Πανεπιστήμιο Δυτικής Αττικής

Σχολή Μηχανικών

Τμήμα Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών
Πρόγραμμα Μεταπτυχιακών Σπουδών «Διαδίκτυο Των Πραγμάτων Και Ευφυή Περιβάλλοντα»

Μεταπτυχιακή Διπλωματική Εργασία

# Ανάπτυξη ευέλικτης και ασφαλούς πύλης δικτύου ανοικτού κώδικα για την υλοποίηση έξυπνου μετρητή ενέργειας

**Μεταπτυχιακός Φοιτητής: Ευθύμιος Τσερέπας**
ΑΜ: 19009

Επιβλέπων:

**Παναγιώτης Παπαγέωργας**
Καθηγητής

Αιγάλεω - Αθήνα, Σεπτέμβριος, 2022


University of Wests Attica

Faculty Of Engineering

Department of Electrical and Electronic Engineering
Master of Science in «Internet of Things and Intelligent Environments»

MSc Thesis

# Development of flexible and secure open source network gateway for smart energy meter implementation

**Student: Efthymios Tserepas**
Registration Number: 19009

Supervisor:

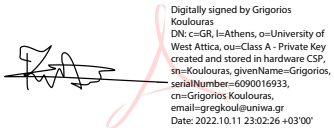
**Panagiotis Papageorgas**
Professor

Egaleo - Athens, September, 2022

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:


Παναγιώτης Παπαγέωργας
Καθηγητής


Διονύσιος Κανδρής
Καθηγητής




Γρηγόριος Κουλούρας
Αναπληρωτής Καθηγητής

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.
Πανεπιστήμιο Δυτικής Αττικής, Ευθύμιος Τσερέπας 2022

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.
Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Δυτικής Αττικής.

**ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος **Ευθύμιος Τσερέπας του Δημητρίου**, με αριθμό μητρώου **msciot19009** φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ, στο ΠΜΣ «ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ ΚΑΙ ΕΥΦΥΗ ΠΕΡΙΒΑΛΛΟΝΤΑ» δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.»

Ο Δηλών

.....

Ευθύμιος Τσερέπας
Τμήμα Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών
Πανεπιστήμιο Δυτικής Αττικής

# Ευχαριστίες

Εκφράζω τις ευχαριστίες μου στους καθηγητές του Μεταπτυχιακού Προγράμματος «Διαδίκτυο Των Πραγμάτων Και Ευφυή Περιβάλλοντα» του Πανεπιστημίου Δυτικής Αττικής για τις πολύτιμες γνώσεις που μου προσκόμισαν. Θέλω να ευχαριστήσω προσωπικά τον επιβλέποντα καθηγητή της διπλωματικής μου εργασίας τον κ. Παπαγέωργα Παναγιώτη για την καθοδήγηση καθ' όλη τη διάρκεια της έρευνας μου, την μετάδοση γνώσεων και εμπειριών από ενδιαφέρουσες τεχνολογίες που ήταν καθοριστικές για την ολοκλήρωση της εργασίας μου καθώς και στο έναυσμα για την συνέχεια των αναζητήσεων στο χώρο της μελέτης και της έρευνας. Θέλω να ευχαριστήσω τον τεχνικό διευθυντής της εταιρίας MAOSCO Ltd, Chris Torr για την πλήρη υποστήριξη στο τομέα προγραμματισμού του Multos Trust Core. Επίσης ένα μεγάλο ευχαριστώ στον πατέρα μου Δημήτριο όπου έβαλε το λιθαράκι να ασχοληθώ με τον συγκεκριμένο τομέα, δίνοντας την ευκαιρία από μικρή ηλικία να γνωρίσω τους υπολογιστές με τον Commodore 64 και τις ασύρματες επικοινωνίες με τον ραδιοφωνικό δέκτη Sony ICF-6700W αποκτώντας εξοικείωση με βασικές ορολογίες.

# Περίληψη

Στον τομέα της ενέργειας με τα πρόσφατα γεγονότα του πολέμου στην Ουκρανία, δημιουργείται αβεβαιότητα και δυστυχώς το σενάριο για μια επερχόμενη ενεργειακή κρίση βρίσκεται όλο και πιο κοντά. Η τεχνολογία του IoT στον τομέα της ενέργειας αναπτύσσετε στα έξυπνα δίκτυα (smart grid's) και μια άμεση συνέπεια των έξυπνων δικτύων είναι η ύπαρξη ενός ηλεκτρικού μοντέλου που θα είναι ικανό να διαχειρίζεται διαφορετικές συσκευές παραγωγής και αποθήκευσης με αποτελεσματικό και αποκεντρωμένο τρόπο, αναπτύσσοντας μια προηγμένη υποδομή μέτρησης (AMI - Advanced Metering Infrastructure). Μέρος της AMI είναι η πύλη έξυπνου μετρητή ενέργειας (SMGW) όπου μπορεί να βοηθήσει στην εξοικονόμηση της ενέργειας. Στην παρούσα διπλωματική εργασία προτείνεται και αξιολογείται η ανάπτυξη ευέλικτης και ασφαλής πύλης δικτύου ανοικτού κώδικα για την υλοποίηση έξυπνου μετρητή ενέργειας. Θα χρησιμοποιηθεί κατάλληλη αναπτυξιακή πλατφόρμα που θα ενσωματώνει τεχνολογία LPWAN για την υποστήριξη διαδικασιών τηλεμετρίας και τεχνολογία Blockckain για διαδικασίες διασφάλισης της ακεραιότητας των δεδομένων. Επειδή το Blockchain από μόνο του δεν είναι σε θέση να διασφαλίσει πλήρως μια συναλλαγή και στις περισσότερες περιπτώσεις τα δεδομένα πρέπει να είναι ασφαλή από το σημείο παραγωγής θα χρησιμοποιηθεί Secure Element για να καλύψει την συγκεκριμένη ευπάθεια. Θα χρησιμοποιηθεί το Azure IoT Hub και το Secure Element για την ασφαλή διασύνδεση με την SMGW για μελλοντικές επεκτάσεις στο πεδίο των εφαρμογών. Επίσης η SMGW θα έχει εγκατεστημένο το metamask ώστε να γίνουν μελλοντικές επεκτάσεις στο πεδίο των smart contracts.

**Λέξεις Κλειδιά:** Έξυπνα δίκτυα,προηγμένη υποδομή μέτρησης, πύλη έξυπνου μετρητή ενέργειας, τεχνολογίες LPWAN,δίκτυο LoRa,τεχνολογίες Blockchain,ασφαλές στοιχείο,κόμβος Azure IoT

# Abstract

In the energy sector with the recent events of the war in Ukraine, uncertainty is created and unfortunately the scenario for an upcoming energy crisis is getting closer. The IoT technology in the energy sector is developing in smart grids and a direct consequence of smart grids is the existence of an electrical model that will be able to manage different production and storage devices in an efficient and decentralized way, developing an advanced infrastructure (AMI - Advanced Metering Infrastructure). Part of AMI is the Smart Energy Meter Gateway (SMGW) where it can help save energy. In this thesis, the development of a flexible and secure open source network gateway for the implementation of a smart energy meter is proposed and evaluated. An appropriate development platform will be used that incorporates LPWAN technology to support telemetry processes and Blockckain technology to ensure data integrity processes. Because Blockchain alone is not able to fully secure a transaction and in most cases data must be secure from the point of production Secure Element will be used to cover this particular vulnerability. Azure IoT Hub and Secure Element will be used to securely interface with SMGW for future expansions of the application domain. Also, SMGW will have metamask installed so that future expansions in the field of smart contracts can be made.

**Keywords:** Smart Grid's,AMI,SMGW,LPWAN technologies,LoRA network,Blockchain technologies,Secure Element,Azure IoT Hub

# Πίνακας περιεχομένων

# Πίνακας σχημάτων

# Πίνακας πινάκων

# Πίνακας κωδίκων

ΚΕΦΑΛΑΙΟ 1

# Εισαγωγή

Στο Εισαγωγικό Κεφάλαιο θα συζητηθούν ο σκοπός της εργασίας, η αναγκαιότητα και σπουδαιότητα της έρευνας, τα βασικά ερευνητικά ερωτήματα καθώς και η διάρθρωση της εργασίας σε κεφάλαια

## 1.1 Σκοπός έρευνας

Ο σκοπός της έρευνας είναι η μελέτη και η διερεύνηση, σύμφωνα με λογισμικό ανοιχτού κώδικα, με σκοπό την ανάπτυξη ευέλικτης και ασφαλούς πύλης για την υλοποίηση έξυπνου μετρητή ενέργειας (SMGW). Η SMGW Θα χρησιμοποιηθεί κατάλληλη αναπτυξιακή πλατφόρμα που θα ενσωματώνει τεχνολογία LPWAN για την υποστήριξη διαδικασιών τηλεμετρίας και για την κρυπτογράφηση από άκρο σε άκρο θα χρησιμοποιηθεί τεχνολογία Blockchain για την διασφάλιση και την ακεραιότητα των δεδομένων. Το Blockchain από μόνο του δεν είναι σε θέση να διασφαλίσει πλήρως μια συναλλαγή γιατί εγγυάται μόνο το αμετάβλητο των δεδομένων, ενώ στις περισσότερες περιπτώσεις τα δεδομένα πρέπει να είναι ασφαλή από το σημείο παραγωγής. Επομένως, για να καλυφθεί αυτό το κενό ασφαλείας, συνιστάται η χρήση των Secure Elements για τη δημιουργία μιας «ρίζας εμπιστοσύνης». Η προτεινόμενη πλατφόρμα θα επιτρέπει την ενσωμάτωση των Ανανεώσιμων πηγών ενέργειας για την καταγραφή την παραγόμενης ενέργειας.

## 1.2 Αναγκαιότητα και σπουδαιότητα έρευνας

Η εξάπλωση του διαδικτύου των πραγμάτων (IoT) αυξάνεται με ταχείς ρυθμούς κάθε χρόνο. Σύμφωνα με το άρθρο [1] οι συνδεδεμένες συσκευές IoT σήμερα είναι πάνω από 42 δισεκατομμύρια και εκτιμάται ότι μέχρι το 2025 θα αυξηθεί σε πάνω από 75 δισεκατομμύρια. Η ραγδαία ανάπτυξη των διασυνδεδεμένων συσκευών διεγείρει προβλήματα σε θέματα ασφάλειας και ακεραιότητας των δεδομένων.

Στον τομέα της ενέργειας με τα πρόσφατα γεγονότα του πολέμου στην Ουκρανία, δημιουργείται αβεβαιότητα και δυστυχώς το σενάριο για μια επερχόμενη ενεργειακή κρίση βρίσκεται όλο και πιο κοντά.

Είναι σημαντική η ανάπτυξη μιας SMGW που θα διασφαλίζει την ασφάλεια και την ακεραιότητα των δεδομένων ώστε να βοηθήσει τους καταναλωτές να επιτύχουν εξοικονόμηση ενέργειας και τιμής, τις υπηρεσίες κοινής ωφέλειας να έχουν πιο άμεσα εικόνα για την παραγόμενη ενέργεια από ανανεώσιμες πηγές ενέργειας καθώς και δραστικά μέτρα για την εξοικονόμηση της ενέργειας.

Ο συνδυασμός Secure Element με το σωστό Blockckain μπορεί να προσφέρει μια βιώσιμη λύση για την υλοποίηση SMGW.

## 1.3 Βασικά ερευνητικά ερωτήματα

- SMGW:
  - Μπορεί να υλοποιηθεί SMGW με χαμηλό κόστος ;
  - Η χρήση του Raspberry pi zero είναι μια οικονομική λύση και επαρκές από μεριάς υπολογιστικής ισχύος για την υλοποίηση της SMGW ;
  - Ποιο LPWAN δίκτυο είναι κατάλληλο για την συνδεσιμότητα της SMGW για την υποστήριξη διαδικασιών τηλεμετρίας;
- Δίκτυα LPWAN :
  - Το δίκτυο LoRa πληρεί της προϋποθέσεις για την σύνδεση της SMGW;
  - Η επέκταση LoRa Node pHAT λειτουργεί ικανοποιητικά με το Raspberry pi zero;
  - Είναι ασφαλής η μετάδοση των δεδομένων από το LoRa Node pHAT στις LoRa GW ;
  - Το TTN ή το Chirpstack είναι πιο ιδανικός LoRaWAN Server για την αποστολή των δεδομένων από την SMGW;
  - Υπάρχει πιο ασφαλής δίκτυο βασισμένο στο LoRaWAN ;
- Blockchain :
  - Το Ethereum ή το Helium blockchain είναι ιδανικό για την αποστολή δεδομένων;
  - Το Ethereum είναι ιδανικό για την δημιουργία smart contracts;
  - Το Helium Blockchain αποτελεί μια καλή λύση για την συνδεσιμότητα της SMGW και την προστασία των δεδομένων;
  - Το Blockchain από μόνο του είναι ιδανικό για την διασφάλιση και την ακεραιότητα των δεδομένων;
- Secure Elements :
  - Η χρήση των Secure Elements πως μπορούν να συμβάλουν στην προστασία και την ακεραιότητα των δεδομένων;
  - Η χρήση του Secure Element μπορεί να συμβάλει σε πιθανές ευπάθειες του LoRa Node pHAT;
  - Το Multos Trust Core λειτουργεί στο Raspberry pi zero ;
  - Ο συνδυασμός Secure Element με το σωστό Blockckain μπορεί να προσφέρει μια βιώσιμη λύση στο σύστημα;
  - Μπορεί να γίνει χρήση του Secure Element για επιπλέον προστασία που θα αφορά το Azure IoT Hub ;

## 1.4 Η διάρθρωση της εργασίας σε Κεφάλαια

Στο 2ο κεφάλαιο θα γίνει θεωρητική μελέτη για την τεχνολογία του διαδικτύου των πραγμάτων (IoT), ποιο συγκεκριμένα θα αναλυθούν οι πιο διαδεδομένες αρχιτεκτονικές και τα πιο διαδεδομένα πρωτόκολλα του IoT. Επίσης θα αναλυθεί η χρήση του IoT στον τομέα της ηλεκτρικής ενέργειας στα λεγόμενα έξυπνα δίκτυα (Smar grid's) καθώς και στα αντικείμενα των smart grid's όπως η προηγμένη υποδομή μέτρησης (AMI - Advanced Metering Infrastructure) που περιλαμβάνει την πύλη έξυπνου μετρητή ενέργειας (SGMW). Στο τέλος του κεφαλαίου θα γίνει μελέτη των πιο διαδεδομένων τεχνολογιών LPWAN δικτύων, σύγκριση και επιλογή για την συνδεσιμότητα της SMGW. Με το συγκεκριμένο κεφάλαιο θα έχουμε αποκομίσει τα βασικά χαρακτηριστικά της τεχνολογίας IoT, την χρήση στα smart grid's την χρησιμότητα της SMGW καθώς και την επιλογή LPWAN για την συνδεσιμότητα της SMGW για την υποστήριξη της τηλεμετρίας. Η επιλογή LPWAN είναι το LoRa για λόγους που θα αναφερθούν.

Στο 3ο κεφάλαιο θα αναλυθεί η τεχνολογία LoRa σε θεωρητικό υπόβαθρο με στόχο την αποκόμιση των βασικών γνώσεων που θα βοηθήσουν στην υλοποίηση ενός LoRa συστήματος. Θα γίνει ανάλυση συσκευών LoRa, τόσο σε κόμβους όσο και σε πύλες και επιλογή που θα χρησιμοποιηθεί στην υλοποίηση της SMGW. Στην συνέχεια θα γίνει σύγκριση των δυο διαδεδομένων LoRaWAN Network Servers του TTN και το Chirpstack και επιλογή για δοκιμές των LoRa συσκευών. Γίνεται επιλογή του Chirpstack για λόγους που θα αναφερθούν στην συνέχεια και πιο συγκεκριμένα στο server του εργαστηρίου restqmlab του Πανεπιστημίου Δυτικής Αττικής. Γίνεται η δοκιμή του LoRa κόμβου που αποτελείται από το raspberry pi zero με την επέκταση LoRa IoT LoRa Node pHAT και λειτουργεί με πολύ ικανοποιητικά αποτελέσματα.

Στο 4ο κεφάλαιο θα γίνει μελέτη σε θέματα που αφορούν την ασφάλεια. Αρχικά θα αναφερθεί η τεχνολογία του Blockchain όπου αποτελεί μια καλή λύση για την ασφάλεια, την ακεραιότητα των δεδομένων αλλά σε μερικές περιπτώσεις η χρήση του μπορεί να είναι ασύμφορη. Ποιο συγκεκριμένα θα εξεταστούν οι δυνατότητες του Ethereum Blockchain όπου είναι ιδανικό για Smart Contracts όχι όμως ιδανικό για την αποστολή δεδομένων και το Helium Blockchain όπου είναι ιδανικό για την αποστολή δεδομένων από κόμβους LoRa. Το blockchain από μόνο του δεν είναι σε θέση να εξασφαλίσει πλήρως μια συναλλαγή γιατί εγγυάται μόνο την αναλλοίωτη κατάσταση των δεδομένων, ενώ στις περισσότερες περιπτώσεις τα δεδομένα πρέπει να είναι ασφαλή στο σημείο παραγωγής. Για να καλυφθεί αυτό το κενό ασφαλείας, προτείνετε η χρήση των Secure Elements (SE) και συγκεκριμένα του Multos Trust Core για τη δημιουργία μιας «ρίζας εμπιστοσύνης» και για την παροχή αξιόπιστων υπολογιστικών πόρων στον LoRa κόμβο για την κρυπτογράφηση του ωφέλιμου φορτίου με κρυπτογράφηση AES-CBC πριν μεταδοθεί στο Helium ακολουθώντας τον ασφαλή σχεδιασμό του μοντέλου. Στο τέλος του κεφαλαίου θα αναλυθούν οι δυνατότητες του Azure IoT Hub που θα βοηθήσει την επεκτασιμότητα της SMGW για μελλοντικές εφαρμογές και θα δοθεί λύση για την επιπλέον ασφάλεια στο Azure IoT Hub με την χρήση του Multos Trust Core.

Από τα προηγούμενα κεφάλαια έχουμε αποκομίσει το θεωρητικό υπόβαθρο για να προχωρήσουμε στο 5ο κεφάλαιο όπου θα γίνει υλοποίηση της SMGW, ενός IoT συστήματος που θα έχει την ονομασία STRESQLab Smart Metering System.

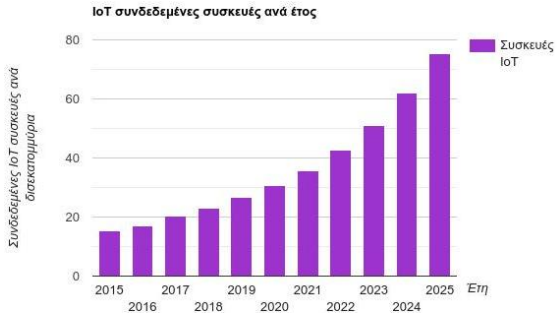
# ΚΕΦΑΛΑΙΟ 2

## Διαδίκτυο των πραγμάτων (IoT), έξυπνα δίκτυα (Smart grid's) και τεχνολογίες LPWAN

Στο συγκεκριμένο κεφάλαιο θα γίνει εν συντομία μια αναφορά για την τεχνολογία του διαδικτύου των πραγμάτων (IoT) και τους τομείς εφαρμογής. Θα συζητηθούν οι πιο διαδεδομένες αρχιτεκτονικές καθώς και τα ποιο διαδεδομένα πρωτόκολλα του IoT. Στην συνέχεια θα αναφερθεί η χρήση του IoT στον τομέα της ηλεκτρικής ενέργειας, στα λεγόμενα έξυπνα δίκτυα (Smart grid's). Από τα αντικείμενα των smart grid's θα αναφερθούν η προηγμένη υποδομή μέτρησης (AMI-Advanced Metering Infrastructure) που περιλαμβάνει την πύλη έξυπνου μετρητή ενέργειας (SGMW). Τέλος του κεφαλαίου θα γίνει μελέτη για τις πιο διαδεδομένες τεχνολογίες LPWAN δικτύων καθώς και σύγκριση και επιλογή για την συνδεσιμότητα της SMGW και την υποστήριξη της τηλεμετρίας.

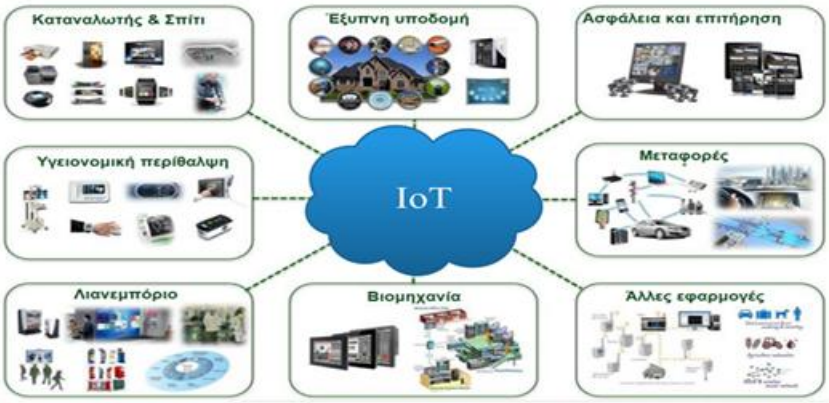
## 2.1 Διαδίκτυο των πραγμάτων (IoT)

Ο όρος διαδίκτυο των πραγμάτων (Internet of Things – IoT) προτάθηκε για πρώτη φορά από τον Βρετανό τεχνολόγο Kevin Ashton το 1999 και αναφέρεται στις συσκευές, ενεργοποιητές, αισθητήρες όπου θα έχουν πρόσβαση στο διαδίκτυο και θα έχουν την δυνατότητα να επικοινωνούν και να ανταλλάσσουν πληροφορίες μεταξύ τους [2]. Οι συνδεδεμένες συσκευές IoT σήμερα είναι πάνω από 42 δισεκατομμύρια και εκτιμάται ότι μέχρι το 2025 θα αυξηθεί σε πάνω από 75 δισεκατομμύρια όπως φαίνεται στην εικόνα 1. Η ραγδαία ανάπτυξη των διασυνδεδεμένων συσκευών αποτελεί πρόκληση σε θέματα ασφάλειας και λύσεις όπως η τεχνολογία του Blockchain και των secrure element παρέχουν επιπλέον ασφάλεια στην ακεραιότητα των δεδομένων.




Εικόνα 1. Αριθμός των συνδεδεμένων IoT συσκευών [1]

Το IoT αναμένεται να έχει πολλά οφέλη σε διάφορους τομείς όπως για παράδειγμα στην βελτίωση της ενεργειακής αποδοτικότητας μέσω των έξυπνων δικτύων (smart grid), την βελτίωση της καθημερινότητας των δημοτών μέσω των έξυπνων πόλεων (smart cities), παρακολούθηση και ασφάλεια στα σπίτια μέσω των έξυπνων σπιτιών (smart homes), στην υγειονομική περίθαλψη, τις μεταφορές από εταιρείες logistics και σε πολλούς άλλους τομείς. Οι εξοπλισμοί του IoT είναι πιο ψηφιοποιημένη και πιο συνδεδεμένοι, δημιουργώντας δίκτυα μεταξύ μηχανών, ανθρώπων και Διαδικτύου, οδηγώντας στη δημιουργία νέων οικοσυστημάτων που επιτρέπουν υψηλότερη παραγωγικότητα, καλύτερη ενεργειακή απόδοση και υψηλότερη κερδοφορία. Οι αισθητήρες συμβάλλουν στην αναγνώριση της κατάστασης των πραγμάτων, με την οποία αποκτούν το πλεονέκτημα της πρόβλεψης ανθρωπίνων αναγκών βάσει των πληροφοριών που συλλέγονται ανά πλαίσιο. Αυτές οι έξυπνες συσκευές όχι μόνο συγκεντρώνουν πληροφορίες από το περιβάλλον τους αλλά είναι επίσης σε θέση να λαμβάνουν αποφάσεις χωρίς ανθρώπινη παρέμβαση [2], [3]. Στην εικόνα 2 απεικονίζονται μερικές εφαρμογές του IoT σε διάφορους τομείς.




Εικόνα 2. Εφαρμογές του IoT

### 2.1.1 Στοιχεία του IoT

Σε αυτήν την ενότητα θα αναφερθούν τα βασικά στοιχεία του IoT που φαίνονται στην εικόνα 3


Εικόνα 3. Στοιχεία του IoT

### Ταυτοποίηση

Η ταυτοποίηση παρέχει μοναδική ταυτότητα για κάθε συνδεδεμένο αντικείμενο στο διαδίκτυο. Για την ταυτοποίηση παρέχονται δυο διαδικασίες, η ονομασία και η διεύθυνση. Η ονομασία αναφέρεται ως όνομα του αντικειμένου ενώ η διεύθυνση είναι η μοναδική διεύθυνση του συγκεκριμένου αντικειμένου [4]. Υπάρχουν πολλές διαθέσιμες μέθοδοι αναγνώρισης για το IoT όπως, οι ηλεκτρονικοί κωδικοί προϊόντων (EPC - electronic product codes) και οι ubiquitous κωδικοί (uCode). [2]

### Ανίχνευση

Στο IoT ανίχνευση ονομάζεται η διαδικασία της συλλογής δεδομένων από τα αντικείμενα εντός του δικτύου και της αποστολής τους σε μια βάση δεδομένων ή ένα cloud.[2]. Τα αντικείμενα μπορεί να είναι έξυπνοι αισθητήρες, ενεργοποιητές, RFID ή φορητές συσκευές ανίχνευσης.Οι υπολογιστές μονής πλακέτας (SBC - Single Board Computers) που είναι ενσωματωμένοι με αισθητήρες,ενσωματωμένο TCP/IP και λειτουργίες ασφαλείας χρησιμοποιούνται συνήθως για την υλοποίηση προϊόντων IoT (π.χ. Arduino, Raspberry Pi, pycom lopy, κ.λπ.).[4]

### Επικοινωνία

Οι τεχνολογίες επικοινωνίας IoT επιτυγχάνουν να συνδέσουν ετερογενή αντικείμενα μεταξύ τους ώστε να επικοινωνούν. Στην επικοινωνία, οι συσκευές ενδέχεται να στέλνουν και να λαμβάνουν μηνύματα, αρχεία και άλλες πληροφορίες. Παραδείγματα πρωτοκόλλων επικοινωνίας που χρησιμοποιούνται για το IoT είναι το WiFi, το Bluetooth, το IEEE 802.15.4, το Z-wave και το LTE-Advanced.[2], [4]

### Υπολογισμός

Ο υπολογισμός είναι μια διαδικασία η οποία εκτελείται στις πληροφορίες που συλλέγονται από τα αντικείμενα χρησιμοποιώντας αισθητήρες με σκοπό την αφαίρεση των περιττών πληροφοριών. Έχουν αναπτυχθεί διάφορες πλατφόρμες υλικού (hardware) και λογισμικού (software). Μερικές διαδεδομένες πλατφόρμες hardware είναι το Arduino, Raspbery Pi, pycom lopy και software τα λειτουργικά συστήματα το Tiny OS, Lite OS, Android κ.λπ

### Υπηρεσίες

Οι υπηρεσίες IoT μπορούν να κατηγοριοποιηθούν σε τέσσερις κατηγορίες.

1. Υπηρεσίες που σχετίζονται με την ταυτότητα : Χρησιμοποιείται για να πάρει την ταυτότητα των αντικειμένων που έχουν στείλει το αίτημα.

2. Υπηρεσίες συγκέντρωσης πληροφοριών : Χρησιμοποιείται για την συλλογή όλων των πληροφοριών από τα αντικείμενα

3. Υπηρεσίες Συνεργατικής επίγνωσης (Collaborative-Aware Services) : Λαμβάνει αποφάσεις σύμφωνα με τις πληροφορίες που συλλέγονται και στέλνει τις κατάλληλες απαντήσεις στις συσκευές.

4. Υπηρεσίες Ubiquitous : Άμεση απόκριση των συσκευών χωρίς προβλήματα οποιαδήποτε στιγμή και οπουδήποτε. [2], [4]

**Σημασιολογία**

Η σημασιολογίας στο IoT είναι η ικανότητα εξαγωγής γνώσης έξυπνα, από διαφορετικά μηχανήματα για την παροχή των απαιτούμενων υπηρεσιών. Η εξαγωγή γνώσης περιλαμβάνει την ανακάλυψη και τη χρήση πόρων, την μοντελοποίηση πληροφοριών, την αναγνώριση και την ανάλυση δεδομένων για να κατανοήσει κανείς τη σωστή απόφαση για την παροχή της ακριβούς υπηρεσίας. [2], [4]

### 2.1.2 Αρχιτεκτονικές IoT

Δεν υπάρχει γενική συμφωνία για μια αρχιτεκτονική IoT. Υπάρχουν διάφορες αρχιτεκτονικές που έχουν προταθεί από ερευνητές. Βασικές αρχιτεκτονικές που θα αναφερθούν είναι η αρχιτεκτονική τριών επιπέδων, τεσσάρων επιπέδων και πέντε επιπέδων.

**Αρχιτεκτονική τριών επιπέδων**

Η αρχιτεκτονική τριών επιπέδων αποτελείτε από επίπεδο αντικειμένων, το επίπεδο δικτύου και το επίπεδο εφαρμογής όπως φαίνεται στην εικόνα 4


Εικόνα 4. Αρχιτεκτονική 3 επιπέδων

Το επίπεδο των αντικειμένων συλλέγει και επεξεργάζεται τα δεδομένα που προέρχονται από τα αντικείμενα. Αυτό το επίπεδο περιλαμβάνει αισθητήρες και ενεργοποιητές (actuators) για την εκτέλεση διαφορετικών λειτουργιών όπως μέτρηση θερμοκρασίας, μέτρηση ηλεκτρικής ενέργειας, ενεργοποίηση συσκευών κλπ.

Το επίπεδο δικτύου μεταφέρει τα δεδομένα που παράγονται από το επίπεδο αντικειμένων στο επίπεδο εφαρμογής. Τα δεδομένα μπορούν να μεταφερθούν μέσω διαφόρων τεχνολογιών όπως για παράδειγμα wifi,ethernet,bluetooth,δικτύων lpwan κλπ.

Το επίπεδο εφαρμογής περιλαμβάνει τις εφαρμογές στις οποίες έχει αναπτυχθεί το IoT. Οι εφαρμογές του IoT μπορεί να είναι έξυπνα σπίτια, έξυπνες πόλεις, έξυπνη υγεία, έξυπνα δίκτυα ηλεκτρικής ενέργειας. Έχει την ευθύνη να παρέχει τις υπηρεσίες στις εφαρμογές. Οι υπηρεσίες μπορεί να διαφέρουν για κάθε εφαρμογή, επειδή οι υπηρεσίες εξαρτώνται από τις πληροφορίες που συλλέγονται από αισθητήρες. [2], [4]

### Αρχιτεκτονική τεσσάρων επιπέδων

Στην αρχιτεκτονική τριών επιπέδων η απευθείας αποστολή δεδομένων από το επίπεδο αντικειμένων στο επίπεδο δικτύου αυξάνει τις πιθανότητες για επιθέσεις από κακόβουλους χρήστες. Στην αρχιτεκτονική τεσσάρων επιπέδων έχει προστεθεί ένα επίπεδο μεταξύ των επιπέδων αντικειμένων και δικτύου το επίπεδο υποστήριξης όπως φαίνεται στην εικόνα 5


Εικόνα 5. Αρχιτεκτονική 4 επιπέδων

Στο συγκεκριμένο επίπεδο αποστέλλονται τα δεδομένα από το επίπεδο των αντικειμένων και έχει δυο ευθύνες. Επιβεβαιώνει ότι οι πληροφορίες αποστέλλονται από τους αυθεντικούς χρήστες (ταυτοποίηση) και προστατεύονται από απειλές. Η πιο συχνά χρησιμοποιούμενη μέθοδος ταυτοποίησης είναι ο έλεγχος ταυτότητας. Υλοποιείται με τη χρήση προ-κοινοποιημένων μυστικών κλειδιών και κωδικών πρόσβασης. [2]

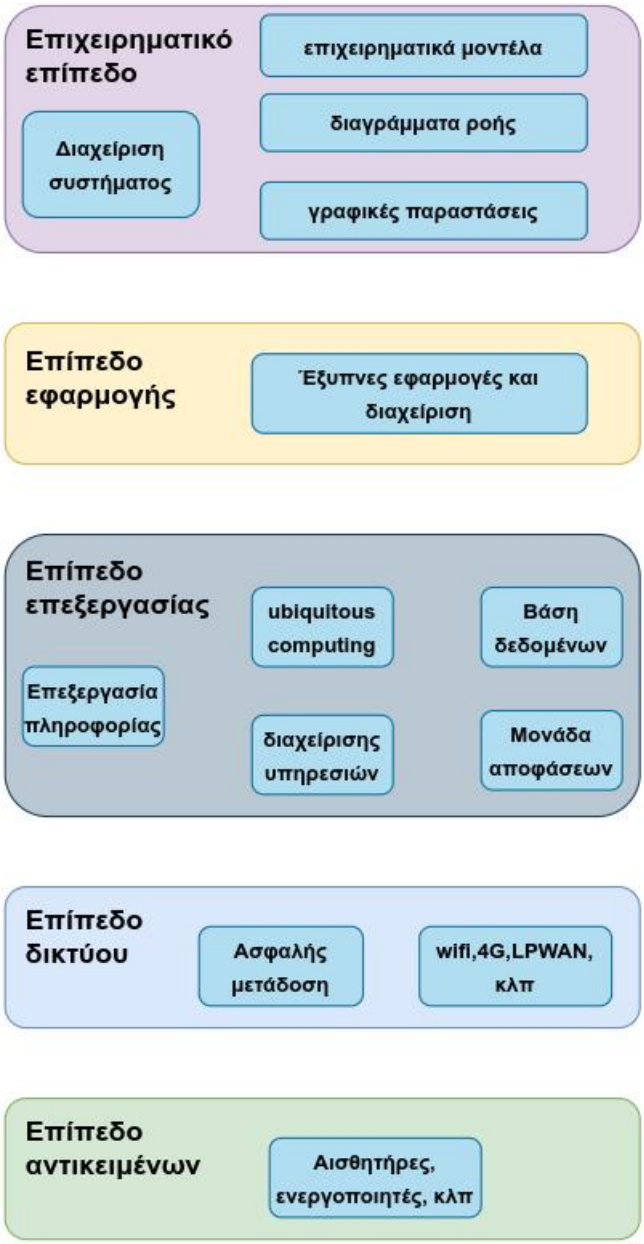
### Αρχιτεκτονική πέντε επιπέδων

Η αρχιτεκτονική πέντε επιπέδων περιλαμβάνει τρία επίπεδα από της προηγούμενες αρχιτεκτονικές που είναι τα επίπεδα, αντικειμένων, επίπεδο δικτύου και εφαρμογής καθώς και άλλα δυο επίπεδα που είναι το επίπεδο επεξεργασίας και το επιχειρηματικό επίπεδο όπως φαίνεται στην εικόνα 6.

Το επίπεδο επεξεργασίας είναι επίσης γνωστό ως επίπεδο ενδιάμεσου λογισμικού (middleware layer). Συλλέγει τις πληροφορίες που αποστέλλονται από το επίπεδο δικτύου, πραγματοποιεί επεξεργασία στις πληροφορίες που συλλέγονται. Έχει την ευθύνη να εξαλείψει επιπλέον πληροφορίες που δεν έχουν νόημα και εξάγει τις χρήσιμες πληροφορίες. Επίσης εκτελεί υπολογισμούς ubiquitous και λαμβάνει αυτόματα αποφάσεις με βάση τα αποτελέσματα.

Το επίπεδο επιχείρησης διαχειρίζεται τις συνολικές δραστηριότητες και υπηρεσίες του συστήματος IoT. Οι αρμοδιότητες αυτού του επιπέδου είναι να δημιουργήσει ένα επιχειρηματικό μοντέλο, γραφήματα, διαγράμματα ροής κ.λπ. με βάση τα δεδομένα που λαμβάνονται από το επίπεδο εφαρμογής. Επίσης σχεδιάζει, αναλύει, εφαρμόζει, αξιολογεί, παρακολουθεί και αναπτύσσει στοιχεία που σχετίζονται με το σύστημα IoT. Επιπλέον, η παρακολούθηση και η διαχείριση των υπόλοιπων τεσσάρων επιπέδων επιτυγχάνεται σε αυτό το επίπεδο [2], [4]–[6].

Η SMWG θα ακολουθεί την αρχιτεκτονική των πέντε επιπέδων.



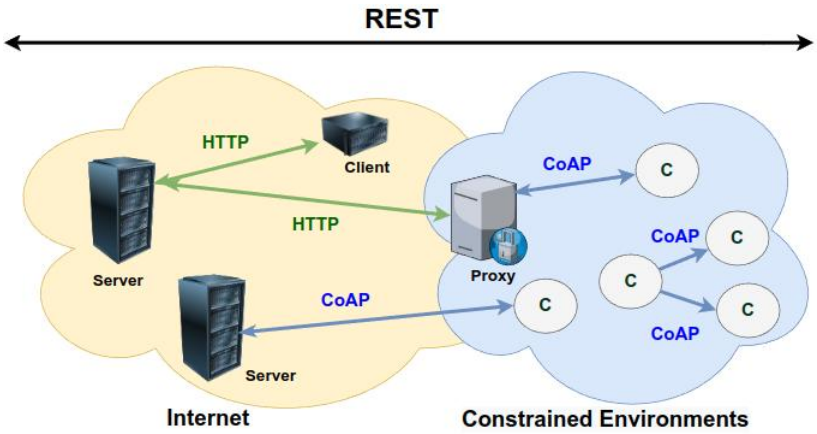

Εικόνα 6. Αρχιτεκτονική 5 επιπέδων

### 2.1.3 Πρωτόκολλα IoT

Για εξοικονόμηση ενέργειας και περιορισμό ισχύος χρειάστηκε να σχεδιαστούν καινούργια πρωτόκολλα (πέρα του TCP/IP) για την διασύνδεση των πραγμάτων με το διαδίκτυο. Με την χρήση των πρωτοκόλλων COAP, MQTT, XMPP οι αισθητήρες των συσκευών όπου έχουν περιορισμένος πόρους (μνήμη, επεξεργαστική ισχύ) μπορούν να συνδεθούν στο διαδίκτυο και να επιτευχθεί χαμηλή κατανάλωση ισχύος.

#### Πρωτόκολλο CoAP

Το πρωτόκολλο CoAP (Constrained Application Protocol) βασίζεται στο πρωτόκολλο του HTTP. Έχει πολλές ομοιότητες με το HTTP αλλά έχει σχεδιαστεί ώστε να επικοινωνεί μηχανή προς μηχανή (M2M), μια συσκευή λειτουργεί σαν client ή σαν server (εικόνα 7).



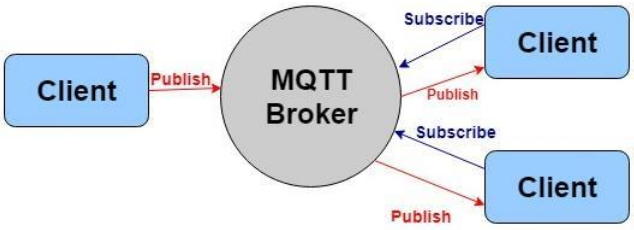

Εικόνα 7. Πρωτόκολλο CoAP

Όπως και στο HTTP μια αίτηση (request) στέλνεται προς ένα διακομιστή (server), αναφερόμενη σε ένα αντικείμενο, με μια μέθοδο αίτησης (request method). Η απάντηση (response) εμπεριέχει την αναπαράσταση του αντικειμένου που ζητήθηκε και τον κωδικό απάντησης (response code). Στο CoAP η επικοινωνία είναι ασύγχρονη χωρίς την δημιουργία κάποιας σύνδεσης και βασίζεται στο UDP. Το UDP δεν είναι τόσο αξιόπιστο στην μετάδοση δεδομένων όποτε το COAP περιλαμβάνει ειδικά μηνύματα και τρόπους ελέγχου ώστε να παρέχει αξιοπιστία στην επικοινωνία. Τα μηνύματα αυτά είναι : Conformable (CON), Non-Conformable (NON), Acknowledgement (ACK), Reset (RST) και σε συνδυασμό με την μέθοδο αίτησης ή τον κωδικό απάντησης που συμπεριλαμβάνονται σε κάποια από τα παραπάνω μηνύματα δηλώνουν αν είναι αίτηση ή απάντηση [7].

**Πρωτόκολλο MQTT**

Το πρωτόκολλο MQTT (Message Queuing Telemetry Transport) είναι διαφορετικό από τον τρόπο λειτουργίας του HTTP. Το MQTT υλοποιεί το μοντέλο Publish/Subscribe για την ανταλλαγή μηνυμάτων όπου ένας κόμβος του δικτύου (Publisher) μπορεί να στέλνει μηνύματα σε έναν άλλο κόμβο ή και περισσότερους (subscriber) με την χρήση ενός ενδιάμεσου διακομιστή (broker). Ο broker φιλτράρει τα μηνύματα και τα αποστέλλει στους ενδιαφερόμενους κόμβους. Οπότε ο publisher και ο subscriber για να πραγματοποιήσουν επικοινωνία δεν είναι απαραίτητο να γνωρίζει ο ένας τον άλλον άλλα πρέπει να γνωρίζουν τον broker (εικόνα 8. Ο τρόπος λειτουργίας είναι ασύγχρονος όπως και στο πρωτόκολλο COAP [8].

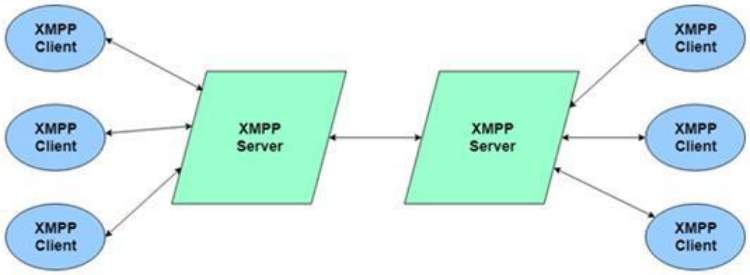
Το MQTT είναι ελαφρύ, ανοιχτού κώδικα και εύκολο στην υλοποίηση,επίσης είναι το επικρατέστερο πρωτόκολλο για ανταλλαγή μηνυμάτων μεταξύ ενός κόμβου και ενός IoT δικτύου. Για τους αναφερθέν λόγους τους λόγους το καθιστά πιο κατάλληλο για την υλοποίηση της SMGW.




Εικόνα 8. Πρωτόκολλο MQTT

### Πρωτόκολλο XMPP

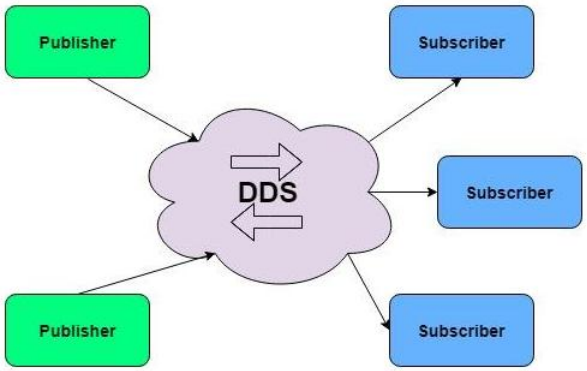
Το πρωτόκολλο XMPP (Extensible Messaging and Presence Protocol ) είναι πρότυπο IETF άμεσων μηνυμάτων (IM) και χρησιμοποιείται για συνομιλία με πολλούς συμμετέχοντες, φωνητικές κλήσεις, βιντεοκλήσεις και τηλεπικοινωνίες. Το XMPP αναπτύχθηκε από την κοινότητα ανοιχτού κώδικα Jabber για να υποστηρίξει ένα ανοικτό, ασφαλές, χωρίς spam και αποκεντρωμένο πρωτόκολλο ανταλλαγής μηνυμάτων. Το XMPP επιτρέπει στους χρήστες να επικοινωνούν μεταξύ τους στέλνοντας άμεσα μηνύματα στο Διαδίκτυο ανεξάρτητα από το λειτουργικό σύστημα που χρησιμοποιούν. Το XMPP προσφέρει στις εφαρμογές άμεσων μηνυμάτων να επιτυγχάνουν έλεγχο ταυτότητας, έλεγχο πρόσβασης, κρυπτογράφηση hop-by-hop και κρυπτογράφηση από άκρο σε άκρο και συμβατότητα με άλλα πρωτόκολλα. Ορισμένες από τις εφαρμογές XMPP περιλαμβάνουν τις Gtalk και Whatsapp. Στην εικόνα 9 δίνεται το διάγραμμα του XMPP [8].




Εικόνα 9. Πρωτόκολλο XMPP

### Πρωτόκολλο DDS

Το DSS (Data Distribution Service) είναι ένα πρωτόκολλο δημοσίευσης-εγγραφής (publish-subscribe) για επικοινωνίες M2M σε πραγματικό χρόνο, το οποίο έχει αναπτυχθεί από την ομάδα διαχείρισης αντικειμένων (Object Management Group - OMG). Σε αντίθεση με άλλα πρωτόκολλα εφαρμογών δημοσίευσης-εγγραφής, όπως το MQTT ή το AMQP, το DDS βασίζεται σε μια αρχιτεκτονική χωρίς μεσάζοντα και χρησιμοποιεί multicasting για να προσφέρει εξαιρετική ποιότητα υπηρεσιών (QoS) και υψηλή αξιοπιστία στις εφαρμογές της. Η αρχιτεκτονική δημοσίευσης- εγγραφής χωρίς μεσάζοντα ταιριάζει με τους περιορισμούς σε πραγματικό χρόνο για τις επικοινωνίες IoT και M2M [8]. Στην εικόνα 10 δίνεται το διάγραμμα του DDS.




