόμενο (αιτούμενο) στο να χάσει σημαντικές προθεσμίες που μπορεί να αποβούν μοιραίες για τα συμφέροντά του. Ετσι, αντιδικώντας με τον Οργανισμό, ο τελευταίος μπορεί να καταστεί υπόλογος για τεράστια οικονομικά συμφέροντα.

1. Λειτουργικά

Προβλέπεται εγκατάσταση όλων των αναβαθμίσεων λογισμικού που κυκλοφορούν στην αγορά. Καμμία αναβάθμιση δεν θα γίνεται χωρίς τη σύμφωνη γνώμη της Δ.Υ.Π. (Διεύθυνση Υπηρεσιών του Ο.Β.Ι.).

2. Εργαλεία

Ο ανάδοχος είναι υποχρεωμένος να εγκαθιστά όλες τις αναβαθμισμένες εκδόσεις των εργαλείων που κυκλοφορούν στην αγορά. Καμμία αναβάθμιση δεν θα γίνεται χωρίς τη σύμφωνη γνώμη της Δ.Υ.Π. (Διεύθυνση Υπηρεσιών του Ο.Β.Ι.). Ο ανάδοχος οφείλει να εγκαταστήσει τις εκδόσεις σε δοκιμαστικό περιβάλλον όπως περιγράφεται στη σχετική παράγραφο και μετά την επιτυχή εγκατάσταση και με τη σύμφωνη γνώμη του Ο.Β.Ι. Προβαίνει στην οριστική αντικατάσταση της έκδοσης.

3. Ο ανάδοχος συντηρεί την τρέχουσα έκδοση της εφαρμογής και παρεμβαίνει για την επίλυση των δυσλειτουργιών.

Ειδοποιήσεις : Ο ανάδοχος ειδοποιείται για τις δυσλειτουργίες είτε από την Δ.Υ.Π. Είτε από το helpdesk του ΟΒΙ και μπορεί να απαιτήσει να έρθει ο ίδιος σε επαφή με το χρήστη που έχει αναφέρει το σφάλμα για λόγους καλύτερης συνεννόησης.

Χρόνοι ανταπόκρισης και επίλυσης

Κατά συνέπεια, πέραν της εφεδρικής λύσης της αυτοπρόσωπης κατάθεσης σε ειδική υπηρεσία του Οργανισμού που έχει θεσπιστεί (υπηρεσία Μίας Στάσης) η αποκατάσταση των δυσλειτουργιών πρέπει να γίνεται το ταχύτερο δυνατόν.

Ελάχιστα αποδεκτά πρέπει να είναι :

- Ανταπόκριση 2 εργάσιμες ώρες ή 4 ημερολογιακές (όπου οι εργάσιμες συμφωνούνται μεταξύ του Οργανισμού και του Αναδόχου συντήρησης).
- Τεχνική δυσλειτουργία : 4 εργάσιμες

Για επίλυση επιχειρησιακών ζητημάτων κ.λ.π. (αλλαγή νομοθεσίας, αλλαγή διαδικασιών του Οργανισμού κ.λ.π.) θεσπίζεται ειδική συμφωνία.

Επίσης, αντικείμενο ειδικής συμφωνίας είναι η τυχόν μείζονες μεταβολές.

3.3. Διαχειριστικά

Τακτικές εργασίες για την εύρυθμη λειτουργία

Προκειμένου η εγκατάσταση να βρίσκεται σε εύρυθμη λειτουργία οι

Διαχειριστές συστήματος (System administrators) φέρουν την ευθύνη της φροντίδας

- Monitoring των vm όπου βρίσκεται εγκατεστημένη η εφαρμογή
 - Ενημέρωση των updates των security patches του λειτουργικού των vm
 - Monitoring της λειτουργίας της εφαρμογής του docker
 - Εκπόνηση σχεδίου και εγκατάσταση και monitoring συστήματος λήψης εφεδρικών αντιγράφων ασφαλείας - Backup
-
- Ενημέρωση του ιστοτόπου με τα security patches και
Ενημερωση του λειτουργικού συστήματος των vm με security patches

Log εγκατάστασης

Παραδίδεται σε χωριστό αρχείο το Log της εγκατάστασης το οποίο περιλαμβάνει τις εγκαταστάσεις και ρυθμίσεις υπηρεσιών λειτουργικού συστήματος ανα VM καθώς και τα αρχεία ρυθμισεων yml όπου διαφοροποιούνται από τις σχετικές αναφορές στο github για την προστασία της εφαρμογής.

Κωδικοί

Παραδίδονται σε χωριστό αρχείο οι κωδικοί των συστημάτων.

3.4. Απαιτούμενες γνώσεις και δεξιότητες του Αναδόχου

Εξαιρετική γνώση των **εφαρμογών**

Formio

Bonita

Alfresco

Payum

Wordpress

MySQL
MongoDB
Solr

Καλή γνώση των παρακάτω **τεχνολογιών**

Bpm
Ecm
Docker
Oauth
Τεχνολογία ψηφιακών υπογραφών
NFS

Πολύ καλή γνώση

- Network administration
- Linux System administration
- Devops

Πολύ καλή γνώση των γλωσσών προγραμματισμού

- Nodejs
- Javascript (jquery)
- Php
- Groovy
- Java

Πολύ καλή γνώση

- Html τεχνολογιών και
- Css (bootstrap)
- UIX

Για την επιχειρησιακή υποστήριξη απαιτούνται :

- Υψηλού επιπέδου γνώσης ανάλυσης επιχειρησιακών απαιτήσεων απαραίτητως με γνώσεις Bpmn.
- Υψηλού επιπέδου ικανότητες κατανόησης του επιχειρησιακού γίνεσθαι καθώς και του νομικού υποβάθρου της εφαρμογής.