Εικόνα 10. Πρωτόκολλο DDS

## 2.2 Έξυπνα δίκτυα (Smart grid's)

Τα δίκτυα ηλεκτρικής ενέργειας έχουν εξελιχθεί την τελευταία δεκαετία για να βελτιώσουν την αποδοτικότητα και την αξιοπιστία τους και να συμπεριλάβουν νέες Ανανεώσιμες Πηγές Ενέργειας που διαδραματίζουν σημαντικό ρόλο για την αντιμετώπιση της κλιματικής αλλαγής και την αντιμετώπιση της επερχόμενης ενεργειακής κρίσης (έως ένα βαθμό) με τα γεγονότα του πολέμου στην Ουκρανία και την ανεπάρκεια της ηγεσίας της Ευρωπαϊκής Ένωσης στο να πάρει μέτρα. Για την επίτευξη αυτών των στόχων, τα έξυπνα δίκτυα εφαρμόζονται τώρα στο δίκτυο ηλεκτρικής ενέργειας σε όλα τα επίπεδα, από τους σταθμούς παραγωγής ενέργειας μέχρι τους τελικούς χρήστες. Ο τελικός στόχος των έξυπνων δικτύων είναι να βελτιώσει την αποδοτικότητα, την αξιοπιστία, την οικονομία και τη βιωσιμότητα της παραγωγής και διανομής ηλεκτρικής ενέργειας σε μεγάλες κλίμακες [9].

Εκτός από τα γεγονότα του πολέμου στην Ουκρανία που δημιουργούν πρόβλημα στην προσφορά και ζήτηση της ηλεκτρικής ενέργειας και η μετάβαση στην ηλεκτροκίνηση δημιουργεί πρόσθετες δυσκολίες, ειδικά στο πλαίσιο των φορτιστών υψηλής ισχύος. Επομένως, προκειμένου να διευκολυνθεί η ισορροπία μεταξύ προσφοράς και ζήτησης ενέργειας και να διασφαλιστεί η ασφαλής λειτουργία του ηλεκτρικού δικτύου, απαιτείται η ευρεία εφαρμογή μέτρων διαχείρισης από την πλευρά της ζήτησης. Ένα ζήτημα σε αυτό το πλαίσιο είναι η επικοινωνία με κατανεμημένους ενεργειακούς πόρους (ΚΕΠ) [10] .Τα συστήματα ΚΕΠ χρησιμοποιούν συνήθως ανανεώσιμες πηγές ενέργειας, συμπεριλαμβανομένων μικρών υδροηλεκτρικών, βιομάζας, βιοαερίου, ηλιακής ενέργειας, αιολικής ενέργειας και γεωθερμικής ενέργειας.

Συνεπώς με την χρήση των έξυπνων δικτύων δημιουργούνται συνθήκες αγοράς ηλεκτρικής ενέργειας από Κατανεμημένους Ενεργειακούς Πόρους (ΚΕΠ) εισάγοντας το μοντέλο επικοινωνίας p2p. Μια άμεση συνέπεια του Έξυπνου Δικτύου είναι η ύπαρξη ενός ηλεκτρικού μοντέλου που θα είναι ικανό να διαχειρίζεται διαφορετικές συσκευές παραγωγής και αποθήκευσης με αποτελεσματικό και αποκεντρωμένο τρόπο, αναπτύσσοντας μια προηγμένη υποδομή μέτρησης (AMI-Advanced Metering Infrastructure).

### 2.2.1 Προηγμένη υποδομή μέτρησης (Advanced metering infrastructure - AMI)

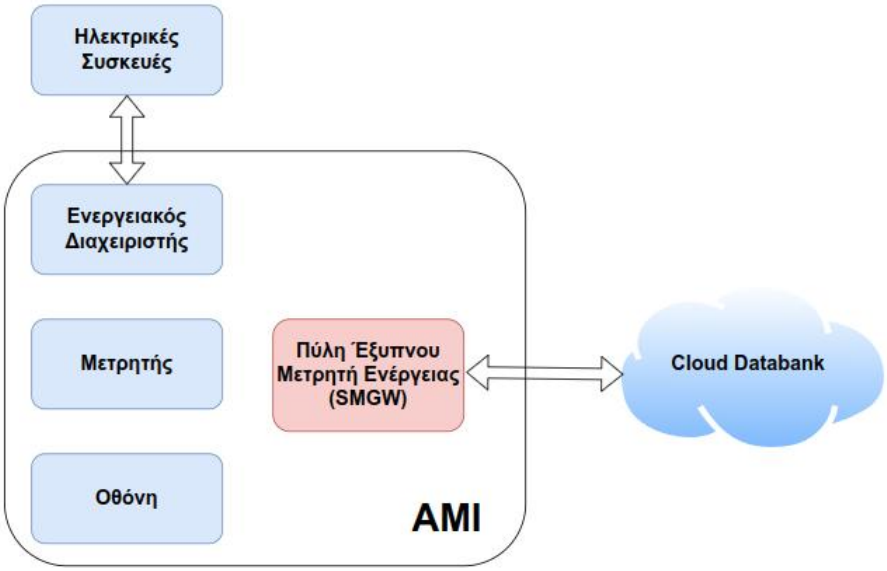
Η AMI είναι μια βελτιωμένη και τροποποιημένη έκδοση της αυτόματης ανάγνωσης του μετρητή,είναι ένα σύστημα που μετρά, συλλέγει, ελέγχει και αξιολογεί την κατανάλωση ενέργειας και μεταδίδει αυτές τις πληροφορίες σε επιχειρήσεις κοινής ωφέλειας, φορείς εκμετάλλευσης δικτύου και πελάτες. Η AMI έχει πολλά καθήκοντα όπως :

- Η ικανότητα self-healing : Για παράδειγμα, η ικανότητα self-healing θα πρέπει να μπορεί να επαναδιαμορφωθεί αυτόματα σε περίπτωση διακοπής λειτουργίας, εκτελώντας αυτόματα την τοποθεσία σφάλματος, την απομόνωση και την αποκατάσταση υπηρεσιών.
- Η προσαρμοστική τιμολόγηση ενέργειας.
- Η διαχείριση από την πλευρά της ζήτησης.
- Η βελτίωση της ενεργειακής απόδοσης.
- Η βελτίωση της αξιοπιστίας του έξυπνου δικτύου.
- Η διαλειτουργικότητα με άλλα συστήματα.
- Η παρακολούθηση και ο έλεγχος της ποιότητας ενέργειας.
- Η διαχείριση διακοπών.
- Η παροχή επικοινωνιών μεταξύ του κεντρικού συστήματος και των έξυπνων μετρητών.

- Η εξοικονόμηση ενέργειας και ενημέρωση του λογισμικού των έξυπνων μετρητών [11].

Το βασικό μπλοκ διάγραμμα της AMI δίνεται στην εικόνα 11 και αποτελείται από τα εξής στοιχεία:

- Ενεργειακός διαχειριστής: λαμβάνει RTP/TOU ή έχει προρυθμιστεί ώστε να παράγει υπό όρους σήματα για τον έλεγχο των συσκευών και της κατάστασης του εξοπλισμού.
- Μετρητής: είναι υπεύθυνος για την κατανάλωση ενέργειας και την παραγωγή μέτρησης και μετάδοσης πληροφοριών σε πραγματικό χρόνο.
- Οθόνη: είναι υπεύθυνη για τη λήψη και την επεξεργασία πληροφοριών και δείχνει την ανάλυση χρήσης ενέργειας.
- Πύλη Έξυπνου Μετρητή Ενέργειας (SMGW) : χρησιμοποιείται για την ανταλλαγή πληροφοριών και σημάτων με εξωτερικούς φορείς και και αν χρειαστεί συνδέει τα στοιχεία εντός της AMI.
- Cloud databank: χρησιμοποιείται για την καταγραφή και αποθήκευση των ιστορικών πληροφοριών ισχύος στις οποίες μπορούν να έχουν πρόσβαση εξωτερικοί παράγοντες και πελάτες [12]




Εικόνα 11. Βασικό μπλοκ διάγραμμα της AMI

Η πολυπλοκότητα και η απλότητα της υποδομής AMI εξαρτώνται από τις απαιτήσεις ασφάλειας δεδομένων, τη διαμόρφωση και την υποδομή σύνδεσης. Διαφορετικοί παράγοντες μπορούν να επωφεληθούν από την AMI όπως:

- Πελάτες: οπτικοποιώντας και παρουσιάζοντας τη χρήση ενέργειας, αυτοί οι πελάτες μπορούν να επιτύχουν εξοικονόμηση ενέργειας και τιμής.
- Υπηρεσίες κοινής ωφέλειας: εξοικονόμηση ανθρώπινου δυναμικού για συλλογή και καταγραφή δεδομένων, πρόταση νέων υπηρεσιών προστιθέμενης αξίας, βελτίωση της πρόγνωσης της ζήτησης.
- Διαχειριστές Συστήματος: προσφέρει στους διαχειριστές καλύτερη γνώση της κατάστασης και των συνθηκών του δικτύου. [12]

Το σύστημα AMI έχει αναπτυχθεί και εφαρμοστεί ευρέως στην Ευρώπη, τις ΗΠΑ και άλλες χώρες.

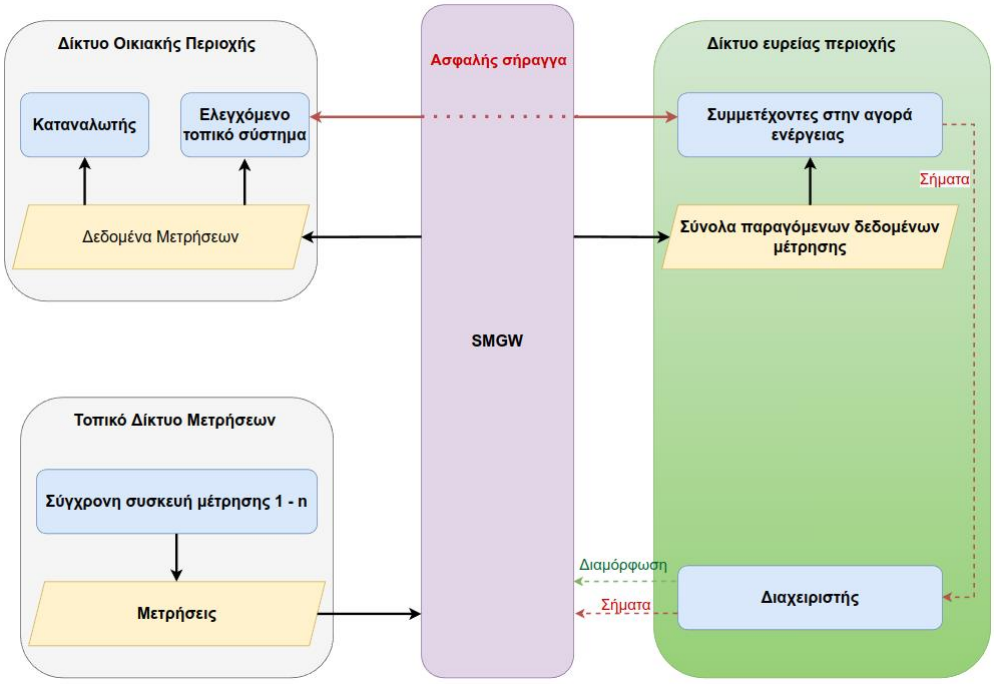
### 2.2.2 Πύλη Έξυπνου Μετρητή (Smart Meter Gateway - SMGW)

Η Διαχείριση από την πλευρά της ζήτησης (Demand Side Management - DSM) έχει υψηλές δυνατότητες για αποτελεσματική αντιστοίχηση ζήτησης και προσφοράς και για αντιμετώπιση σημείων συμφόρησης στο δίκτυο διανομής προσαρμόζοντας την πλευρά της ζήτησης. Τα έξυπνα συστήματα μέτρησης αποτελούν σημαντικό στοιχείο για την υλοποίηση του DSM. Θέτουν τις βάσεις για την παροχή κινήτρων μετατόπισης φορτίου προς την πλευρά της ζήτησης, καθώς και για την επαλήθευση της ανταπόκρισης σε αυτά τα κίνητρα. Η αντίδραση σε κίνητρα μετατόπισης φορτίου, όπως σήματα τιμών, μπορεί να πραγματοποιηθεί με χειροκίνητη ανθρώπινη παρέμβαση. Ωστόσο, οι πιο προηγμένες έννοιες περιλαμβάνουν Συστήματα Διαχείρισης Ενέργειας (Energy Management Systems - EMS) που ανταποκρίνονται αυτόματα στα κίνητρα DSM χρησιμοποιώντας τη διαθέσιμη ευελιξία με οριακή μόνο ανθρώπινη αλληλεπίδραση.

Στη Γερμανία, οι πύλες έξυπνων μετρητών (Smart Meter Gateway - SMGW) διαδραματίζουν κρίσιμο ρόλο στην ενεργοποίηση μέτρων από την πλευρά της ζήτησης (Demand Side Management - DSM), επιτρέποντας αμφίδρομη επικοινωνία με τους κατανεμημένους ενεργειακούς πόρους.

Η SMGW είναι μια συσκευή επικοινωνίας με δύο κύριες λειτουργίες. Μπορεί να μεταφέρει αυτόματα μετρήσεις από συνδεδεμένους έξυπνους μετρητές σε συμμετέχοντες στην αγορά ενέργειας (External Market Participants - EMP) και επιτρέπει στους EMP να στέλνουν κίνητρα ή εντολές για προσαρμογές φορτίου σε τοπικά πλαίσια ελέγχου, όπως τα (Energy Management Systems – EMS). EMS είναι ένα σύστημα εργαλείων με τη βοήθεια υπολογιστή που χρησιμοποιούνται από διαχειριστές δικτύων ηλεκτρικής ενέργειας για την παρακολούθηση, τον έλεγχο και τη βελτιστοποίηση της απόδοσης του συστήματος παραγωγής ή μεταφοράς.[10], [13]

Η SMGW λειτουργεί ως μεσολαβητής μεταξύ τριών διαφορετικών δικτύων επικοινωνίας όπως φαίνεται στην εικόνα 12.




Εικόνα 12. SMGW

Τα δίκτυα αυτά είναι :

- Το οικιακό δίκτυο (Home Area Network - HAN), είναι το δίκτυο επικοινωνίας από την πλευρά του καταναλωτή. Μέσα σε αυτό το δίκτυο υπάρχουν κατανεμημένοι ενεργειακοί πόροι , κουτιά ελέγχου και αποκεντρωμένα συστήματα διαχείρισης ενέργειας, τα οποία ονομάζονται συλλογικά ελεγχόμενο τοπικό σύστημα (Controllable Local System - CLS) στην υποδομή SMGW. Στο οικιακό δίκτυο , το SMGW παρέχει δεδομένα μετρητή στον καταναλωτή και διαγνωστικές πληροφορίες στον τεχνικό εξυπηρέτησης.
- Το τοπικό δίκτυο μετρήσεων περιέχει όλες τις συνδεδεμένες συσκευές μέτρησης.
- Το δίκτυο ευρείας περιοχής (Wide Area Network - WAN) χρησιμοποιείται για κάθε επικοινωνία με τον «διαχειριστή της πύλης Smart Meter» και τους «Εξωτερικούς συμμετέχοντες στην αγορά ενέργειας»

Ο διαχειριστής Smart Meter Gateway (GWA) ενεργεί ως αξιόπιστη αρχή για το SMGW. Κάθε SMGW εκχωρείται σε ένα GWA. Μόνο το GWA είναι εξουσιοδοτημένο να διαμορφώνει το SMGW και να εκτελεί λειτουργίες διαχείρισης, όπως ενημερώσεις λογισμικού. Για να εκτελέσει αυτές τις λειτουργίες, το GWA μπορεί να επικοινωνήσει με ασφάλεια με την SMGW χρησιμοποιώντας το κανάλι διαχείρισης. Όλοι οι άλλοι συνεργάτες επικοινωνίας του SMGW στο WAN είναι εξωτερικοί συμμετέχοντες στην αγορά ενέργειας (External Market Participants - EMP).

Πρέπει να γίνει διάκριση μεταξύ ενεργών και παθητικών EMP. Ενώ το παθητικό EMP λαμβάνει μόνο δεδομένα μέτρησης, το ενεργό EMP (aEMP) είναι εξουσιοδοτημένο να επικοινωνεί με ένα ή περισσότερα CLS στο HAN, χρησιμοποιώντας την ασφαλή σήραγγα δεδομένων που παρέχεται από το SMGW. Επομένως, η διαχείριση από την πλευρά της ζήτησης ενεργεί ως aEMP. Καταναλωτές είναι τα νομικά ή φυσικά πρόσωπα των οποίων η κατανάλωση μετράται. Κάθε καταναλωτής είναι ο κάτοχος των μεμονωμένων δεδομένων του μετρητή του και μπορεί να τα ανακτήσει σε μια καθορισμένη διεπαφή από η SMGW ή τη συσκευή μέτρησης. Για τη συντήρηση και τη διενέργεια διαγνωστικών ελέγχων, ο τεχνικός εξυπηρέτησης έχει τοπική πρόσβαση στο SMGW μέσω της διεπαφής HAN. [10], [13]

## 2.3 LPWAN δίκτυα

Τα LPWAN είναι τεχνολογία ασυρμάτων δικτύων ιδανικά για συσκευές IoT, καθώς περιλαμβάνουν τα εξής χαρακτηριστικά : μεγάλη διάρκεια ζωής μπαταρίας, μεγάλη εμβέλεια και χαμηλό κόστος. Σε αυτήν την ενότητα θα γίνει μια αναδρομή στους προκατόχους των σημερινών LPWAN δικτύων, στην συνέχεια αναφορά στις ποίο διαδεδομένες τεχνολογίες LPWAN καθώς και την επιλογή δικτύου LPWAN για την υλοποίηση της SMGW.

### 2.3.1 Ιστορία των LPWAN δικτύων

Παρόμοια δίκτυα που δεν ονομάζονταν LPWAN είχαν αναπτυχθεί στα τέλη της δεκαετίας του 1980 και στις αρχές της δεκαετίας του 1990 και είναι προκάτοχοι των δικτύων LPWAN (εικόνα 13).

Η Ademco είχε κατασκευάσει ένα ασύρματο δίκτυο το AlarmNet για την παρακολούθηση των πινάκων συναγερμού. Λειτουργούσε στην μπάντα των 928 MHz και έστελνε μικρές ποσότητες δεδομένων με χαμηλούς ρυθμούς. Τα δίκτυα AlarmNet λειτουργούν σε 18 μεγάλες περιοχές στις ΗΠΑ και καλύπτουν περίπου το 65% του αστικού πληθυσμού. Η εταιρεία ανήκει πλέον στη Honeywell.

Η Motorola την δεκαετία του 1980 είχε υλοποιήσει ένα ασύρματο δίκτυο ευρείας περιοχής με την ονομασία Ardis. Το Ardis είχε χαμηλό ρυθμό δεδομένων και χρησιμοποιήθηκε κυρίως για την αυτοματοποίηση πωλήσεων, την παρακολούθηση στόλου, το ηλεκτρονικό ταχυδρομείο και άλλες ηλεκτρονικές συναλλαγές και την ανταλλαγή μηνυμάτων. Απορροφήθηκε από την American

Mobile, η οποία πήρε τις βάσεις πελατών ARDIS και τις μετέφερε σε μεταγενέστερη τεχνολογία.[14]

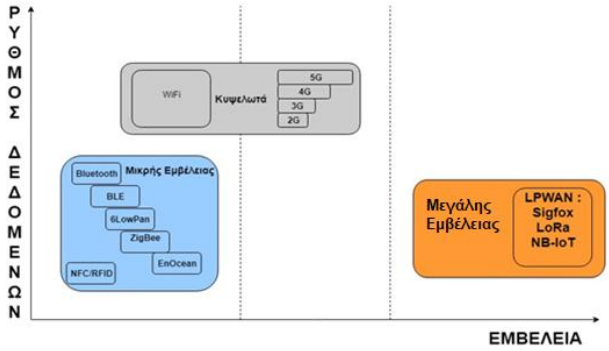

Εικόνα 13. Ιστορία των LPWAN δικτύων [14]

Με την εξάπλωση του IoT όλα και περισσότερες συσκευές συνδέονται στο διαδίκτυο γεγονός που οδήγησε στην επανεμφάνιση των τεχνολογιών LPWAN.

Η SIGFOX υλοποίησε το πρώτο σύγχρονο LPWAN δίκτυο στην Γαλλία το 2010. Χρησιμοποιεί συχνότητες 868-869 MHz και 902-928MHz ανάλογα την περιοχή. [15] Ένα πρωτόκολλο επικοινωνίας που βασίζεται στο LoRa που ονομάζεται LoRaWAN τυποποιήθηκε από τη LoRa–Alliance (πρώτη έκδοση το 2015).Χρησιμοποιεί συχνότητες 863-870 MHz και 433MHz για Ευρώπη. Το NB-IoT είναι μια τεχνολογία IoT στενής ζώνης που καθορίζεται στην ΄Εκδοση 13 του 3GPP τον Ιούνιο του 2016. Το NB-IoT μπορεί να συνυπάρχει με το GSM (παγκόσμιο σύστημα για κινητές επικοινωνίες) και το LTE (long-term evolution) υπό αδειοδοτημένες ζώνες συχνοτήτων (π.χ. 700 MHz, 800 MHz και 900 MHz). [16].

### 2.3.2 Σημερινές τεχνολογίες LPWAN

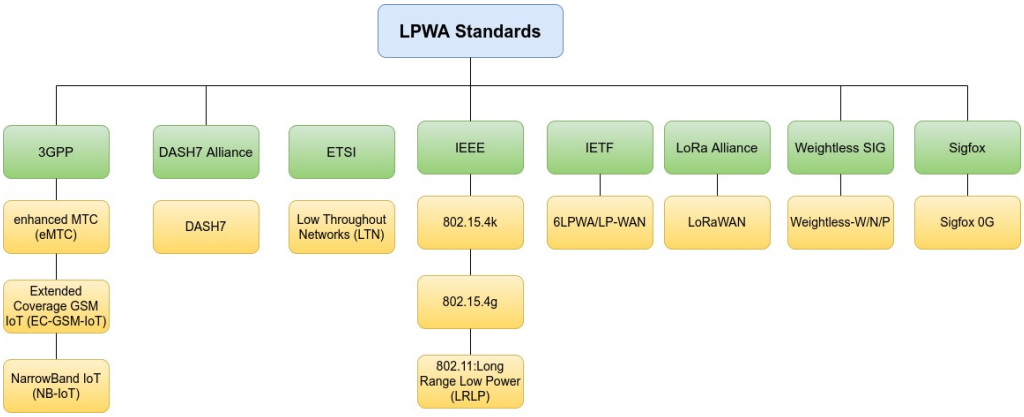
Τα δίκτυα LPWAN εκφράζουν μια νέα τάση στην εξέλιξη των τεχνολογιών IoT. Σε αντίθεση μετά συστήματα 3G / 4G ή WiFi, τα LPWAN δεν εστιάζουν στους υψηλούς ρυθμούς μετάδοσης ανά συσκευή ή την μείωση της καθυστέρησης, αντιθέτως οι βασικές μετρήσεις απόδοσης που ορίζονται για τα LPWAN είναι η ενεργειακή αποδοτικότητα,η επεκτασιμότητα και η κάλυψη. [17] Στην εικόνα 14 φαίνεται ένα block διάγραμμα των ρυθμών δεδομένων και της κάλυψης σε διάφορα ασύρματα δίκτυα. Στο πορτοκαλί πλαίσιο είναι μερικά από τα πιο δημοφιλή LPWAN δίκτυα και όπως φαίνεται στοχεύουν στην κάλυψη μεγαλύτερης εμβέλειας και όχι στο μεγαλύτερο ρυθμό δεδομένων.




Εικόνα 14. Ρυθμός Δεδομένων VS Εμβέλεια

Τα δίκτυα LPWAN λαμβάνουν όλο και περισσότερο δημοτικότητα στις βιομηχανικές και ερευνητικές κοινότητες λόγο των χαρακτηριστικών επικοινωνίας χαμηλής ισχύος, μεγάλης εμβέλειας

και χαμηλού κόστους. Παρέχει επικοινωνία μεγάλης εμβέλειας από 10 έως 40 χλμ σε αγροτικές ζώνες και από 1 έως 5 χλμ. σε αστικές ζώνες. Βασικά πλεονεκτήματά είναι η ενεργειακή απόδοση (διάρκεια ζωής της μπαταρίας 10+ χρόνια) καθώς και το χαμηλό κόστος το οποίο είναι 5E και ένα λειτουργικό κόστος 1E ανά ετός για κάθε συσκευή. Τα LPWAN προσαρμόζονται ιδανικά για εφαρμογές IoT όπου χρειάζονται την μετάδοση μικρών δεδομένων σε μεγάλο εύρος. Πολλές τεχνολογίες LPWAN έχουν προκύψει στο αδειοδοτημένο αλλά και στο μη αδειοδοτημένο εύρος ζώνης συχνοτήτων. [8], [18] Στην εικόνα 15 δίνονται τα πρότυπα των δικτύων LPWA.




Εικόνα 15. LPWA Standards

Τα πιο διαδεδομένα LPWAN είναι το NB-IoT, το LoRa, το LTE-M και το Sigfox. Είναι πρωτοποριακές τεχνολογίες που περιλαμβάνουν πολλές τεχνικές.

### 2.3.3 NB-IoT

Το NB-IoT είναι ένα από τα πιο διαδεδομένα δίκτυα ευρείας περιοχής χαμηλής ισχύος (LPWAN) που έχει αναπτυχθεί ώστε να επιτρέπει τη σύνδεση μιας ευρείας γκάμας συσκευών και υπηρεσιών χρησιμοποιώντας ζώνες συχνοτήτων κυψελοειδών δικτύων. Από τεχνική άποψη, το NB-IoT έχει σχεδιαστεί για συσκευές χαμηλού κόστους που λειτουργούν σε εύρος ζώνης χαμηλής συχνότητας 180 KHz για να προσφέρουν κάλυψη δικτύου άνω των 160 dB με latency περίπου 10 δευτερόλεπτα. Με αυτά τα χαρακτηριστικά, το NB-IoT μπορεί να στοχεύσει σε συσκευές IoT που είναι ανεκτικές σε καθυστέρηση ή βρίσκονται σε περιοχές όπου η μετάδοση σήματος είναι κακή. Η πρώτη έκδοση του NB-IoT κυκλοφόρησε το 2016 με την παρουσία της 13 έκδοσης της προδιαγραφής 3GPP και στην συνέχεια επεκτάθηκε στην έκδοση 14 με βελτιώσεις. Το NB-IoT μπορεί να αξιοποιήσει το υπάρχον δίκτυο του LTE, οπότε και τις τεχνικές του LTE όπως OFDMA για το κανάλι καθόδου, SC-FDMA για το κανάλι ανόδου, κωδικοποίηση καναλιού, προσαρμογή ρυθμού. Η ιδέα πίσω από το NB-IoT είναι να προσαρμοστεί το LTE πρωτόκολλο για να χρησιμοποιεί λιγότερη ισχύ από αυτή που χρειάζεται σε ένα κυψελωτό δίκτυο κινητής, ενώ παράλληλα διατηρεί τη χρήση παρόμοιων τεχνολογιών για ευκολότερη ενσωμάτωση με την ήδη υπάρχουσα υποδομή. Ένα NB-IoT δίκτυο μπορεί να συνυπάρχει με το LTE δίκτυο, και πιθανώς να χρησιμοποιεί μέρη από την υποδομή, περιλαμβανομένου του eNodeB hardware και του Δικτύου κορμού. [19], [20]

Οι βασικοί στόχοι του NB-IoT είναι :

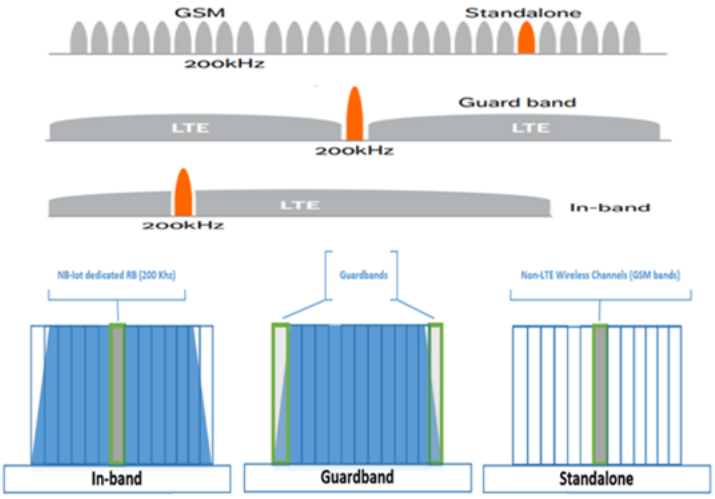
- Επέκταση της κάλυψης : 164 dB , 20 dB αυξημένη σε σχέση με το GPRS
- Μείωση της πολυπλοκότητας και του κόστους των UE's : μικρότερη από 5 E ανά μονάδα
- Μεγάλη διάρκεια ζωής μπαταρίας: Μεγαλύτερη από 10 χρόνια.
- Αυξημένη χωρητικότητα : 40 συσκευές ανά νοικοκυριό, και περίπου 55 χιλιάδες συσκευές ανά κυψέλη

- Ετερογένεια (συμβατότητα με άλλα δίκτυα)
- Αξιοπιστία
- Ασφάλεια [21]

Το μήκος ωφέλιμου φορτίου είναι μέχρι 1600 bytes. Το NB-IoT καταλαμβάνει ένα εύρος ζώνης (BW) 180 KHz τόσο για την κάτω ζεύξη (Downlink) όσο και για την άνω ζεύξη (Uplink) η οποία αντιστοιχεί σε ένα μπλοκ πόρων στην μετάδοση του LTE και μπορεί να αναπτυχθεί με τρεις τρόπους:

1. Αυτόνομη λειτουργία (Stand Alone Operation): Αξιοποίηση των συχνοτήτων του GSM που χρησιμοποιούνται σήμερα, διαθέτοντας ένα BW 200 KHz υπάρχει διαθέσιμο ένα διάστημα προστασίας (guard interval) 10 KHz που παραμένει και στις δύο πλευρές του φάσματος.
2. Λειτουργία ζώνης προστασίας (Guard Band Operation) : Αξιοποίηση των μπλοκ πόρων που δεν χρησιμοποιούνται μέσα σε μια ζώνη ασφαλείας του φορέα LTE.
3. Λειτουργία εντός της ζώνης (In-band operation): Χρήση μπλοκ πόρων που χρησιμοποιούνται από ένα φορέα LTE για την επικοινωνία του eNodeB.

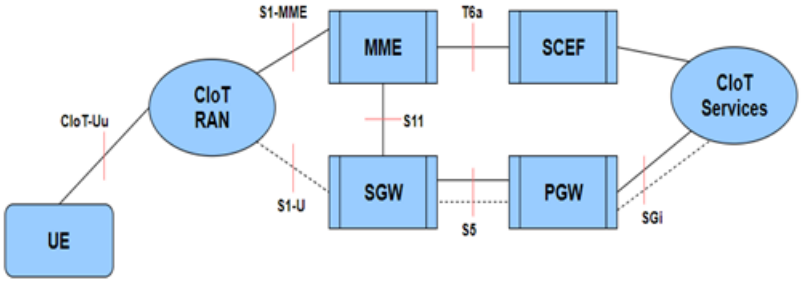
Στην εικόνα 16 δίνονται οι επιλογές ανάπτυξης NB-IoT.




Εικόνα 16. Επιλογές ανάπτυξης NB-IoT [22]

Στην εικόνα 17 δίνεται η δομή του δικτύου

Το δίκτυο κορμού του NB-IoT βασίζεται στο εξελιγμένο σύστημα πακέτων (EPS) και καθορίστηκαν δύο βελτιστοποιήσεις για το κυψελοειδές διαδίκτυο των πραγμάτων (CIoT). Η πρώτη βελτιστοποίηση αφορά στο επιπέδου χρήστη CIoT EPS και η δεύτερη στο επιπέδου ελέγχου CIoT EPS. Και τα δύο επίπεδα επιλέγουν την καλύτερη διαδρομή για τα πακέτα δεδομένων ελέγχου και χρήστη, για την άνω και κάτω ζεύξη. Η διαδρομή βελτιστοποίησης για το επιλεγμένο επίπεδο είναι ευέλικτη για το πακέτο δεδομένων που παράγεται από το κινητό σύνολο. Η διαδικασία πρόσβασης κυψέλης μιας τερματική συσκευής NB-IoT είναι παρόμοια με αυτή του LTE. Στο επίπεδο ελέγχου CIoT EPS, το εξελιγμένο δίκτυο επίγειας ραδιοεπικοινωνίας UMTS (E-UTRAN) χειρίζεται τις ραδιοεπικοινωνίες μεταξύ UE και MME και αποτελείται από τους εξελιγμένους σταθμούς βάσης που ονομάζονται eNodeB ή eNB. Το MME διαχειρίζεται την σηματοδοσία ελέγχου που σχετίζεται με την κινητικότητα και την ασφάλεια του UE. Αυτό συμπεριλαμβάνει τη διαχείριση της πρόσβασης του UE και στις συνδέσεις και τους πόρους του δικτύου. Όταν το UE απαιτεί μια μεταπομπή (handover) σε μια κυψέλη νέας περιοχής, το MME εκκινεί τη μεταφορά.




Εικόνα 17. Δομή του δικτύου

Στη συνέχεια, τα δεδομένα μεταβιβάζονται στην πύλη δικτύου πακέτων δεδομένων (PGW) μέσω μιας κεντρικής πύλης εξυπηρέτησης (SGW). Η PGW εξασφαλίζει την επικοινωνία μεταξύ του EPC και των εξωτερικών δικτύων IP, όπως το internet μέσω της διεπαφής SGi. Η SGW είναι το σημείο διασύνδεσης μεταξύ του ραδιοδικτύου και του EPC. Εξυπηρετεί κάθε UE λειτουργώντας ως δρομολογητής, αναλαμβάνει την δρομολόγηση στα πακέτα δεδομένων της συσκευής από το σταθμό βάσης προς την αντίστοιχη κεντρική πύλη του αντίστοιχου PGW.

Για πακέτα δεδομένων που δεν είναι IP, θα μεταφερθούν σε ένα νέο κόμβο, τον service capability exposure function (SCEF), ο οποίος μπορεί να παραδώσει δεδομένα τύπου μηχανής πάνω από το επίπεδο ελέγχου και να παρέχει μια «θεωρητική» διεπαφή για τις υπηρεσίες δικτύου (authentication and authorization, discovery and access network capabilities). Με τη βελτιστοποίηση επιπέδου χρήστη EPS CIoT , τόσο τα πακέτα IP όσο και τα μη IP μπορούν να μεταδοθούν μέσω RF φορέων (radio bearers) διαμέσου του SGW και PGW στο διακομιστή εφαρμογών. Συνοπτικά, για την NB-IoT, η υπάρχουσα αρχιτεκτονική δικτύου E-UTRAN μπορεί να επαναχρησιμοποιηθεί. [22]–[24]

Για τις ζώνες συχνοτήτων χρησιμοποιούνται οι ίδιες συχνότητες του LTE με ένα υποσύνολο για την χρήση στο NB-IoT. Για την έκδοση 13 του 3GPP οι ζώνες συχνοτήτων δίνονται στον Πίνακα 1

Πίνακας 1. Ζώνες συχνοτήτων για το NB-IoT 3GPP έκδοση 13 [25]

| Αριθμός Μπάντας | Εύρος συχνοτήτων για την άνω ζεύξη σε MHz | Εύρος συχνοτήτων για την Κάτω ζεύξη σε MHz |
|---|---|---|
| 1 | 1920 - 1980 | 2110 - 2170 |
| 2 | 1850 - 1910 | 1930 - 1990 |
| 3 | 1710 - 1785 | 1805 - 1880 |
| 5 | 824-849 | 869 - 894 |
| 8 | 880 - 915 | 925 - 960 |
| 12 | 699 - 716 | 729 - 746 |
| 13 | 777 - 787 | 746 - 756 |
| 17 | 704 - 716 | 734 - 746 |
| 18 | 815 - 830 | 860 - 875 |
| 19 | 830 - 845 | 875 - 890 |
| 20 | 832 - 862 | 791 - 821 |
| 26 | 814 - 849 | 859 - 894 |
| 28 | 703 - 748 | 758 - 803 |
| 66 | 1710 - 1780 | 2110 - 2200 |

Στην έκδοση 14 του 3GPP προστέθηκαν οι μπάντες που δίνονται στον Πίνακα 2.

Πίνακας 2. Ζώνες συχνοτήτων για το NB-IoT 3GPP έκδοση 14 [26]

| Αριθμός Μπάντας | Εύρος συχνοτήτων για την άνω ζεύξη σε MHz | Εύρος συχνοτήτων για την Κάτω ζεύξη σε MHz |
|---|---|---|
| 11 | 1427.9 - 1447.9 | 1475.9 – 1495.9 |
| 25 | 1850 - 1915 | 1930 - 1995 |
| 31 | 452.5 – 457.5 | 462.5 – 467.5 |
| 70 | 1695 - 1710 | 1995 - 2020 |

Αξίζει να σημειωθεί ότι οι περισσότερες συχνότητες βρίσκονται στο χαμηλότερο εύρος των υφιστάμενων LTE ζωνών. Αυτό είναι πιο ιδανικό για τις επικοινωνίες τύπου μηχανής διότι υπάρχουν πολλές συσκευές που αναμένεται να λειτουργούν υπό δύσκολες RF συνθήκες.

### 2.3.4 Sigfox

Η Sigfox έχει καταστεί μία από τις κύριες τεχνολογίες δικτύου χαμηλής ισχύος (LPWAN), καθώς έχει προσελκύσει την προσοχή των βιομηχανιών, των ακαδημαϊκών κοινοτήτων και των οργανισμών ανάπτυξης προτύπων τα τελευταία χρόνια. Αυτή η τεχνολογία αναπτύχθηκε το 2009 από την εταιρεία Sigfox (Labège, Γαλλία). Η Sigfox προσφέρει κάλυψη δικτύου που επιτρέπει την αμφίδρομη επικοινωνία για συσκευές IoT σε περισσότερες από 50 χώρες, καλύπτοντας πληθυσμό 1 δισεκατομμυρίων κατοίκων. [27]

Ο αριθμός των μηνυμάτων μέσω της άνω ζεύξης (αποστολή από κόμβους) περιορίζεται σε 140 μηνύματα την ημέρα και 4 για την κάτω ζεύξη (λήψη από σταθμό βάσης). Το μέγιστο μήκος ωφέλιμου φορτίου για κάθε μήνυμα άνω ζεύξης είναι 12 byte και 8 byte για την κάτω ζεύξη. Οι συγκεκριμένοι περιορισμοί έχουν ως αποτέλεσμα την χαμηλότερη κατανάλωση ενέργειας του συστήματος. Χρησιμοποιεί διαμόρφωση D-BPSK (Differential Binary Phase-Shift Keying) για την οποία τα μήνυμα των 4, 8 ή 12 byte έχουν εύρος ζώνης 100 Hz και ταχύτητα 100 bps στην Ευρώπη και 600 Hz με ταχύτητα 600bps στην Αμερική. Η D-BPSK αποδίδει καλή προστασία από παρεμβολές (π.χ. jamming), καθώς η λαμβανόμενη ισχύς συγκεντρώνεται σε πολύ στενό εύρος ζώνης και φτάνει σε ένα υψηλό λαμβανόμενο επίπεδο ισχύος οδηγώντας σε πολύ χαμηλή κατανάλωση ενέργειας. Αυτή η τεχνική διαμόρφωσης είναι μέρος της κατηγορίας διαμορφώσεων UNB (Ultra-Narrow Band), όπως η CSS (Chirp Spread Spectrum) που χρησιμοποιείται στο σχήμα διαμόρφωσης LoRa. [28] Η τεχνική UNB παρέχει αποτελεσματική χρήση φάσματος με αποτέλεσμα την αυξημένη χωρητικότητα και εμβέλεια δικτύου καθώς και χαμηλή κατανάλωση ενέργειας [29]. Ο προϋπολογισμός ζεύξης είναι 160db με εμβέλεια κάλυψης 10 χμ σε αστικό περιβάλλον και 40 χμ σε αγροτικό περιβάλλον.[30], [31] Στην Ευρώπη χρησιμοποιεί την ζώνη συχνοτήτων SRD860 και στις ΗΠΑ την ζώνη συχνοτήτων ISM900. Η μέγιστη ισχύς μετάδοσης άνω ζεύξης είναι 25 mW στην Ευρώπη (158 mW στις ΗΠΑ), ενώ η μέγιστη ισχύς εκπομπής στη κάτω ζεύξη είναι 500 mW στην Ευρώπη (4 W στις ΗΠΑ). [32], [33] Στον Πίνακα 3 δίνονται τα RF χαρακτηριστικά του Sigfox. [34]

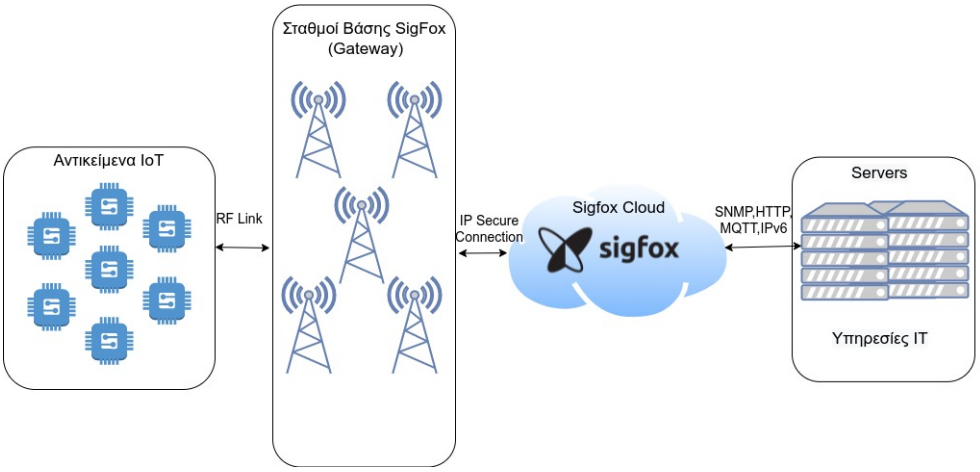
Η Sigfox αναπτύσσει τους ιδιόκτητους σταθμούς βάσης της εξοπλισμένους με RF που καθορίζονται από λογισμικό . Καθώς οι σταθμοί βάσης μπορούν να λαμβάνουν ταυτόχρονα μηνύματα σε όλα τα κανάλια, ο κάθε κόμβος μπορεί να επιλέξει τυχαία ένα κανάλι συχνότητας για να μεταδώσει τα μηνύματά του. Οι σταθμοί βάσης (gateways) προωθούν τα μηνύματα στο cloud που καταλήγουν στους servers που παρέχουν υπηρεσίες IT.

Τα πλεονεκτήματα του Sigfox είναι :

- Χαμηλή κατανάλωση ισχύος
- Λειτουργεί καλά για απλές συσκευές που μεταδίδουν σπάνια, επειδή στέλνει πολύ μικρές ποσότητες δεδομένων με αργό ρυθμό δεδομένων.

Πίνακας 3. Sigfox : Τεχνικές λεπτομέρειες RF

| | RC1 | RC2 | RC3 | RC4 | RC5 | RC6 | RC7 |
|---|---|---|---|---|---|---|---|
| **Κεντρική συχνότητα άνω ζεύξης (MHz)** | 868.130 | 902.200 | 923.200 | 920.800 | 923.300 | 865.200 | 868.800 |
| **Κεντρική συχνότητα κάτω ζεύξης (MHz)** | 869.525 | 905.200 | 922.200 | 922.300 | 922.300 | 866.300 | 869.100 |
| **Ρυθμός δεδομένων άνω ζεύξης (bit/s)** | 100 | 600 | 100 | 600 | 100 | 100 | 100 |
| **Ρυθμός δεδομένων κάτω ζεύξης (bit/s)** | 600 | 600 | 600 | 600 | 600 | 600 | 600 |
| **Προτεινόμενη EIRP (dBm)** | 16 | 24 | 16 | 24 | 14 | 16 | 16 |
| **Ιδιαιτερότητες** | Κύκλος λειτουργίας 1% * | Αναπήδηση συχνότητας ** | Listen Before Talk *** | Αναπήδηση συχνότητας ** | Listen Before Talk ** | | Κύκλος λειτουργίας 1% * |
| | * Ο κύκλος λειτουργίας είναι 1% του χρόνου ανά ώρα (36 δευτερόλεπτα). Για ωφέλιμο φορτίο 8 έως 12 byte, αυτό σημαίνει 6 μηνύματα την ώρα, 140 την ημέρα. | **Η συσκευή εκπέμπει κάθε μήνυμα 3 φορές σε 3 διαφορετικές συχνότητες. Μέγιστος χρόνος ενεργοποίησης 400 ms ανά κανάλι. Καμία νέα εκπομπή πριν από 20 δευτερόλεπτα. | ***Οι συσκευές πρέπει να επαληθεύσουν ότι το κανάλι 200 kHz που λειτουργεί από τη Sigfox είναι απαλλαγμένο από σήμα ισχυρότερο από -80 dBm πριν από τη μετάδοση. | | | | |




Εικόνα 18. Αρχιτεκτονική δικτύου Sigfox

- Υποστηρίζει μια ευρεία περιοχή κάλυψης στις περιοχές όπου δραστηριοποιείται.

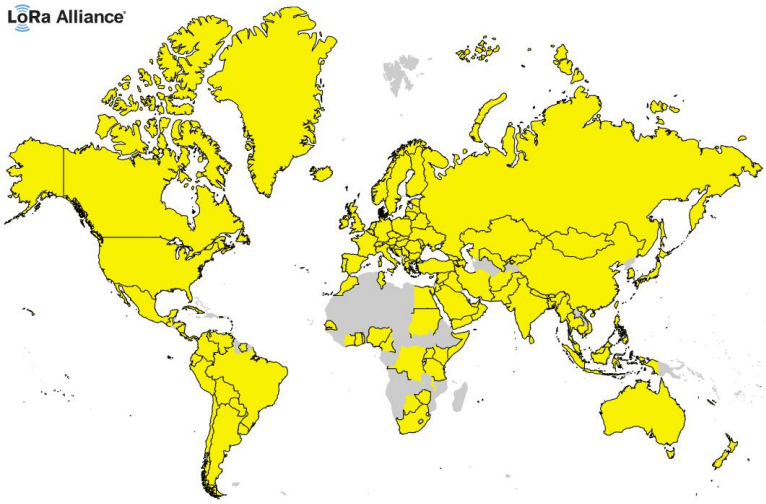
Τα μειονεκτήματα του Sigfox είναι :

- Δεν έχει αναπτυχθεί παντού, επομένως δεν θα λειτουργήσει για μεγάλο φάσμα περιπτώσεων χρήσης προς το παρών
- Η επικοινωνία κατευθύνεται καλύτερα από το τελικό σημείο στο σταθμό βάσης. Έχει αμφίδρομη λειτουργικότητα αλλά η χωρητικότητα του από το σταθμό βάσης προς το τελικό σημείο είναι περιορισμένη.
- Η κινητικότητα είναι δύσκολη με τις συσκευές Sigfox.

### 2.3.5 LoRa

Το δίκτυο LoRa είναι ένα από τα πιο ευρέως υιοθετημένα LPWAN πρότυπα διεθνώς. Το όνομα του προέρχεται από το Long Range (μακρά εμβέλεια) και αρχικά αναπτύχθηκε από την Cycleo το 2009. Το 2012 εξαγοράστηκε από την Semtech και έθεσε υπό έλεγχο την LoRa Alliance το 2014.[16] Σε παγκόσμια κλίμακα 166 παρόχοι, παρέχουν υπηρεσίες χρησιμοποιώντας το LPWAN

δίκτυο LoRa [35]. Οι χώρες που δραστηριοποιούνται οι πάροχοι χρησιμοποιώντας LoRa δίνονται στην εικόνα 19:




Εικόνα 19. Με κίτρινο χρώμα οι χώρες που δραστηριοποιούνται παρόχοι που χρησιμοποιούν LoRa [35]

Η τεχνολογία LoRa επιτρέπει τη σύνδεση επικοινωνίας μεγάλης εμβέλειας και ανήκει στο φυσικό επίπεδο (PHY) του LoRaWAN, το οποίο καθορίζει το πρωτόκολλο επικοινωνίας και την αρχιτεκτονική του δικτύου και ανήκει στο επίπεδο ελέγχου πρόσβασης μέσων (MAC). [36]

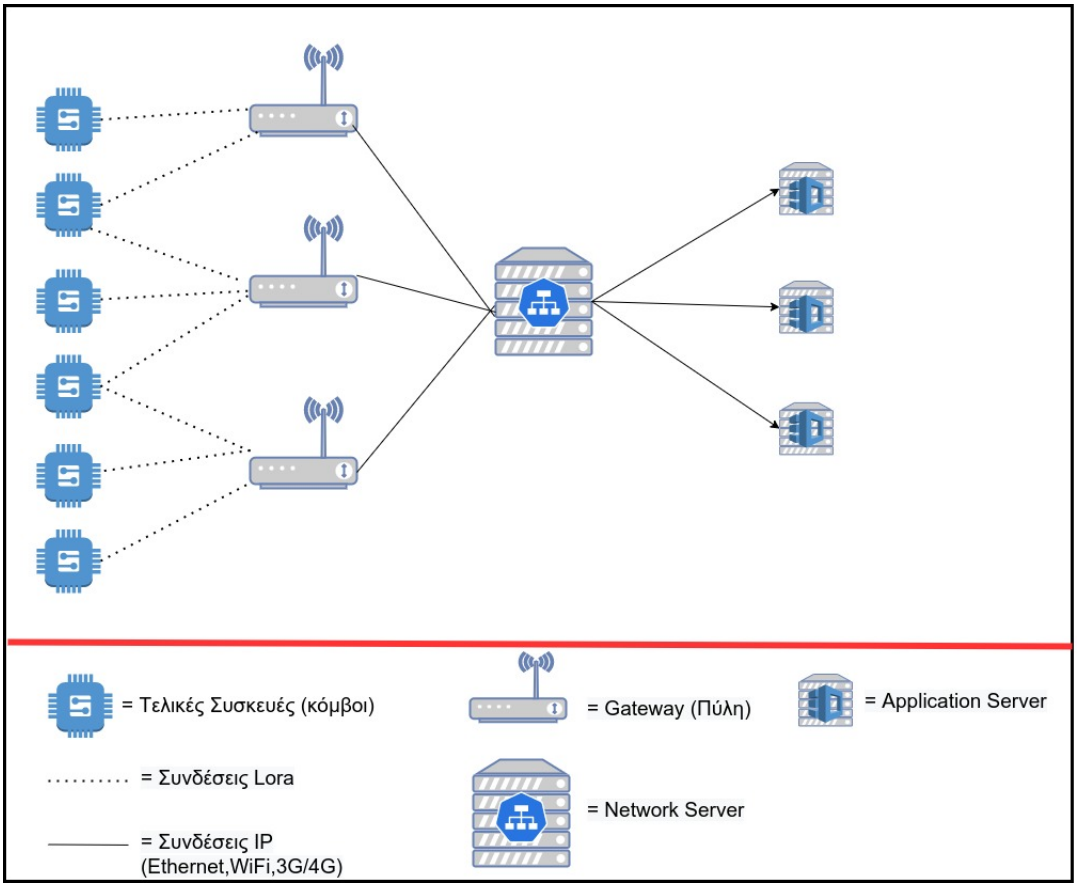
Το LoRa χρησιμοποιεί όπως και το Sigfox ζώνες ISM χωρίς άδεια, συγκεκριμένα για την Ασία 433 MHz, για την Ευρώπη 868 MHz, στην Βόρεια Αμερική 915 MHz και διαμορφώνει τα σήματα χρησιμοποιώντας μια ιδιόκτητη διαμόρφωση διασποράς φάσματος (Chirp Spread Spectrum - CSS). Μέσω της διαμόρφωσης διασποράς φάσματος εξασφαλίζεται αμφίδρομη επικοινωνία όπου ένα σήμα στενής ζώνης το εξαπλώνει σε ένα ευρύτερο εύρος ζώνης καναλιού. Το σήμα που προκύπτει έχει χαμηλά επίπεδα θορύβου επιτρέποντας επαρκή προστασία παρεμβολών.

Διαθέτει έξι παράγοντες διασποράς που οδηγούν σε προσαρμοστικούς ρυθμούς δεδομένων. Αυτή η δυνατότητα επιτρέπει την ταυτόχρονη μετάδοση πολλαπλών διαφορετικών σημάτων στο ίδιο κανάλι συχνότητας. Ο ρυθμός δεδομένων κυμαίνεται ανάμεσα σε 300 bps και 50 kbps και το μέγιστο μήκος ωφέλιμου φορτίου για κάθε μήνυμα είναι 243 byte.

Λόγω της χρήσης ζωνών χωρίς άδεια, το δίκτυο LoRa είναι ανοιχτό σε πελάτες που δεν έχουν εξουσιοδότηση από ρυθμιστές ραδιοσυχνοτήτων. Ως αποτέλεσμα, το δίκτυο LoRa είναι εύκολο να αναπτυχθεί σε εμβέλεια άνω των πολλών χιλιομέτρων και εξυπηρετεί πελάτες με ελάχιστο κόστος επένδυσης και συντήρησης.

Το δίκτυο LoRa υποστηρίζει τοπολογία πλέγματος mesh. Η προτεινόμενη τοπολογία και η πιο διαδεδομένη για το LoRa είναι η τοπολογία τύπου αστέρα. Με αυτήν την τοπολογία αστέρα αυξάνεται η διάρκεια ζωής της μπαταρίες των τελικών συσκευών (κόμβων). Στην εικόνα 20 δίνεται η αρχιτεκτονική ενός δικτύου LoRa.

Οι τελικές συσκευές (κόμβοι), είναι αισθητήρες, ενεργοποιητές όπου επικοινωνούν με πύλες στέλνοντας τα δεδομένα. Οι πύλες παρέχουν αμφίδρομη επικοινωνία. Ο κάθε κόμβος δεν επικοινωνεί με μια συγκεκριμένη πύλη, συνεπώς τα δεδομένα που μεταδίδονται από ένα κόμβο είναι εφικτό να λαμβάνονται από πολλαπλές πύλες. Η κάθε πύλη προωθεί τα ληφθέντα δεδομένα στον διακομιστή δικτύου όπου έχει σύνδεση IP (Ethernet,Wifi,3G,4G). Ο διακομιστής δικτύου υλοποιεί το πρωτόκολλο LoRaWAN, επικυρώνει την αυθεντικότητα και την ακεραιότητα των συσκευών, φιλτράρει τα περιττά ληφθέντα πακέτα, επιλέγει τις πύλες που χρησιμοποιούνται για downlinks και στέλνει εντολές ADR για τη βελτιστοποίηση του ρυθμού δεδομένων των συσκευών και της κατανάλωσης ενέργειας στο δίκτυο. Ο διακομιστής εφαρμογών είναι υπεύθυνος για την αποκω-




Εικόνα 20. Αρχιτεκτονική Δικτύου Lora

δικοποίηση των πακέτων που αποστέλλονται από τις συσκευές και τη δημιουργία των πακέτων που πρέπει να σταλούν πίσω στις συσκευές. Τα δεδομένα μπορούν να ενσωματωθούν σε υπάρχοντα συστήματα διαχείρισης δεδομένων ή πλατφόρμες IoT όπως το AWS, το Azure και το Google Cloud. [36]–[38]

Τα πλεονεκτήματα του LoRa είναι :

- Είναι ιδανικό για εφαρμογές κτιρίου
- Είναι εφικτό να γίνει ρύθμιση και διαχείριση ενός δικτύου από τον χρήστη
- Ανάπτυξη με χαμηλό κόστος
- Δεν υπάρχουν περιορισμοί στην αποστολή μηνυμάτων
- Ικανοποιητικός ρυθμός δεδομένων
- Οι συσκευές LoRa λειτουργούν καλά όταν βρίσκονται σε κίνηση, γεγονός που τις καθιστά χρήσιμες για την παρακολούθηση περιουσιακών στοιχείων σε κίνηση, όπως αποστολές (OTAA ενεργοποίηση).
- Μεγάλη διάρκεια ζωής μπαταριών.

Πίνακας 4. Βασικά χαρακτηριστικά δικτύων Sigfox, LoRa, NB-IoT [16], [39]

| | Sigfox | LoRa | NB-IoT |
|---|---|---|---|
| Μέγιστη<br><br>Απώλεια Ζεύξης (MCL) | 160 dB | 157 dB | 164 dB |
| Εύρος κάλυψης | 10 χλμ (αστική περιοχή) ,<br>40 χλμ (αγροτική περιοχή) | 5 χλμ (αστική περιοχή) ,<br>20 χλμ (αγροτική περιοχή) | 1 χλμ (αστική περιοχή) ,<br>10 χλμ (αγροτική περιοχή) |
| Διαμόρφωση | BPSK | CSS | QPSK |
| Αδειοδοτημένο φάσμα | ΟΧΙ | ΟΧΙ | ΝΑΙ |
| Περιορισμοί κύκλου εργασιών | ΝΑΙ | ΝΑΙ | ΟΧΙ |
| Περιορισμοί ισχύος εξόδου | Ναι (14dBm=25mW) | Ναι (14dBm=25mW) | Όχι (23 dBm=200mW) |
| Εύρος ζώνης καναλιού | 100 Hz | 125, 250 και 500 KHz | 200 KHz |
| Ρυθμός δεδομένων για την κάτω ζεύξη (downlink) | 0.1 kbps | 0.3-50 kbps | 0.5-200 kbps |
| Ρυθμός δεδομένων για την άνω ζεύξη (uplink) | 0.1 kbps | 0.3-50 kbps | 0.5-200 kbps |
| Προσαρμοστικός ρυθμός δεδομένων | ΟΧΙ | ΝΑΙ | ΟΧΙ |
| Μήκος οφέλιμου φορτίου (payload) | 12 bytes uplink / 8 bytes downlink | 243 bytes | 1600 bytes |
| Όριο μηνυμάτων ανά ημέρα | 140 uplink / 14 downlink | ΟΧΙ (χωρίς χρέωση) | ΟΧΙ (ανάλογα την χρέωση) |
| Ασφάλεια | Χαμηλή (AES-128) | Μεσαία (AES-128) | Υψηλή (LTE Security) |
| Standardization | Σε συνεργασία με<br>ETSI | LoRa-Alliance | 3GPP |
| Επιτρέπεται το ιδιωτικό δίκτυο | ΟΧΙ | ΝΑΙ | ΟΧΙ |
| Διάρκεια ζωής μπαταρίας (σε χρόνια) | 10+ | 10+ | 10+ |
| Κόστος φάσματος | ΟΧΙ (Ελεύθερο) | ΟΧΙ (Ελεύθερο) | >500 Μ€ / MHz |
| Κόστος εγκατάστασης | >4000€ / Σταθμό Βάσης | >100€/ Gateway<br>>1000 / Σταθμό Βάσης | >15000€ / Σταθμό Βάσης |
| Κόστος τελικής συσκευής | <2€ | 3-5€ | >20€ |

## 2.4 Επιλογή LPWAN για την υλοποίηση της SMGW

Το κάθε δίκτυο LPWAN έχει τα πλεονεκτήματα και τα μειονεκτήματα του. Δεν υπάρχει ”καλύτερο” LPWAN δίκτυο, υπάρχει η καλύτερη επιλογή δικτύου σύμφωνα με τις απαιτήσεις ενός IoT συστήματος που θέλουμε να υλοποιήσουμε. Στον Πίνακα 4 δίνονται τα βασικά χαρακτηριστικά των δικτύων Sigfox, LoRa, NB-IoT όπως αναλύσαμε στις προηγούμενες ενότητες.

Η SMGW θα στέλνει τα δεδομένα από τον έξυπνο μετρητή ενέργειας. Το Sigfox για την συγκεκριμένη εφαρμογή έχει κάποιους περιορισμούς σε σχέση με το LoRa και NB-IoT όπως το μήκος ωφέλιμου φορτίου για την ανοδική ζεύξη είναι 12 bytes, ένα τυπικό μέγεθος μηνύματος έξυπνου μετρητή είναι περίπου 100 byte [40], η τακτική παρακολούθηση του δικτύου βοηθούν τους παρόχους ηλεκτρικής ενέργειας να λάβουν αποφάσεις σε περιόδους αιχμής φορτίου και σε διακοπές οπότε ένας υψηλός ρυθμός δεδομένων και χαμηλότερη καθυστέρηση (latency) βοηθούν στην επίτευξη του στόχου και το LoRa μαζί με το NB-IoT παρέχουν υψηλότερους ρυθμούς δεδομένων και χαμηλότερη καθυστέρηση. Ακόμα ένας βασικός παράγοντας είναι πρός το παρών η μικρή κάλυψη του Sigfox στην Ελλάδα. Για τους παραπάνω λόγους το LoRa και το NB-IoT είναι πιο κατάλληλο για εφαρμογές έξυπνου μετρητή ενέργειας.

Το NB-IoT έχει υψηλότερο ρυθμό δεδομένων , καλή κάλυψη δικτύου. Ωστόσο μπορεί να υπάρχουν περιοχές που δεν υπάρχει κάλυψη. Το NB-IoT δεν είναι ελεύθερο και για οικονομική λύση θα πρέπει να χρησιμοποιήσουμε τις υποδομές των παρόχων κινητής τηλεφωνίας. Ένα επίσης πλεονέκτημα του NB-IoT είναι η παροχή μεγαλύτερης ασφάλειας

Το LoRa εχει χαμηλότερους ρυθμούς δεδομένων από το NB-IoT αλλά είναι ικανοποιητικοί για το σύστημα που θα γίνει υλοποίηση. Επίσης είναι ελεύθερο δεν απαιτεί άδεια φάσματος και το κόστος εγκατάστασης είναι χαμηλό. Παρέχει χαμηλότερη ασφάλεια σε σχέση με το NB-IoT, υπάρχουν όμως λύσεις για την εξασφάλιση της επιπλέον ασφάλειας στο LoRa.

Για την υλοποίηση της SMGW θα γίνει επιλογή για LPWAN δίκτυο το LoRa. Θέλουμε το σύστημα να είναι ανοιχτού κώδικα, παραμετροποιήσιμο, να έχει χαμηλό κόστος εγκατάστασης και να δοθεί έμφαση στην ασφάλεια και την ακεραιότητα των δεδομένων. Όπως θα δούμε στην συνέχεια η χρήση secure element και του blockchain Helium θα δώσουν λύσεις για την επιπλέον ασφάλεια του LoRa.

Επίσης το NB-IoT και το LoRa μπορούν να συνυπάρχουν και να γίνεται παράλληλα ανάπτυξη

για χρήση σε SMGW. Οπότε για Πανελλαδική κάλυψη μπορούν να αξιοποιηθούν και οι 2 τεχνολογίες.

Στο κεφάλαιο 3 που ακολουθεί θα γίνει ανάλυση στο θεωρητικό και πρακτικό υπόβαθρο που αφορά την τεχνολογία του LPWAN δικτύου LoRa.

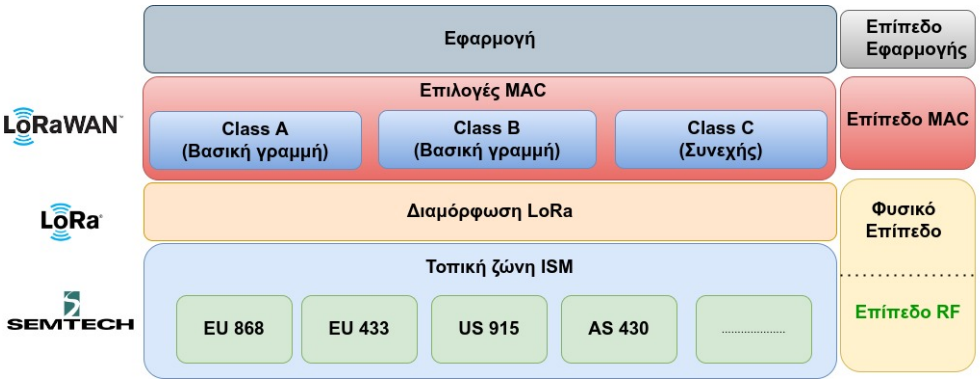
# ΚΕΦΑΛΑΙΟ 3

# LoRa

Σε αυτό το κεφάλαιο θα γίνει ανάλυση της τεχνολογίας LoRa σε θεωρητικό υπόβαθρο και στην συνέχεια σε πρακτικό. Οι πιο διαδεδομένοι LoRa Network Servers είναι το TTN και το Chirpstack. Για δοκιμές των LoRa συσκευών θα γίνει στο Chirpstack για λόγους που θα αναφερθούν στην συνέχεια και πιο συγκεκριμένα στο server του εργαστηρίου restqmlab του Πανεπιστημίου Δυτικής Αττικής. Στο παράρτημα 2 δίνονται

Στο παράρτημα 2 οι οδηγίες εγκατάστασης του ChirpStack για το Raspberry (ίδια βήματα για όλα τα linux συστήματα).

Στην εικόνα 21 δίνεται η Στοίβα Πρωτοκόλλου LoRa που θα γίνει ανάλυση στην συνέχεια.




Εικόνα 21. Στοίβα πρωτοκόλλου LoRa

## 3.1 Διαμόρφωση Φυσικού Επιπέδου

### 3.1.1 Διαμόρφωση LoRa

Στην τεχνική διαμόρφωσης του LoRa η εξάπλωση του φάσματος πραγματοποιείτε με την δημιουργία ενός σήματος chirp που διαφέρει διαρκώς σε συχνότητα. Ένα σήμα chirp αντιστοιχεί σε μια σάρωση στη συχνότητα στο αντίστοιχο εύρος ζώνης. Το εύρος ζώνης συχνότητας του chirp είναι ισοδύναμο με το φασματικό εύρος ζώνης του διαμορφωμένου σήματος LoRa. [41]. Για κάθε φέρουσα συχνότητα η οποία συμβολίζετε με $f_c$, η στιγμιαία συχνότητα ενός άνω chirp αυξάνεται από $f_c - \frac{BW}{2}$ σε $f_c + \frac{BW}{2}$ ενώ η στιγμιαία συχνότητα ενός κάτω chirp μειώνεται από $f_c + \frac{BW}{2}$ σε

$f_c - \frac{BW}{2}$. Η χρονική διάρκεια ενός chirp καθορίζεται από τον παράγοντα διασποράς (spreading factor) και το εύρος ζώνης (bandwidth), και δίνεται από την σχέση 3.1 : [42]

$$t = \frac{2^{SF}}{BW} \tag{3.1}$$

Η εξάπλωση του φάσματος χωρίζει κάθε bit δεδομένων ωφέλιμου φορτίου σε πολλαπλά σύμβολα. Αυτό μπορεί να οδηγήσει σε βελτίωση της προστασίας από θόρυβο στον τομέα χρόνου καθώς και να επιφέρει κέρδος επεξεργασίας για τον δέκτη όπως φαίνεται στον πίνακα 2.

Ο υπολογισμός της χρονικής διάρκειας ενός συμβόλου εξαρτάται από τον συντελεστή διασποράς SF, την ελάχιστη συχνότητα $f_{min}$ και τη μέγιστη συχνότητα $f_{max}$ δηλαδή το εύρος ζώνης (BW) καθώς και τα bit εισόδου. Υπολογίζεται από την σχέση 3.2 : [43]

$$t_s = \frac{SF}{BW} = \frac{SF}{f_{max} - f_{min}} \tag{3.2}$$

### 3.1.2 Συντελεστής διασποράς SF

Υπάρχουν εφτά συντελεστές διασποράς (SF) για τον έλεγχο του ρυθμού bit, τη βελτίωση της εμβέλειας και τη μείωση της κατανάλωσης ενέργειας με τιμές {6,7,8,9,10,11,1}. Ο συντελεστής διασποράς είναι η παράμετρος που ελέγχει τη χρονική κατανομή κάθε bit δεδομένων. Το SF αντιστοιχεί στον αριθμό των bits ανά σύμβολο και αν επιλεγεί υψηλότερος συντελεστής διασποράς, κάθε bit δεδομένων ωφέλιμου φορτίου θα κατανεμηθεί σε περισσότερα σύμβολα. Το σύνολο των διαδοχικών bit SF αποτελεί σύμβολο. Για κάθε SF, υπάρχουν δύο κωδικοί, ένας για το σύμβολο '0' και ένας για το σύμβολο '1'. Όσο περισσότερο αυξάνεται το SF, τόσο μεγαλύτερο είναι το μήκος του κώδικα και επομένως τόσο χαμηλότερος είναι ο ρυθμός δεδομένων bit.

Ο αριθμός των πιθανών συμβόλων είναι ίσος με $M = 2^{SF}$. Επομένως, η σχέση μεταξύ του ρυθμού bit $D_b$ και του ρυθμού συμβόλων $D_s$ μπορεί να γραφτεί ως: $D_s = D_b/SF$ . Το φάσμα εξάπλωσης λαμβάνεται μέσω του chirp. Όταν η παράγωγος της διακύμανσης της συχνότητας είναι θετική τότε έχουμε να κάνουμε με ένα επάνω chirp, αντίστροφα είναι ένα κάτω chirp. [44]

### 3.1.3 Εύρος ζώνης BW

Το Εύρος ζώνης (BW) είναι το εύρος ζώνης που χρησιμοποιείται κατά τη διάρκεια μιας μετάδοσης. Εκφράζεται σε Hertz. Για τη διαμόρφωση LoRa, είναι δυνατή η αποστολή ενός τσιρπ ανά δευτερόλεπτο και ανά Hertz εύρους ζώνης. Δηλαδή, στην επιλογή ενός εύρος ζώνης 125 kHz, είναι δυνατή η αποστολή 125000 τσιρπ ανά δευτερόλεπτο. [45]

### 3.1.4 Κώδικας διόρθωσης σφαλμάτων προώθησης (CR)

Το LoRa διαθέτει έναν κωδικό διόρθωσης σφαλμάτων προώθησης (CR) για να να αντιμετωπιστεί η επιζήμια επίδραση της παρεμβολής RF. Ο ρυθμός διόρθωσης σφαλμάτων (CR) ισούται με 4 / ( 4 + n ) , με $n \in \{1, 2, 3, 4\}$. Η προεπιλεγμένη τιμή του CR είναι ίση με 4/5. [37], [46]

Γνωρίζοντας το SF,BW,CR που αναλύσαμε πριν μπορούμε να υπολογίσουμε τον ρυθμό δεδομένων $R_b$ σύμφωνα με την σχέση 3.3: [37]

$$R_b = SF * \frac{BW}{2^{SF}} * CR \tag{3.3}$$

Ένα παράγειγμα υπολογισμού ρυθμού δεδομένων για ζώνη συχνοτήτων Ευρώπης 868MHz, προτεινόμενο SF=7, BW=125KHz και CR=4/5 ο ρυθμός δεδομένων υπολογίζεται σύμφωνα με την σχέση 3.4:

Πίνακας 5. Ρυθμός bit ή ρυθμός δεδομένων για τελική συσκευή EU LoRaWAN.

| Ρυθμός Δεδομένων | Ρυθμίσεις | Ενδεικτικός φυσικός ρυθμός bit [bits/sec] |
|---|---|---|
| 0 | SF12/125KHz | 250 |
| 1 | SF11/125KHz | 440 |
| 2 | SF10/125KHz | 980 |
| 3 | SF9/125KHz | 1760 |
| 4 | SF8/125KHz | 3125 |
| 5 | SF7/125KHz | 5470 |
| 6 | SF6/125KHz | 11000 |

$$R_b = 7 * \frac{125 * 10^3}{2^7} * \frac{4}{5} = 5,47kbps \tag{3.4}$$

Στον Πίνακα 5 δίνεται ο ρυθμός δεδομένων ανάλογα την επιλογή του SF και BW. [47]

Ένα μεγαλύτερο SF αυξάνει τον χρόνο στον αέρα, γεγονός που αυξάνει την κατανάλωση ενέργειας, μειώνει τον ρυθμό μετάδοσης δεδομένων και βελτιώνει το εύρος επικοινωνίας. Για επιτυχή επικοινωνία, όπως καθορίζεται από το SF, η μέθοδος διαμόρφωσης πρέπει να αντιστοιχεί μεταξύ ενός πομπού και ενός δέκτη για ένα δεδομένο πακέτο. [48] Στον Πίνακα 6 δίνεται το SNR στον δέκτη LoRa ανάλογα με την επιλογή SF. [45]

Πίνακας 6. Εύρος συντελεστή διασποράς SF

| Συντελεστής διασποράς SF | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|
| Chirps ανά σύμβολο | 64 | 128 | 256 | 512 | 1024 | 2048 | 4096 |
| SNR δέκτη LoRa | -5 dB | -7.5 dB | -10 dB | -12.5 dB | -15 dB | -17.5 dB | -20 dB |

### 3.1.5 Ισχύς μετάδοσης (TP)

Η ισχύς μετάδοσης (TP) προσδιορίζει την ενέργεια που ακτινοβολείται από την κεραία του κόμβου LoRa. Μπορεί να κυμαίνεται από − 4 dBm έως + 20 dBm, αλλά διαφορετικές περιοχές θα μπορούσαν να έχουν διαφορετικά όρια ισχύος (για παράδειγμα, στην Ευρώπη, το ανώτερο όριο είναι + 14 dBm).

### 3.1.6 Sync Word

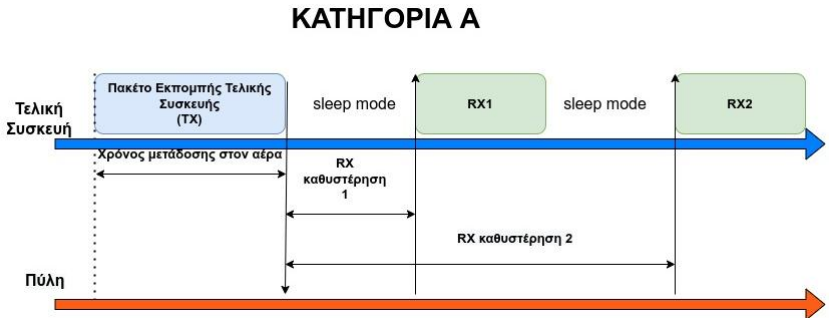
Το Sync Word είναι μια παράμετρος τιμής 1 byte που ορίζεται από τα δύο τελευταία up-chip του Προοιμίου (preable) του LoRa και χρησιμοποιείται για τη ταξινόμηση των δικτύων LoRa χρησιμοποιώντας τις ίδιες ζώνες συχνοτήτων. Όποια συσκευή έχει διαμορφωθεί με ένα δεδομένο Sync Word θα απορρίψει οποιαδήποτε εισερχόμενη μετάδοση, εάν το Sync Word δεν ταιριάζει με το δικό του. Πιο συγκεκριμένα, οι προεπιλεγμένες τιμές που λαμβάνονται από την τιμή byte του Sync Word για ιδιωτικά δίκτυα LoRa είναι 0x12 για συσκευές Semtech SX127x και 0x1424 για συσκευές SX126x. Αντίθετα, τα δημόσια δίκτυα LoRa (όπως το The Things Network, TTN) θα εκφράζονται με τιμές ίσες με 0x34 για συσκευές Semtech SX127x και με 0x3444 για συσκευές SX126x [46]

## 3.2 Επίπεδο MAC - LoRaWAN

Η LoRa Alliance έχει δημιουργήσει το πρωτόκολλο LoRaWAN που ανήκει στο επίπεδο MAC και χρησιμοποιεί το φυσικό επίπεδο LoRa Το LoRaWAN ορίζει το πρωτόκολλο επικοινωνίας και την αρχιτεκτονική του συστήματος για το δίκτυο. Η πρώτη έκδοση 1.0 του LoRaWAN κυκλοφόρησε το 2015, η έκδοση 1.0.1 το 2016, η έκδοση 1.1 το 2017, η έκδοση 1.0.3 το 2018 και η έκδοση 1.0.4 το 2020. [49] Οι τελικές συσκευές (κόμβοι) μπορούν να διαμορφωθούν σε τρεις διαφορετικές κατηγορίες, όπου η κάθε μια έχει διαφορετικά χαρακτηριστικά ανάλογα με τις απαιτήσεις της εφαρμογής IoT. Αυτές οι κατηγορίες διαφοροποιούν τον χρόνο καθυστέρησης (latency) επικοινωνίας κατερχόμενης ζεύξης σε σχέση με τη διάρκεια ζωής της μπαταρίας. Αυτές οι κατηγορίες ονομάζονται κατηγορία A, κατηγορία B, κατηγορία C και αναλύονται παρακάτω. [46], [49]

### 3.2.1 Κατηγορία A

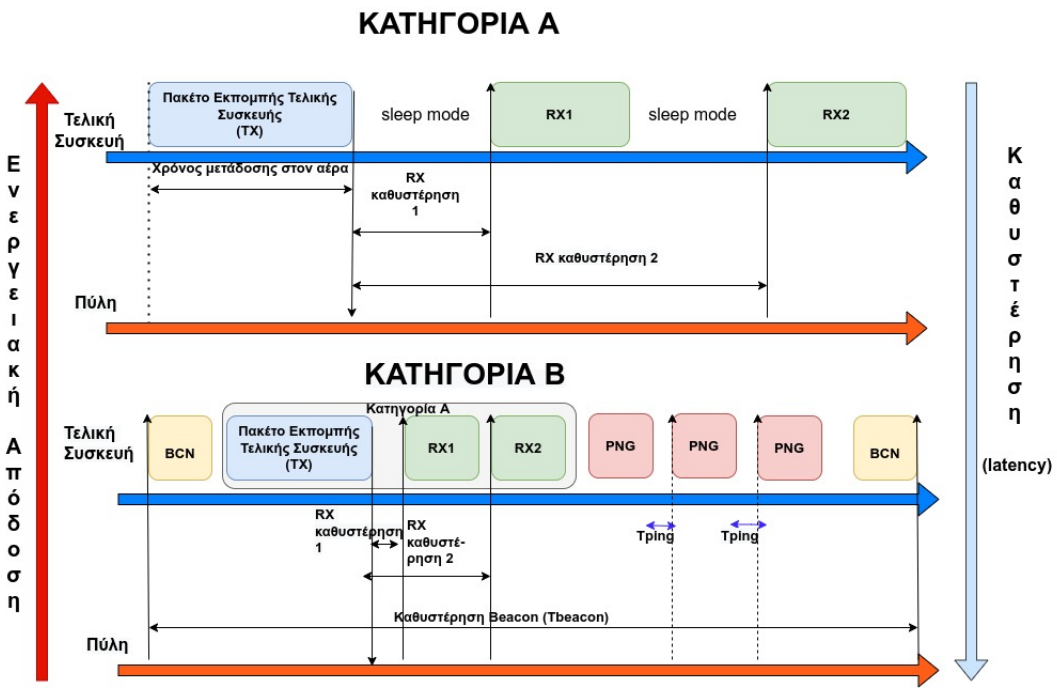
Σε αυτήν την κατηγορία ανήκουν οι τελικές συσκευές (αισθητήρες) που έχουν για πηγή τροφοδοσίας μπαταρία. Οι τελικές συσκευές εκπέμπουν τα μηνύματα μέσω της ανερχόμενης ζεύξης (uplink) προς την πύλη (πύλες) ενώ τα ληφθέντα μηνύματα μέσω της κατερχόμενης ζεύξης από την πύλη (πύλες) προς την τελική συσκευή (συσκευές) πραγματοποιούνται μόνο κατά την διάρκεια δυο συγκεκριμένων slot λήψεων σε συγκεκριμένες χρονικές περιόδους και πάντα μετά από την μετάδοση ανερχόμενης ζεύξης. Πιο συγκεκριμένα όπως φαίνεται στην εικόνα 22, κατά την πρώτη χρονική περίοδο καθυστέρησης RX1 μετά από το τέλος της περιόδου μετάδοσης TX, μια συσκευή κατηγορίας A ανοίγει ένα σύντομο slot λήψης RX1 (1sec) ακολουθώντας την ίδια ζώνη συχνοτήτων που χρησιμοποιήθηκε για την μετάδοση (ανοδική ζεύξη) του προηγούμενου μηνύματος. Στην περίπτωση που δεν ληφθεί μήνυμα κατερχόμενης ζεύξης από την πύλη, η τελική συσκευή ανοίγει ένα δεύτερο slot λήψης RX2 (2sec). Η πύλη μπορεί να ανταποκριθεί εντός της πρώτης υποδοχής λήψης ή της δεύτερης υποδοχής λήψης, αλλά όχι και στις δύο. Οι συσκευές κατηγορίας A έχουν την μέγιστη διάρκεια ζωής μπαταρίας. [18], [46], [49]




Εικόνα 22. Κατηγορία A

### 3.2.2 Κατηγορία B

Οι συσκευές κατηγορίας B είναι κατάλληλες για εφαρμογές που χρειάζονται περισσότερο την κατερχόμενη ζεύξη καθώς ανοίγονται επιπλέον slot λήψης. Όπως φαίνεται στην μέση της εικόνας 23, οι τελικές συσκευές στην αρχή εντάσσονται στο δίκτυο ως κατηγορία A και στην συνέχεια αποφασίζουν να αλλάξουν σε κατηγορία B. Ένα σήμα beacon (BCN) εκπέμπετε περιοδικά μέσω της πύλης και οι τελικές συσκευές κατηγορίας B πρέπει περιοδικά να λαμβάνουν ένα από τα σήματα BCN ώστε να συγχρονιστούν στο δίκτυο. Όταν η τελική συσκευή λαμβάνει το σήμα BCN μπορεί να ανοίξει ένα σύντομο slot λήψης που ονομάζεται slot ping (PNG). Οποιαδήποτε BCN μπορούν να χρησιμοποιηθούν από το LoRa Server για την έναρξη επικοινωνίας κατερχόμενης ζεύξης. [18], [36], [46]




Εικόνα 23. Κατηγορία Β

### 3.2.3 Κατηγορία C

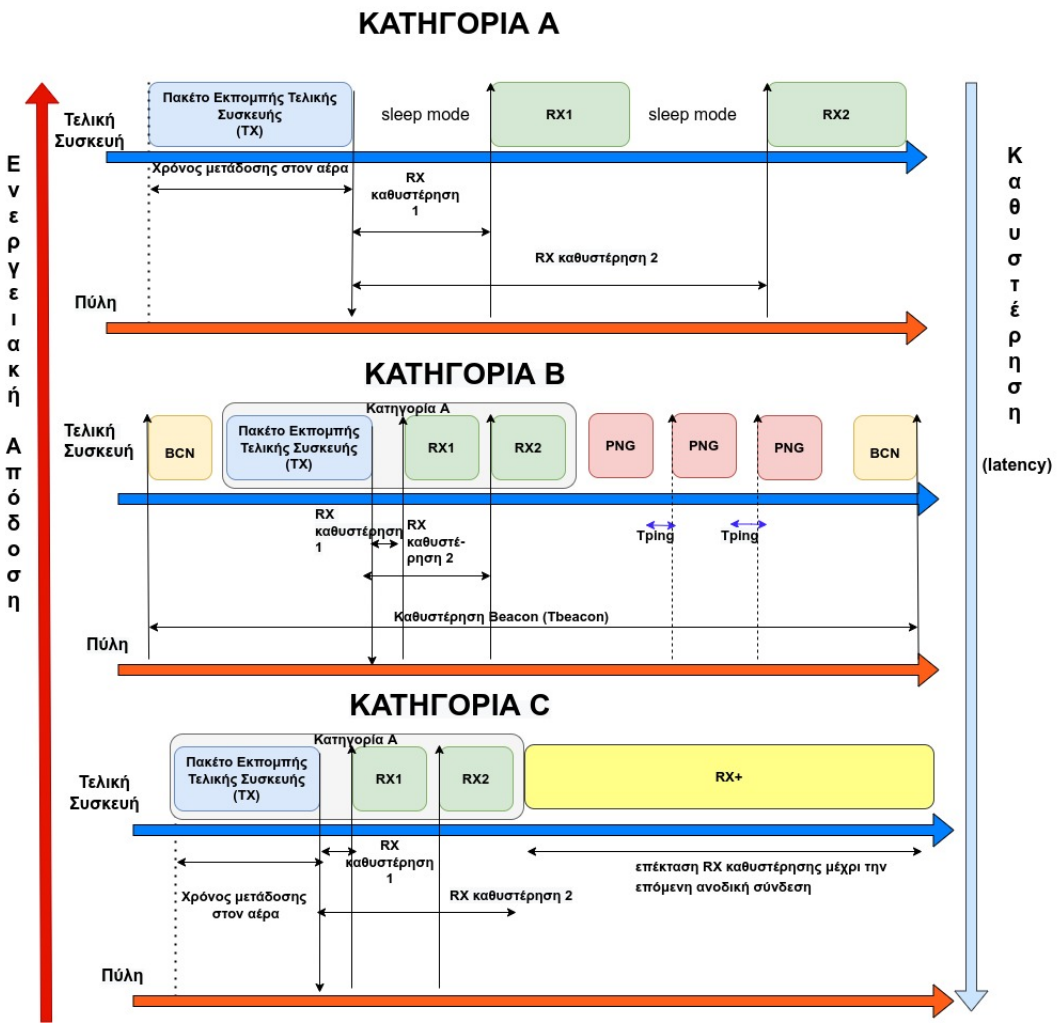
Οι τελικές συσκευές κατηγορίας C έχουν την χαμηλότερη καθυστέρηση (latency) και την μεγαλύτερη κατανάλωση ενέργειας. Δεν είναι επαρκής η τροφοδοσία μέσω μπαταρίας και απαιτούν κύρια τροφοδοσία. Λειτουργούν ως κατηγορία Α διατηρώντας τα ίδια slot λήψης RX1,RX2 αλλά στην συνέχεια παραμένει ανοιχτό ένα slot λήψης μέχρι να την επόμενη ανοδική σύνδεση, δηλαδή μέχρι να σταλεί επόμενο πακέτο από την τελική συσκευή όπως φαίνεται στην εικόνα 24. Οι τελικές συσκευές κατηγορίας C δεν μπορούν να μεταβούν σε κατηγορία B. [46], [49]

### 3.2.4 Κύκλος λειτουργίας - Duty cycle

Η προδιαγραφή LoRaWAN αναφέρει ότι τα δίκτυα LoRaWAN θα πρέπει να χρησιμοποιούν ζώνες συχνοτήτων ISM. Αυτές οι ζώνες υπόκεινται σε κανονισμούς σχετικά με τη μέγιστη ισχύ μετάδοσης και τον κύκλο λειτουργίας. [37] Όταν γίνεται αποστολή ενός σήματος, χρειάζεται ένα ορισμένο χρονικό διάστημα πριν ο δέκτης λάβει αυτό το σήμα. Αυτή η ώρα ονομάζεται Time on Air (ToA). Ο κύκλος λειτουργίας (duty cycle) είναι η αναλογία του χρόνου κατά τη διάρκεια του οποίου λειτουργεί ένα εξάρτημα, συσκευή ή σύστημα. Ο κύκλος λειτουργίας μπορεί να εκφραστεί ως αναλογία ή ως ποσοστό. Για την Ευρώπη υπάρχει κύκλος λειτουργίας 0,1% και 1,0% ανά ημέρα ανάλογα με το κανάλι. Αν ο χρόνος ToA είναι 530ms και πρέπει να εφαρμοστεί duty cycle 1% πρέπει να περιμένουμε 99x530ms = 52,47s πριν στείλουμε ένα νέο μήνυμα. [50] Στον Πίνακα 7 δίνεται ο υπολογισμός Airtime για payload 100 bytes.

Πίνακας 7. Υπολογισμός Airtime LoRaWAN για payload 100 bytes [51]

| Data Rate | DR6 | DR5 | DR4 | DR3 | DR2 | DR1 | DR0 |
|---|---|---|---|---|---|---|---|
| Airtime | 94.8 ms | 189.7 ms | 338.4 ms | 615.4 ms | 1,108.0 ms | 2,461.7 ms | 4,431.9 ms |
| 1% max duty cycle | 9.5 sec<br>379 msg/hour | 19.0 sec<br>189 msg /hour | 33.8 sec<br>106 msg /hour | 61.5 sec<br>58 msg /hour | 110.8 sec<br>32 msg /hour | 246.2 sec<br>14 msg/hour | 443.2 sec<br>8 msg /hour |
| fair access policy | 273.2 sec (avg)<br>13.2 avg/hour<br>316 msg/24h | 546.3 sec (avg)<br>6.6 avg /hour<br>158 msg /24h | 974.7 sec (avg)<br>3.7 avg /hour<br>88 msg /24h | 1,772.4 sec (avg)<br>2.0 avg/hour<br>48 msg/24h | 3,190.9 sec (avg)<br>1.1 avg /hour<br>27 msg /24h | 7,089.7 sec (avg)<br>0.5 avg/hour<br>12 msg/24h | 12,763.8 sec (avg)<br>0.3 avg/hour<br>6 msg/24h |




Εικόνα 24. Κατηγορία C

## 3.3 Ασφάλεια LoRaWAN

Η ασφάλεια του LoRaWAN έχει υλοποιηθεί ώστε να τηρεί τα στάνταρ των δικτύων LPWAN, δηλαδή χαμηλή κατανάλωση ισχύος, χαμηλή πολυπλοκότητα υλοποίησης, χαμηλό κόστος και υψηλή επεκτασιμότητα. Η υλοποίηση ασφάλειας περιλαμβάνει κρυπτογραφικούς αλγορίθμους καθώς και ασφάλεια από άκρο σε άκρο. Θεσπίζεται αμοιβαίος έλεγχος ταυτότητας μεταξύ μιας τελικής συσκευής LoRaWAN και του δικτύου LoRaWAN ως μέρος της διαδικασίας σύνδεσης δικτύου. Η συγκεκριμένη διαδικασία εξασφαλίζει ότι μόνο γνήσιες και εξουσιοδοτημένες συσκευές θα συνδέονται σε γνήσια και αυθεντικά δίκτυα. Το LoRaWAN MAC και τα μηνύματα εφαρμογής διαθέτουν έλεγχο ταυτότητας προέλευσης, προστασία ακεραιότητας, προστασία επανάληψης και κρυπτογράφηση. Με αυτήν την διαδικασία εξασφαλίζετε ότι η κυκλοφορία του δικτύου δεν έχει τροποποιηθεί και προέρχεται από μια νόμιμη συσκευή. Η ασφάλεια LoRaWAN εφαρμόζει επιπλέον κρυπτογράφηση από άκρο σε άκρο για ωφέλιμα φορτία εφαρμογών που ανταλλάσσονται μεταξύ των τελικών συσκευών και των διακομιστών εφαρμογών. [52]

### 3.3.1 Κλειδιά LoRaWAN

Το LoRaWAN χρησιμοποιεί κρυπτογραφικούς αλγορίθμους AES σε συνδυασμό με διάφορους τρόπους λειτουργίας :

- CMAC για προστασία ακεραιότητας και
- CTR για κρυπτογράφηση.

Για να συμμετέχει μια τελική συσκευή σε ένα δίκτυο LoRaWAN θα πρέπει να προσαρμοστεί και να ενεργοποιηθεί. Η ενεργοποίηση μιας τελικής συσκευής μπορεί να πραγματοποιηθεί με δυο τρόπους :

- ABP (Activation By Personalization) και
- OTAA (Over-The-Air Activation).