3.5. Troubleshooting

Καταγράφονται ορισμένα προβλήματα ενδεικτικά σε πρώτη φάση και ο κατάλογος θα πρέπει να συμπληρώνεται περιοδικά από τις καταγραφές του helpdesk.

- **Λάθος στοιχεία χρήστη** : Σβήνεται η εγγραφή και καλείται ο χρήστης να κάνει επανεγγραφή. Η απαλοιφή πρέπει να γίνει από δύο σημεία από το formio και εφόσον υπάρχει ανοιχτό case στο bonita σβήνεται και από εκεί
- **Δεν έχει φτάσει ποτέ το email** : Ο έλεγχος γίνεται στο λογαριασμό του outlook. Αναζητείται το μήνυμα στα εξερχόμενα και αποστέλλεται εκ νέου χειροκίνητα
- **“Χτύπησε” ο mail server του office**: Αυτό διαπιστώνεται επειδή θα έχει χτυπήσει και το αντίστοιχο case στο bonita. Ο τεχνικός θα πρέπει να μπει στο bonita και να επανεκκινήσει το case.
- Ο αιτών ειδοποιεί ότι έχει βάλει **λάθος στοιχεία**. : Αντιμετωπίζεται όπως όλες οι περιπτώσεις μεταβολής δεδομένων κατά τον έλεγχο κανονικότητας. Δηλαδή, ο χρήστης αιτείται και ο αρμόδιος υπάλληλος διορθώνει τα στοιχεία.
- Ο αιτών υποβάλλει την αίτηση και για διαφόρους λόγους το **Bonita δεν λειτουργεί**. : Αυτό διαπιστώνεται είτε γιατί δεν εμφανίζεται η δυνατότητα πληρωμής είτε γιατί η αίτηση παραμένει αδρανής πέραν του διαστήματος που προβλέπεται. Καταρχήν επανεκκινείται το bonita εάν δεν έχει ήδη γίνει. Εφόσον η λειτουργία αποκατασταθεί ο τεχνικός “κάνει κλήσεις” προς τις αντίστοιχες ροές του bonita ανάλογα την περίπτωση συμβουλευόμενος το διάγραμμα.
- Πληρωμές - Ο αιτών υποβάλλει παράπονο ότι ενώ έχει πληρώσει **η αίτησή του φαίνεται απλήρωτη**. : Η πρώτη διάγνωση γίνεται είτε γιατί το “κουμπί” πληρωμής επανεμφανίζεται δίπλα από την αίτηση. είτε γιατί το σύστημα στέλνει email υπενθύμισης πληρωμής.
Πιθανά αίτια:
 1. Το bonita τη συγκεκριμένη στιγμή ήταν εκτός λειτουργίας
 2. Το σύστημα της Τράπεζας ήταν σε δυσλειτουργία
 Γίνεται σύγκριση με το extrait της Τράπεζας και αναλόγως λύνεται κατά περίπτωση. Εάν ο αιτών έχει πληρώσει κανονικά τότε πρέπει ο τεχνικός να τρέξει χειροκίνητα το task της πληρωμής στο bonita κάνοντας ένα post με τα στοιχεία της πληρωμής πάνω στο lane με τα στοιχεία της.

Παρατίθεται αρχείο - καταγραφής και επίλυσης προβλημάτων (Σημειώνεται ότι **έχει εκπονηθεί εξ ολοκλήρου από τους συναδέλφους του helpdesk**)

- https://docs.google.com/document/d/19Gf_0VXZvi8r2PduMAmUr1QBlm-Bi4ZGYlp4R3mZDFM/edit?usp=sharing
-
-

3.6. Διάγραμμα bonita

Παραδίδεται το διάγραμμα διαδικασιών το οποίο στο παρόν πρέπει να αντικαθίσταται μετά από κάθε αναθεώρηση.

3.7. Καταγραφή δυσλειτουργιών

Για κάθε αναφορά προς το helpdesk είτε προς τους τεχνικούς συμπληρώνεται αρχείο, το οποίο υπάρχει σε ηλεκτρονική μορφή. Στο ίδιο αρχείο (παρατίθεται ανωτέρω δείγμα) καταγράφεται και η επιλυση.

Σκοπός των διαδικασιών αυτών είναι μεταξύ άλλων:

- Η διευκόλυνση της λειτουργίας του helpdesk
- Η δημιουργία αρχείου faq
- Η εν γένει αξιολόγηση της αξιοπιστίας της εφαρμογής
- Η εν γένει αξιολόγηση της επάρκειας του αναδόχου
- Η ανάδειξη αναγκαιοτήτων αναθεώρησης της εφαρμογής.