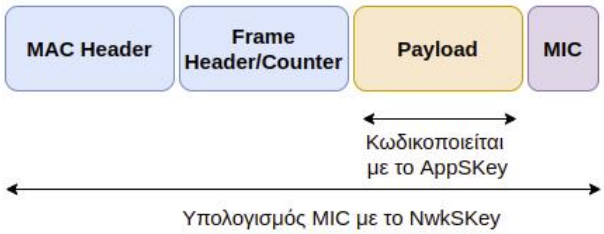
Για την ταυτοποίηση των συσκευών LoRaWAN κάθε συσκευή έχει προσαρμοσμένο ένα μοναδικό κλειδί AES μήκους 128 bit που ονομάζεται AppKey (χρησιμοποιείται στην μέθοδο OTAA) και ένα μοναδικό αναγνωριστικό 64-bit που παραχωρείτε σε μια τελική συσκευή από τον κατασκευαστή και ονομάζεται DevEUI (χρησιμοποιείται στην μέθοδο OTAA και ABP). [52], [53]

Το AppSKey δημιουργείται από το AppKey και χρησιμοποιείται για την κρυπτογράφηση και αποκρυπτογράφηση του ωφέλιμου φορτίου. Το AppSKey είναι συγκεκριμένο για κάθε τελική συσκευή. Οι διακομιστές δικτύου αντιμετωπίζονται ως αξιόπιστα μέρη για να μεταδίδουν με ασφάλεια τα κρυπτογραφημένα μηνύματα σε κάθε άκρο. [54], [55]

Το NwkSKey δημιουργείται επίσης από το AppKey και χρησιμοποιείται για την επιβεβαίωση της ακεραιότητας κάθε μηνύματος από τον Κώδικα Ακεραιότητας Μηνύματος (έλεγχος MIC). Το MIC είναι παρόμοιο με ένα άθροισμα ελέγχου, με τη διαφορά ότι αποτρέπει την σκόπιμη παραβίαση ενός μηνύματος. Για αυτό, το LoRaWAN χρησιμοποιεί AES-CMAC. [55]

Τα συγκεκριμένα κλειδιά συνεδρίας AppSKey και NwKSey είναι μοναδικά ανά συσκευή και ανά συνεδρία. Εάν η ενεργοποίηση της τελικής συσκευής γίνει με την μέθοδο OTAA τα κλειδιά δημιουργούνται ξανά σε κάθε ενεργοποίηση ενώ με την μέθοδο ABP τα κλειδιά παραμένουν ίδια μέχρι να τα αλλάξουμε. [55]

Στην εικόνα 25 φαίνεται η δομή ενός πακέτου LoRaWAN η προστασία του.




Εικόνα 25. Δομή ενός πακέτου LoRaWAN και η προστασία του

### 3.3.2 Ενεργοποίηση OTAA

Στην μέθοδο ενεργοποίησης OTAA πραγματοποιείτε μια διαδικασία σύνδεσης με αίτημα σύνδεσης (Join-request) και ανταλλαγή μηνυμάτων σύνδεσης αποδοχής (join-accept) για κάθε νέα σύνδεση. Σύμφωνα με το μήνυμα join-accept οι τελικές συσκευές λαμβάνουν τα νέα κλειδιά συνεδρίας (NwkSkey και AppSKey). Στην μέθοδο ABP τα δυο κλειδιά συνεδρίας αποθηκεύονται κατευθείαν στις τελικές συσκευές. [37] Παρακάτω θα αναλύσουμε την διαδικασία ενεργοποίησης OTAA για την έκδοση 1.0.x του LoRaWAN καθώς και για την έκδοση 1.1.

**OTAA LoRaWAN 1.0.x**

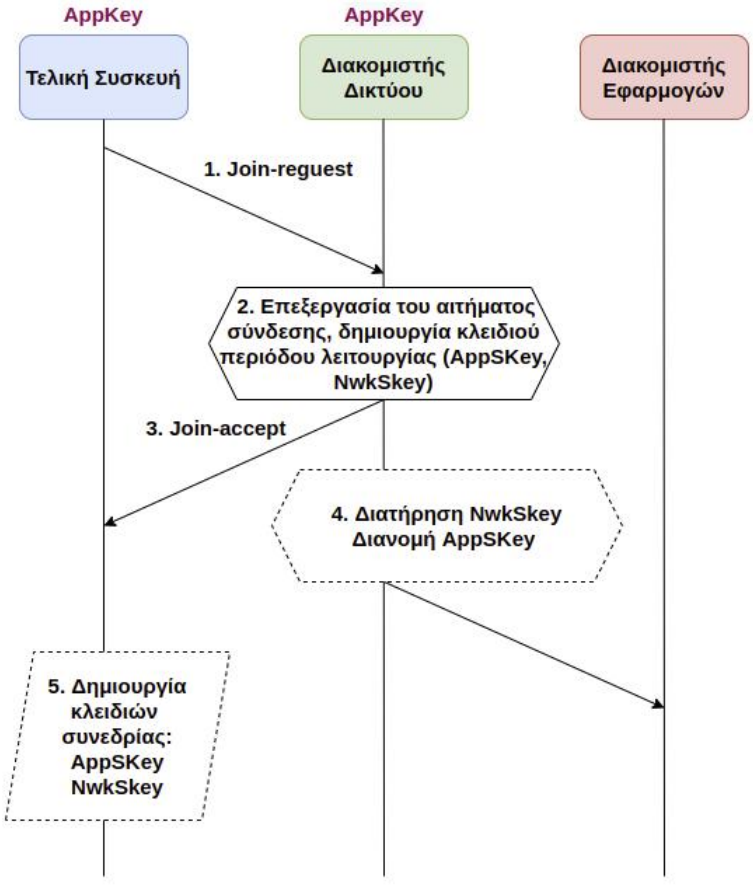
Για την έκδοση LoRaWAN 1.0.x η διαδικασία σύνδεσης απαιτεί την ανταλλαγή δυο μηνυμάτων MAC ανάμεσα στην τελική συσκευή και τον διακομιστή δικτύου :

- Join-request : από την τελική συσκευή στον διακομιστή δικτύου
- Join-accept : από το διακομιστή δικτύου έως την τελική συσκευή

Πριν από την ενεργοποίηση πρέπει να αποθηκεύονται στην τελική συσκευή τα εξής:

- AppEUI – ένα παγκοσμίως μοναδικό αναγνωριστικό εφαρμογής 64-bit στον χώρο διευθύνσεων IEEE EUI64 που προσδιορίζει μοναδικά την οντότητα που μπορεί να επεξεργαστεί το πλαίσιο Join-req.
- DevEUI – ένα παγκοσμίως μοναδικό αναγνωριστικό συσκευής 64-bit στον χώρο διευθύνσεων IEEE EUI64 που προσδιορίζει μοναδικά την τελική συσκευή.
- Το AppKey είναι ένα μυστικό κλειδί AES-128 bit γνωστό ως root κλειδί. Το ίδιο AppKey θα πρέπει να παρέχεται στο δίκτυο όπου πρόκειται να εγγραφεί η τελική συσκευή.

Στην εικόνα 26 φαίνονται τα βήματα της διαδικασίας ενεργοποίησης OTAA για το LoRaWAN 1.0.x




Εικόνα 26. Ροή μηνυμάτων σύνδεσης OTAA στο Lorawan 1.0

Βήμα 1 : Η διαδικασία σύνδεσης ξεκινά πάντα από την τελική συσκευή. Η τελική συσκευή στέλνει το μήνυμα αίτησης σύνδεσης (Join-request) στο δίκτυο που πρόκειται να συνδεθεί. Το μήνυμα Join-request αποτελείται από τα ακόλουθα πεδία:

- AppEUI
- DevEUI
- DevNonce : Υλοποιείτε από την τελική συσκευή και είναι μια μοναδική τυχαία τιμή μήκους 2 byte. Ο διακομιστής δικτύου χρησιμοποιεί το DevNonce κάθε τελικής συσκευής για να παρακολουθεί τα αιτήματά τους για σύνδεση. Εάν μια τελική συσκευή στείλει ένα αίτημα σύνδεσης με ένα DevNonce που χρησιμοποιήθηκε στο παρελθόν (αυτή η κατάσταση είναι γνωστή ως επίθεση επανάληψης), ο διακομιστής δικτύου απορρίπτει το αίτημα σύνδεσης και δεν επιτρέπει σε αυτήν την τελική συσκευή να εγγραφεί στο δίκτυο. Το DevNonce είναι ένας μετρητής που ξεκινά από το 0 όταν η συσκευή τροφοδοτείται αρχικά και αυξάνεται με κάθε αίτημα σύνδεσης από τη συσκευή τερματισμού.

Ο Κωδικός Ακεραιότητας Μηνύματος (MIC) υπολογίζεται σε όλα τα πεδία του μηνύματος Join-request χρησιμοποιώντας το AppKey. Το υπολογιζόμενο MIC προστίθεται στη συνέχεια στο μήνυμα Join-request.

Βήμα 2: Στην συνέχεια ο διακομιστής δικτύου επεξεργάζεται το μήνυμα Join-request και δημιουργεί τα δυο κλειδιά περιόδου λειτουργίας (NwkSKey και AppSKey) και το μήνυμα Join-accept εάν η τελική συσκευή επιτρέπεται να συνδεθεί σε ένα δίκτυο.

Το μήνυμα Join-accept περιλαμβάνει τα ακόλουθα πεδία:

- AppNonce : Παραχωρείτε από τον διακομιστή δικτύου, είναι ένα μοναδικό αναγνωριστικό μήκους 3 bytes που χρησιμοποιείται από την τελική συσκευή για την εξαγωγή των κλειδιών περιόδου λειτουργίας (AppSKey και NwkSKey)
- NetID : Αποτελείται από το αναγνωριστικό δικτύου (NwkID), τα πιο σημαντικά 7 bit.
- DevAddr : Μια διεύθυνση συσκευής 32-bit που εκχωρείται από τον διακομιστή δικτύου για την αναγνώριση της τελικής συσκευής εντός του τρέχοντος δικτύου.
- DLSettings : Ένα πεδίο 1 byte που αποτελείται από ρυθμίσεις κατερχόμενης ζεύξης που πρέπει να χρησιμοποιήσει η τελική συσκευή.
- RxDelay : Περιλαμβάνει την καθυστέρηση ανάμεσα σε εκπομπή (TX) και λήψη (RX).
- CFList : Μια προαιρετική λίστα συχνοτήτων καναλιού για το δίκτυο στο οποίο συνδέεται η τελική συσκευή. Αυτές οι συχνότητες είναι συγκεκριμένες ανάλογα την περιοχή.

Ο Κωδικός Ακεραιότητας Μηνύματος (MIC) υπολογίζεται σε όλα τα πεδία στο μήνυμα Join-accept χρησιμοποιώντας το AppKey. Το υπολογιζόμενο MIC προστίθεται στη συνέχεια στο μήνυμα Join-accept. Στη συνέχεια, το ίδιο το μήνυμα Join-accept κρυπτογραφείται με το AppKey. Ο διακομιστής δικτύου χρησιμοποιεί μια λειτουργία αποκρυπτογράφησης AES σε λειτουργία ECB για να κρυπτογραφήσει το μήνυμα Join-accept.

Βήμα 3 : Ο διακομιστής δικτύου στέλνει το κρυπτογραφημένο μήνυμα Join-accept πίσω στην τελική συσκευή ως κανονική κατερχόμενη σύνδεση.

Βήμα 4 : Ο διακομιστής δικτύου διατηρεί το NwkSKey και διανέμει το AppSKey στον διακομιστή εφαρμογών.

Βήμα 5 : Η τελική συσκευή αποκρυπτογραφεί το μήνυμα Join-accept χρησιμοποιώντας την λειτουργία κρυπτογράφησης AES. Η τελική συσκευή χρησιμοποιεί το AppKey και το AppNonce για να παράξει τα δύο κλειδιά περιόδου λειτουργίας, το κλειδί συνεδρίας δικτύου (NwkSKey) και το κλειδί συνεδρίας εφαρμογής (AppSKey).

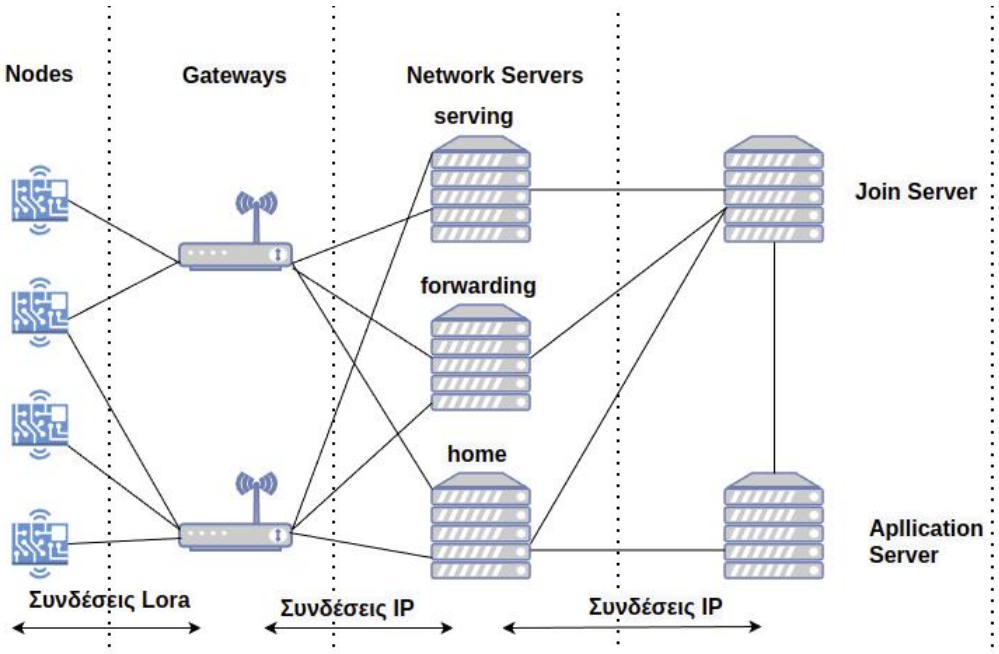
Η τελική συσκευή είναι πλέον ενεργοποιημένη στο Δίκτυο. Μετά την ενεργοποίηση, οι ακόλουθες πληροφορίες αποθηκεύονται στην τελική συσκευή:

1. DevAddr
2. NwkSKey
3. AppSKey

[55], [56]

**OTAA LoRaWAN 1.1**

Στην έκδοση LoRaWAN 1.1 χρησιμοποιείτε ένας νέος διακομιστής που ονομάζεται ”Join Server” (εικόνα 27) και σκοπός του είναι να παρέχει περισσότερη ασφάλεια. Στην συγκεκριμένη έκδοση συναντάμε τρείς διακομιστές δικτύου έναντι ενός σε σχέση με την προκατόχου της. Η ονομασία των διακομιστών δικτύου είναι, home, forwarding, serving με σκοπό να επιτραπεί το roaming σε όλη την πόλη, την Χώρα και γενικά σε όλο τον κόσμο.




Εικόνα 27. Αρχιτεκτονική δικτύου LoRaWAN v1.1.

Όπως και στην προηγούμενη έκδοση πριν την ενεργοποίηση πρέπει να αποθηκεύονται στην τελική συσκευή τα DevEUI, AppKey, καθώς και άλλα στοιχεία :

- Το JoinEUI το μοναδικό αναγνωριστικό του Join Server και είναι ενσωματωμένο στις τελικές συσκευές κατά την κατασκευή. Αυτό το αναγνωριστικό χρησιμοποιείται κατά τη διαδικασία Join. Το JoinEUI απαιτείται μόνο στην μέθοδο OTAA.
- Το NwkKey

Δεν χρησιμοποιείτε το AppEUI όπως στην προηγούμενη έκδοση και έχει αντικατασταθεί με το JoinEUI. Τα AppKey, NwkKey καθώς και τα DevEUI, JoinEUI θα πρέπει να γνωστοποιούνται στον Join Server που θα βοηθήσει στην επεξεργασία της διαδικασίας σύνδεσης και της παραγωγής των κλειδιών περιόδου λειτουργίας. Το AppKey και το NwkKey είναι τα βασικά κλειδιά (root keys) AES-128 bit από τα οποία παράγονται τα κλειδιά περιόδου λειτουργίας (session keys).

Τα κλειδία περιόδου συνεδρίας που δημιουργούνται είναι τα εξής :

- Κλειδί ακεραιότητας περιόδου λειτουργίας προώθησης δικτύου - Forwarding Network Session Integrity Key(FNwkSIntKey) : Χρησιμοποιείται για τον υπολογισμό του MIC όλων των μηνυμάτων δεδομένων ανερχόμενης ζεύξης για τη διασφάλιση της ακεραιότητας των δεδομένων. Αυτό το κλειδί θεωρείται ”δημόσιο” και μπορεί να είναι κοινόχρηστο με έναν διακομιστή forwarding.

- Κλειδί ακεραιότητας περιόδου λειτουργίας δικτύου εξυπηρέτησης - Serving Network Session Integrity Key (SNwkSIntKey) : είναι ένα κλειδί συνεδρίας δικτύου που είναι ξεχωριστά για κάθε τελική συσκευή που χρησιμοποιείται για τον υπολογισμό του MIC όλων των μηνυμάτων δεδομένων κατερχόμενης ζεύξης για να διασφαλιστεί η ακεραιότητα των δεδομένων. Αυτό το κλειδί θεωρείται ”ιδιωτικό” και δεν πρέπει να μοιράζεται με διακομιστές forwarding.

- Κλειδί κρυπτογράφησης συνεδρίας δικτύου - Network Session Encryption Key (NwkSEncKey) : είναι ένα κλειδί συνεδρίας δικτύου που είναι συγκεκριμένο για κάθε τελική συσκευή και χρησιμοποιείται για την κρυπτογράφηση/αποκρυπτογράφηση των εντολών του στρώματος MAC.

- Όπως και στην προηγούμενη έκδοση το AppSKey δημιουργείτε από το βασικό κλειδί AppKey και χρησιμοποιείτε για την κρυπτογράφηση και αποκρυπτογράφηση του ωφέλιμου φορτίου.

Επίσης χρησιμοποιούνται δυο κλειδιά συνεδρίας διάρκειας ζωής για την επανασύνδεση στο δίκτυο μετά την αποσύνδεση. Κατά τη δημιουργία αυτών των κλειδιών, εκτός από το βασικό κλειδί του NwkKey, χρησιμοποιείται επίσης το DevEUI. Αυτά τα κλειδιά σύνδεσης χρησιμοποιούνται επίσης από την μέθοδο OTAA και όχι την ABP. Αυτά τα κλειδία είναι τα εξής:

- JSIntKey : Χρησιμοποιείται για το MIC των μηνυμάτων τύπου 1 Rejoin-Request και των σχετικών απαντήσεων Join-Accept.

- JSEncKey : Χρησιμοποιείται για την κρυπτογράφηση των μηνυμάτων Join-Accept ως απόκριση σε μηνύματα Rejoin-Request.

Στο σχήμα 28 φαίνονται τα βήματα της διαδικασίας ενεργοποίησης OTAA για το LoRaWAN 1.1

Βήμα 1: Η τελική συσκευή στέλνει το μήνυμα αίτησης σύνδεσης (Join-request) στο δίκτυο που πρόκειται να συνδεθεί. Το μήνυμα αίτησης Join-request αποτελείται από το JoinEUI, DevEUI, DevNonce. Ο Κωδικός Ακεραιότητας Μηνύματος (MIC) υπολογίζεται σε όλα τα πεδία στο μήνυμα Join-accept χρησιμοποιώντας το NwkKey (στην προηγούμενη έκδοση το AppKey). Το υπολογιζόμενο MIC προστίθεται στη συνέχεια στο μήνυμα Join-accept.

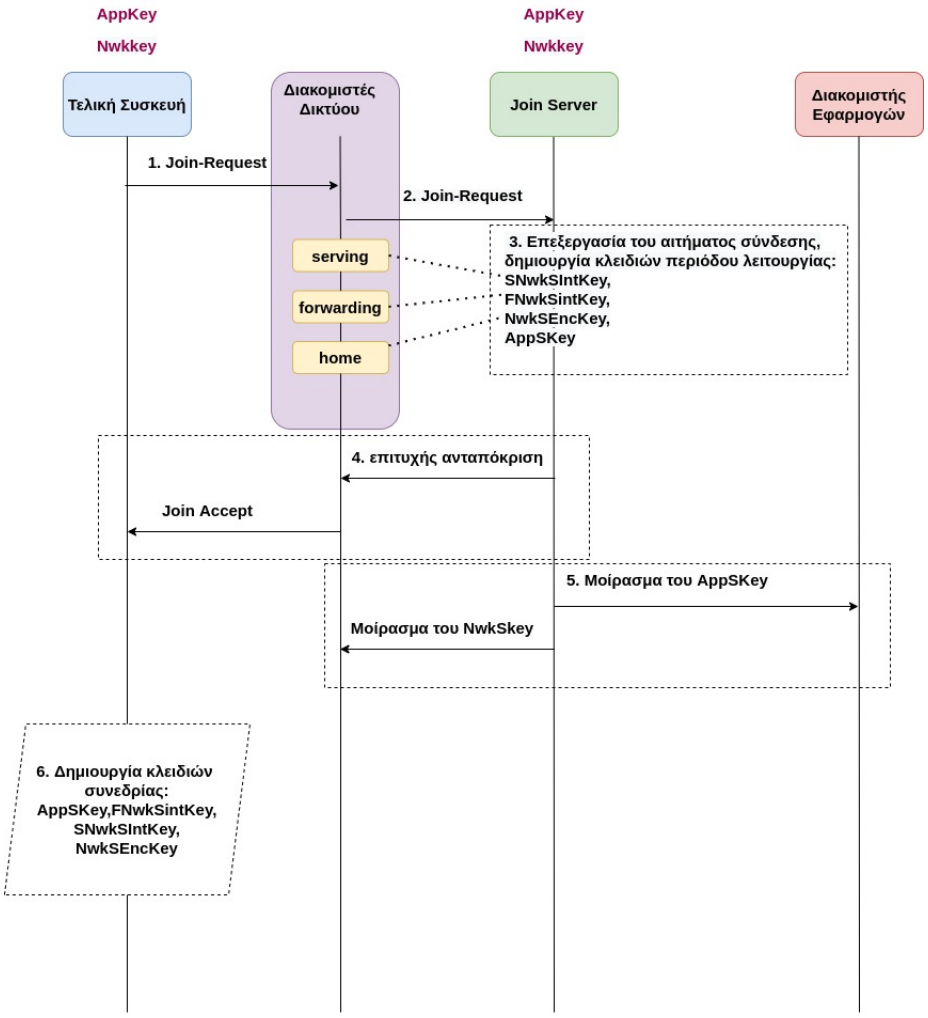
Βήμα 2: Ο διακομιστής δικτύου προωθεί το μήνυμα Join-request στον αντίστοιχο Join Server.

Βήμα 3: Ο Join server επεξεργάζεται το μήνυμα Join-reques. Ο Join Server θα δημιουργήσει όλα τα κλειδιά περιόδου λειτουργίας (AppSKey, FNwkSIntKey, SNwkSIntKey και NwkSEncKey).

Βήμα 4: Εάν η παραπάνω διαδικασία είναι επιτυχής τότε ο διακομιστής δικτύου δημιουργεί το μήνυμα Join-accept. Το μήνυμα Join-accept αποτελείται από τα εξής πεδία:

- JoinNonce : μια τιμή μετρητή συγκεκριμένης συσκευής που παρέχεται από τον Join Server και χρησιμοποιείται από την τελική συσκευή για την εξαγωγή των κλειδιών περιόδου λειτουργίας, FNwkSIntKey, SNwkSIntKey, NwkSEncKey και AppSKey.

- NetID : ένα μοναδικό αναγνωριστικό δικτύου 24-bit




Εικόνα 28. Ροή μηνυμάτων σύνδεσης OTAA στο Lorawan 1.1

- Όπως και στην προηγούμενη έκδοση τα πεδία - DevAddr, DLSettings, RxDelay, CFList.

Ο κωδικός ακεραιότητας μηνύματος (MIC) υπολογίζεται σε όλα τα πεδία στο μήνυμα Join-accept χρησιμοποιώντας NwkKey ή το JSIntKey στην περίπτωση επανασύνδεσης. Το υπολογιζόμενο MIC προστίθεται στη συνέχεια στο μήνυμα Join-accept. Ο διακομιστής δικτύου χρησιμοποιεί μια λειτουργία αποκρυπτογράφησης AES σε λειτουργία ECB για να κρυπτογραφήσει το μήνυμα αποδοχής σύνδεσης. Το μήνυμα Join-accept κρυπτογραφείται με το NwkKey στην περίπτωση που ενεργοποιείται από το αίτημα σύνδεσης ή το JSEncKey εάν ενεργοποιείται από το αίτημα επανασύνδεσης. Στη συνέχεια, ο διακομιστής δικτύου στέλνει το κρυπτογραφημένο μήνυμα Join-accept πίσω στην τελική συσκευή ως κανονική κατερχόμενη ζεύξη.

Βήμα 5: Ο Join Server στέλνει το AppSKey στο Application Server και τα τρία κλειδιά περιόδου λειτουργίας δικτύου (FNwkSIntKey, SNwkSIntKey και NwkSEncKey) στον διακομιστή δικτύου.

Βήμα 6: Η τελική συσκευή αποκρυπτογραφεί το μήνυμα Join-accept χρησιμοποιώντας τη λειτουργία κρυπτογράφησης AES. Η τελική συσκευή χρησιμοποιεί τα AppKey, NwkKey και JoinNonce για τη δημιουργία των κλειδιών περιόδου λειτουργίας.

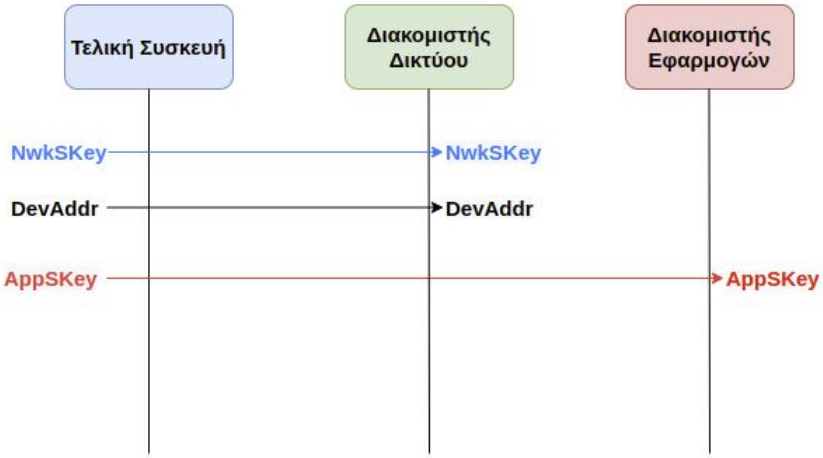
Η τελική συσκευή είναι πλέον ενεργοποιημένη στο Δίκτυο. Μετά την ενεργοποίηση, οι ακόλουθες πληροφορίες αποθηκεύονται στην τελική συσκευή: DevAddr,FNwkSIntKey, SNwkSIntKey, NwkSEncKey, AppSKey

### 3.3.3 Ενεργοποίηση ABP

Στην μέθοδο ενεργοποίησης ABP τα κλειδία περιόδου και η τιμή του Devaddr αποθηκεύονται κατευθείαν στην τελική συσκευή και παραμένουν ίδια σε όλη την διάρκεια της σύνδεσης της τελικής συσκευής. Παρακάτω θα αναλύσουμε την διαδικασία ενεργοποίησης ABP για την έκδοση 1.0.x του LoRaWAN καθώς και για την έκδοση 1.1.

#### ABP LoRaWAN 1.0.x

Το DevAddr και τα δύο κλειδιά περιόδου λειτουργίας NwkSKey και AppSKey αποθηκεύονται απευθείας στην τελική συσκευή αντί για το DevEUI, το AppEUI και το AppKey σε σχέση με την μέθοδο OTAA. Κάθε τελική συσκευή θα πρέπει να έχει ένα μοναδικό σύνολο NwkSKey και AppSkey. Το ίδιο DevAddr και NwkSKey θα πρέπει να αποθηκευτούν στον διακομιστή δικτύου και το AppSKey θα πρέπει να αποθηκευτεί στον διακομιστή εφαρμογών όπως φαίνεται στην εικόνα 29.




Εικόνα 29. ABP LoRaWAN 1.0.x

#### ABP LoRaWAN 1.1

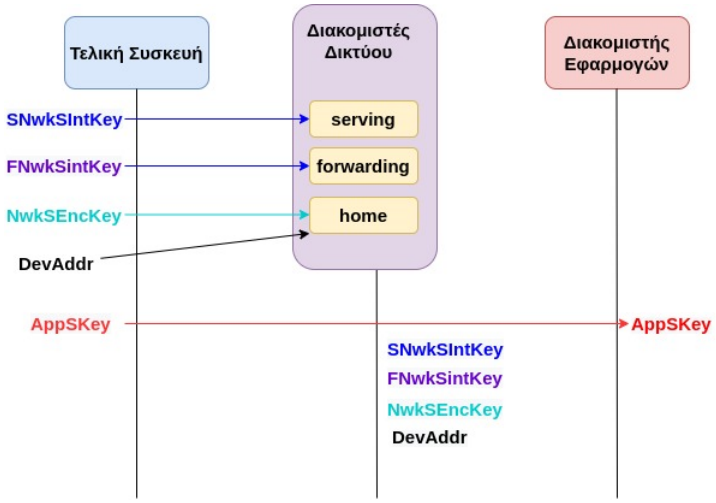
Το DevAddr και τα τέσσερα κλειδιά περιόδων λειτουργίας FNwkSIntKey, SNwkSIntKey, NwkSEncKey και AppSKey αποθηκεύονται απευθείας στην τελική συσκευή αντί για τα DevEUI, JoinEUI, AppKey και NwkKey στην μέθοδο OTAA 1.1. Το ίδιο DevAddr, FNwkSIntKey, SNwkSIntKey και NwkSEncKey θα πρέπει να αποθηκευτούν στους διακομιστές δικτύου και πιο συγκεκριμένα :

- Στον home το NwkSEncKey και DevAddr,
- Στον forwarding το FNwkSIntKey,
- Στον serving SNwkSIntKey

και το ίδιο AppSKey θα πρέπει να αποθηκευτεί στον διακομιστή εφαρμογής όπως φαίνεται στην εικόνα 30.

### 3.3.4 Σύγκριση ενεργοποίησης ABP με OTAA

Η ενεργοποίηση OTAA παρέχει κάποια πλεονεκτήματα σε σχέση με την ενεργοποίηση ABP και είναι τα εξής:




Εικόνα 30. ABP LoRaWAN 1.1

- Τα κλειδιά περιόδου λειτουργίας δημιουργούνται μόνο όταν απαιτείται, επομένως δεν μπορούν να παραβιαστούν πριν από την ενεργοποίηση.
- Εάν η συσκευή αλλάξει σε ένα νέο δίκτυο, μπορεί να συνδεθεί ξανά για να δημιουργήσει τα νέα κλειδιά - αντί να χρειάζεται να επαναπρογραμματιστεί. Επομένως υποστηρίζεται η κινητικότητα.
- Οι ρυθμίσεις δικτύου όπως το RxDelay και το CFList μπορούν να καθοριστούν κατά τη στιγμή της σύνδεσης.

Από τα αναφερθέντα πλεονεκτήματα το OTAA παρέχει έναν πιο ευέλικτο και ασφαλή τρόπο δημιουργίας κλειδιών περιόδου λειτουργίας με τους διακομιστές και συνιστάται για εφαρμογές που απαιτούν υψηλότερη ασφάλεια.

## 3.4 Συσκευές LoRaWAN

Σε αυτήν την ενότητα θα γίνει αναφορά στις συσκευές LoRaWAN κατηγοριοποιώντας σε δυο κατηγορίες

- Gateways (Πύλες)
- Nodes (Κόμβοι - τελικές συσκευές)

Καθώς και η επιλογή για την υλοποίση του IoT συστήματος.

### 3.4.1 Πύλες - Gateways

Οι πύλες λαμβάνουν τα απεσταλμένα δεδομένα των κόμβων και στην συνέχεια διαβιβάζουν τα δεδομένα στον αντίστοιχο LoRa Network Server. Θα δούμε περιληπτικά τα χαρακτηριστικά σε διάφορές πύλες.

#### Dragino LG-02

Το LG02 (εικόνα 31) είναι μια πύλη LoRa διπλού καναλιού ανοιχτού κώδικα. Σας επιτρέπει να γεφυρώσετε το ασύρματο δίκτυο LoRa σε ένα δίκτυο IP μέσω WiFi, Ethernet ή 3G/4G κινητής τηλεφωνίας μέσω προαιρετικής μονάδας LTE. Το LG02 ενσωματώνει Linux OS για τον άμεσο

Πίνακας 8. Χαρακτηριστικά Dragino LG02 [58]

| Σύστημα Hardware | Διεπαφές | WiFi χαρακτηριστικά | Χαρακτηριστικά LoRa | |
|---|---|---|---|---|
| 400Mhz ar9331 processor | 10M/100M RJ45 Ports x 2 | IEEE 802.11 b/g/n | Εύρος συχνότητας:<br>Band 1 (HF): 862 ~1020 Mhz<br>Band 2 (LF): 410 ~528 Mhz | Χαμηλό RX ρεύμα 10.3 mA, 200 nA register retention. |
| 64MB RAM | WiFi : 802.11 b/g/n | Ζώνη συχνοτήτων: 2.4 ~2.462GHz | | Πλήρως ενσωματωμένοsynthesizer με ανάυση 61 Hz. |
| 16MB Flash | LoRa Wireless | Ισχύς εκπομπής:<br>11n tx power : mcs7/15: 11db mcs0 : 17db<br>11b tx power: 18db<br>11g 54M tx power: 12db<br>11g 6M tx power: 18db | 168 dB μέγιστο link budget | FSK, GFSK, MSK, GMSK,<br>LoRaTM και OOK διαμόρφωση |
| | Power Input: 12V DC | Ευαισθησία Wi-Fi:<br>11g 54M : -71dbm<br>11n 20M : -67dbm | +20 dBm - 100 mW constant RF output vs | Ενσωματωμένος συγχρονιστής bit για ανάκτηση ρολογιού |
| | USB 2.0 host connector x 1 | | +14 dBm high efficiency PA | Preamble ανίχνευση |
| | USB 2.0 host internal interface x 1 | | Προγραμματιζόμενο bit rate πάνω από 300 kbps | Δυναμικό εύρος RSSI 127 dB |
| | 2 x LoRa Interfaces | | Υψηλή ευαισθησία: κάτω από -148 dBm | Αυτόματο RF Sense και CAD με εξαιρετικά γρήγορο AFC |
| | | | et-proof front end: IIP3 = -12.5 dBm | Packet engine έως 256 byte με CRC |
| | | | Εξαιρετικό blocking immunity | Ενσωματωμένος αισθητήρας θερμοκρασίας και ένδειξη χαμηλής μπαταρίας. |

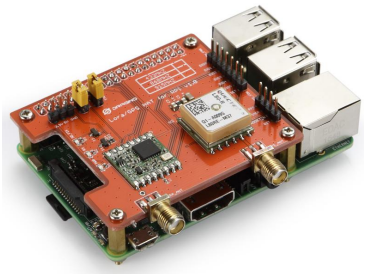
έλεγχο δύο μονάδων sx1276/sx1278 LoRa που επιτρέπει στο LoRa να λειτουργεί σε λειτουργία full duplex LoRa και να αυξάνει την αποτελεσματικότητα της επικοινωνίας. Το LG02 μπορεί να χρησιμοποιηθεί για την παροχή μιας χαμηλού κόστους ασύρματης λύσης IoT για την υποστήριξη 50 ∼ 300 κόμβων αισθητήρων. [57]


Εικόνα 31. Dragino LG02 GW

Στον Πίνακα 8 δίνονται τα βασικά χαρακτηριστικά της πύλης LG02.

### Dragino Lora/GPS Hat

Το Dragino Lora/GPS Hat (εικόνα 32) είναι μια επέκταση για LoRa και GPS που συνδέετε στο Raspberry pi 2/3/4. Μπορεί να χρησιμοποιηθεί σαν πύλη καθώς και σαν κόμβος. Βασίζεται στον πομποδέκτη SX1276/SX1278. Το πρόσθετο στο L80 GPS (βασίζεται στο MTK MT3339) έχει σχεδιαστεί για εφαρμογές που χρησιμοποιούν GPS συνδεδεμένο μέσω των σειριακών θυρών στο Raspberry Pi, όπως εφαρμογές χρονισμού ή γενικές εφαρμογές που απαιτούν GPS πληροφορίες. [59]



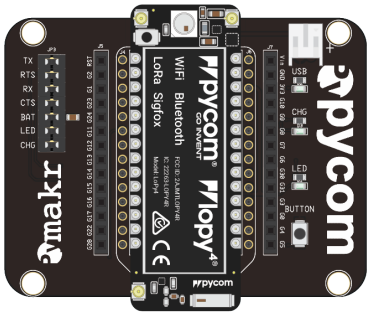
Εικόνα 32. Dragino Lora/GPS Hat

Στον Πίνακα 9 δίνονται τα βασικά χαρακτηριστικά του Dragino Lora/GPS Hat

Πίνακας 9. Χαρακτηριστικά Dragino Lora/GPS Hat [59]

| LoRa Χαρακτηριστικά | |
|---|---|
| 168 dB μέγιστο link budget | Πλήρως ενσωματωμένοsynthesizer με ανάυση 61 Hz. |
| +20 dBm - 100 mW constant RF output vs | FSK, GFSK, MSK, GMSK,LoRaTM και OOK διαμόρφωση |
| +14 dBm high efficiency PA | Ενσωματωμένος συγχρονιστής bit για ανάκτηση ρολογιού |
| Προγραμματιζόμενο bit rate πάνω από 300 kbps | Preamble ανίχνευση |
| Υψηλή ευαισθησια: κάτω από -148 dBm | Δυναμικό εύρος RSSI 127 dB |
| et-proof front end: IIP3 = -12.5 dBm | Αυτόματο RF Sense και CAD με εξαιρετικά γρήγορο AFC |
| Εξαιρετικό blocking immunity | Packet engine έως 256 byte με CRC |
| GPS Χαρακτηριστικά | |
| Βασισμένο στο MT3339 | Ακρίβεια Οριζόντιας Θέσης: Αυτόνομη <2,5 m CEP. |
| Λήψη ισχύος: 25 mA, Παρακολούθηση ισχύος: 20 mA. | Ακρίβεια χρονισμού: 1 PPS από 10 ns, Χρόνος επανάκτησης <1 δευτ |
| Συμβατό με GPS, SBAS. | Ακρίβεια ταχύτητας χωρίς βοήθημα <0,1 m/s, Ακρίβεια επιτάχυνσης χωρίς βοήθημα 0,1 m/s² |
| Προγραμματιζόμενο bit rate πάνω από 300 kbps | Απόκτηση ευαισθησίας -148 dBm, Παρακολούθηση -165 dBm, Επανάκτηση -160 dBm. |
| Σειριακές διεπαφές UART: Ρυθμιζόμενη 4800∼115200 bps, Προεπιλογή: 9600bps. | Περιβαλλοντικό: Θερμοκρασία λειτουργίας -40°C έως 85°C, Θερμοκρασία αποθήκευσης -45°C έως 125°C. |
| Ρυθμός ενημέρωσης: 1Hz (Προεπιλογή), έως 10 Hz. | Μέγιστο υψόμετρο δυναμικής απόδοσης 18000 m, μέγιστη ταχύτητα 515 m/s, μέγιστη Επιτάχυνση 4G |
| I/O Voltage:2.7V ∼2.9V | Δέκτης ζώνης L1 (1575,42 MHz) Κανάλι 22 (Παρακολούθηση) /66 (Απόκτηση). |
| Πρωτόκολλα: NMEA 0183, PMTK | |

### Pycom lopy4




Εικόνα 33. Pycom Lopy4

Το lopy4 της pycom (εικόνα 33) είναι ένας μικροελεγκτής που προγραμματίζεται με micropython και υποστηρίζει συνδεσιμότητα LoRa, Sigfox, Wifi και Bluetooth. Για το LoRa μπορεί να χρησιμοποιηθεί σαν πύλη και σαν κόμβος. Χρησιμοποιεί το chip SX1276 της Semtech[60]

Στον πίνακα 10 δίνονται τα βασικά χαρακτηριστικά του Pycom lopy4

Πίνακας 10. Χαρακτηριστικά pycom lopy4 [60]

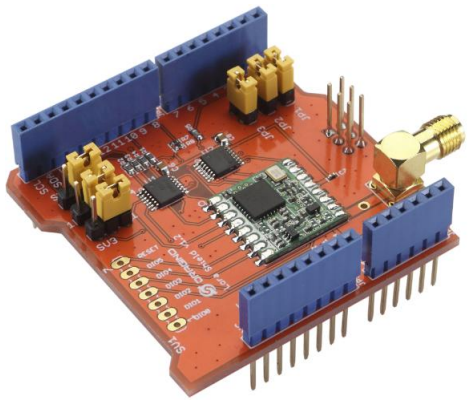
| | |
|---|---|
| LoPy4 Features | Ισχυρή CPU, BLE και ραδιόφωνο WiFi τελευταίας τεχνολογίας,<br>Μπορεί επίσης να χρησιμοποιηθεί ως πύλη Nano LoRa<br>MicroPython enabled,<br>Ταιριάζει σε τυπικό breadboard (με κεφαλίδες), |
| Επεξεργασία | Espressif ESP32 chipset,<br>Dual processor WiFi radio System on chip,<br>Ο επεξεργαστής δικτύου χειρίζεται τη συνδεσιμότητα WiFi<br>και τη στοίβα IPv6,<br>Ο κύριος επεξεργαστής είναι εντελώς δωρεάν για την εκτέλεση<br>της εφαρμογής χρήστη,<br>Ένας επιπλέον συνεπεξεργαστής ULP που μπορεί να παρακολουθεί GPIOs,<br>τα κανάλια ADC ελέγχουν τα περισσότερα από τα εσωτερικά περιφερειακά κατά<br>to sleep-mode ενώκαταναλώνει μόνο 25uA |
| Λειτουργικές συχνότητες Sigfox | RCZ1 – 868MHz (Ευρώπη),<br>RCZ2 – 902MHz (ΗΠΑ, Καναδά και Μεξικό),<br>RCZ3 – (Ιαπωνία και Κορέα) |
| Προδιαγραφές Sigfox | Class 0 device. Maximum Tx power:<br>14dBm (Ευρώπη),<br>20dBm (Αμερική),<br>20dBm 20dBm (Αυστραλία και Νέα Ζηλανδία),<br>Εμβέλεια κόμβου: Έως 50 χλμ,<br>Sigfox certified |
| Χαρακτηριστικά LoRa | Εμβέλεια κόμβου: Έως 40 χλμ,<br>Νανο-πύλη: Έως 22 χλμ,<br>Χωρητικότητα Nano-gateway:<br>Έως 100 κόμβοι |
| Διεπαφές | 2 x UART, SPI, 2 x I2C, I2S, micro SD card,<br>Αναλογικά κανάλια: 8_12 bit ADC,<br>Χρονοδιακόπτες: 4_16 bit με PWM και λήψη εισόδου,<br>GPIO: Έως 24 |
| Μνήμη | RAM: 4MB,<br>External flash: 8MB,<br>Hardware floating point acceleration,<br>Python multi-threading |
| WiFi | 802.1b/g/n 16mbps |
| Bluetooth | Χαμηλή ενέργεια και κλασικό |
| RTC | Λειτουργεί στα 32KHz |

| | |
|---|---|
| Power | Λειτουργεί στα 32KHz,<br>Είσοδος: 3.3V 5.5V,<br>Έξοδος 3v3 με δυνατότητα τροφοδοσίας<br>έως και 400 mA,<br>WiFi: 12mA σε ενεργή λειτουργία,<br>5uA σε αναμονή,<br>Lora: 15mA σε ενεργή λειτουργία,<br>1-uA σε αναμονή,<br>Sigfox (Ευρώπη): 12 mA σε λειτουργία Rx<br>42mA σε λειτουργία Tx και 0,5uA σε αναμονή,<br>Sigfox (Αυστραλία, Νέα Ζηλανδία και Νότια<br>(Αμερική): 12 mA σε λειτουργία Rx, 120 mA<br>σε Tx 120 mA in Txλειτουργία και 0,5uA<br>σε αναμονή |

### 3.4.2 Κόμβοι - Nodes

Οι κόμβοι γνωστοί και ως τελικές συσκευές, είναι αισθητήρες, ενεργοποιητές όπου επικοινωνούν με τις πύλες στέλνοντας τα δεδομένα. Όπως αναφέραμε προηγουμένως οι πύλες στην συνέχεια διαβιβάζουν τα δεδομένα στον αντίστοιχο LoRa Network Server. Θα γίνει περιληπτικά αναφορά σε κόμβους που χρησιμοποιούνται ευρέως.

**Dragino LoRa Shield**



Εικόνα 34. Dragino LoRa Shield

Το Dragino LoRa Shield είναι μια επέκταση για LoRa που συνδέετε στο Arduino (εικόνα 34). Βασίζεται σε βιβλιοθήκη ανοιχτού κώδικα, παρέχει μεγάλη εμβέλεια και χαμηλή κατανάλωση ρεύματος. Χρησιμοποιεί το τσιπ Semtech SX1276/SX1278 και αποσκοπεί σε εφαρμογές ασύρματου δικτύου αισθητήρων.[61]

Στον Πίνακα 11 δίνονται τα βασικά χαρακτηριστικά του Dragino LoRa shield

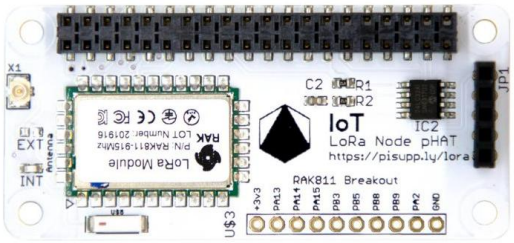
Πίνακας 11. Χαρακτηριστικά Dragino LoRa shield [61]

| Χαρακτηριστικά |
|---|
| 168 dB μέγιστο link budget |

| |
|---|
| +20 dBm - 100 mW constant<br>RF output vs |
| +14 dBm high efficiency PA |
| Προγραμματιζόμενο bit rate up<br>πάνω από 300 kbps |
| Υψηλή ευαισθησια: κάτω<br>από -148 dBm |
| Bullet-proof front end:<br>IIP3 = -12.5 dBm |
| Εξαιρετικό blocking immunity |
| Χαμηλό RX ρεύμα 10.3 mA,<br>200 nA register retention |
| Πλήρως ενσωματωμένο synthesizer με<br>ανάλυση of 61 Hz |
| FSK, GFSK, MSK, GMSK,<br>LoRaTM και OOK διαμόρφωση |
| Ενσωματωμένος συγχρονιστής bit για<br>ανάκτηση ρολογιού |
| Preamble detection |
| 127 dB Dynamic Range<br>RSSI. |
| Packet engine έως 256 byte με CRC |
| Ενσωματωμένος αισθητήρας θερμοκρασίας και<br>ένδειξη χαμηλής μπαταρίας |

### IoT LoRa Node pHAT

Η συγκεκριμένη επέκταση LoRa συνδέεται στο Raspberry pi (εικόνα 35). Υποστηρίζει όλα τα μοντέλα του Raspberry καθώς και την σειρά zero. Η υποστήριξη της σειράς pi zero δίνει μεγάλο πλεονέκτημα στο συγκεκριμένο module καθώς μπορούν να υλοποιηθούν κόμβοι LoRa με χαμηλό κόστος. Χρησιμοποιεί το τσιπ RAK811 το οποίο βασίζεται στο Semtech SX1276.




Εικόνα 35. IoT LoRa Node pHAT

Στον Πίνακα 12 δίνονται τα βασικά χαρακτηριστικά του IoT LoRa Node pHAT

Πίνακας 12. Χαρακτηριστικά IoT LoRa Node pHAT

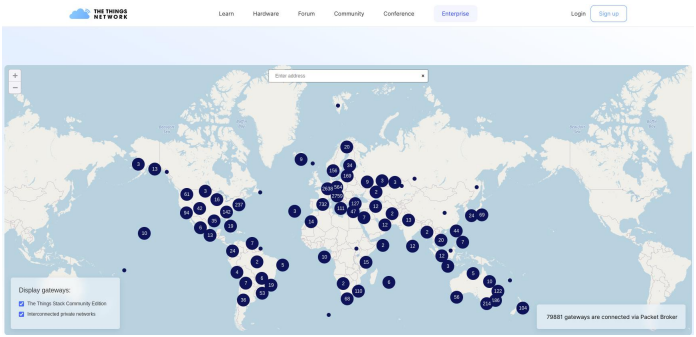
| Χαρακτηριστικά | Υποστήριξη Συχνοτήτων |
|---|---|
| Χρησιμοποιεί το chip RAK811 LoRa Radio με πλήρη LoRaWAN Stack | EU 863 - 870 MHz |
| Επικοινωνεί με το Raspberry Pi μέσω UART | US 902 - 928 MHz |
| Υποστηρίζει LoRaWAN συνδέσεις και LoRaP2P λειτουργίες | AU 915 - 928 MHz |
| Χαμηλή ισχύς - χρησιμοποιεί λιγότερο από 50 mA κατά τη μετάδοση | AS 923 MHz |
| Μέγιστη ισχύς εξόδου 100MW (20dBm) / ρυθμιζόμενο από 5 έως 20dBm | IN 865 - 867MHz |
| Υψηλή ευαισθησία:: -148dBm επιτρέποντας εξαιρετικά μεγάλη συνδεσιμότητα εμβέλειας. | KR 920 MHz |

## 3.5 LoRaWAN Network Servers

Όπως έγινε αναφορά στις προηγούμενες ενότητες οι κόμβοι επικοινωνούν με τις πύλες στέλνοντας τα δεδομένα και οι πύλες στην συνέχεια διαβιβάζουν τα δεδομένα στον αντίστοιχο LoRa Network Server. Σε αυτήν την ενότητα θα γίνει αναφορά στο The Things Network (TTN) και στο Chirpstack.

### 3.5.1 The Things Network (TTN)

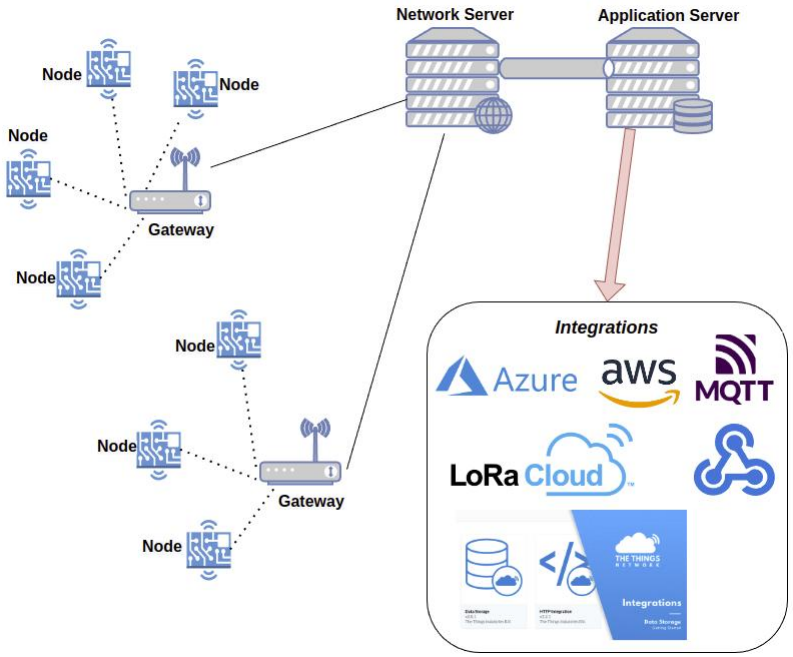
Το Things Network γνωστό ως TTN, είναι μια υποδομή ανοιχτού κώδικα που στοχεύει στην παροχή δωρεάν κάλυψης ασύρματου δικτύου LoRaWAN. Η κοινότητα αναπτύσσεται σε διάφορες χώρες και στην εικόνα 36 φαίνονται οι ενεργές πύλες gateways που είναι 79.881.




Εικόνα 36. Χάρτης ενεργών πυλών TTN

Με την χρήση του TTN το οποίο παρέχει ένα σύνολο ανοιχτών εργαλείων καθιστά εφικτό την δημιουργία εφαρμογών IoT με χαμηλό κόστος, με ασφάλεια και επεκτασιμότητα. Η ελεύθερη έκδοση έχει κάποιους περιορισμούς όπως το μέγιστο ωφέλιμο φορτίο στο κανάλι ανόδου ανά μήνυμα είναι 51 bytes για κάθε κόμβο και τα μηνύματα στο κανάλι καθόδου περιορίζονται σε 10 μηνύματα την ημέρα για κάθε κόμβο. [62]

Στην εικόνα 37 δίνεται η αρχιτεκτονική του δικτύου TTN.




Εικόνα 37. Η αρχιτεκτονική του TTN

Οι πύλες επικοινωνούν με τους κόμβους και διαβιβάζουν τα δεδομένα στον διακομιστή δικτύου. Ο διακομιστής εφαρμογών επεξεργάζεται τα μηνύματα δεδομένων για συγκεκριμένη εφαρμογή που λαμβάνονται από τις τελικές συσκευές. Επίσης, δημιουργεί όλα τα ωφέλιμα φορτία κατερχόμενης ζεύξης επιπέδου εφαρμογής και τα στέλνει στις συνδεδεμένες τελικές συσκευές μέσω του διακομιστή δικτύου. [63]

### 3.5.2 Chirpstack

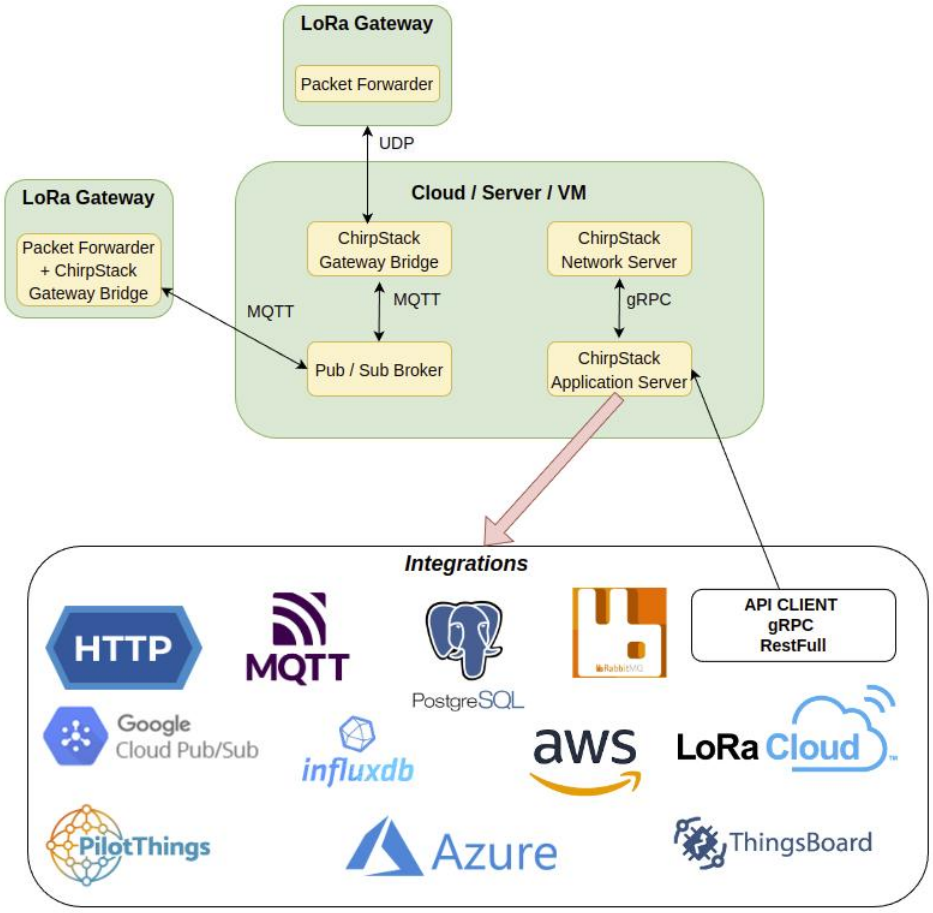
Το Chirpstack παρέχει δομικά στοιχεία ανοιχτού κώδικα για την δημιουργία ασύρματου δικτύου LoRaWAN. Προσφέρει μια καλή λύση για δημιουργία IoT εφαρμογών καθώς είναι φιλικό στον χρήστη. Όλα τα εξαρτήματα έχουν άδεια χρήσης βάσει της άδειας MIT και μπορούν να χρησιμοποιηθούν για εμπορικούς σκοπούς. Το Chirpstack αποτελείται από τα εξής δομικά στοιχεία :

- ChirpStack Gateway Bridge
- ChirpStack Network Server
- ChirpStack Application Server

Στην εικόνα 38 της επόμενης σελίδας δίνεται η αρχιτεκτονική του Chirpstack.

#### ChirpStack Gateway Bridge

Αναλαμβάνει την μετατροπή των πακέτων δεδομένων που λαμβάνονται από την πύλη, σε μορφή Semtech UDP packet-forwarder σε μορφή όπως json ή protobuf. Η μορφή που προκύπτει στην συνέχεια γίνεται publish σε έναν MQTT Broker σε ένα θέμα που καθορίζει τον τύπο του μηνύματος που στάλθηκε από την συσκευή. Ο διακομιστής δικτύου είναι subscriber σε τέτοια θέματα.




Εικόνα 38. Η αρχιτεκτονική του ChirpStack

#### ChirpStack Network Server

Λειτουργεί ως ενδιάμεσο μεταξύ της πύλης και του διακομιστή εφαρμογών. Παρέχει έναν μηχανισμό επαλήθευσης πακέτων για να διασφαλίσει τη μοναδικότητα του πακέτου.

#### Application Server

Χειρίζεται τα αιτήματα σύνδεσης και την κρυπτογράφηση ωφέλιμου φορτίου. Επιπλέον, προσφέρει διεπαφή χρήστη (UI) για εύκολη διαμόρφωση και διαχείριση ολόκληρης της στοίβας LoRaWAN. [64], [65]

Στο παράρτημα 2 δίνονται οι οδηγίες εγκατάστασης του ChirpStack για το Raspberry (ίδια βήματα για όλα τα linux συστήματα)

### 3.5.3 Επιλογή LoRaWAN Network Server

Πριν την τελική υλοποίηση της SMGW συστήματος επιλέχθηκε το ChirpStack για να γίνουν δοκιμές στους κόμβους και στις πύλες. Οι λόγοι που επιλέχθηκε το ChirpStack είναι οι εξής:

- Χρησιμοποιεί όλο το εύρος στο ωφέλιμο φορτίο (243 bytes) του ανοδικού καναλιού και δεν περιορίζεται στα 51 bytes ανά μήνυμα όπως στο TTN. Αυτό οδηγεί σε πιο ιδανική λύση για έξυπνους μετρητές οι αισθητήρες μετρήσεων ηλεκτρικής ενέργειας.

- Δεν υπάρχουν περιορισμοί στο κανάλι καθόδου όπως στο TTN που υπάρχει περιορισμός στα 10 μηνύματα την ημέρα ανά κόμβο. Αυτό οδηγεί σε μια πιο ιδανική λύση για ένα έξυπνο δίκτυο όπου οι παρόχοι ηλεκτρικής ενέργειας θα μπορούν να λάβουν αποφάσεις σε περιόδους αιχμής φορτίου και σε διακοπές.
- Το περιβάλλον του ChirpStack είναι πιο παραμετροποιήσιμο και ήταν μια ευκαιρία να υλοποιήσω το θεωρητικό υπόβαθρο του LoRaWAN.
- Είναι εγκατεστημένο το ChirpStack στο server του εργαστηρίου restqmlab του πανεπιστημίου Δυτικής Αττικής με δυνατότητα απομακρυσμένης διαχείρισης.

Οι δοκιμές των συσκευών LoRa έγιναν και τοπικά στην πύλη του Raspberry pi και στο server του εργαστηρίου restqmlab του Πανεπιστημίου Δυτικής Αττικής (ΠΑΔΑ). Στην ενότητα που ακολουθεί δίνονται τα αποτελέσματα των συσκευώ LoRa στο server του εργαστηρίου restqmlab.

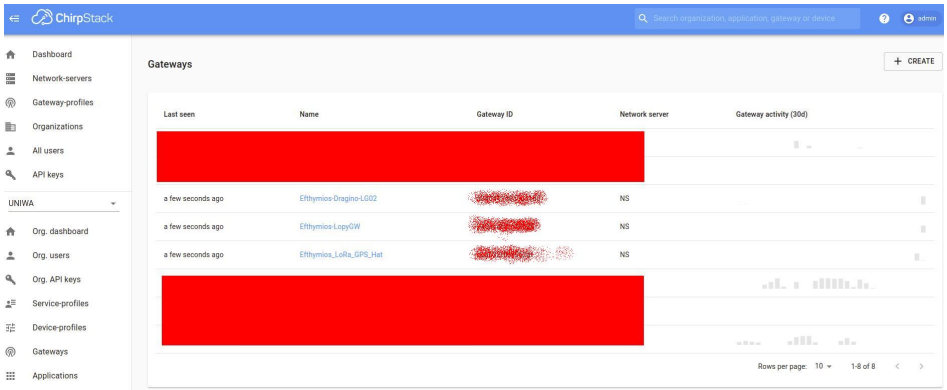
## 3.6 Αποτελέσματα συσκευών LoRa στο server του εργαστηρίου restqmlab

Σε αυτήν την ενότητα δίνονται τα αποτελέσματα από τις ενεργές πύλες καθώς και την αποστολή ενός απλού μηνύματος από τους κόμβους.

### 3.6.1 Ενεργές πύλες στο server του εργαστηρίου restqmlab

Στην εικόνα 39 φαίνονται οι ενεργές πύλες που είναι οι:

- Dragino LG02
- Lopy 4
- Dragino Lora/GPS Hat

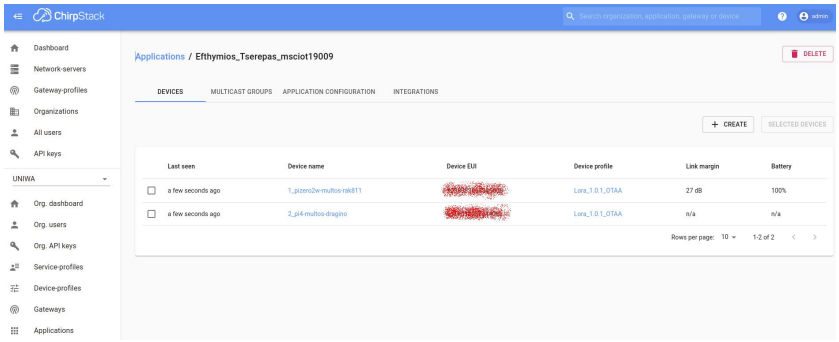



Εικόνα 39. Ενεργές πύλες

### 3.6.2 Ενεργοί κόμβοι στο server του εργαστηρίου restqmlab

Στην εικόνα 40 φαίνονται οι ενεργοί κόμβοι που είναι οι:

- IoT LoRa Node pHAT
- Dragino Lora/GPS Hat




Εικόνα 40. Ενεργοί κόμβοι

## 3.7 Επιλογές για την υλοποίηση της SMGW

Θα δοθεί έμφαση στην ασφάλεια και την ακεραιότητα των δεδομένων στο τελικό IoT σύστημα που θα υλοποιηθεί στο κεφάλαιο 5. Η χρήση του Blockckain προσφέρει ασφάλεια και εμπιστοσύνη σε ένα σύστημα IoT καθώς κάθε νέο μπλοκ συνδέεται με όλα τα μπλοκ πριν από αυτό σε μια κρυπτογραφική αλυσίδα στο με τέτοιο τρόπο που είναι σχεδόν αδύνατο να παραβιαστεί. [66]. Όπως θα δούμε στην συνέχεια θα γίνει χρήση του Helium Blockchain όπου είναι το μεγαλύτερο δημόσιο αποκεντρωμένο δίκτυο LoRaWAN.

Το Blockchain από μόνο του δεν είναι σε θέση να εξασφαλίσει πλήρως μια συναλλαγή γιατί εγγυάται μόνο την αναλλοίωτη κατάσταση των δεδομένων, ενώ στις περισσότερες περιπτώσεις τα δεδομένα πρέπει να είναι ασφαλή στο σημείο παραγωγής. Προκειμένου να αντιμετωπιστεί το κενό ασφαλείας, το ωφέλιμο φορτίο από τον κόμβο θα κρυπτογραφηθεί με κρυπτογράφηση AES-CBC χρησιμοποιώντας το Secure Element : Multos Trust Core προτού μεταδοθεί στο δίκτυο Helium.

Το Multos Trust Core συνδέεται με ευκολία σε raspberry pi καθώς και σε raspberry pi zero παρέχοντας πολλές δυνατότητες. Για αυτό τον λόγο θα χρησιμοποιηθεί για τον κόμβο LoRa-Helium η επέκταση IoT LoRa Node pHAT όπου θα συνδεθεί στο raspberry pizero2w ρίχνοντας σημαντικά το κόστος υλοποίησης. Το Raspberry pi zero 2w αποτελεί μια οικονομική λύση με κόστος που κυμαίνεται 15-20 €, διαθέτει τετραπύρηνο επεξεργαστή Arm Cortex-a53 όπου είναι επαρκής για την υλοποίηση της SMGW από μεριάς υπολογιστικής ισχύος. Επίσης το Raspberry είναι linux embedded system με πλεονέκτημα την επεκτασιμότητα.

Τα δοκιμαστικά αποτελέσματα αποστολής δεδομένων στον server restqmlab από τον κόμβο ήταν ικανοποιητικά χωρίς να υπάρξει πρόβλημα.

Στην ενότητα που ακολουθεί θα γίνει η εγκατάσταση του IoT LoRa Node pHAT για το pizero2w.

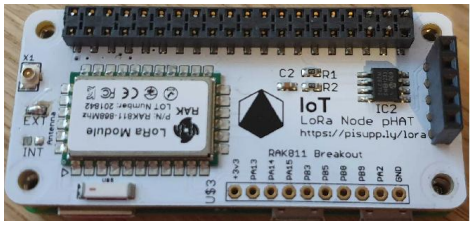
### 3.7.1 Εγκατάσταση IoT LoRa Node pHAT

Σε αυτήν την ενότητα θα γίνει εγκατάσταση του IoT LoRa Node pHAT για το pizero2w (ίδια βήματα και για το pi) Στο παράρτημα 1 δίνονται οδηγίες για την εγκατάσταση του λειτουργικού Raspbian OS και των βασικών ρυθμίσεων.

**Hardware:**

Τοποθετούμε την πλακέτα pHAT του κόμβου LoRa πάνω από το pizero2w με τις κεφαλίδες στραμμένες προς τα πάνω. Επιβεβαιώνουμε ότι οι ακίδες GPIO ευθυγραμμίζονται και πιέζονται σταθερά προς τα κάτω (εικόνα 41). [67]

Το pHAT PIS-1128 επικοινωνεί με το Raspberry Pi μέσω UART μόνο χρησιμοποιώντας 3 GPIO [68]. Η εικόνα 42 δείχνει τις GPIO που χρησιμοποιούνται από το module:




Εικόνα 41. Σύνδεση του pHAT LoRa στο pizero2w [67]


Εικόνα 42. GPIO που χρησιμοποιούνται από το module [69]

**Software:**

Πρώτα ενεργοποιούμε το hardware serial στο Raspberry Pi και απενεργοποιούμε το serial console ακολουθώντας τα παρακάτω βήματα :

- στο terminal εκτελούμε την εντολή sudo raspi-config
- Επιλέγουμε την επιλογή (3) interface option
- στην συνέχεια I6 Serial Port
- Επιλέγουμε NO και μετά
- Επιλέγουμε YES.

Στην συνέχεια κάνουμε επανεκκίνηση.

```
sudo reboot
```

Στην συνέχεια μέσω terminal εγκαθιστουμε τη βιβλιοθήκη RAK811 Python:

```
$ sudo apt install python3-pip
$ sudo pip3 install rak811
```

[67]

**Κώδικας ενεργοποίησης για την μέθοδο OTAA**

Ο κόμβος θα ενεργοποιηθεί μέσω της μεθόδου OTAA. Το Helium Blockchain που θα χρησιμοποιηθεί στην συνέχεια υποστιρίζει την μέθοδο OTAA, δεν υποστιρίζει ABP. Η μέθοδος OTAA έχει πλεονεκτήματα που έχουν αναφερθεί στην ενότητα 3.3.4 : Σύγκριση ενεργοποίησης ABP με OTAA.

Αρχικά βρίσκουμε το dev_eui της συσκευής για να το χρησιμοποιήσουμε στον κώδικα 1 Μέσω terminal εκτελούμε τις εντολές:

```
$ rak811 hard-reset
$ rak811 get-config dev_eui
```

```
from sys import exit
from time import sleep
from rak811.rak811 import Mode, Rak811
lora = Rak811()

lora.hard_reset()
lora.mode = Mode.LoRaWan
lora.band = 'EU868'

lora.set_config(dev_eui='xxxxxxxxxxxxxxx',
app_eui='xxxxxxxxxxxxxxx',
app_key='xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
lora.join_otaa()


lora.dr = 5

print('Sending packets every 3 sec - Interrupt to cancel loop')
print('You can send downlinks from the Chirpstack console')
try:
    while True:
        print('Send packet')
        # Cayenne lpp random value as analog
        lora.send('Hello World!')

        while lora.nb_downlinks:
            print('Received', lora.get_downlink()['data'].hex())

        sleep(3)
except:  # noqa: E722
    pass

print('Cleaning up')
lora.close()
exit(0)
```

Κώδικας 1. Activation OTAA - sending a simple message

[67], [70]

## 3.8 Σύνοψη

Σε αυτό το κεφάλαιο έγινε θεωρητική μελέτη του δικτύου LoRa, επιλογή ενσωματωμένου συστήματος, LoRa επέκτασης και δοκιμές στο Chirpstack Server του εργαστηρίου restqmlab του Πανεπιστημίου Δυτικής Αττικής. Τα αποτελέσματα είναι ικανοποιητικά και στο επόμενο κεφάλαιο θα συζητηθούν θέματα ασφάλειας και λύσεις που θα βοηθήσουν στην ασφάλεια της SMGW.

# ΚΕΦΑΛΑΙΟ 4

## Ασφάλεια : Blockchain, Secure Element και Azure IoT Hub

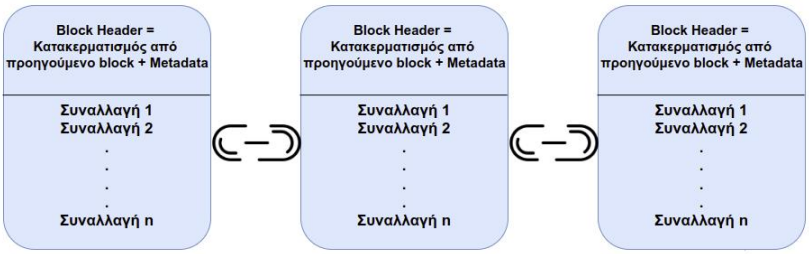
Σε αυτό το κεφάλαιο θα αναλυθούν θέματα ασφάλειας. Το blockchain αποτελεί μια καλή λύση για την ασφάλεια, την ακεραιότητα των δεδομένων αλλά σε μερικές περιπτώσεις η χρήση του μπορεί να είναι ασύμφορη. Για το IoT σύστημα μας το blockchain Helium είναι μια βιώσιμη λύση. Το blockchain από μόνο του δεν είναι σε θέση να εξασφαλίσει πλήρως μια συναλλαγή γιατί εγγυάται μόνο την αναλλοίωτη κατάσταση των δεδομένων, ενώ στις περισσότερες περιπτώσεις τα δεδομένα πρέπει να είναι ασφαλή στο σημείο παραγωγής. Για να καλυφθεί αυτό το κενό ασφαλείας, προτείνετε η χρήση των Secure Elements (SE) και συγκεκριμένα του Multos Trust Core για τη δημιουργία μιας «ρίζας εμπιστοσύνης» και για την παροχή αξιόπιστων υπολογιστικών πόρων τις IoT συσκευές για την κρυπτογράφηση του ωφέλιμου φορτίου με κρυπτογράφηση AES-CBC πριν μεταδοθεί στο Helium ακολουθώντας τον ασφαλή σχεδιασμό του μοντέλου. Το Azure IoT Hub προσφέρει μια αξιόπιστη και ασφαλή αμφίδρομη επικοινωνία μεταξύ συσκευών και του cloud και λειτουργεί ως κεντρικός κόμβος μηνυμάτων για την επικοινωνία μεταξύ μιας εφαρμογής IoT και των συνδεδεμένων συσκευών της. Θα δοθεί λύση για την επιπλέον ασφάλεια στο Azure IoT Hub με την χρήση του Multos Trust Core.

## 4.1 Blockchain

### 4.1.1 Εισαγωγή

Το Blockchain, είναι μια σειρά από μπλοκ (που περιέχουν δεδομένα) στα οποία τα μπλοκ συνδέονται μεταξύ τους μέσω κρυπτογραφικών λειτουργιών σχηματίζοντας μια ισχυρή αλυσίδα. Αυτό το βοηθά να καταγράφει τις συναλλαγές μέσω μιας ασφαλούς και επαληθεύσιμης διαδικασίας χωρίς κανέναν μεσάζοντα. Στις περισσότερες περιπτώσεις, μόνο ο κατακερματισμός του προηγούμενου μπλοκ περιλαμβάνεται στο τρέχον μπλοκ κατά τον σχηματισμό του (εικόνα 43). [71]

Το blockchain έγινε γνωστό από την κυκλοφορία του Bitcoin. Το Bitcoin δημιουργήθηκε το 2008 από ένα άτομο ή μια ομάδα με ονομασία Satoshi Nakamoto και κυκλοφόρησε το 2009 όταν ο πηγαίος κώδικάς του δόθηκε στο διαδίκτυο σαν ελεύθερο λογισμικό. Όπως και στην εικόνα 43 αποτελείται από διαδοχικά αλυσιδωτά μπλοκ (consecutive chained blocks) που αναπαράγονται και αποθηκεύονται από τους κόμβους ενός δικτύου peer-to-peer, όπου τα μπλοκ δημιουργούνται με έναν κατανεμημένο τρόπο μέσω ενός αλγορίθμου συναίνεσης (consensus algorithm). Ένας τέτοιος αλγόριθμος, μαζί με τη χρήση μηχανισμών κρυπτογράφησης, παρέχει δύο διακριτικές ιδιότητες του blockchain: αποκέντρωση (decentralisation) και δημοκρατικό έλεγχο (democratic control)




Εικόνα 43. Βασική δομή blockchain

των δεδομένων. Αυτό διασφαλίζει ότι τα δεδομένα στην αλυσίδα δεν μπορούν να παραβιαστούν με κακόβουλο τρόπο, ότι οι λειτουργίες στην αλυσίδα δεν μπορούν να απορριφθούν και ότι η προέλευσή τους παρακολουθείται πλήρως. Με την χρήση του Bitcoin πραγματοποιούνται συναλλαγές με ταχύτητα σε όλο τον κόσμο χωρίς να απαιτεί υποδομές πέρα από ένα ψηφιακό πορτοφόλι που είναι δωρεάν σε μορφή λογισμικού που μπορεί να εγκατασταθεί σε υπολογιστή (Windows, Linux, Mac) ή σε tablet-smartphone (Android,IOS,Harmony OS) και σύνδεση στο διαδίκτυο.

΄Ολα αυτά επιτυγχάνονται μέσω του μηχανισμού συναίνεσης που ονομάζεται Proof-of-Work (PoW). [72] Το PoW είναι μια κρυπτογραφική υπολογιστική διαδικασία κατακερματισμού (hashing) που δημιουργεί μπλοκ με τη συναίνεση όλων των κόμβων του δικτύου. Το PoW αποτελεί τη βάση και για πολλά άλλα κρυπτονομίσματα όπως και το Ethereum (θα αλλάξει μηχανισμό συναίνεσης όταν γίνει αναβάθμιση σε Ethereum 2) , επιτρέποντας την ασφαλή, αποκεντρωμένη συναίνεση. Συνοπτικά το PoW :

- Είναι ένας αποκεντρωμένος μηχανισμός συναίνεσης που απαιτεί από τα μέλη ενός δικτύου να καταβάλουν προσπάθεια για να λύσουν ένα αυθαίρετο μαθηματικό παζλ για να εμποδίσουν τον οποιοδήποτε να δημιουργήσει πρόβλημα στο σύστημα

- Χρησιμοποιείται ευρέως στην εξόρυξη κρυπτονομισμάτων, για την επικύρωση συναλλαγών και την εξόρυξη νέων token.

- Λόγω του PoW , οι συναλλαγές Bitcoin και άλλων κρυπτονομισμάτων μπορούν να υποβληθούν σε επεξεργασία peer-to-peer με ασφαλή τρόπο χωρίς την ανάγκη αξιόπιστου τρίτου μέρους.

- Το PoW σε κλίμακα απαιτεί τεράστιες ποσότητες ενέργειας, οι οποίες αυξάνονται μόνο καθώς περισσότεροι ανθρακωρύχοι (miners) εντάσσονται στο δίκτυο. [73]

Εκτός από το PoW υπάρχουν και διάφοροι μηχανισμοί συναίνεσης όπως για παράδειγμα το Proof of Stack (PoS) που χρησιμοποιούν μερικά από τα πιο διαδεδομένα blockchains όπως Polkadot, Avalanche, Cardano και με την αναβάθμιση του Ethereum στην έκδοση 2 θα χρησιμοποιεί το PoS, το Proof of History (PoW) που χρησιμοποιεί το blockchains Solana, το Proof of Coverage (PoC) που χρησιμοποιεί το blockchains Helium.

Το κάθε blockchain κυκλοφορεί και το δικό του κρυπτονόμισμα (token). Για παράδειγμα το blockchain του Ethereum έχει το token Ether, το Helium το HNT. Το blockchain και το token (κρυπτονόμισμα) είναι διαφορετικές έννοιες. Για να καταλάβουμε την έννοια του token θα δοθεί ένα παράδειγμα για τον φυσικό κόσμο. Ένα εισιτήριο για το θέατρο μπορεί να θεωρηθεί ως token. Έγινε μια συναλλαγή σε ευρώ για την απόκτηση του εισιτηρίου token, η υπηρεσία (για παράδειγμα το blockchain) είναι η τέχνη το θέατρο και το token το εισιτήριο για να παρακολουθήσουμε την θεατρική παράσταση.

### 4.1.2 Δομή του blockchain

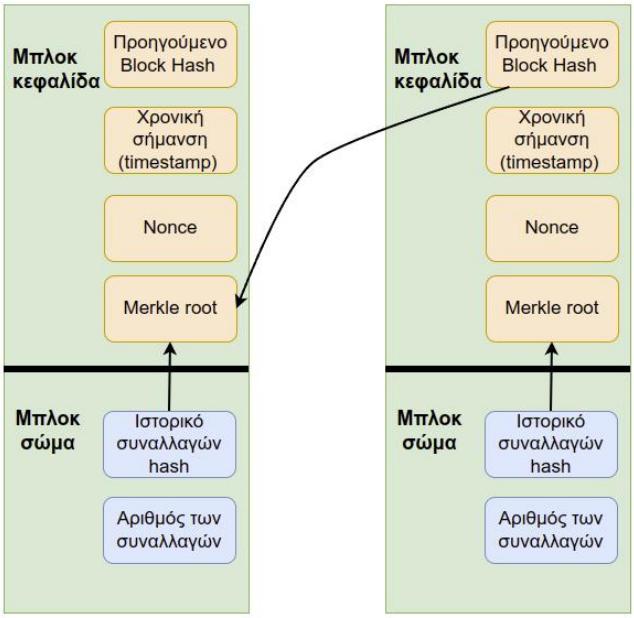
Συνολικά, το blockchain αποτελείται από τρία βασικά μέρη:

1. Block: αυτό μπορεί να θεωρηθεί ως μια λίστα λογαριασμών που δεν μπορούν να τροποποιηθούν. Μόλις γίνει εγγραφή κάτι σε μπλοκ, κάθε κόμβος μπορεί να αναζητήσει τα πάντα σε αυτό το μπλοκ. Το μέγεθος, η περίοδος και τα συμβάντα ενεργοποίησης των μπλοκ εξαρτώνται από τον τύπο της αλυσίδας μπλοκ (blockchain).

2. Chain : έχει τη λειτουργία της σύνδεσης μιας λίστας μπλοκ. Το Blockchain βασίζεται στη σύνδεση όλων των μπλοκ.

3. Network: αυτό είναι ένα σύνολο κόμβων. Σε ένα παραδοσιακό δίκτυο, οι δρομολογητές (routes) θεωρούνται κόμβοι (nodes) , ενώ σε ένα δίκτυο blockchain, τα μπλοκ (blocks) είναι οι κόμβοι (nodes).

Αναλυτικά, η δομή μιας αλυσίδας μπλοκ (blockchain) δίνεται στην εικόνα 44.




Εικόνα 44. Δομή του blockchain

Ένα μπλοκ μπορεί να χωριστεί σε δύο μέρη:

- Κεφαλίδα (header)
- Σώμα (body)

Η κεφαλίδα μπλοκ περιλαμβάνει :

1. κατακερματισμό (hash) πριν από το μπλοκ
2. μια χρονική σήμανση (timestamp) που υποδεικνύει τον χρόνο εγγραφής των δεδομένων μπλοκ
3. Nonce : η τιμή του οποίου προσαρμόζεται από τους εξορύκτες (miners) έτσι ώστε ο κατακερματισμός του μπλοκ να είναι μεγαλύτερος από τον κατακερματισμό του επόμενου μπλοκ.

4. Merkle root : γρήγορη σύνοψη και επαλήθευση της ύπαρξης και της ακεραιότητας των δεδομένων του μπλοκ.

Το σώμα του μπλοκ καταγράφει τις λεπτομέρειες πληροφοριών συναλλαγής και τον αριθμό των συναλλαγών.

Μέσω της τιμής δυσκολίας της διαδικασίας PoW, βρίσκεται πρώτα το σωστό Nonce και οι εξορύκτες (miners) που έχουν επαληθευτεί από όλους τους εξορύκτες θα λάβουν τα τρέχοντα δικαιώματα λογιστικής μπλοκ. Αυτές οι εγγραφές δημιουργούνται από τη διαδικασία κατακερματισμού (hash), της ρίζας Merkle για μέτρηση στην κεφαλίδα του μπλοκ (block header). ΄Οταν συμβαίνει μια νέα συναλλαγή, οι πληροφορίες μεταδίδονται σε όλους τους συμμετέχοντες στο δίκτυο. ΄Ολοι οι εξορύκτες που έχουν λάβει τη συναλλαγή θα την επαληθεύσουν επικυρώνοντας την υπογραφή της συναλλαγής. Οι εξορύκτες παραγγέλνουν και συσκευάζουν τις συναλλαγές σε μπλοκ χρονοσήμανσης (timestamp blocks). Στη συνέχεια, μεταδίδουν τα μπλοκ πίσω στο δίκτυο. Οι κόμβοι δικτύου (network nodes) επαληθεύουν ότι τα μπλοκ περιέχουν έγκυρες συναλλαγές και αναφέρονται στο προηγούμενο μπλοκ της αλυσίδας αναπτύσσοντας ένα είδος αλγορίθμου κατακερματισμού. Τα μπλοκ που δεν επαληθεύονται από κόμβους θα απορριφθούν. Επομένως, ένα blockchain παρέχει μια αξιόπιστη στρατηγική για σενάρια ανταλλαγής πληροφοριών ή συναλλαγών πόρων. [74]

### 4.1.3 Τύποι κόμβων blockchain

Οι διάφοροι τύποι κόμβων blockchain περιλαμβάνουν τα ακόλουθα:

- Miner: Οι εξορύκτες (miners) είναι ειδικοί κόμβοι που συσκευάζουν τις συναλλαγές σε μπλοκ και εκτελούν τους αλγόριθμους συναίνεσης (consensus algorithms) που ικανοποιούν τις απαιτήσεις του συστήματος. Σύμφωνα με το PoW οι εξορύκτες στο δίκτυο διαθέτουν την υψηλότερη υπολογιστική ισχύ.

- Πλήρης κόμβος (Full node) : Οι πλήρεις κόμβοι κατεβάζουν (download) ολόκληρο το blockchain και επαληθεύουν συνεχώς την ακεραιότητα όλων των συναλλαγών, καθιστώντας την υποδομή διαχωρίσιμη και αποκεντρωμένη. Απαιτείται επαρκής αποθήκευση και υπολογιστική ισχύς για την εκτέλεση ενός πλήρους κόμβου.

- Thin client: Οι Thin client πραγματοποιούν λήψη μόνο των κεφαλίδων μπλοκ που περιέχουν τους κατακερματισμούς (hashes) των συναλλαγών εντός του μπλοκ. Επομένως, είναι δυνατή η αλληλεπίδραση με την αλυσίδα μπλοκ με ελάχιστες απαιτήσεις αποθήκευσης και υπολογιστικής ισχύς. Αυτή η προσέγγιση ονομάζεται απλοποιημένη επαλήθευση πληρωμής στο Bitcoin και light client στο Ethereum.

- Διακομιστής εμπιστοσύνης πελάτη (Server-trusting client) : Η διεπαφή προγραμματισμού εφαρμογής πελάτη Bitcoin (Bitcoin client application programming interface - BCCAPI) προτείνεται για τη δημιουργία ασφαλών, ελαφρών πελατών (lightweight clients) για συστήματα με περιορισμένους πόρους. Με το BCCAPI, είναι δυνατό για έναν πελάτη να συνδέσει έναν διακομιστή που περιέχει το blockchain και να εκτελέσει ερωτήματα (queries) εναντίον του. Εδώ, ο διακομιστής έχει μόνο δημόσια κλειδιά πελατών και δεν μπορεί να δημιουργήσει μια συναλλαγή χωρίς την έγκριση του πελάτη. [75]

### 4.1.4 Ορολογίες που συναντάμε στο blockchain

#### Αποκέντρωση - Decentralization

Στο blockchain, η αποκέντρωση αναφέρεται στη μεταφορά του ελέγχου και της λήψης αποφάσεων από μια κεντρική οντότητα (άτομο, οργανισμός ή ομάδα αυτών) σε ένα κατανεμημένο

δίκτυο. Τα αποκεντρωμένα δίκτυα προσπαθούν να μειώσουν το επίπεδο εμπιστοσύνης που πρέπει να έχουν οι συμμετέχοντες ο ένας στον άλλο και να αποτρέψουν την ικανότητά τους να ασκούν εξουσία ή να ελέγχουν ο ένας τον άλλον με τρόπους που υποβαθμίζουν τη λειτουργικότητα του δικτύου. [76]

#### Καθολικό - Ledger

Το blockchain είναι μια μορφή δημόσιου καθολικού, το οποίο είναι μια σειρά (ή αλυσίδα) μπλοκ στα οποία καταγράφονται οι λεπτομέρειες συναλλαγής μετά από κατάλληλο έλεγχο ταυτότητας και επαλήθευση από τους καθορισμένους συμμετέχοντες στο δίκτυο. [77]

#### Πορτοφόλι - Wallet

Το πορτοφόλι blockchain είναι ένα πορτοφόλι κρυπτονομισμάτων που επιτρέπει στους χρήστες να διαχειρίζονται διαφορετικά είδη κρυπτονομισμάτων. Ένα πορτοφόλι blockchain βοηθά κάποιον να ανταλλάσσει χρήματα εύκολα. Οι συναλλαγές είναι ασφαλείς, καθώς είναι κρυπτογραφικά υπογεγραμμένες. [78]

#### Byzantine Fault Tolerance (BFT)

Το Byzantine Fault Tolerance (BFT) είναι ένα χαρακτηριστικό αποκεντρωμένων συστημάτων χωρίς άδεια, τα οποία είναι ικανά να εντοπίζουν και να απορρίπτουν με επιτυχία ατελείς ή λανθασμένες πληροφορίες.

Σε ένα αποκεντρωμένο σύστημα χωρίς άδεια, ο καθένας μπορεί να εγγραφεί στο δίκτυο και να ξεκινήσει τη μετάδοση πληροφοριών. Χωρίς το Byzantine Fault Tolerance, οποιοδήποτε μέλος ενός δικτύου θα μπορούσε να τροφοδοτήσει μη έγκυρες πληροφορίες σε ένα δίκτυο και να υπονομεύσει την αξιοπιστία του δικτύου. Στο πλαίσιο του Bitcoin, ένας κόμβος μπορεί εύκολα να ενταχθεί στο δίκτυο και να ξεκινήσει τη μετάδοση μπλοκ και συναλλαγών. Για παράδειγμα, ένας κόμβος θα μπορούσε να μεταδώσει δύο συναλλαγές που ξοδεύουν το ίδιο bitcoin—μια διπλή δαπάνη. Έτσι, το Bitcoin χρειάζεται έναν τρόπο ώστε οι κόμβοι να προσδιορίζουν την εγκυρότητα των δεδομένων που λαμβάνουν από άλλους κόμβους.

Το Bitcoin είναι Byzantine Fault Tolerant γιατί κάθε κόμβος μπορεί να επαληθεύσει κάθε συναλλαγή και να αποκλείσει ανεξάρτητα και με αντικειμενικό τρόπο. Εάν ένας κόμβος εκπέμπει μη έγκυρα μπλοκ ή συναλλαγές, όλοι οι άλλοι κόμβοι θα τα αναγνωρίσουν και θα τα απορρίψουν, αποτρέποντας την είσοδο μη έγκυρων συναλλαγών στο blockchain. [79]

## 4.2 Ethereum

Εάν το bitcoin έφερε την αποκέντρωση του χρήματος, το ethereum έχει ως στόχο να αποτελέσει την αποκέντρωση όλου του internet. [80] Το blockchain Ethereum δημιουργήθηκε το 2013 από τον Vitalik Buterin. Ο βασικός σκοπός είναι να δημιουργήσει ένα εναλλακτικό πρωτόκολλο για τη δημιουργία αποκεντρωμένων εφαρμογών και όχι ένα δίκτυο πληρωμών. Το token του Ethereum blockchain είναι το ether. [81]

Το Ethereum και το bitcoin έχουν κοινά στοιχεία όπως :

- Είναι δίκτυα P2P
- Εχουν αλγόριθμο συναίνεσης Byzantine Fault-Tolerant (BFT) για συγχρονισμό του καθολικού
- Κάνουν χρήση κρυπτογραφικών εργαλείων όπως ψηφιακές υπογραφές και κατακερματισμοί

- Έχουν το δικό τους ψηφιακό νόμισμα (token).

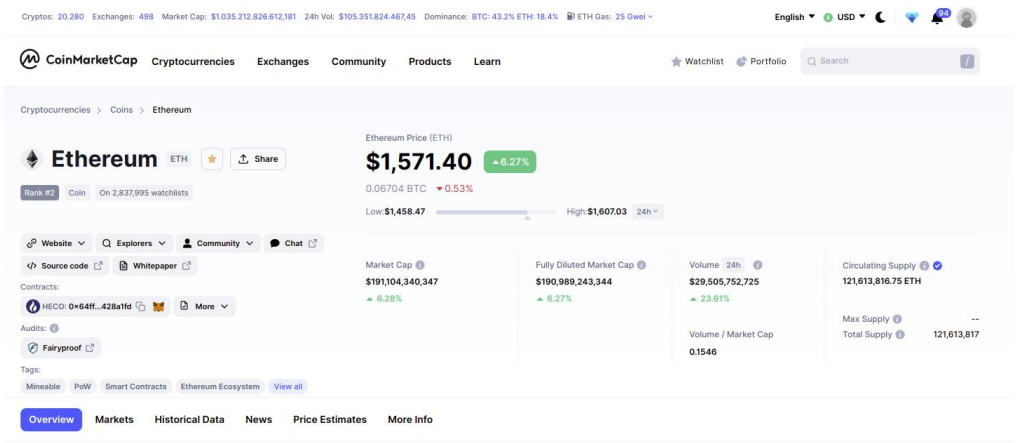
Το Ethereum ενσωματώνει μια πλήρη γλώσσα Turing (όπως solidity) για την δημιουργία και την χρήση έξυπνων συμβολαίων (smart contracts), κομμάτια κώδικα που είναι μόνιμα αποθηκευμένα στο blockchain και ικανά να ανταποκρίνονται στα αιτήματα των χρηστών [82]

Τα έξυπνα συμβόλαια επιτρέπουν στους συμμετέχοντες να συναλλάσσονται μεταξύ τους χωρίς μια αξιόπιστη κεντρική αρχή. Τα αρχεία συναλλαγών είναι αμετάβλητα, επαληθεύσιμα και διανέμονται με ασφάλεια σε όλο το δίκτυο, δίνοντας στους συμμετέχοντες πλήρη ιδιοκτησία και ορατότητα στα δεδομένα συναλλαγών. Οι συναλλαγές αποστέλλονται και λαμβάνονται από λογαριασμούς Ethereum που έχουν δημιουργηθεί από χρήστες. Ένας αποστολέας πρέπει να υπογράφει συναλλαγές και να ξοδεύει το ether ως κόστος επεξεργασίας συναλλαγών στο δίκτυο. [83] Τα έξυπνα συμβόλαια εκτελούνται σε έναν υπολογιστή που ονομάζεται εικονική μηχανή Ethereum (EVM). Κάθε κόμβος στο δίκτυο έχει ένα τοπικό αντίγραφο του EVM για να επικυρώσει την εκτέλεση της σύμβασης.

Το Ethereum δίνει επιπλέον την δυνατότητα για την δημιουργία αποκεντρωμένων εφαρμογών (dapp). Μια αποκεντρωμένη εφαρμογή είναι μια εφαρμογή που βασίζεται σε ένα αποκεντρωμένο δίκτυο που συνδυάζει ένα έξυπνο συμβόλαιο και μια διεπαφή χρήστη διεπαφής. Στο Ethereum, τα έξυπνα συμβόλαια είναι προσβάσιμα και διαφανή – όπως τα ανοιχτά API – έτσι ώστε το dapp να μπορεί να περιλαμβάνει ακόμη και ένα έξυπνο συμβόλαιο που έχει γράψει κάποιος άλλος. [84]

### 4.2.1 Ether και υποδιαιρέσεις

Μέσω του site Coinmarketcap βλέπουμε πληροφορίες σχετικά με τα tokens και στο συγκεκριμένο link `https://https://coinmarketcap.com/currencies/ethereum/` πληροφορίες για το token ether όπως φαίνεται στην εικόνα 45




Εικόνα 45. Coinmarketcap Ether

Μερικές πληροφορίες περιλαμβάνουν ,την αξία ενός HNT που είναι 1,569.49$ (η τιμή ενδέχεται να έχει απότομες διακυμάνσεις τόσο αυξητικές όσο και φθίνουσες όπως και σε όλα τα κρυπτονομίσματα), την κεφαλαιοποίηση του που είναι περίπου στο 190 δισεκατομμύρια δολάρια και συγκεκριμένα 190,983,162,360 $ επίσης φαίνονται και τα διαθέσιμα ανταλλακτήρια που υποστηρίζουν το HNT. **Οι συγκεκριμένες πληροφορίες δεν αποτελούν επενδυτικές συμβουλές.**

Στον Πίνακα 13 φαίνονται οι υποδιαιρέσεις του ether:

Πίνακας 13. Υποδιαιρέσεις του ether

| Μονάδα | Τιμή wei | wei | ether value |
|---|---|---|---|
| wei | 1 wei | 1 | 10^-18 ETH |
| kwei | 10^3 wei | 1,000 | 10^-15 ETH |
| mwei | 10^6 wei | 1,000,000 | 10^-12 ETH |
| gwei | 10^9 wei | 1,000,000,000 | 10^-9 ETH |
| microether | 10^12 wei | 1,000,000,000,000 | 10^-6 ETH |
| milliether | 10^15 wei | 1,000,000,000,000,000 | 10^-3 ETH |
| ether | 10^18 wei | 1,000,000,000,000,000,000 | 1 ETH |

### 4.2.2 Δομή μιας συναλλαγής στο Ethereum

Μια συναλλαγή στο Ethereum είναι ένα σειριακό δυαδικό μήνυμα που αποτελείται από τα παρακάτω στοιχεία:

- Nonce : Αύξων αριθμός, που εκδόθηκε από τον αρχικό λογαριασμό, που χρησιμοποιείται για την αποτροπή επανάληψης του μηνύματος.
- Gas price : Η τιμή του gas (σε wei) που προτίθεται να πληρώσει ο εντολέας.
- Gas limit : Το μέγιστο ποσό σε gas που προτίθεται να πληρώσει ο εντολέας για τη συναλλαγή.
- Recipient : Η διεύθυνση Ethereum του αποδέκτη.
- Value : Η ποσότητα σε ether που θα σταλεί στον προορισμό.
- Data : Το ωφέλιμο φορτίο δυαδικών δεδομένων μεταβλητού μήκους.
- v,r,s : Τα τρία στοιχεία μιας ψηφιακής υπογραφής του αρχικού λογαριασμού.

### 4.2.3 Εργαλεία για Ethereum

Σε αυτήν την υποενότητα θα γίνει αναφορά μερικών εφαρμογων για το ehtereum που θα χρησιμοποιούμε και ποιο συγκεκριμένα :

- Metamask : Ψηφιακό πορτοφόλι
- Ganache : Δοκιμαστικό περιβάλλον με εικονικά ether.
- Remix : Ανάπτυξη και διαχείριση έξυπνων συμβολαίων για το Ethereum

#### Metamask

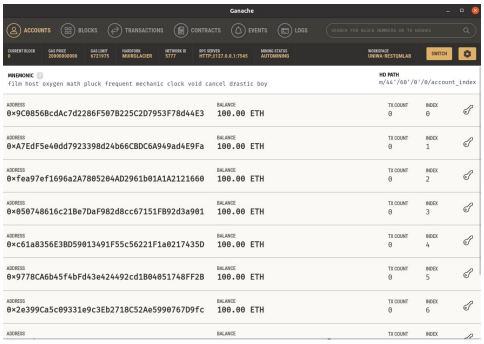
Το MetaMask είναι ένα δωρεάν crypto wallet που εκτελείτε μέσω web ή κινητού και επιτρέπει στους χρήστες να αποθηκεύουν και να ανταλλάσσουν κρυπτονομίσματα, να αλληλεπιδρούν με το οικοσύστημα blockchain Ethereum και να φιλοξενούν μια αυξανόμενη σειρά αποκεντρωμένων εφαρμογών (dApps) [85]. Στο παράρτημα 4 δίνονται οδηγίες για την εγκατάσταση του metamask.

#### Ganache - Truffle Suite

Το Ethereum Ganache είναι ένα τοπικό blockchain στη μνήμη που έχει σχεδιαστεί για ανάπτυξη και δοκιμή. Προσομοιώνει τα χαρακτηριστικά ενός πραγματικού δικτύου Ethereum, συμπεριλαμβανομένης της διαθεσιμότητας ενός αριθμού λογαριασμών που χρηματοδοτούνται με δοκιμαστικά Ether. [86] Στο παράρτημα 4 δίνονται οδηγίες για την εγκατάσταση του Ganache.

### Σύνδεση Metamask και Ganache

Αφού ακολουθήσαμε τα βήματα του παραρτήματος 4 διαθέτουμε 10 ψηφιακά πορτοφόλια από 100 εικονικά ether το καθένα (εικόνα 46)

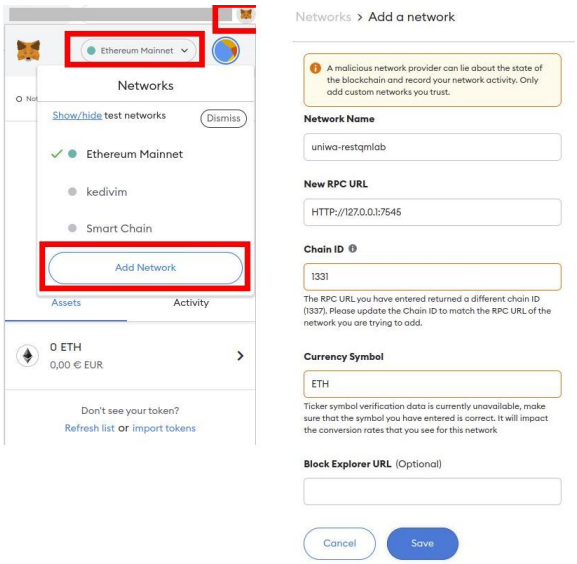



Εικόνα 46. Ganache

Ανοίγουμε τον browser που κάναμε εγκατάσταση το metamask (παράρτημα 4) κάνουμε click στο εικονίδιο metamask μετά Ethereum Mainnet και Add Network. Στην συνέχεια συμπληρώνουμε τα στοιχεία ως εξής:

- Network Name : uniwa-restqmlab (ορίζουμε ένα όνομα που θέλουμε)
- New RPC URL : HTTP://127.0.0.1:7545 (το βρίσκουμε στο περιβάλλον του ganache)
- Chain ID : 1331
- Currency Symbol : ETH

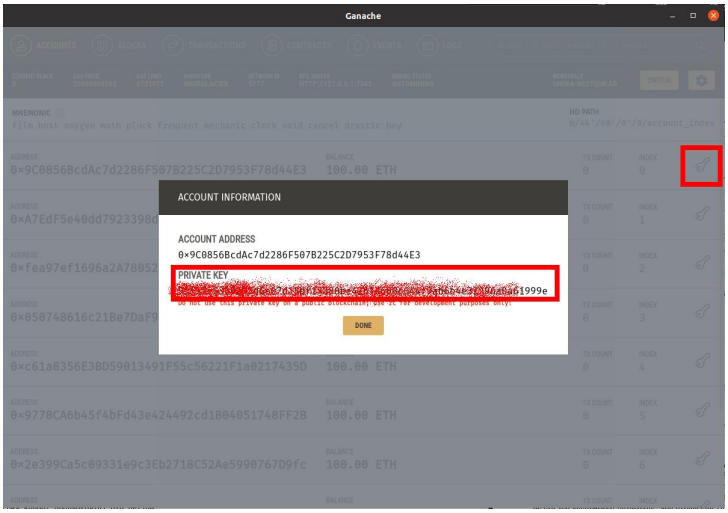
Και μετά save όπως την εικόνα 47.




Εικόνα 47. Save Network

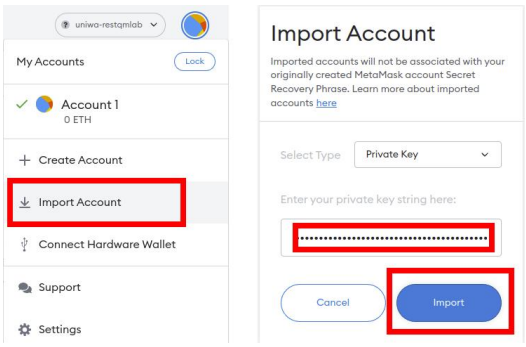
Έχουμε συνδέσει επιτυχώς το Ganache με το Metamask και τώρα θα συνδέσουμε 2 εικονικούς λογαριασμούς.

Από το περιβάλλον του Ganache επιλέγουμε το εικονίδιο κλειδί από τους διαθέσιμους λογαριασμούς και μετά copy to private key του επιλεγμένου λογαριασμού (εικόνα 48)



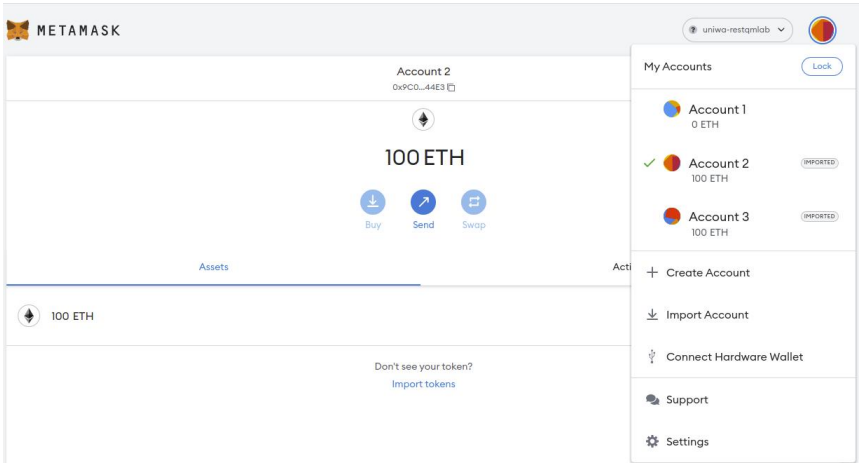

Εικόνα 48. PRIVATE KEY

Μετά από το Metamask επιλέγουμε Import Account κάνουμε paste το private key του επιλεγμένου λογαριασμού και τέλος επιλέγουμε Import (εικόνα 49).




Εικόνα 49. Import Account

Στην συνέχεια εμφανίζονται τα εικονικά ether του λογαριασμού. Με την ίδια λογική προσθέτουμε και αλλους εικονικούς λογαριασμούς (εικόνα 50).

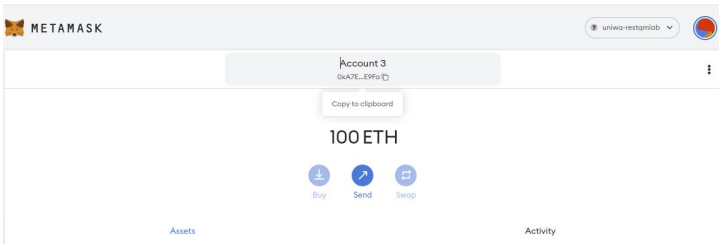



Εικόνα 50. Εικονικά ether

### Metamask : Συναλλαγή

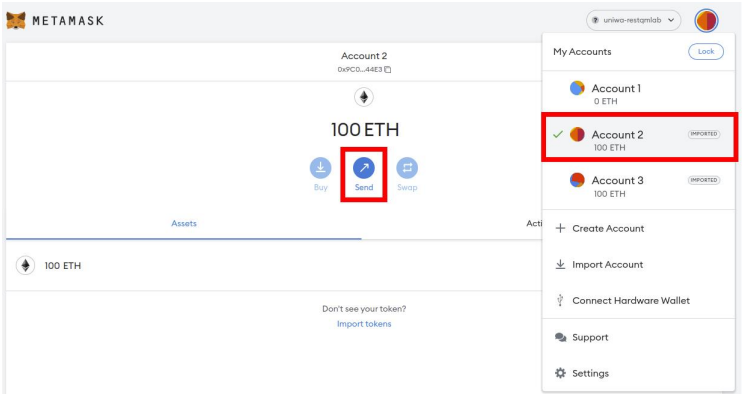
Στην προηγούμενη ενότητα συνδέσαμε το Ganache με το Metamask και προσθέσαμε 2 λογαριασμούς με εικονικά Ether. Σε αυτήν την ενότητα θα κάνουμε μια συναλλαγή και πιο συγκεκριμένα θα στείλουμε εικονικά Ether από τον ένα λογαριασμό του Ganache στον άλλον.

Επιλέγουμε τον λογαριασμό 3 και κάνουμε copy την διεύθυνση του πορτοφολιού (εικόνα 51).



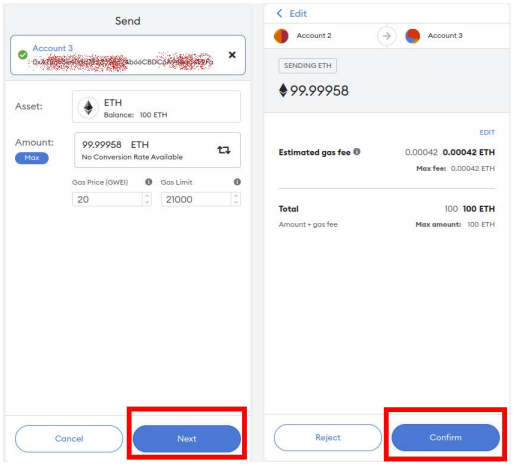

Εικόνα 51. Διεύθυνση πορτοφολιού

Στην συνέχεια επιλέγουμε τον λογαριασμό 2 κάνουμε click στην επιλογή send (εικόνα 52).



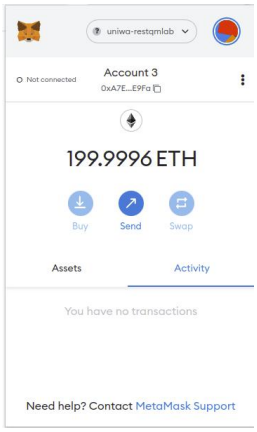

Εικόνα 52. Send

Μετά κάνουμε paste την διεύθυνση του πορτοφολιού, ορίζουμε το ποσό που θα στείλουμε και επιλέγουμε Confirm (εικόνα 53) όπου φαίνεται η τιμή Gas Price και Gas Limit.




Εικόνα 53. Confirm

Η συναλλαγή έγινε κράτησε τέλη 0.00042 ether σε σημερινή αντιστοιχία με ευρώ περίπου 0.64 (εικόνα 54).

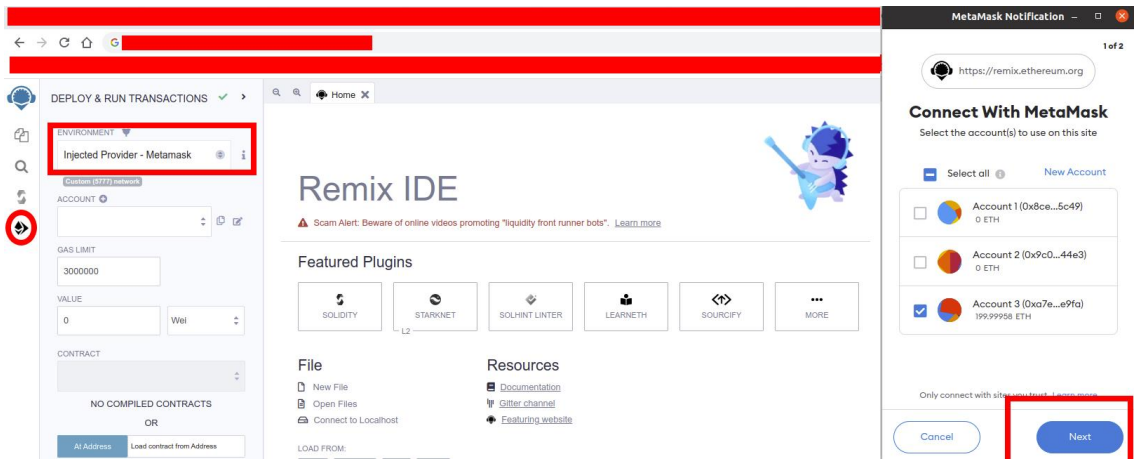



Εικόνα 54. Επιτυχής συναλλαγή

### Remix

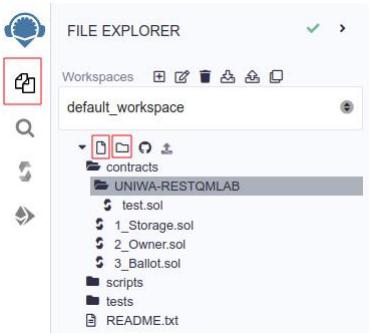
Το Remix, γνωστό ως Remix IDE, είναι ένα Ethereum IDE ανοιχτού κώδικα που χρησιμοποιήσετε για τη σύνταξη, τη μεταγλώττιση και τον εντοπισμό σφαλμάτων κώδικα Solidity. Το Remix IDE χρησιμοποιείται την ανάπτυξη έξυπνων συμβολαίων από χρήστες σε κάθε επίπεδο γνώσης. Δεν απαιτεί εγκατάσταση, προωθεί έναν γρήγορο κύκλο ανάπτυξης και έχει ένα πλούσιο σύνολο προσθηκών με εύχρηστα GUI. Το IDE διατίθεται σε 2 εκδόσεις (web app ή desktop app) και ως επέκταση VSCode. [87], [88]

Ανοίγουμε το link `https://remix.ethereum.org/` , κάνουμε click στο εικονίδιο του ethereum (Deploy run transactions) και επιλέγουμε στο μενού ENVIRONMENT το Inject Provider – Metamask για να συνδέσουμε το Metamask με το account του Ganache. Εμφανίζει μενου το metamask επιλέγουμε τον λογαριασμό που θέλουμε να κάνουμε add, μετά next και connect (εικόνα 55).




Εικόνα 55. Σύνδεση Metamask με Remix IDE

Στην συνέχεια πάμε στο file explorer, δημιουργούμε έναν φάκελο με την ονομασία UNIWA-RESTQMLAB και μέσα στο φάκελο δημιουργούμε αρχεία με τα smart contracts που θέλουμε να υλοποιήσουμε, πατάμε add ένα αρχείο με την ονομασία test.sol (εικόνα 56). Και γράφουμε τον κώδικα 2 για ένα απλό μήνυμα Hello World!

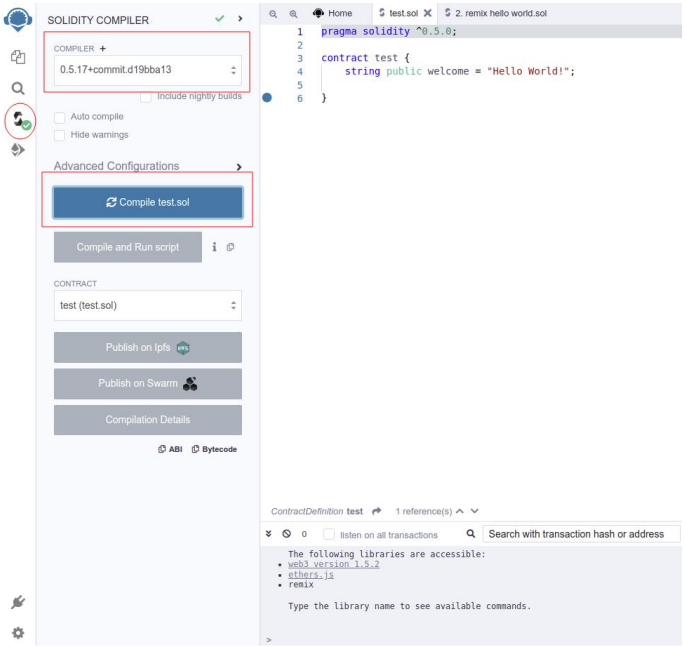



Εικόνα 56. Προσθήκη φακέλων και αρχείων στο Remix IDE

```
1
2 pragma solidity ^0.8.0;
3
4 contract test {
5   string public welcome = "Hello World!";
6
7 }
```

Κώδικας 2. Solidity Hello World!

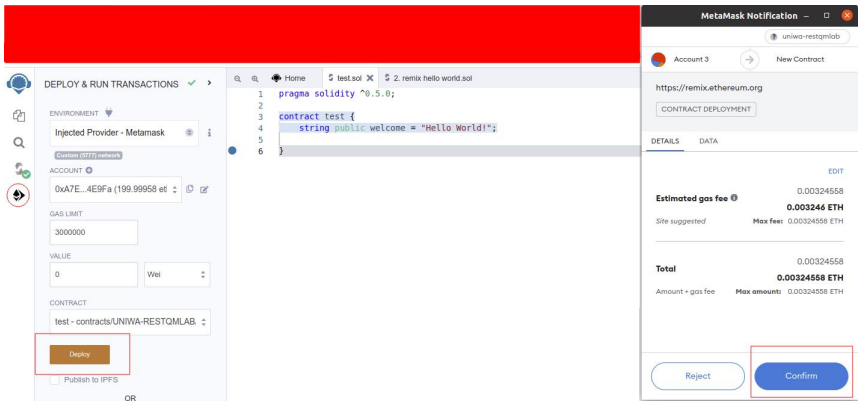
Στην συνέχεια πάμε στο compile επιλέγουμε COMPILER + 0.5.17+commit.d19bba13 και μετά Compile test.sol αν γίνει επιτυχώς το compile εμφανίζει πράσινο βελάκι (εικόνα 57).




Εικόνα 57. Compile test.sol

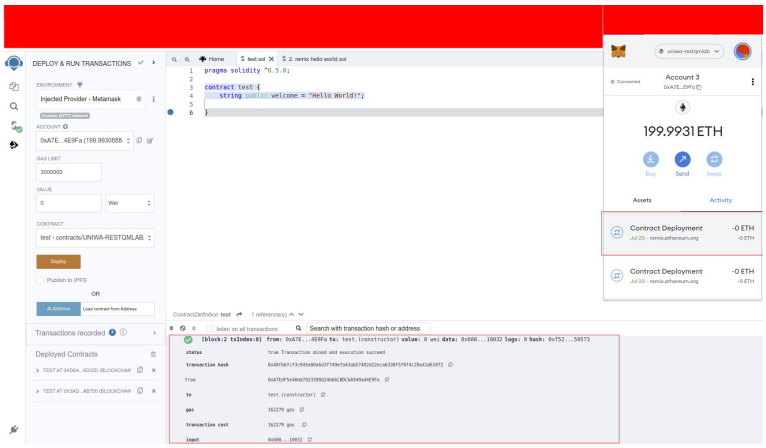
Στην συνέχεια πάμε στο Deploy  run transactions επιλέγουμε deploy. Εμφανίζει μήνυμα με τα εικονικά χρήματα που χρειάζεται για την συναλλαγή και επιλέγουμε Confirm (εικόνα 58). Τα τέλη συναλλαγής είναι 0.003246 ether σε σημερινή αντιστοιχία με ευρώ περίπου 5. Αν θέλουμε να στείλουμε απλά δεδομένα από αισθητήρες λόγο υψηλών τελών δεν είναι προς το παρών καλή επιλογή του Ethereum blockchain. Το Ethereum 2 θα μεταβεί από τον αλγόριθμο proof-of-work στον αλγόριθμο proof-of-stake θα είναι μια μεγάλη αναβάθμιση που θα στοχεύει σε χαμηλά τέλη όπως το Solana Blockchain. Το Solana είναι ανερχόμενο blockchain με χαμηλά τέλη και χρησιμο-

ποιεί έναν καινούργιο αλγόριθμο το proof-of-history άλλα μέχρι στιγμής έχει παρουσιάσει κενό ασφαλείας και έχει πέσει 2 φορές το δίκτυο για μεγάλη χρονική διάρκεια. Για έξυπνα συμβόλαια, dapps το Ethereum είναι καλή λύση λόγο της σταθερότητας και της μεγάλης ασφάλειας που παρέχει. Με την επερχόμενη αναβάθμιση σε Ethereum 2 που ίσως γίνει το 2023 θα πέσουν τα τέλη και θα αναπτυχθούν πολλές αποκεντρωμένες εφαρμογές. Το blockchain Helium είναι ιδανικό για την αποστολή δεδομένων από αισθητήρες, έχει χαμηλά τέλη και θα γίνει ανάλυση στην ενότητα 4.3




Εικόνα 58. Deploy

Στην εικόνα 59 φαίνεται ότι έγινε επιτυχώς η συναλλαγή καθώς και πληροφορίες του block. Η αποκωδικοποίηση συναλλαγής δεν είναι διαθέσιμη για το chainId 1331 επειδή είναι δοκιμαστικό.




Εικόνα 59. Επιτυχής συναλλαγή

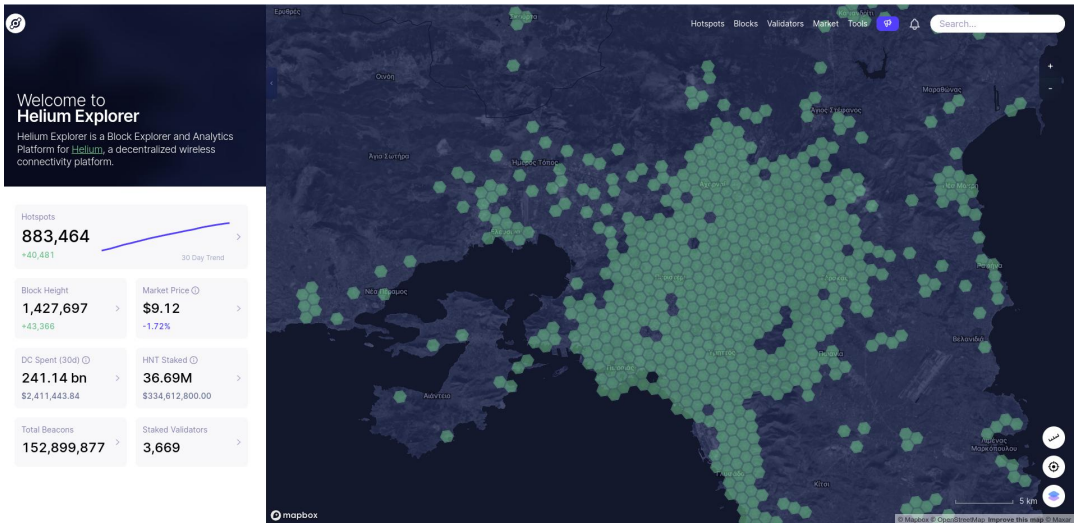
## 4.3 Helium Blockchain

Το Helium Blockchain είναι το μεγαλύτερο δημόσιο αποκεντρωμένο δίκτυο LoRaWAN στο κόσμο, ένα blockchain που επιτρέπει σε συσκευές οπουδήποτε στον κόσμο να συνδέονται ασύρματα στο διαδίκτυο και να εντοπίζονται γεωγραφικά χωρίς την ανάγκη εξοπλισμού δορυφορικής τοποθεσίας που απαιτεί ενέργεια ή ακριβά σχέδια κινητής τηλεφωνίας. Το blockchain λειτουργεί με ένα νέο πρωτόκολλο συναίνεσης, που ονομάζεται Πρωτόκολλο Συναίνεσης Helium (Helium Consensus Protocol), και ένα νέο είδος απόδειξης που ονομάζεται Απόδειξη Κάλυψης Proof-of-Coverage (PoC). [89]

Για την υλοποίηση του IoT συστήματος επιλέχτηκε το Helium Blockchain για τους εξής λόγους :

- Οι LoRa κόμβοι συνδέονται στο Helium.
- Έχει χαμηλά τέλη συναλλαγής. Κατά την αποστολή η λήψη δεδομένων από τον αισθητήρα για κάθε 24 bytes είναι 0.00001 $ Δηλαδή για αποστολή δεδομένων ενός έξυπνου μετρητή (ή αισθητήρα) το ετήσιο κόστος για αποστολή δεδομένων κάθε 15 λεπτά είναι 0.3504 $
- Το Helium διασφαλίζει ότι όλες οι επικοινωνίες στο blockchain έχουν κρυπτογράφηση από άκρο σε άκρο, καθιστώντας το πιο κατάλληλο για ευαίσθητες πληροφορίες.
- Υπάρχει μεγάλη κάλυψη δικτύου από hotspot όπου μπορούμε να συνδέσουμε την IoT συσκευή μας. Αυτό οδηγεί σε μείωση κόστος υλικού αφού δεν απαιτεί να διαθέτουμε hotspot.

Στο link `https://explorer.helium.com/hotspots` μπορούμε να δούμε τα ενεργά hotspot και την κάλυψη παγκοσμίως. Στο σχήμα 60 δίνεται η κάλυψη από hotspots για την περιοχή της Αττικής.




Εικόνα 60. Helium Hotspots στην Αττική

### 4.3.1 Βασικά στοιχεία του Helium δικτύου

Τα Βασικά στοιχεία του Helium δικτύου είναι :

- Το Proof-of-Coverage (PoC) : είναι ο αλγόριθμος που χρησιμοποιεί το Helium και δίνει την δυνατότητα στους miners να αποδείξουν ότι παρέχουν ειλικρινά την ασύρματη κάλυψη για την τοποθεσία που έχουν κατοχυρώσει στα hotspots τους.
- Proof-of-Location : είναι ένα σύστημα που χρησιμοποιεί το WHIP για να ερμηνεύσει τη φυσική γεωγραφική θέση μιας συσκευής χωρίς να απαιτείται ενεργοβόρο και ακριβό υλικό δορυφορικής τοποθεσίας. Οι συσκευές μπορούν να κάνουν ασφαλείς, αμετάβλητες και επαληθεύσιμες αξιώσεις σχετικά με την τοποθεσία τους σε μια συγκεκριμένη χρονική στιγμή που καταγράφετε στο blockchain.
- WHIP : Το WHIP είναι ένα πρωτόκολλο ασύρματου δικτύου ανοιχτού κώδικα και συμβατό με πρότυπα, κατασκευασμένο για συσκευές χαμηλής κατανάλωσης σε μεγάλες περιοχές. Λειτουργεί σε υπάρχοντα τσιπ ραδιοφώνου εμπορευμάτων που διατίθενται από δεκάδες κατασκευαστές που δεν διαθέτουν αποκλειστικές τεχνολογίες ή σχήματα διαμόρφωσης που απαιτούνται.

- Helium Consensus Mechanism : Το πρωτόκολλο συναίνεσης του Helium λειτουργεί με ορισμένες αρχές. Πρώτον, το δίκτυο Helium επιτρέπει στα Hotspots να λειτουργούν σύμφωνα με τους κανόνες συναίνεσης του Helium και τις προδιαγραφές δικτύου δωρεάν συμμετοχή στο δίκτυο. Δεύτερον, το δίκτυο δεν παρέχει κίνητρα για μόχλευση σε παράγοντες όπως το φθηνό ενεργειακό κόστος ή η ανάπτυξη πρόσθετου υλικού στην ίδια τοποθεσία. Τρίτον, το πρωτόκολλο θα πρέπει να έχει την ικανότητα να αντέχει τις βυζαντινές αποτυχίες με τέτοιο τρόπο ώστε να είναι ακόμη εφικτή η συναίνεση εφόσον οι συμμετέχοντες ενεργούν με ειλικρίνεια. Για το σκοπό αυτό, το δίκτυο Helium χρησιμοποιεί το HoneyBadgerBFT, μια παραλλαγή του BFT. Τέλος, τα Hotspot δεν έχουν τη δυνατότητα λογοκρισίας, επιλογής ή απόρριψης συναλλαγών που θα συμπεριληφθούν στο μπλοκ.

- Helium Decentralized Wire Network (DWN) : παρέχει ασύρματη πρόσβαση στο διαδίκτυο για συσκευές μέσω πολλαπλών ανεξάρτητων miners. Καθορίζει επίσης τις προδιαγραφές δικτύου Helium και WHIP με τις οποίες συμμορφώνονται οι συμμετέχοντες στο δίκτυο. Οι δρομολογητές πληρώνουν αυτό το δίκτυο των miners για την αποστολή δεδομένων προς και από το διαδίκτυο και οι miners ανταμείβονται με νέα tokens για την παροχή κάλυψης δικτύου και την παράδοση δεδομένων συσκευής στο Διαδίκτυο. [89], [90]

### 4.3.2 Βασικά στοιχεία του DWN

Το DWN αποτελείται από 3 μέρη, τους συμμετέχοντες, το blockchain και την φυσική υλοποίηση

#### Συμμετέχοντες

Οι συμμετέχοντες αποτελούνται από :

- Συσκευές : στέλνουν και λαμβάνουν κρυπτογραφημένα δεδομένα από το διαδίκτυο χρησιμοποιώντας υλικό συμβατό με το WHIP. Τα δεδομένα που αποστέλλονται από τις συσκευές είναι δακτυλικά αποτυπώματα (fingerprinted) και αυτά αποθηκεύονται στο blockchain.

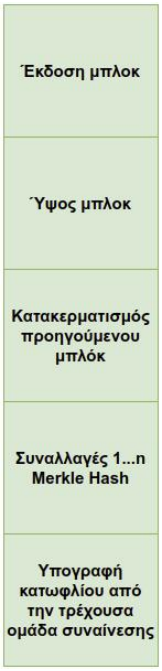
- Miners : παρέχουν στο Helium κάλυψη δικτύου μέσω εξειδικευμένου υλικού που είναι γνωστό ως Hotspots. Αυτά τα hotspots παρέχουν μια γέφυρα μεγάλης εμβέλειας μεταξύ των συσκευών και του διαδικτύου. Οι χρήστες μπορούν να ενταχθούν στο δίκτυο Helium ως miners μέσω της αγοράς ή της κατασκευής ενός hotspot που συμμορφώνεται με το WHIP.

- Routers : είναι εφαρμογές διαδικτύου που αγοράζουν κρυπτογραφημένα δεδομένα συσκευής από τους miners. Μπορούν να πληρώσουν αρκετούς miners για να αποκτήσουν άφθονα αντίγραφα ενός πακέτου για να εντοπίσουν γεωγραφικά μια συσκευή χωρίς τη βοήθεια υλικού δορυφορικής τοποθεσίας. Αυτά τα δεδομένα είναι διάσημα ως Proof-of-Location σε μέρη όπου οι miners είναι πολλοί. Οι δρομολογητές είναι υπεύθυνοι για την επιβεβαίωση στα Hotspots ότι τα δεδομένα της συσκευής παραδόθηκαν στον σωστό προορισμό και ότι ο miner πρέπει να πληρωθεί για την υπηρεσία τους. [89], [90]

#### Blockchain Helium

Το blockchain Helium είναι ένα κατανεμημένο καθολικό που έχει σχεδιαστεί για να παρέχει έναν οικονομικά αποδοτικό τρόπο εκτέλεσης του λογικού πυρήνα εφαρμογής στη λειτουργία ενός DWN, αποθήκευσης αμετάβλητων δακτυλικών αποτυπωμάτων δεδομένων συσκευής και παροχής συστήματος συναλλαγών. Το δίκτυο Helium είναι μια αμετάβλητη λίστα συναλλαγών μόνο σε παράρτημα που επιτυγχάνει συναίνεση χρησιμοποιώντας το Πρωτόκολλο Συναίνεσης Helium Οι χρήστες εσωτερικοί και εξωτερικοί στο DWN έχουν πρόσβαση στο blockchain, το οποίο είναι ένα

νέο πρωτόκολλο που δημιουργήθηκε από την αρχή ειδικά για το DWN. Το blockchain αποτελείται από μπλοκ που περιέχουν μια κεφαλίδα και μια λίστα συναλλαγών. Στην εικόνα 61 δίνεται η δομή ενός δεδομένου μπλόκ.




Εικόνα 61. Δομή ενός δεδομένου μπλόκ Helium

Το proof-of-coverage είναι παράγοντας ζωτικής σημασίας για το δίκτυο και γιαυτό οι miners υποβάλλουν τα αποδεικτικά τους σε τακτά χρονικά διαστήματα. Ολοι οι Miners έχουν μια βαθμολογία, η οποία μειώνεται με την πάροδο του χρόνου και ενισχύεται με την υποβολή αποδείξεων κάλυψης στο blockchain.

Σε μια σταθερή περίοδο επιλέγεται μια ομάδα συναίνεσης HoneyBadgerBFT από τους miners με την υψηλότερη βαθμολογία. Για την συγκεκριμένη περίοδο όλες οι συναλλαγές κρυπτογραφούνται και υποβάλλονται στην ομάδα συναίνεσης για την ενσωμάτωση στο blockchain. Η ομάδα συναίνεσης είναι υπεύθυνη για την αποκρυπτογράφηση συναλλαγών χρησιμοποιώντας αποκρυπτογράφηση κατωφλίου, τη συμφωνία σχετικά με την εγκυρότητα και την ταξινόμηση των συναλλαγών, τη διαμόρφωση τους σε μπλοκ και την προσάρτησή τους στο blockchain για το οποίο τα μέλη της ομάδας συναίνεσης λαμβάνουν ανταμοιβή. Δεδομένου ότι η ομάδα συναίνεσης επικυρώνει τις συναλλαγές χωρίς να χρειάζεται να παρέχει ένα σχετικό block-proof (πέρα από μια υπογραφή κατωφλίου), πρακτικά δεν υπάρχει χρόνος διακανονισμού και η απόδοση των συναλλαγών είναι εξαιρετικά υψηλή σε σύγκριση με μια αλυσίδα μπλοκ Nakamoto Consensus όπως το Bitcoin ή το Ethereum.

### Φυσική υλοποίηση

Το δίκτυο Helium είναι επίσης ένα ασύρματο δίκτυο. Οι συμμετέχοντες στο δίκτυο Helium μπορούν να θεωρηθούν ως εξής:

- WHIP : Το δίκτυο Helium χρησιμοποιεί ένα νέο ασύρματο πρωτόκολλο (ανοιχτού κώδικα), που ονομάζεται WHIP. Το WHIP είναι ένα πρωτόκολλο ασύρματου δικτύου μεγάλης εμβέλειας, χαμηλής κατανάλωσης. Το WHIP χρησιμοποιεί ισχυρή κρυπτογραφία δημόσιου κλειδιού και ο έλεγχος ταυτότητας πραγματοποιείται χρησιμοποιώντας την αλυσίδα μπλοκ

Helium και τα δεδομένα κρυπτογραφούνται από άκρο σε άκρο μεταξύ της συσκευής και του αντίστοιχου δρομολογητή που φιλοξενείται στο Διαδίκτυο.

- Hotspots : Είναι οι miners, φυσικές συσκευές δικτύου που παρέχουν ασύρματη κάλυψη ευρείας περιοχής και συμμετέχουν στο δίκτυο Helium. Τα hotspots μεταδίδουν δεδομένα μεταξύ δρομολογητών στο διαδίκτυο και συσκευών ενώ παράγουν αποδείξεις κάλυψης (PoC) για το δίκτυο Helium. Τα hotspots μπορούν να συνεργάζονται και να εντοπίζουν γεωγραφικά τις Συσκευές χρησιμοποιώντας το δίκτυο Helium χωρίς πρόσθετο απαιτούμενο υλικό. Κάθε Hotspot μπορεί να υποστηρίξει χιλιάδες συνδεδεμένες Συσκευές και να παρέχει κάλυψη σε μεγάλη εμβέλεια.

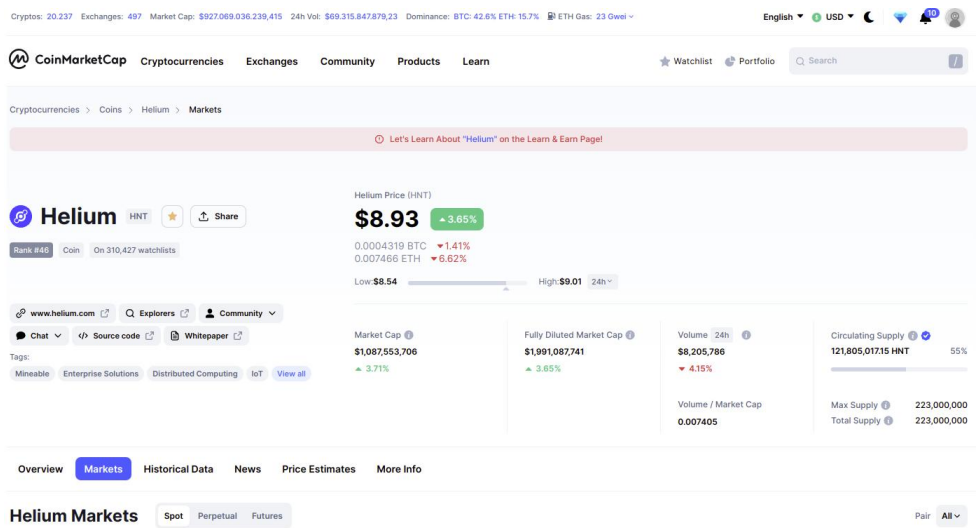
- Οι συσκευές υπάρχουν με τη μορφή προϊόντων υλικού που περιέχουν έναν ραδιοφωνικό πομποδέκτη, συμβατό με WHIP και επικοινωνούν με hotspots στο δίκτυο Helium. Οι κόμβοι loRa που χρησιμοποιήσαμε στο chirpstack μπορούν επιτυχώς να συνδεθούν στο Helium. Το WHIP έχει σχεδιαστεί για να διευκολύνει τη μετάδοση και τη λήψη δεδομένων χαμηλής ισχύος, έτσι συνήθως οι συσκευές υπάρχουν με τη μορφή αισθητήρων που τροφοδοτούνται από μπαταρίες που μπορούν να λειτουργήσουν για αρκετά χρόνια χρησιμοποιώντας τυπικές μπαταρίες.

### 4.3.3 Helium (HNT) Token

Το token HNT είναι το κρυπτονόμισμα του blockchain Helium. Το πρώτο token εμφανίστηκε με το μπλοκ genesis στις 29 Ιουλίου 2019. Με το HNT token επιβραβεύονται οι miners για την παροχή κάλυψης ασύρματου δικτύου, τις χρεώσεις από συναλλαγές καθώς και για την επικύρωση της ακεραιότητας του δικτύου Helium.

Εκτός από τη χρήση του token HNT, το Helium χρησιμοποιεί ένα άλλο token, το οποίο δεν μπορεί να ανταλλαγεί και το οποίο οι χρήστες μπορούν να χρησιμοποιήσουν για να πληρώσουν τα τέλη συναλλαγών, τα Data Credits (Πιστωτικές Μονάδες Δεδομένων) αναφέρονται σαν DC.

Μέσω του site Coinmarketcap βλέπουμε πληροφορίες σχετικά με τα tokens και στο συγκεκριμένο link https://coinmarketcap.com/currencies/helium/markets/ πληροφορίες για το token HNT όπως φαίνεται στην εικόνα 62.




Εικόνα 62. Coinmarketcap HNT

Μερικές πληροφορίες περιλαμβάνουν ,την αξία ενός HNT που είναι 8.93$ (η τιμή ενδέχεται να έχει απότομες διακυμάνσεις τόσο αυξητικές όσο και φθίνουσες όπως και σε όλα τα κρυπτονο-

μίσματα), την κεφαλαιοποίηση του που είναι περίπου στο 1 δισεκατομμύριο δολάρια και συγκεκριμένα 1,078,920,055 $ επίσης φαίνονται και τα διαθέσιμα ανταλλακτήρια που υποστηρίζουν το HNT. **Οι συγκεκριμένες πληροφορίες δεν αποτελούν επενδυτικές συμβουλές**.

### 4.3.4 Εγκατάσταση Helium Hotspot

Σε αυτήν την ενότητα θα γίνει εγκατάσταση ενός Helium Hotspot. Επιλέχθηκε το SenseCAP M1 (εικόνα 63) λόγο διαθεσιμότητας για αγορά και της μεγάλης μνήμης RAM 8 GB. Θα ενταχθούμε στο blockchain με το συγκεκριμένο Hotspot και γίνουμε miner παρέχοντας ασύρματη κάλυψη δικτύου και μια γέφυρα μεγάλης εμβέλειας μεταξύ του κόμβου (κόμβων) και του διαδικτύου.


Εικόνα 63. SenseCAP M1

Τα χαρακτηριστικά του SenseCAP M1 δίνονται παρακάτω:

- Επεξεργαστής : Raspberry Pi 4 (Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz)
- RAM : 8GB
- Αποθήκευση : 64GB MicroSD Card
- EU868 Version: 863MHz 870MHz ,AS923 Version: 902MHz 928MHz,US915 Version: 902MHz 928MHz
- Ευαισθησία : -125dBm @125KHz/SF7, -139dBm @125KHz/SF12
- Μετάδοση ισχύος (TX Power) : Έως 26 dBm
- Κέρδος κεραίας: US915: 2.6 dBi ,AS923: 2.6 dBi,EU868: 2.8 dBi
- Αντίσταση κεραίας: 50 Ohm
- Μοτίβο ακτινοβολίας κεραίας: Omni-Directional
- Wi-Fi : 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless
- Bluetooth : Bluetooth 5.0, BLE
- LoRaWAN : Supports Class A, C

- Τάση εισόδου: DC 5V – 3A
- Διεπαφές : USB Type-C (Power Supply), 1 Ethernet RJ45 ,1 RP-SMA Female Antenna Connector
- Θερμοκρασία λειτουργίας : 0°C έως 50°C
- Τάση εισόδου : 0% – 90% (χωρίς συμπύκνωση)
- Διάχυση θερμότητας : Περίβλημα αλουμινίου, ψύκτρα και ανεμιστήρας ψύξης
- IP Grade : IP20
- Πιστοποιήσεις : FCC / CE / RCM / ROHS / UKCA
- Ασφάλεια και αξιοπιστία: Ενσωματωμένο κρυπτοτσιπ ECC608, έλεγχος ταυτότητας υψηλής ασφάλειας και αξιόπιστη συνδεσιμότητα.

**Η εγκατάσταση του SenseCAP είναι πολύ απλή ακολουθώντας τα παρακάτω βήματα:**

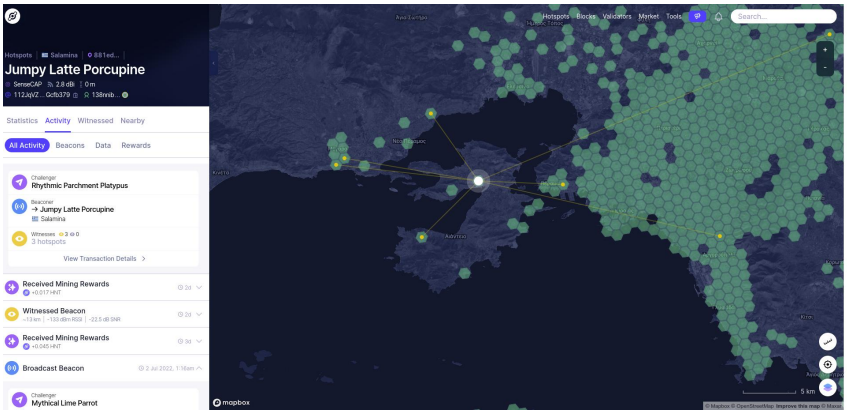
- Κατεβάζουμε από το GooglePlay ή AppStore την εφαρμογή Helium App και δημιουργούμε πορτοφόλι (Helium Wallet)
- Συνδέουμε την κεραία και το ρεύμα στο hotspot
- Πατάμε στο κουμπί στο πίσω μέρος του hotspot μέχρι να αναβοσβήνει το led (περίπου 5 δευτερόλεπτα)
- Επιλέγουμε από την εφαρμογή Set up Hotspot το Sensecap M1
- Επιλέγουμε το wifi βάζουμε το password.

Η διεύθυνση του πορτοφολιού που έχω δημιουργήσει είναι η :
138nnibtuUJESMMKUYZ9aR8vVNfxRBNvRtVgvdVbiFfcesvx6Bo
και πηγαίνοντας στο site `https://explorer.helium.com/` βλέπουμε την διαθεσιμότητα των hotspots και βάζοντας την διεύθυνση του πορτοφολιού μας, εμφανίζει το hotspot μας (ή τα hotspots μας αν έχουμε πάνω από 1). Στην εικόνα 64 φαίνεται το hotspot που δημιουργήσαμε (Από την στιγμή της ενεργοποίησης απαιτείται 24-48 ώρες μέχρι το hotspot να κατεβάσει το blockchain και να ενεργοποιηθεί).


Εικόνα 64. Hotspot Jumpy Latte Porcupine

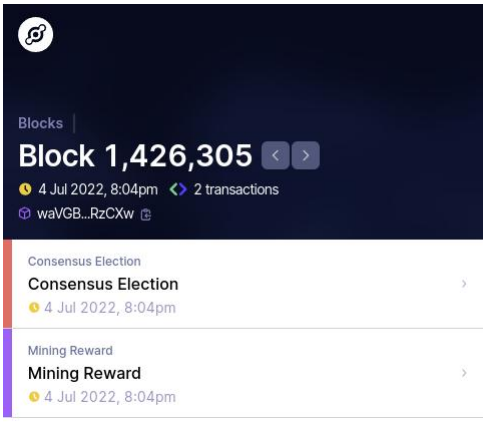
Στην εικόνα 65 φαίνεται η οπτική επαφή με άλλα hotspot και τα minings rewards που κερδίζουμε από τον αλγόριθμο Proof-of-Coverange για την κάλυψη του ασυρμάτου δικτύου, ανταλλάζοντας beacons. Το Beacon είναι ένα πακέτο PoC που στέλνει ο πομπός ενός hotspot (Challengee),

χωρίς να στοχεύει κάποιον συγκεκριμένο δέκτη hotspot. Ουσιαστικά, ένα Hotspot αναζητά και έως και 25 Hotspots που λαμβάνουν το Challengee και αναφέρονται ως Witness. Beacons χρησιμοποιεί και το wifi στο router του σπιτιού μας που περιέχουν πληροφορίες όπως το ssid και το password για να συνδεθούν οι συσκευές. Όπως ισχύει και σε όλα τα ασύρματα δίκτυα όσο βρίσκετε η κεραία του πομπού σε μεγαλύτερο υψόμετρο υπάρχει μεγαλύτερη κάληψη. **Οι συγκεκριμένες πληροφορίες δεν αποτελούν επενδυτικές συμβουλές.**




Εικόνα 65. Witnesses

Μέσω της εφαρμογής Helium στο κινητό επιλέγοντας τα Mining Rewards και κάνοντας click στο view block βλέπουμε το block στο blockchain όπως φαίνεται στην εικόνα 66

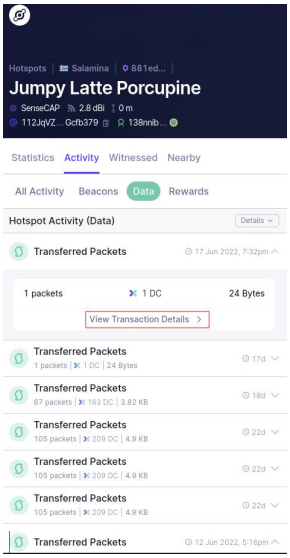



Εικόνα 66. View Block

- Consensus Election : Στην εφαρμογή του Helium, μια συναινετική ομάδα εκλεγμένων Validators (Επικυρωτών) λαμβάνει κρυπτογραφημένες συναλλαγές ως εισροές και προχωρά στην επίτευξη κοινής συμφωνίας σχετικά με την παραγγελία αυτών των συναλλαγών προτού σχηματίσει ένα μπλοκ και προσθέσει στο blockchain. [91]

- Mining Rewards : Το συνολικό ποσό του κέρδους του block από το mining από όλα τα hotspot.

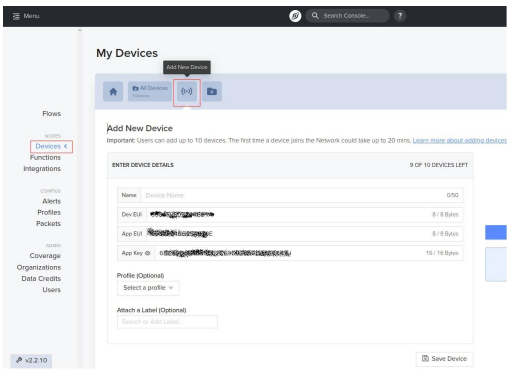
Στο μενού data μπορούμε να δούμε τις συναλλαγές του hotspot μας που αφορούν δεδομένα από κόμβους και στο υπομενού View Transaction Details λεπτομέρειες της συναλλαγής (εικόνα 67).




Εικόνα 67. Hotspot Activity (Data)

### 4.3.5 Προσθήκη κόμβου στο δίκτυο Helium

Για να ενεργοποιήσουμε τον κόμβο (τους κόμβους) ανοίγουμε την σελίδα https://console.helium.com/ επιλέγουμε στο menu devices και μετά Add New Device όπως φαίνεται στην εικόνα 68.




Εικόνα 68. Add New Device

Η ενεργοποίηση ακόλουθα την μέθοδο OTAA. Επιλέγουμε Save Device για να αποθηκεύσουμε την συσκευή.

Στο κεφάλαιο της υλοποίησης θα δωθούν πληροφορίες και οι κώδικες για την αποστολή και την λήψη των δεδομένων του IoT συστήματος.

## 4.4 Secure Element

Το Secure Element (SE) είναι ένας προγραμματιζόμενος μικροελεγκτής που είναι ανθεκτικός σε παραβιάσεις και παρέχει ένα αξιόπιστο περιβάλλον εκτέλεσης και αποθήκευσης. Γενικα τα SE είναι μικρά σε μέγεθος (25 $mm^2$) και έχουν σχεδιαστεί ώστε να εξασφαλίζουν λειτουργίες ασφάλειας όπως, συναρτήσεις κατακερματισμού-digest functions (SHA1, SHA2, MD5, κλπ.) και κρυπτογραφικές λειτουργίες (ECC,RSA, AES κλπ.). [92]

Η εφαρμογή SE περιλαμβάνει πολλούς διαφορετικούς τομείς. Χρησιμοποιούνται στο IoT (για

την εγγύηση της προέλευσης δεδομένων), για τη δημιουργία TLS tunnel, για αναγνώριση, για πιστοποίηση/επαλήθευση και εγγραφές προστασίας από παραβιάσεις.

[93], [94] Τα secure elements τα συναντάμε στους παρακάτω τύπους:

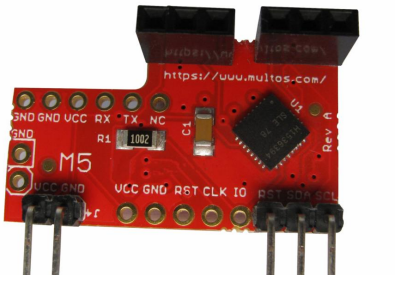
- UICC : κοινώς γνωστή ως κάρτα SIM
- Ως ενσωματωμένο (embedded) SE (eSE).
- Υποδοχή κάρτας SD.
- Ως υπηρεσία cloud. [95]

Η ασφαλής από τη σχεδίαση προσέγγισή του το καθιστά απαραβίαστο όχι μόνο για τα δεδομένα που διατηρεί ή το πρόγραμμα που εκτελεί, αλλά και για το σύστημα στο σύνολό του όπου λειτουργεί ως πρόσθετο HSM (Hardware Security Module). [94]

Οι SE είναι πιστοποιημένες σύμφωνα με τα διεθνή πρότυπα Κοινών Κριτηρίων - Επίπεδο Διασφάλισης Αξιολόγησης (CC-EAL) από EAL1 (χαμηλότερο) έως EAL7 (υψηλότερο). Υπάρχουν ορισμένες SE (κάτω από EAL7+) που λέγεται ότι έχουν διασφάλιση αξιολόγησης υψηλότερη από το επίπεδο EAL7. [93], [94] Συνήθως οι SE έχουν λιγότερο από 1 MB ROM (200-500 KB) και λιγότερο από 15 KB RAM (2-5 KB δημόσια RAM, 8-10 KB δυναμική RAM). Η επικοινωνία των SE πραγματοποιείται μέσω APDU (Application Protocol Data Unit ISO-7816), μέσω σειριακής/NFCδιεπαφής ,SPI (Serial Peripheral Interface) και I2C (Inter-Integrated Circuit).

### 4.4.1 Secure Element : Multos Trust Core

Το MULTOS Trust Core (εικόνα 69) είναι ένας ενσωματωμένος μικροελεγκτής υψηλής ασφάλειας που παρέχει μια μονάδα ασφαλείας υλικού για έξυπνες και συνδεδεμένες συσκευές. Προσφέρει προστασία υλικού από την αρχή της εμπιστοσύνης (Root of Trust (RoT)) καθώς δημιουργεί και προστατεύει κλειδιά και εκτελεί κρυπτογραφικές λειτουργίες μέσα στο ασφαλές περιβάλλον του.Συνδέεται με Raspberry pi και Arduino. [96], [97]




Εικόνα 69. Multos Trust Core

Το Multos Trust Core υποστηρίζει τις εξής λειτουργίες :

- Δημιουργία & αποθήκευση κλειδιού RSA
- Δημιουργία και επαλήθευση υπογραφής RSA
- Κρυπτογράφηση/αποκρυπτογράφηση RSA
- Δημιουργία & αποθήκευση κλειδιού ECC
- Συμφωνία κλειδιού ECC Diffie Hellman
- Δημιουργία και επαλήθευση υπογραφής ECC DSA

- TLS/DTLS 1.2
- Δημιουργία κλειδιών AES
- Κρυπτογράφηση και αποκρυπτογράφηση σε λειτουργία AES CBC & GCM
- Πραγματική παραγωγή τυχαίων αριθμών με βάση το υλικό (TRNG)
- Κατακερματισμός SHA-1 και SHA-256
- HMAC με βάση SHA-1 και SHA-256
- Λειτουργίες διαχείρισης κλειδιών (παραγωγή, εξαγωγή,εισαγωγή, συγχώνευση) [96], [97]

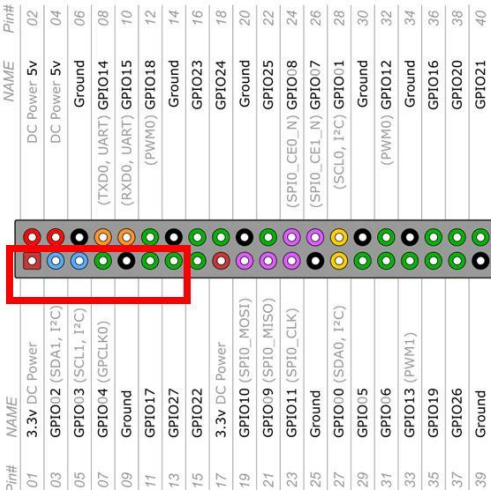
Το Trust Core χρησιμοποει MULTOS OS την έκδοση 4.5.3 (M5-P22), τον Secure μικροελεγκτή Infineon SLE78CUFX5000PHM, FREE RAM 9.900 byte, κοινόχρηστα μεταξύ MULTOS και εφαρμογών. Η επικοινωνία είναι μέσω I2C [97] Τα χαρακτηριστικά του Trust Core δίνονται στον Πίνακα 14.

### 4.4.2 Εγκατάσταση Multos Trust Core στο Raspberry Pi

Σε αυτήν την ενότητα θα γίνει εγκατάσταση του Multos Trust Core για raspberry pi (Ίδια μέθοδο για το pi και το pizero). [98]

**Hardware:**

Η επικοινωνία του Raspberry με το Multos Trust Core πραγματοποιείτε μέσω I2C και η σύνδεση γίνεται εύκολα όπως φαίνεται στην εικόνα 70 μέσω των GPIO.




Εικόνα 70. Σύνδεση Multos Trust Core στις GPIO του pi

- pin Vcc(+) στην GPIO 3,3v
- pin SDA στην GPIO02 (SDA, I2C)
- pin SCL στην GPIO03 (SCL1, I2C)
- pin Groud στην GPIO Ground
- pin RST στην GPIO017
- pin SCL στην GPIO027

Πίνακας 14. Χαρακτηριστικά Multos Trust Core [97]

| **Χαρακτηριστικά της συσκευής** | |
|---|---|
| MULTOS OS | MULTOS 4.5.3 |
| Micro-controller | Ασφαλές λειτουργικό σύστημα: MULTOS v4.5.3 (M5-P22)<br>Ασφαλής μικροελεγκτής: Infineon SLE78CUFX5000PHM 16 bit<br>με Integrity Guard με ολοκληρωμένη ανίχνευση σφαλμάτων,<br>διπλή CPU αυτοελέγχου και πλήρως κρυπτογραφημένη διαδρομή δεδομένων<br>συμπεριλαμβανομένου κρυπτογραφημένου υπολογισμού στην CPU. |
| **Κρυπτογράφηση** | |
| Κρυπτογραφία εφαρμογών | RNG, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, DES, 3DES,<br>AES, SEED, RSA (up to 4096 bit keys), ECC (up to 521 bit curves) |
| **Υποστηριζόμενα γενικά χαρακτηριστικά** | |
| AWS support | AWS IoT Greengrass running on Raspbian |
| Azure IoT Hub support | Azure IoT Hub running on Raspbian |
| Υποστήριξη προγραμματισμού | TLS 1.2 API (Προγραμματισμός C/C++/Python 3)<br>PKCS#11 v2.40 (Προγραμματισμός C/C++)<br>Trust Core χαμηλού επιπέδου API<br>Προσαρμοσμένες ασφαλείς εφαρμογές MULTOS (προγραμματισμός C) |
| Συμβατότητα | Σχεδιασμένο και δοκιμασμένο με πλακέτες Raspberry Pi™ Model 3B (inc.3B+)<br>και Arduino συμβατές με τη διάταξη κεφαλίδας UNO R3.<br>Είναι δυνατή η χρήση με άλλες μάρκες και μοντέλα πλακετών<br>(για παράδειγμα RPi Model 4) |
| ISO 7816 interface | T=0, T=1, up to 447k |
| Αντικατάσταση εφαρμογής | Δυνατότητα αντικατάστασης εφαρμογών με ένα μόνο ALC,<br>η νέα εφαρμογή μπορεί να κληρονομήσει δεδομένα από την<br>εφαρμογή που αντικαταστάθηκε |
| Υποστήριξη εφαρμογής | Υποστηρίζονται όλα τα MULTOS M4-P18 και M5-P19 primitives,<br>νέα ενσωματωμένα primitives, υποστηρίζεται βελτιστοποιημένο<br>σύνολο εντολών |
| Reset pin | Reset pin for chip reset |
| Serial IO interface | Ενιαία σειριακή θύρα μετάδοσης/λήψης έως 57.600 baud |
| I2C interface | Single master and slave port |
| Επεξεργασία σφαλμάτων | Δυνατότητα για εφαρμογές να επεξεργάζονται σφάλματα συστήματος και<br>να παύουν ή να επανεκκινούν το MULTOS |
| On-chip debugging | Πλήρης εντοπισμός σφαλμάτων στο chip όταν βρίσκεται σε λειτουργία εντολών<br>και χρήση του περιβάλλοντος εντοπισμού σφαλμάτων SmartDeck Eclipse<br>(απαιτείται USB/Serial adapter) |
| Δημόσιο μέγεθος | 3,200 bytes |
| Free RAM | 9.900 byte, κοινόχρηστα μεταξύ MULTOS και εφαρμογών<br>(session data and stack) |
| Free NVM for applications | Τουλάχιστον 250 χιλ |
| Delays | Λειτουργία καθυστέρησης με προαιρετικό jitter |
| Timers | Οκτώ μετρητές αντίστροφης μέτρησης και<br>οκτώ χρονόμετρα αντίστροφης μέτρησης |
| Watchdog | One hardware watchdog |
| Πολλαπλοί τομείς ισχύος | Ξεχωριστό Vcc, GPIO και ISO power domains που<br>υποστηρίζουν λειτουργία εξαιρετικά χαμηλής ισχύος<br>όταν χρησιμοποιείτε το ISO 14443 interface. |
| Αντίμετρα ασφαλείας | Εκτεταμένα αντίμετρα ασφαλείας υλικού και λογισμικού για την προστασία<br>του κώδικα και των δεδομένων της εφαρμογής |

**Software:**

Ενεργοποιούμε το I2C interface στο raspberry ακολουθώντας τα παρακάτω βήματα :

- στο terminal εκτελούμε την εντολή sudo raspi-config
- Επιλέγουμε την επιλογή (3) interface option

- στην συνέχεια I5 I2C
- Would you like the ARM I2C interface to be enabled? Επιλέγουμε YES
- μετά finish

Στην συνέχεια δημιουργούμε έναν φάκελο με το όνομα trust_core με την εντολή :

```
mkdir trust_core
```

Πάμε στον φάκελο trust_core :

```
cd trust_core
```

Κατεβάζουμε το λογισμικό για να γίνει εγκατάσταση του multos με την παρακάτω εντολή:

```
wget https://multos.com/wp-content/
uploads/2021/02/trustcore.gz
```

Αποσυμπιέζουμε με την εντολή :

```
tar -xf trustcore.gz
```

και μετά εκτελούμε το install script :

```
sudo bash install
```

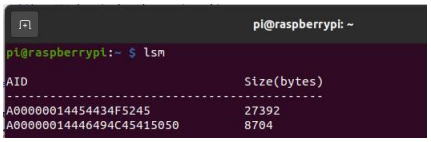
Αυτό εγκαθιστά τα ακόλουθα στοιχεία :

- Πλήρες i2c driver, multosI2CInterface (ρυθμίζεται να εκτελείται αυτόματα κατά την εκκίνηση) και libmultosio.so
- Βιβλιοθήκη PKCS#11 και περιλαμβάνει αρχεία, libmultos-cryptoki.so, pkcs11.h, pkcs11f.h και pkcs11t.h
- Εργαλείο διαχείρισης κλειδιών, p11keyman
- Εργαλείο δημιουργίας κλειδιών, p11keygen
- Εργαλείο διαχείρισης PIN, p11pinman
- Application loader, loadm
- Εργαλείο διαγραφής εφαρμογής, διαγραφή
- Κατάλογος εφαρμογών, delete

Μετά εγκαθιστούμε το πακέτο libncurses5 με την εντολή :

```
sudo apt-get install libncurses5
```

Με την εντολή lsm ελέγχουμε την εγκατάσταση του driver:
Η εγκατάσταση του driver ολοκληρώθηκε επιτυχώς. [98]




Εικόνα 71. check the driver installation

## 4.5 Θέματα ασφάλειας και χρήση του Multos Trust Core

Το Blockchain από μόνο του δεν είναι σε θέση να εξασφαλίσει πλήρως μια συναλλαγή γιατί εγγυάται μόνο την αναλλοίωτη κατάσταση των δεδομένων, ενώ στις περισσότερες περιπτώσεις τα δεδομένα πρέπει να είναι ασφαλή στο σημείο παραγωγής. Προκειμένου να αντιμετωπιστεί το κενό ασφαλείας, το ωφέλιμο φορτίο από τον κόμβο θα κρυπτογραφηθεί με κρυπτογράφηση AES-CBC χρησιμοποιώντας το Secure Element : Multos Trust Core προτού μεταδοθεί στο δίκτυο Helium.

## 4.6 Ευπάθεια και αντιμετώπιση για τον κόμβο LoRa

Όπως είχαμε αναφέρει στο κεφάλαιο 3 και στην ενότητα 3.3.2 Ενεργοποίηση OTAA κατά την ενεργοποίηση με την μέθοδο OTAA πραγματοποιείτε μια διαδικασία σύνδεσης με αίτημα σύνδεσης (Join-request) και ανταλλαγή μηνυμάτων σύνδεσης αποδοχής (join-accept) για κάθε νέα σύνδεση. Σύμφωνα με το μήνυμα join-accept οι τελικές συσκευές λαμβάνουν τα νέα κλειδιά συνεδρίας (NwkSkey και AppSKey).

Ο χώρος που αποθηκεύονται τα κλειδιά NwkSkey και AppSKey στις τελικές συσκευές είναι πιθανό να μην είναι ασφαλείς και αν κάποιος κακόβουλος χρήστης καταφέρει να “κλέψει” τα συγκεκριμένα κλειδιά να καταφέρει να αποκρυπτογραφήσει το ωφέλιμο φορτίο. Όσο αφορά την πύλη Helium δηλαδή το hotspot sensecap m1 έχει ενσωματωμένο τσιπ κρυπτογράφησης ECC608 και τα κλειδιά αποθηκεύονται σε αφαλές χώρο.

Σε αυτήν την περίπτωση για να αντιμετωπιστεί η συγκεκριμένη ευπάθεια είναι καλό τα κλειδία στους κόμβους να αποθηκεύονται σε Secure Element για να μην μπορούν να υποκλαπούν.

Ανατρέχοντας στην βιβλιοθήκη του LoRa Node pHAT και συγκεκριμένα για την βιβλιοθήκη του chip RAK811 βλέπουμε στα αρχεία cli.py και RAK811.py ότι τα κλειδία φορτώνονται στην EEPROM του LoRa Node pHAT χρησιμοποιώντας τη μέθοδο set_config(). Το LoRa Node pHAT έχει κάποιο κρυπτογραφικό κώδικα (πιθανώς AES) ενσωματωμένο στο firmware της και κάνει την κρυπτογράφηση εσωτερικά όταν καλεί τη μέθοδο send(); Σε αυτήν την περίπτωση δεν μπορεί να χρησιμοποιηθεί secure element για την αποθήκευση των κλειδιών.

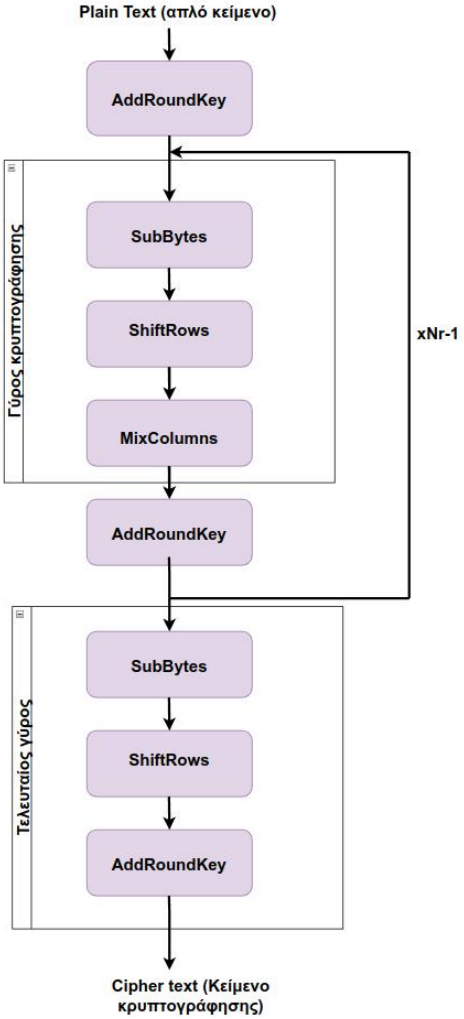
Το RAK811 πιθανώς δεν παρέχει ασφαλές περιβάλλον για την αποθήκευση ή τη χρήση αυτών των κλειδιών. Για να αντιμετωπίσουμε την συγκεκριμένη ευπάθεια προτείνετε με την χρήση του Multos να κρυπτογραφηθεί το ωφέλιμο φορτίο πριν σταλεί στο Helium με κρυπτογράφηση AES-CBC ώστε να μην βασιζόμαστε στην αδύναμη ασφάλεια του RAK811. Το ωφέλιμο φορτίο δεν θα μπορεί να αποκρυπτογραφεί ακόμα και αν ο κακόβουλος χρήστης έχει τα κλειδιά NwkSkey και AppSKey.

### 4.6.1 Κρυπτογράφηση AES-CBC

Γενικά ο AES είναι ένας αλγόριθμος συμμετρικού κλειδιού όπου κατά την κρυπτογράφηση και την αποκρυπτογράφηση χρησιμοποιείτε ένα ίδιο κλειδί. Ο AES θέτει το μήκος του μπλοκ δεδομένων σε 128 bit και τα μήκη κλειδιού σε 128,192 ή 256 bit. Είναι ένας επαναληπτικός αλγόριθμος και κάθε επανάληψη ονομάζεται γύρος (Nr). Ο AES χρησιμοποιεί διαφορετικό αριθμό γύρων για καθένα από τα καθορισμένα μήκη κλειδιών. Όταν χρησιμοποιείτε κλειδί 128 bit ο συνολικός αριθμός των γύρων είναι 10, για κλειδί 192 bit 12 γύροι και για κλειδί 256 bit 14 γύροι. Κάθε γύρος

στο AES, εκτός από τον τελικό γύρο, αποτελείται από τέσσερις μετασχηματισμούς (εικόνα 72):

- SubBytes : Ένας πίνακας αντικατάστασης (s-box) χρησιμοποιείται για την αντικατάσταση κάθε byte. Αυτή η λειτουργία έχει υψηλή επίδραση μη γραμμικότητας στη διαδικασία κρυπτογράφησης.
- ShiftRows : Οι τρεις τελευταίες σειρές του πίνακα 4x4 byte μετατοπίζονται κυκλικά αφήνοντας την πρώτη σειρά αναλλοίωτη. Αυτή η διαδικασία προσθέτει περισσότερη σύγχυση και κρυπτογράφηση στη διαδικασία κρυπτογράφησης.
- MixColumns : Θεωρώντας κάθε byte ως πολυώνυμο και όχι ως αριθμό, κάθε στήλη πολλαπλασιάζεται με έναν σταθερό πίνακα που ικανοποιεί το πεδίο Galois (8η δύναμη του 2).
- AddRoundKey : Σε αυτό το βήμα, η λειτουργία XOR υλοποιείται μεταξύ του αρχικού μπλοκ εισόδου και του κλειδιού.

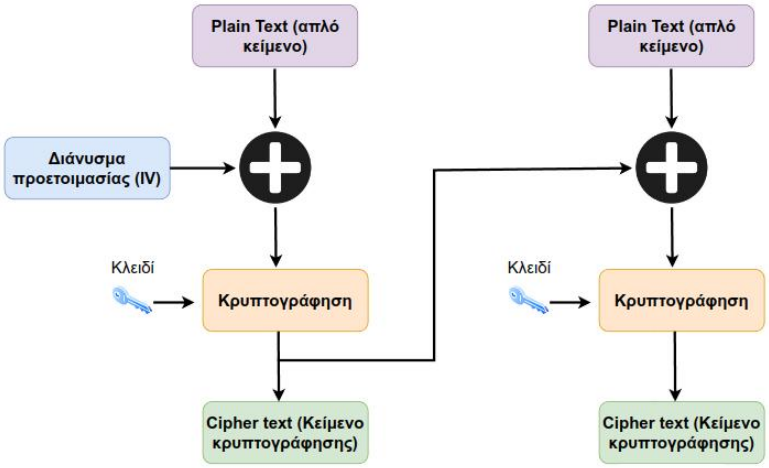



Εικόνα 72. Διάγραμμα ροής κρυπτογράφησης AES

Ο τελικός γύρος δεν έχει τον μετασχηματισμό MixColumns, η ροή αποκρυπτογράφησης είναι απλώς το αντίστροφο της ροής κρυπτογράφησης και κάθε λειτουργία είναι το αντίστροφο της αντίστοιχης στη διαδικασία κρυπτογράφησης. Ο μετασχηματισμός γύρου του AES και τα βήματά του λειτουργούν σε κάποια ενδιάμεσα αποτελέσματα, που ονομάζονται κατάσταση. Η κατάσταση

μπορεί να απεικονιστεί ως μια ορθογώνια μήτρα με τέσσερις σειρές. Ο αριθμός των στηλών στην κατάσταση συμβολίζεται με Nb και είναι ίσος με το μήκος του μπλοκ σε bit διαιρούμενο με το 32. Για ένα μπλοκ δεδομένων 128 bit (16 bytes) η τιμή του Nb είναι 4, επομένως η κατάσταση αντιμετωπίζεται ως 4×4 μήτρα και κάθε στοιχείο στη μήτρα αντιπροσωπεύει ένα byte.

Όλες οι λειτουργίες AES έχουν τα ίδια βήματα που αναφέρθηκαν παραπάνω, αλλά διαφέρουν στις περαιτέρω τεχνικές. Η λειτουργία αλυσιδωτής μπλοκ κρυπτογράφησης (Cipher Block Chaining - CBC) διαφέρει από άλλες λειτουργίες AES ανάλογα με το Διάνυσμα αρχικοποίησης (Initialization Vector - IV) στην αρχή της διαδικασίας κρυπτογράφησης. Η λειτουργία CBC πραγματοποιείτε με την διαδικασία XOR του πρώτου μπλοκ απλού κειμένου (b1) με το διάνυσμα αρχικοποίησης πριν το κρυπτογραφήσει. Η λειτουργία CBC περιλαμβάνει επίσης αλυσίδα μπλοκ καθώς κάθε επόμενο μπλοκ απλού κειμένου επεξεργάζεται με την διαδικασία XOR με το κρυπτογραφημένο κείμενο του προηγούμενου μπλοκ [99], [100]. Κάθε μπλοκ κρυπτογραφείται ανάλογα με το IV του προηγούμενου μπλοκ όπως φαίνεται στην εικόνα 73




Εικόνα 73. Κρυπτογράφηση σε λειτουργία CBC

Αν συνοψίσουμε αυτή τη διαδικασία σε έναν τύπο θα μοιάζει με τον τύπο 4.1

$$C_i = E_K(B_i \bigoplus C_{i-1}) \tag{4.1}$$

Όπου το EK υποδηλώνει τον αλγόριθμο κρυπτογράφησης μπλοκ χρησιμοποιώντας το κλειδί K, και το Ci-1 είναι η κρυπτογράφηση που αντιστοιχεί στο Bi-1. Υποθέτουμε ότι το C0 είναι το διάνυσμα αρχικοποίησης (IV).

Ομοίως, η αποκρυπτογράφηση με χρήση του CBC μπορεί να αποτυπωθεί από τον τύπο 4.2

$$B_i = D_K(C_i) \bigoplus (C_{i-1}) \tag{4.2}$$

Όπου το DK υποδηλώνει τον αλγόριθμο αποκρυπτογράφησης μπλοκ χρησιμοποιώντας το κλειδί K. Το ίδιο διάνυσμα αρχικοποίησης (C0) θα χρησιμοποιηθεί για την αποκρυπτογράφηση [101].

Με τη λειτουργία CBC ένα πλεονέκτημα είναι ότι τα ίδια μπλοκ δεν έχουν το ίδιο κρυπτογράφημα. Αυτό συμβαίνει επειδή το διάνυσμα αρχικοποίησης προσθέτει έναν τυχαίο παράγοντα σε κάθε μπλοκ. Επομένως, για τα ίδια μπλοκ σε διαφορετικές θέσεις θα έχουν διαφορετικά κρυπτογραφήματα. Αν και η λειτουργία CBC είναι πιο ασφαλής, η κρυπτογράφηση της δεν είναι ανθεκτική σε απώλειες μπλοκ. Αυτό συμβαίνει επειδή τα μπλοκ εξαρτώνται από τα προηγούμενα μπλοκ τους για κρυπτογράφηση. Έτσι, εάν χαθεί το μπλοκ Bi, η κρυπτογράφηση όλων των επόμενων μπλοκ δεν θα είναι δυνατή. Αυτή η αλυσιδωτή συμπεριφορά σημαίνει επίσης ότι η κρυπτογράφηση των μπλοκ πρέπει να γίνεται διαδοχικά, όχι παράλληλα. Ωστόσο, αυτά τα μειονεκτήματα

δεν επεκτείνονται στην αποκρυπτογράφηση, η οποία μπορεί να γίνει παράλληλα, εάν όλα τα μπλοκ κρυπτογραφημένου κειμένου είναι διαθέσιμα και μπορούν να ανεχθούν απώλειες μπλοκ [101]. Για τον προγραμματισμό του Multos ώστε να υποστηρίζει τις κρυπτογραφικές δυνατότητες του AES-CBC θα χρησιμοποιηθεί η βιβλιοθήκη PCKS#11. Στην επόμενη ενότητα και πριν γίνει υλοποίηση στο Multos θα γίνει μια σύντομη αναφορά για τα Πρότυπα Κρυπτογραφίας Δημόσιου Κλειδιού (PKCS).

### 4.6.2 Πρότυπα Κρυπτογραφίας Δημόσιου Κλειδιού - PKCS

Τα Πρότυπα Κρυπτογραφίας Δημόσιου Κλειδιού (PKCS) αναπτύχθηκαν από την RSA Security Inc. για την παροχή ενός προτύπου που επιτρέπει τη διαλειτουργικότητα και τη συμβατότητα μεταξύ συσκευών και υλοποιήσεων προμηθευτών. Τα πρότυπα PKCS καλύπτουν μια πληθώρα διαδικασιών κρυπτογραφίας δημοσίου κλειδίου όπως :

- PKCS#1 : RSA Encryption Standard
- PKCS#11 : Cryptographic Token Interface Standard
- PKCS#8 : Private-key Information Syntax Standard

Συνεπώς το PKCS#11 είναι ένα από τα Πρότυπα Κρυπτογραφίας Δημόσιου Κλειδιού, ορίζει την διεπαφή προγραμματισμού (API) Cryptoki που έχει σχεδιαστεί για να είναι μια διεπαφή μεταξύ εφαρμογών και κρυπτογραφικών συσκευών όπως έξυπνες κάρτες, μονάδες ασφαλείας υλικού (Hardware Security Modules – HSM), PCMCIA και USB key tokens. Οι σχεδιαστές του PKCS#11 περιέγραψαν τους σχεδιαστικούς στόχους ως εξής: να «παρέχουν μια τυπική διεπαφή μεταξύ εφαρμογών και (φορητών) κρυπτογραφικών συσκευών» και ταυτόχρονα να «επιτρέπουν την κοινή χρήση πόρων» (μια σχέση πολλών προς πολλά μεταξύ εφαρμογών και συσκευές). Στην ορολογία PKCS#11, ένα διακριτικό (token) είναι μια συσκευή που αποθηκεύει αντικείμενα (π.χ. κλειδιά, δεδομένα και πιστοποιητικά) και εκτελεί κρυπτογραφικές λειτουργίες. Όταν κάποιος χρησιμοποιεί ένα διακριτικό (ή να επικοινωνήσει με αυτό), πρέπει πρώτα να δημιουργήσει μια περίοδο λειτουργίας με το διακριτικό, το οποίο απαιτεί από τον χρήστη να «συνδεθεί» και να πιστοποιηθεί η ταυτότητα στη συσκευή. Στη συνέχεια, ο χρήστης μπορεί να κάνει χρήση της λειτουργικότητας που παρέχεται από το διακριτικό πραγματοποιώντας κλήσεις μέσω της διεπαφής ή του API [102], [103]. Στο link PKCS #11 Cryptographic Token Interface Current Mechanisms Specification Version φαίνονται οι κρυπτογραφικοί μηχανισμοί που υποστηρίζονται από το PKCS#11. Στην επόμενη ενότητα ακολουθεί η εγκατάσταση build για την υποστήριξη των διαδικασιών κρυπτογράφησης-αποκρυπτογράφησης AES-CBC.

### 4.6.3 Εγκατάσταση AES-CBC build για το Multos

Θα εγκαταστήσουμε πρώτα ένα build το pymultosP11. Αυτό το build περιλαμβάνει 2 αρχεία, το pkcs11.h για τη χρήση της βιβλιοθήκης PKCS#11 για την κλήση της συνάρτησης aesEncCbc και το aesDecCbc για την κρυπτογράφηση και αποκρυπτογράφηση του ωφέλιμου φορτίου και το αρχείο Python.h για τη δημιουργία του Python module.

Πηγαίνουμε στον φάκελο trust_core :

```
cd trust_core
```

Μπορούμε να κατεβάσουμε το συγκεκριμένο build με την εντολή

```
wget --no-check-certificate 'https://drive.google.com/file
/d/1DOuGz6ZqRL_mtwumXVwJHxXQpdNXu5ja/view?usp=sharing' -O pymultosP11.zip
```

Αποσυμπιέζουμε με την εντολή:

```
tar -xf pymultosP11.zip
```

Για την εγκατάσταση του build εκτελούμε την εντολή:

```
sudo bash buildpymultosP11
```

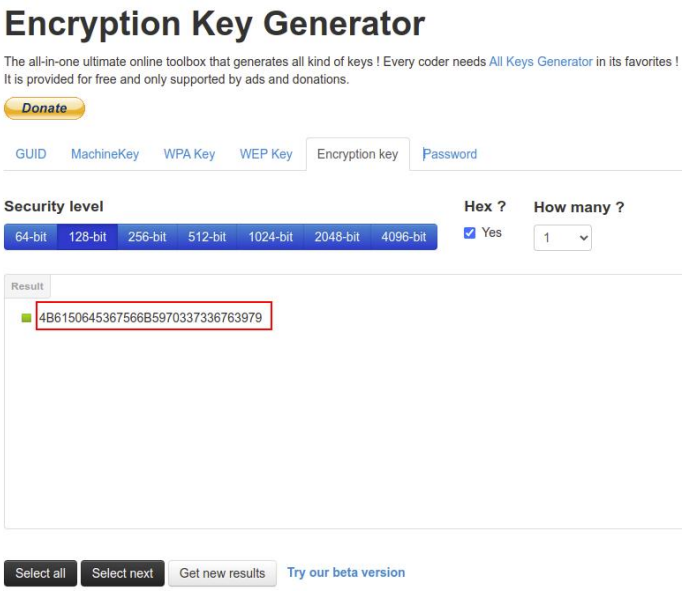
### 4.6.4 Multos AES Key Management και εισαγωγή κλειδιού

Το Trust Core διαθέτει 10 υποδοχές για συμμετρικά κλειδιά (γενικά κλειδιά AES ή κλειδιά που σχετίζονται με συνεδρία TLS). Τα κλειδιά που σχετίζονται με TLS δημιουργούνται αυτόματα από τους σχετικούς μηχανισμούς.

Τα κλειδιά AES για γενική χρήση μπορούν να δημιουργηθούν με το εργαλείο p11keyman. Το p11keyman μπορούμε να το χρησιμοποιήσουμε για :

- Δημιουργία κλειδιών
- Εισαγωγή κλειδιού (όπου το κλειδί είναι τυλιγμένο από ένα υπάρχον κλειδί AES με άδεια UNWRAP)
- Εισαγωγή κλειδιού από τρία στοιχεία
- Εξαγωγή κλειδιού (χρησιμοποιώντας ένα υπάρχον κλειδί AES ή RSA με άδεια WRAP)
- Διαγραφή κλειδιού

Θα χρησιμοποιήσουμε την εντολή p11keyman -i για να εισάγουμε το κλειδί που θα χρησιμοποιηθεί για την κρυπτογράφηση του ωφέλιμου φορτίου. Αν θέλουμε να δημιουργήσουμε κλειδί πριν το εισάγουμε στο Multos μπορούμε από το site https://www.allkeysgenerator.com/Random/Security-Encryption-Key-Generator.aspx και επιλέγουμε 128 bit (μπορούμε και 256 bit) στο HEX ? επιλέγουμε YES και δημιουργείτε το κλειδί (εικόνα 74).

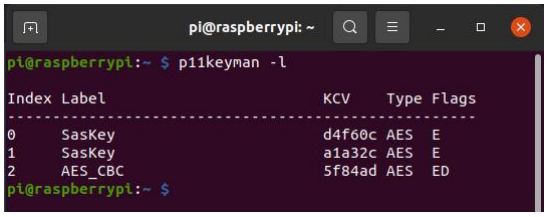



Εικόνα 74. Encryption Key Generator

Από το terminal εισάγουμε το κλειδί στο multos με την εντολή:

- p11keyman -i 32 ED AES_CBC
- Στην περίπτωση που δεν τρέξει η εντολή p11keyman και έχουμε μήνυμα access denied πάμε στον φάκελο bin με cd /bin και μετά εκτελούμε την εντολή chmod +x p11keyman. Μετά εκτελούμε ξανά την εντολή p11keyman -i 32 E SasKey.
- Βάζουμε το pin (Το default είναι 1234)
- Enter component 1 Βάζουμε το επιθυμητό κλειδί
- Enter component 2 ffffffffffffffffffffffffffffffff ffffffffffffffffffffffffffffffff
- Enter component 3 ffffffffffffffffffffffffffffffff ffffffffffffffffffffffffffffffff

Στο terminal θα δούμε Key imported OK. Με την εντολή p11keyman -l βλέπουμε τα κλειδιά που έχουμε εισάγει στο multos (εικόνα 75.




Εικόνα 75. p11keyman -l

## 4.7 Δοκιμή του pymultosP11

Σε αυτήν την ενότητα θα γίνουν δοκιμές του build pymultosP11

Πρώτα θα γίνει δοκιμή του pymultosP11 και θα κρυπτογραφήσουμε ένα μήνυμα Node1 : 280 KW/h και μετά θα το αποκρυπτογραφήσουμε. Στην συνέχεια θα στείλουμε το κρυπτογραφημένο μήνυμα Node1 : 280 KW/h στον server του εργαστηρίου restqmlab.

### 4.7.1 Κρυπτογράφηση του μηνύματος Node1 : 280 KW/h

Στον κώδικα3 γίνεται η κρυπτογράφηση του μηνύματος Node1 : 280 KW/h. Προσοχή το μήνυμα που θα κρυπτογραφηθεί θα πρέπει να είναι πολλαπλάσιο του 16.

```
1
2  import pymultosP11
3
4  import warnings
5  import time
6
7  warnings.filterwarnings("ignore", category=DeprecationWarning)
8
9  # One time init
10
11 pkcs11_handle = pymultosP11.init()
12
13 # 16 byte ICV needed for AES-CBC
14
15 icv = bytes(b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
       )
```

```
16
17 # ID of key slot containing the AES key to use
18
19 key_slot = 2
20
21 # Data to encrypt as a bytes-like object
22
23 data_string = 'Node1 : 280 KW/h' #Must be multiple to 16 because of AES-CBC
       encyption
24
25 data_bytes = bytes(data_string,'utf-8')
26
27 # Do the encryption. Returns a bytes-like object
28
29 cipher_text = pymultosP11.aesEncCbc(pkcs11_handle,data_bytes,icv,key_slot)
30
31 # view cipher text as hex
32
33 print (cipher_text.hex())
```

Κώδικας 3. Κρυπτογράφηση AES-CBC του μηνύματος Node1 : 280 KW/h

Το αποτέλεσμα είναι 4cf3831de1329771d5a2c1be430d121b και φαίνεται στην εικόνα 76


Εικόνα 76. Κρυπτογράφηση μηνύματος Node1 : 280 KW/h

### 4.7.2 Αποκρυπτογράφηση του μηνύματος 4cf3831de1329771d5a2c1be430d121b

Στον κώδικα 4 γίνεται η κρυπτογράφηση του μηνύματος Node1 : 280 KW/h. Προσοχή το μήνυμα που θα κρυπτογραφηθεί θα πρέπει να είναι πολλαπλάσιο του 16.

```
1 import pymultosP11
2
3 import warnings
4
5 warnings.filterwarnings("ignore", category=DeprecationWarning)
6
7 # One time init
8
9 pkcs_handle = pymultosP11.init()
10
11 # 16 byte ICV needed for AES-CBC
12
13 icv = bytes(b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
       )
14
15 # ID of key slot containing the AES key to use
16
17 key_slot = 2
18
19 payload = '4cf3831de1329771d5a2c1be430d121b'
```

```
20
21 # Data to encrypt as a bytes-like object
22
23 cipher_text = bytearray.fromhex((payload))
24
25 # Do the decryption. Returns a bytes-like object
26
27 plain_text = pymultosP11.aesDecCbc(pkcs_handle,cipher_text,icv,key_slot)
28
29 # view cipher text as hex
30
31 print(plain_text.decode())
```

Κώδικας 4. Αποκρυπτογράφηση του μηνύματος 4cf3831de1329771d5a2c1be430d121b

Το αποτέλεσμα είναι Node1 : 280 KW/h και φαίνεται στην εικόνα 77


Εικόνα 77. Αποκρυπτογράφηση του μηνύματος

Τα αποτελέσματα είναι ικανοποιητικά λειτουργεί το pymultosP11 χωρίς πρόβλημα. Στην συνέχεια θα στείλουμε το κρυπτογραφημένο μήνυμα Node1 : 280 KW/h στον server του εργαστηρίου restqmlab.

### 4.7.3 Αποστολή κρυπτογραφημένου μηνύματος (Node1 : 280 KW/h) στον server του εργαστηρίου restqmlab

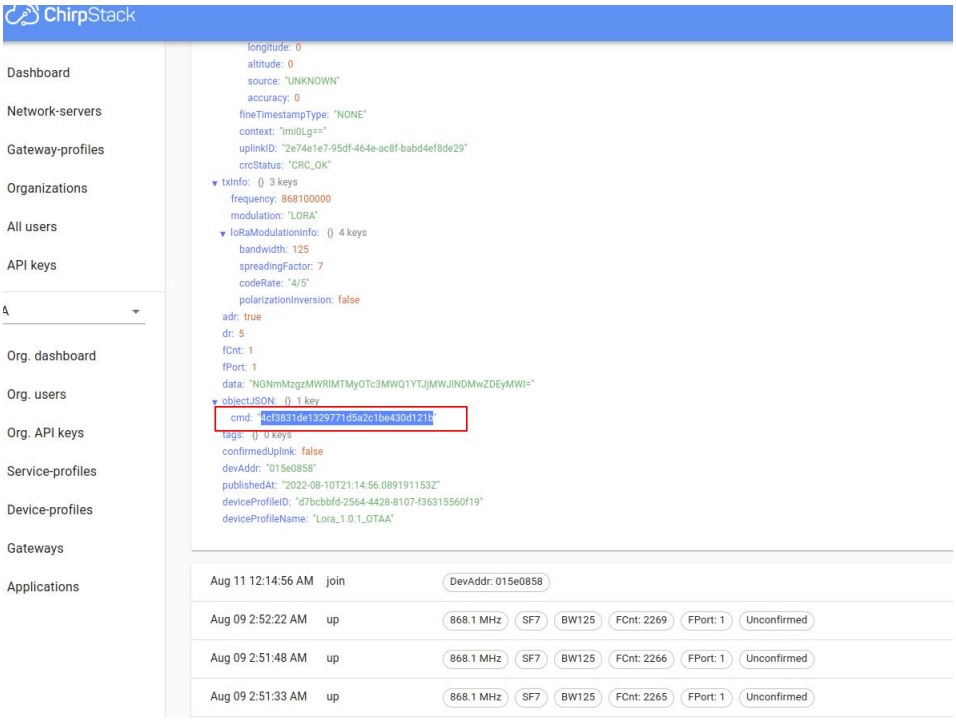
Στον παρακάτω κώδικα 5 γίνεται η κρυπτογράφηση του μηνύματος Node1 : 280 KW/h και αποστολή στον server του εργαστηρίου restqmlab κάθε 900 δευτερόλεπτα (15 λεπτά).

```
1
2 import pymultosP11
3
4 import warnings
5
6 warnings.filterwarnings("ignore", category=DeprecationWarning)
7
8
9
10
11 #from random import randint
12 from sys import exit
13 from time import sleep
14 from rak811.rak811 import Mode, Rak811
15 lora = Rak811()
16
17 lora.hard_reset()
18 lora.mode = Mode.LoRaWan
19 lora.band = 'EU868'
20
21 lora.set_config(dev_eui='3038383663385608',
22 app_eui='3038383663385608',
23 app_key='D78D5F5942C4218A78BE9BAA75453D6F')
```

```
lora.join_otaa()

lora.dr = 5



# One time init

pkcs11_handle = pymultosP11.init()


# 16 byte ICV needed for AES-CBC

icv = bytes(b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
    )



# ID of key slot containing the AES key to use

key_slot = 2

# Data to encrypt as a bytes-like object

data_string = 'Node1 : 280 KW/h'

data_bytes = bytes(data_string,'utf-8')




# Do the encryption. Returns a bytes-like object

cipher_text = pymultosP11.aesEncCbc(pkcs11_handle,data_bytes,icv,key_slot)

# view cipher text as hex

#print (cipher_text.hex())

print('Sending packets every 900 seconds - Interrupt to cancel loop')
print('You can send downlinks from the Chirpstack')
try:
    while True:
        print('Send packet')
        # Cayenne lpp random value as analog
        lora.send(cipher_text.hex())

        while lora.nb_downlinks:
            print('Received', lora.get_downlink()['data'].hex())

        sleep(900)
except:  # noqa: E722
    pass

print('Cleaning up')
lora.close()
exit(0)
```

Κώδικας 5. Αποστολή κρυπτογραφημένου μηνύματος Node1 : 280 KW/h στον server του εργαστηρίου restqmlab

Στην εικόνα 78 φαίνεται η επιτυχής αποστολή κρυπτογραφημένων μηνυμάτων στον server του εργαστηρίου restqmlab.




Εικόνα 78. Επιτυχής αποστολή κρυπτογραφημένων μηνυμάτων στον server του εργαστηρίου restqmlab

## 4.8 Azure IoT Hub

Σε αυτήν την ενότητα θα γίνει χρήση του Mutlos Trust Core για την αποστολή δεδομένων στο Azure IoT Hub. Στο κεφάλαιο 5 όπως θα δούμε θα χρησιμοποιήσουμε το Azure IoT Hub. Στο 3ο Παράρτημα δίνονται οδηγίες για την εγκατάσταση του Azure IoT Hub

Το Azure IoT Hub προσφέρει μια αξιόπιστη και ασφαλή αμφίδρομη επικοινωνία μεταξύ συσκευών και του cloud και λειτουργεί ως κεντρικός κόμβος μηνυμάτων για την επικοινωνία μεταξύ μιας εφαρμογής IoT και των συνδεδεμένων συσκευών της. [104]

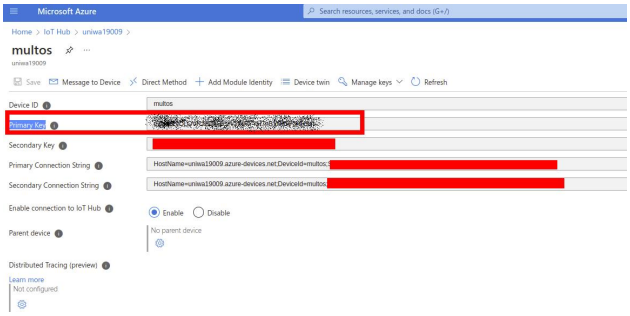
Υποστηρίζει διάφορα δημοφιλή πρωτόκολλα IoT για την μετάδοση των δεδομένων όπως των HTTPS, AMQP,AMQP μέσω WebSockets, MQTT και MQTT μέσω Websockets. Υπάρχει η δυνατότητα να διαχειριστούν και άλλα πρωτόκολλα μέσω της μετατροπής πρωτοκόλλου στο άκρο εντός του Azute IoT edge ή εκτελώντας μετατροπή πρωτοκόλλου στο cloud μέσω ανάπτυξης μιας προσαρμοσμένης πύλης πρωτοκόλλου Azure IoT (χρησιμοποιώντας Azure Service Fabric, Azure Cloud Services ή Virtual Machines των Windows). [105]

Επίσης παρέχει και συμπληρωματικές δυνατότητες για μηχανική μάθηση (machine learning) , ανάλυση ροής (stream analytics) , διαχείριση συμβάντων (event management) και οπτικοποίηση (visualization). [106]

Προστατεύει τα δεδομένα κρυπτογραφώντας τα αυτόματα πριν τα διατηρήσει στο cloud. Υποστηρίζει κρυπτογράφηση δεδομένων σε κατάσταση ηρεμίας χρησιμοποιώντας κλειδιά διαχείρισης από τον πελάτη (CMK), γνωστά και ως Bring your own key (BYOK). Το Azure IoT Hub παρέχει κρυπτογράφηση δεδομένων σε κατάσταση ηρεμίας και κατά τη μεταφορά. Τα δεδομένα κρυπτογραφούνται όταν διαβάζονται και αποκρυπτογραφούνται όταν γράφονται. [107]

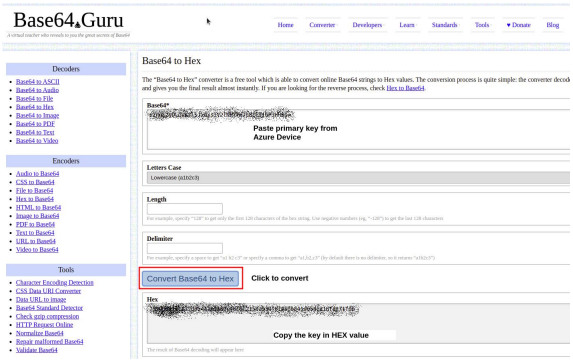
### 4.8.1 Εισαγωγή κλειδιού στο Multos

Πρώτα πρέπει να εισάγουμε κλειδί στο Multos. Θα χρησιμοποιήσουμε την εντολή p11keyman -i για να εισάγουμε το Primary Key από την συσκευή μας στο Azure IoT Hub αφού πρώτα μετατρέψουμε από base64 σε HEX. Από το Azure Portal πάμε στην συσκευή και κάνουμε copy το primary key (εικόνα 79).




Εικόνα 79. Primary Key

Στην συνέχεια πάμε στο link `https://base64.guru/converter/decode/hex` κάνουμε paste το primary key στο πεδίο Base64, μετά click στο πεδίο Convert Base64 to HEX, και copy το κλειδί στην μορφή HEX (εικόνα 80).




Εικόνα 80. Convert Base64 to HEX

Από το terminal εισάγουμε το primary key στο multos με την εντολή:

- p11keyman -i 32 E SasKey
- Στην περίπτωση που δεν τρέξει η εντολή p11keyman και έχουμε μήνυμα access denied πάμε στον φάκελο bin με cd /bin και μετά εκτελούμε την εντολή chmod +x p11keyman. Μετά εκτελούμε ξανά την εντολή p11keyman -i 32 E SasKey.
- Βάζουμε το pin (Το default είναι 1234)
- Enter component 1 paste το κλειδί σε μορφή HEX
- Enter component 2 ffffffffffffffffffffffffffffffff
  ffffffffffffffffffffffffffffffffffffffff
- Enter component 3 ffffffffffffffffffffffffffffffff
  ffffffffffffffffffffffffffffffffffffffff

Στο terminal θα δούμε Key imported OK. Με την εντολή p11keyman -l βλέπουμε τα κλειδία που έχουμε δημιουργήσει.

### 4.8.2 Πρόσθετα αρχεία για την λειτουργία του Multos Trust Core

Θα χρειαστεί να κατεβάσουμε κάποια τροποποιημένα αρχεία για να λειτουργήσει σωστά η αποστολή δεδομένων στο azure μέσω του multos. Κάνοντας click εδώ κατεβάζουμε τα τροποποιημένα αρχεία.

Τα αρχεία που συμπεριλαμβάνονται στον συμπιεσμένο φάκελο είναι :

- p11keyman : Ανανεωμένη έκδοση του εργαλείου p11keyman
- signing_mechanism.py : Τροποποιημένο αρχείο της βιβλιοθήκης azure-iot-sdk-python για να γίνεται η αναγνώριση του κλειδιού από τα slot του multos.
- simple_send_message.py : Τροποποιημένο αρχείο της βιβλιοθήκης azure-iot-sdk-python για να στείλουμε μήνυμα.
- ένας φάκελος pymultosAzure : Περιέχει ένα build για ένα module στην python, για να κάνουμε import pymultosAzure καθώς και τις απαραίτητες βιβλιοθήκες pkcs11#.

Αποσυμπιέζουμε το αρχείο τον φάκελο trust_core με την εντολή:

```
tar -xf extrafiles.tar.xz -C trust_core
```

Πάμε στον φάκελο trust_core_pymultosAzure

```
cd trust_core/pymultosAzure
```

Και εκτελούμε το

```
sudo bash buildpymultosAzure
```

Στην συνέχεια αντιγράφουμε το αρχείο signing_mechanism.py στις ακόλουθες διαδρομές με τις εντολές:

```
$ sudo cp signing_mechanism.py /usr/local/lib/python3.9/dist-packages/
azure/iot/device/common/auth
```

```
$ sudo cp signing_mechanism.py /home/pi/azure-iot-sdk-python/
azure-iot-device/azure/iot/device/common/auth
```

Το συγκεκριμένο αρχείο θα επιτρέψει όταν γίνει ο συγχρονισμός με το Azure να εξάγει το primary key που είναι αποθυκευμένο στο chip του multos. Το multos trust core διαθέτει 10 slots για αποθήκευση κλειδιών.

Στην ενότητα 9.4 του 3ου Παραρτήματος δίνεται ο κώδικας 10 που αφορά την αποστολή ενός απλού μηνύματος Hello World! στο IoT Hub. Στην 11 γραμμή του κώδικα 10 έχει μια μεταβλητή CONNECTION_STRING που έχει την εξής μορφή:

```
CONNECTION_STRING = "HostName=uniwa19009.azure-devices.net;DeviceId=mypi4;
SharedAccessKey=
ThisIsnoTrue/TestonlyN/EEVYMkpGHCykox9FSJJuHs="
```

Το SharedAccessKey είναι το primary key της συσκευής που είναι συνδεδεμένη με το IoT Hub. Στον τροποποιημένο κώδικα 4.5 για την αποστολή ενός μηνύματος στο IoT Hub με την χρήση του Multos, η τιμή SharedAccessKey θα είναι 0 που ισοδυναμεί με το primary key που είναι αποθηκευμένο στο τσιπ του multos στο Slot 0.

Το IoT Hub επικυρώνει την υπογραφή χρησιμοποιώντας το αντίγραφο του primary key. Ένα διακριτικό (token) που είναι υπογεγραμμένο με το συμμετρικό κλειδί μιας ταυτότητας συσκευής χορηγεί μόνο το δικαίωμα DeviceConnect για τη συσχετισμένη ταυτότητα συσκευής. Το διακριτικό ασφαλείας (security token) έχει την ακόλουθη μορφή:

```
SharedAccessSignature sig={signature-string}&se={expiry}&skn={policyName}
&sr={URL-encoded-resourceURI}
```

Ποιο συγκεκριμένα :

- {signature} : Μια συμβολοσειρά υπογραφής HMAC-SHA256. Για μεμονωμένες εγγραφές, αυτή η υπογραφή δημιουργείται χρησιμοποιώντας το συμμετρικό κλειδί (πρωτεύον ή δευτερεύον) για την εκτέλεση του κατακερματισμού. Για ομάδες εγγραφής (enrollment groups) , ένα κλειδί που προέρχεται από το κλειδί της ομάδας εγγραφής (enrollment group) χρησιμοποιείται για την εκτέλεση του κατακερματισμού. Ο κατακερματισμός εκτελείται σε ένα μήνυμα της μορφής: URL-encoded-resourceURI + ”\n” + expiry. Σημαντικό: Το κλειδί πρέπει να αποκωδικοποιηθεί από το base64 πριν χρησιμοποιηθεί για την εκτέλεση του υπολογισμού HMAC-SHA256. Επίσης, το αποτέλεσμα της υπογραφής πρέπει να έχει κωδικοποίηση URL.
  **Με το trust core, το κλειδί αποθηκεύεται και χρησιμοποιείται με ασφάλεια μέσα στο τσιπ του multos**

- {resourceURI}: URI του τελικού σημείου εγγραφής στο οποίο μπορεί να προσπελαστεί με αυτό το διακριτικό, ξεκινώντας με το αναγνωριστικό εύρους για την παρουσία της υπηρεσίας παροχής συσκευών. Για παράδειγμα, Scope ID/registrations/Registration ID

- {expiry} : Συμβολοσειρές UTF8 για αριθμό δευτερολέπτων από την εποχή 00:00:00 UTC την 1η Ιανουαρίου 1970.

- {URL-encoded-resourceURI} : Κωδικοποίηση URL με πεζά γράμματα του URI πόρου πεζών

- {policyName} : Το όνομα της πολιτικής κοινής πρόσβασης (shared access policy) στην οποία αναφέρεται αυτό το διακριτικό. Το όνομα πολιτικής που χρησιμοποιείται κατά την παροχή (provisioning) συμμετρικής βεβαίωσης κλειδιού (symmetric key attestation ) είναι εγγραφή (registration) [108].

Όταν μια συσκευή βεβαιώνει με μια μεμονωμένη εγγραφή, η συσκευή χρησιμοποιεί το συμμετρικό κλειδί που ορίζεται στην καταχώριση μεμονωμένης εγγραφής για να δημιουργήσει την κατακερματισμένη υπογραφή για το διακριτικό SAS.

Στην περίπτωση που δεν χρησιμοποιήσουμε το trust core το κλειδί αποθηκεύεται στο software των συσκευών, με αυτόν τον τρόπο είναι ευάλωτο στο να γίνει εξαγωγή με συνέπεια οι συσκευές να γίνουν claimed δηλαδή να ελέγχονται από εφαρμογές αλλών συσκευών. Το multos αποτρέπει αυτήν την ευπάθεια επειδή το primary key είναι αποθηκευμένο στο τσιπ και όχι στο software της συσκευής.

Ακολουθεί ο κώδικας 6 :

```
1
2  import os
3  import asyncio
4  import pymultosAzure
5  from azure.iot.device.aio import IoTHubDeviceClient
6
7  async def main():
8      # Set up the MULTOS Trust Core HSM
9      if pymultosAzure.init() <= 0:
10         print("Failed to initialise MULTOS HSM")
11         return
12
13     # Key value is in the HSM slot 0
14     connection_string = "HostName=uniwa19009.azure-
15     devices.net;DeviceId=multos;SharedAccessKey=0"
16
17     # Create instance of the device client using
18     the connection string
19     device_client = IoTHubDeviceClient.create_from_
20     connection_string(connection_string)
21
22     # Connect the device client.
23     await device_client.connect()
24
25     # Send a single message
26     print("Sending message...")
27     await device_client.send_message
28     ("Hello World!")
29     print("Message successfully sent!")
30
31     # Finally, shut down the client
32     await device_client.shutdown()
33
34
35 if __name__ == "__main__":
36     asyncio.run(main())
37
38     # If using Python 3.6 or below, use the
39     following code instead of asyncio.run(main()):
40     # loop = asyncio.get_event_loop()
41     # loop.run_until_complete(main())
42     # loop.close()
```

Κώδικας 6. Αποστολή μηνύματος στο IoT Hub με την χρήση του Multos

Εκτελώντας τον κώδικα 6 εμφανίζει μήνυμα :

```
Sending message...
Message successfully sent!
```

## 4.9 Σύνοψη

Σε αυτό το κεφάλαιο εξετάστηκαν οι δυνατότητες του Ethereum Blockchain όπου είναι ιδανικό για Smart Contracts και το Helium Blockchain όπου είναι για την αποστολή δεδομένων από κόμβους LoRa.

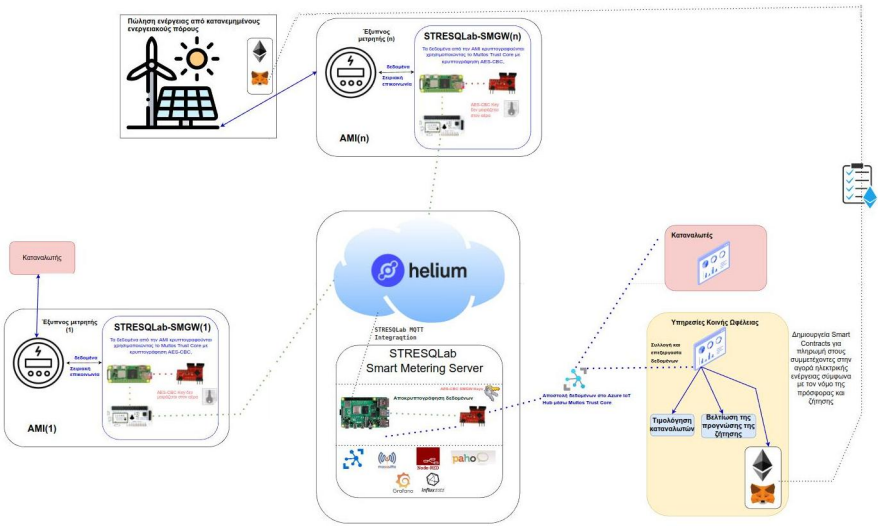
Επίσης εξετάστηκαν οι δυνατότητες του Secure Element Mutlos Trust Core για δυο χρήσεις. Η πρώτη αφορά την επιπλέον κρυπτογράφηση του ωφέλιμου φορτίου από τον κόμβο LoRa-Helium πριν αποσταλεί στο Helium Blockchain και η δεύτερη χρήση αφορά για την αποστολή δεδομένων στο Azure IoT Hub. Και με τις δυο χρήσεις αντιμετωπίστηκαν δυο πιθανές ευπάθειες.

Στα προηγούμενα κεφάλαια έγινε μελέτη που αφορά στο θεωρητικό υπόβαθρο και στο επόμενο κεφάλαιο θα γίνει η υλοποίηση της SMGW.

# ΚΕΦΑΛΑΙΟ 5

## Υλοποίηση SMGW

Στα προηγούμενα κεφάλαια έγινε μελέτη στο θεωρητικό υπόβαθρο που αφορούσε στην ανάπτυξη ασφαλούς και ευέλικτης πύλης έξυπνου μετρητή ενέργειας. Σε αυτό το κεφάλαιο θα γίνει υλοποίηση της SMGW ενός IoT συστήματος που θα έχει την ονομασία STRESQLab Smart Metering System και τα δυο βασικά επίπεδα θα είναι το STRESQLab-SMGW και το STRESQLab Smart Metering Server. Το STRESQLab-SMGW θα στέλνει τα δεδομένα με ασφάλεια στον STRESQLab Smart Metering Server όπου θα τα επεξεργάζεται για να δημιουργεί εφαρμογές σε υπηρεσίες κοινής οφέλειας, καταναλωτές, εξωτερικούς συμμετέχοντες στην αγορά ενέργειας (όπως από κατανεμημένους ενεργειακούς πόρους) κλπ. Για να επιτευχθεί η ασφαλής επικοινωνία θα χρησιμοποιηθεί το Secure Element Trust Core όπου θα κωδικοποιεί το ωφέλιμο φορτίο με κωδικοποίηση AES-CBC πριν το στείλει στο Helium Blockchain. Τα δεδομένα από το Helium Blockchain θα λαμβάνονται στον STRESQLab Smart Metering Server μέσω του MQTT Broker όπου θα γίνεται η αποκρυπτογράφηση του ωφέλιμου φορτίου με την χρήση άλλου Multos Trust Core. Το κλειδί κρυπτογράφησης AES-CBC δεν θα μοιράζεται στον αέρα και θα εισαχθεί στο Multos για να γίνει η αποκρυπτογράφηση του ωφέλιμου φορτίου. Στην υλοποίηση του συστήματος θα δοθεί έμφαση στην ασφάλεια της επικοινωνίας της STRESQLab-SMGW. Το μπλοκ διάγραμμα του STRESQLab Smart Metering System δίνεται στην εικόνα 81.



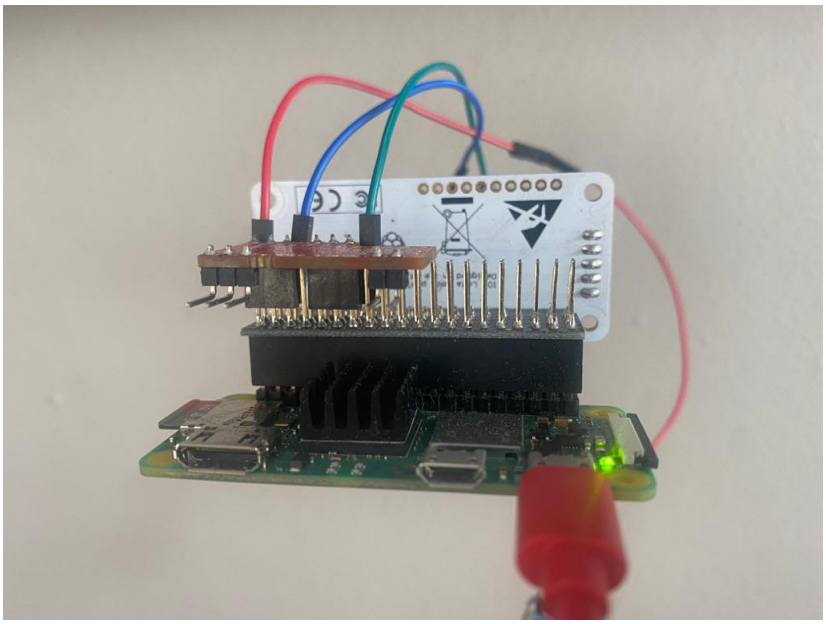

Εικόνα 81. STRESQLab Smart Metering System

## 5.1 STRESQLab-SMGW

Στην εικόνα 82 δίνεται η πύλη έξυπνου μετρητή ενέργειας (STRESQLab-SMGW) και αποτελείται από :

- Ενσωματωμένο σύστημα : Raspberry pi zero 2w (OS : Raspbian GNU/Linux 11 (bullseye))
- Secure element : Multos Trust Core
- LoRa node hat : Pi Supply / pHAT PIS-1128.
- Raspberry gpio επέκταση : (RaidSonic Icy Box IB-RPA101)

Το Raspberry pi zero 2w επιλέχθηκε επειδή έχει χαμηλό κόστος και εύκολη σύνδεση με το Secure Element Multos Trust Core. Το LoRa node hat : Pi Supply / pHAT Το PIS-1128 είναι το μόνο hat που λειτουργεί στο pi zero και λειτουργεί πολύ ικανοποιητικά. Με τη χρήση επέκτασης gpio RaidSonic Icy Box IB-RPA101 είναι πιο ευέλικτο να συνδεθεί το LoRa node hat , Multos και τα pin της σειριακής επικοινωνίας με τον έξυπνο μετρητή.



Εικόνα 82. STRESQLab-SMGW

Για την επικοινωνία του STRESQLab-SMGW με το Helium χρησιμοποιήθηκε το Hotspot SenseCAP-M1. Για πληροφορίες σχετικά με την εγκατάσταση του Helium Hotspot SenseCAP-M1 έχουν δωθεί στην ενότητα 4.3.4 του 4ου Κεφαλάιου. Για την εισαγωγή κλειδιού στο Multos έχουν δοθεί οδηγίες στην ενότητα 4.6.2 του 4ου Κεφαλάιου.

Στον πίνακα 15 δίνονται οι ρυθμίσεις και τα λειτουργικά χαρακτηριστικά του κόμβου LoRa.

Πίνακας 15. Ρυθμίσεις και λειτουργικά χαρακτηριστικά του κόμβου

| Freguency (MHz) | Spreading Factor | Bandwidth (KHz) | Code Rate | Bitrate (Bits/s) | Throughput (Bits/s) | Maxinun Payload Size (bytes) | Latency (s) | Cost per 24 bytes ($) | airtime | *1% max duty cycle* | fair access policy |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 868 | SF7 | 125KHz | 4/5 | 5470 | 4844.7 | 242 | 1 | 0.00001 | 189.7 ms | 19.0 sec<br>189 msg/hour | 546.3 sec (avg)<br>6.6 avg/hour<br>158 msg /24h |

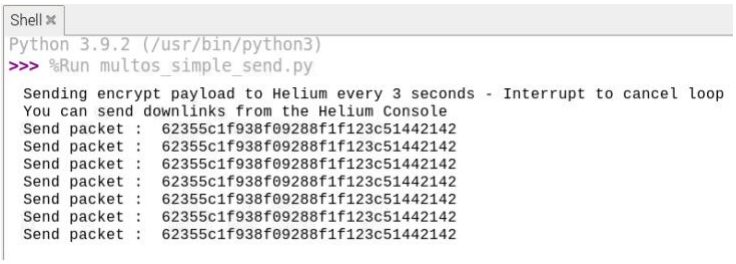
Στον κώδικα 7 γίνεται η κρυπτογράφηση του μηνύματος SGMW1 : 280 KW/h με την χρήση του Multos Trust Core και αποστολή στο Helium blockchain κάθε 900 δευτερόλεπτα (15 λεπτά).

```
import pymultosP11
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

from sys import exit
from time import sleep
from rak811.rak811 import Mode, Rak811
lora = Rak811()

lora.hard_reset()
lora.mode = Mode.LoRaWan
lora.band = 'EU868'
xxxxxxxxxxxxxxx
lora.set_config(dev_eui='xxxxxxxxxxxxxxx',
app_eui='xxxxxxxxxxxxxxx',
app_key='xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
lora.join_otaa()

lora.dr = 5

# One time init

pkcs11_handle = pymultosP11.init()


# 16 byte ICV needed for AES-CBC

icv = bytes(b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
    )


# ID of key slot containing the AES key to use

key_slot = 2

# Data to encrypt as a bytes-like object

data_string = 'SGMW1 : 280 KW/h'

data_bytes = bytes(data_string,'utf-8')

# Do the encryption. Returns a bytes-like object

cipher_text = pymultosP11.aesEncCbc(pkcs11_handle,data_bytes,icv,key_slot)

# view cipher text as hex

#print (cipher_text.hex())

print('Sending encrypt payload to Helium every 900 seconds - Interrupt to
    cancel loop')
print('You can send downlinks from the Helium Console')
try:
    while True:
        print('Send packet')
        # Cayenne lpp random value as analog
        lora.send(cipher_text.hex())

        while lora.nb_downlinks:
            print('Received', lora.get_downlink()['data'].hex())

        sleep(900)
```

```
except:  # noqa: E722
    pass

print('Cleaning up')
lora.close()
exit(0)
```

Κώδικας 7. Αποστολή κρυπτογραφημένου μηνύματος SMWG1 : 280 KW/h στο Helium

Στην εικόνα 83 φαίνεται η αποστολή του κρυπτογραφημένου μηνύματος SMGW1.




Εικόνα 83. Αποστολή κρυπτογραφημένου μηνύματος στο Helium Blockchain

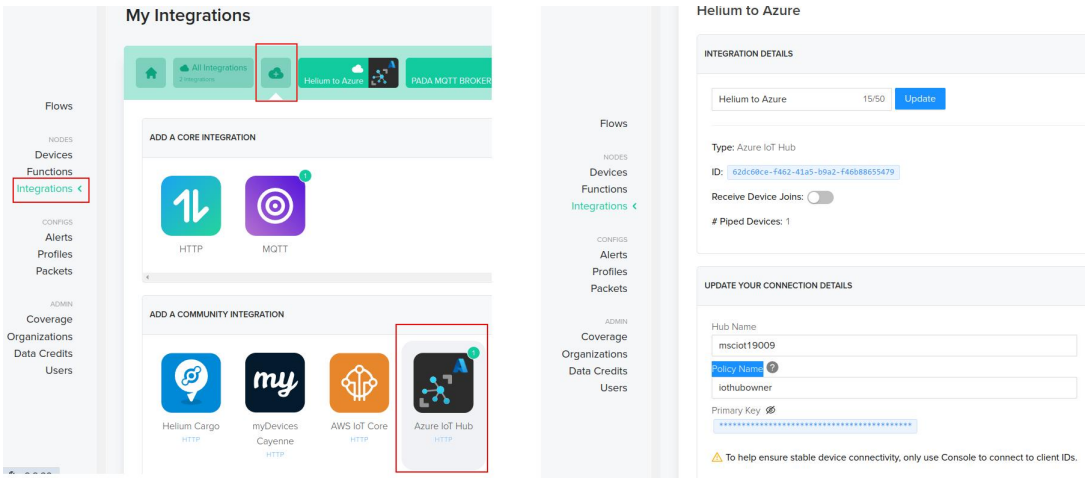
### 5.1.1 Integrations Helium Console

Για την προσθήκη κόμβου στο Helium Console έχουν δοθεί οδηγίες στην ενότητα 4.3.5 του 4ου Κεφαλάιου. Σε αυτήν την ενότητα θα δοθούν οδηγίες για την προσθήκη integrations για MQTT και Azure IoT Hub.

#### Azure IoT Hub Integration

Στο Παράρτημα 3 δίνονται οδηγίες για την εγκατάσταση του Azure IoT Hub.

Για την προσθήκη του Azure IoT Hub στο Helium Console επιλέγουμε Integrations→Add New Integration →Azure IoT Hub και στην συνέχεια εμφανίζει ένα menu όπως φαίνεται στην εικόνα 84



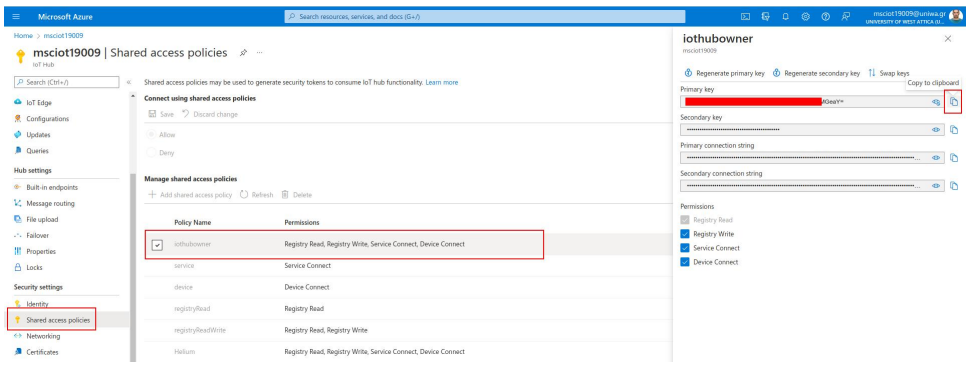

Εικόνα 84. Integration : Azure IoT Hub

Στο INTEGRATION DETAILS προσθέτουμε τα εξής :

- Ορίζουμε όνομα του integration πχ Helium to Azure

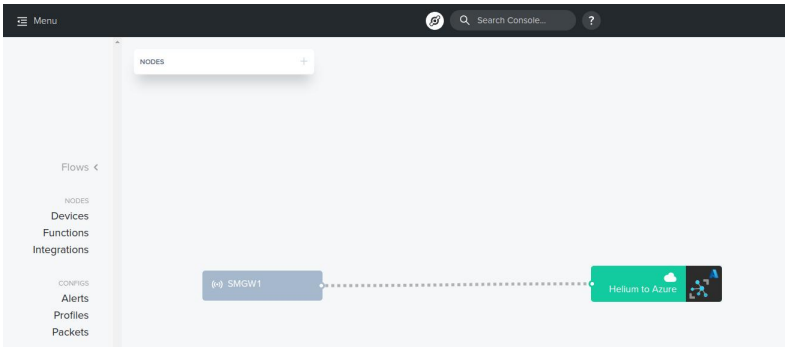
- Hub Name : το όνομα του iot hub (msciot19009)
- Policy Name : Δικαιώματα (iothubowner)
- Primary key : Copy paste το primary key

Τα συγκεκριμένα στοιχεία τα βρίσκουμε από το Azure Portal → στην επιλογή IoT Hub →Shared access policies όπως φαίνεται στην εικόνα 85




Εικόνα 85. Shared access policies

Στο μενού flows που παρέχει ένα φιλικό περιβάλλον εργασίας που παραπέμπει στο node-red συνδέουμε το SGMW1 στο Azure IoT Hub όπως φαίνεται στην εικόνα 86




Εικόνα 86. Flow Node + Azure IoT Hub

### MQTT Integration

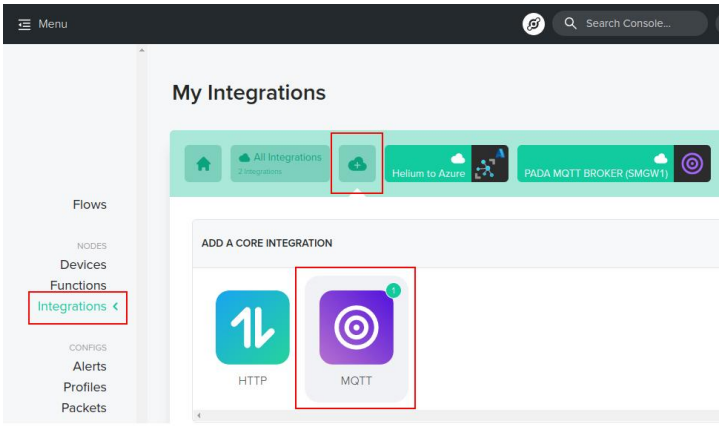
Σε αυτήν την ενότητα θα γίνει προσθήκη MQTT. Θα χρησιμοποιηθεί ο MQTT Broker του εργαστηρίου restqmlab που παρέχει remote access. Για την προσθήκη του MQTT στο Helium Console επιλέγουμε Integrations→Add New Integration →MQTT όπως φαίνεται στην εικόνα 88

Στην συνέχεια εμφανίζει ένα menu όπως φαίνεται στην εικόνα 84

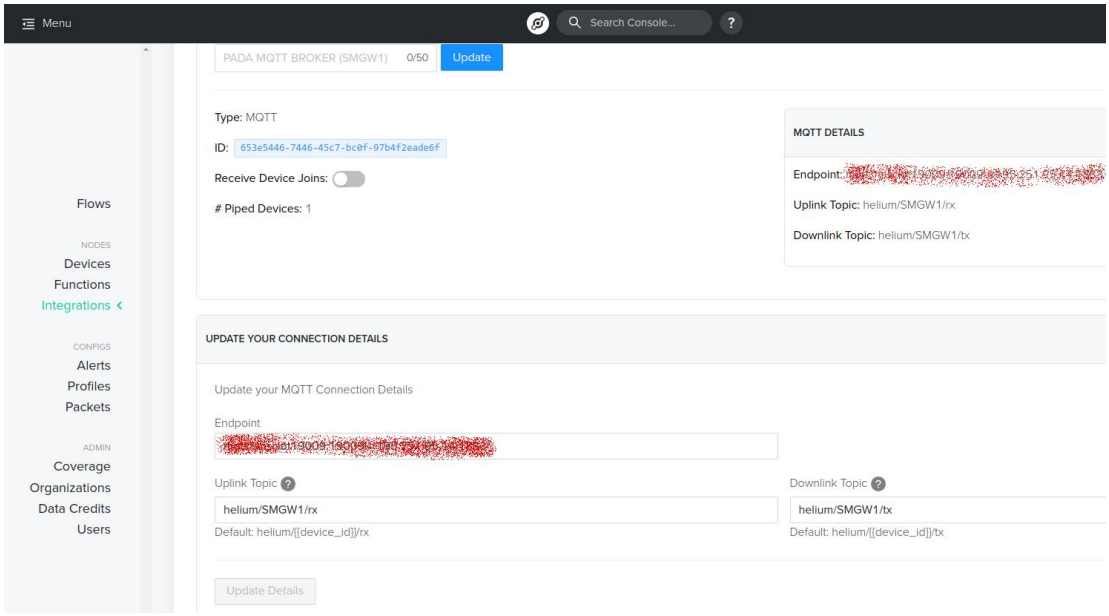
Στο INTEGRATION DETAILS προσθέτουμε τα εξής :

- Ορίζουμε όνομα του integration πχ PADA MQTT BROKER (SMGW1)
- Endpoint : Θα έχει την μορφή mqtt://username:password@την ip του MQTT Broker:1883 ένα παράδειγμα mqtt://test:testpassw@192.168.2.122:1883
- Uplink Topic : helium/SMGW1/rx
- Downlink Topic : helium/SMGW1/tx

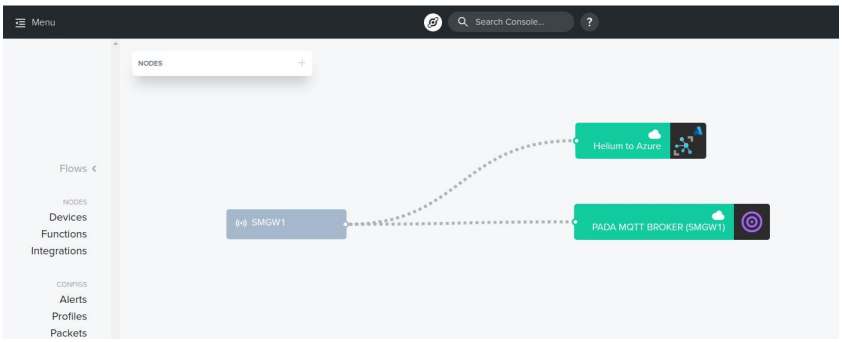
Στο μενού flows συνδέουμε το SGMW1 με το MQTT όπως φαίνεται στην εικόνα 86




Εικόνα 87. Integration : MQTT




Εικόνα 88. Integration : MQTT




Εικόνα 89. Flow Node + MQTT

### Επιλογή Integration

Η επιλογή του Integration θα είναι ο MQTT Broker του εργαστηρίου restqmlab όπου παρέχει απομακρυσμένη διαχείριση. Το Azure IoT Hub θα το αξιοποιήσουμε στο STRESQLab Smart

Metering Server. Δόθηκαν πληροφορίες στην ενότητα Azure IoT Hub Integration επειδή είναι πιο εύκολο στην υλοποίηση. Στο MQTT για απομακρυσμένη διαχείριση θα χρειαστεί port forwarding.

## 5.2 STRESQLab Smart Metering Server

Ο STRESQLab Smart Metering Server δίνεται στην εικόνα 90


Εικόνα 90. Application Server

και αποτελείται από:

- Ενσωματωμένο σύστημα : Raspberry pi4 4GB (OS : Raspbian GNU/Linux 11 (bullseye))
- Software : Node-Red, Azure IoT Hub, Mosquitto broker, Paho-MQTT
- Secure element : Multos Trust Core
- Επέκταση GPIO : Model B3X40 Pin GPIO Adapter Extension Board 1 to 3 GPIO Module

### 5.2.1 MQTT Broker

Το λογισμικό που θα χρησιμοποιηθεί για MQTT Broker είναι το Mosquitto. Το Mosquitto είναι ένας MQTT Broker ανοιχτού κώδικα (με άδεια EPL/EDL) που εφαρμόζει τις εκδόσεις 5.0, 3.1.1 και 3.1 του πρωτοκόλλου MQTT [109]. Στην ενότητα 7.2 του 1ου Παραρτήματος δίνονται οδηγίες για την εγκατάσταση του Mosquitto MQTT Broker.

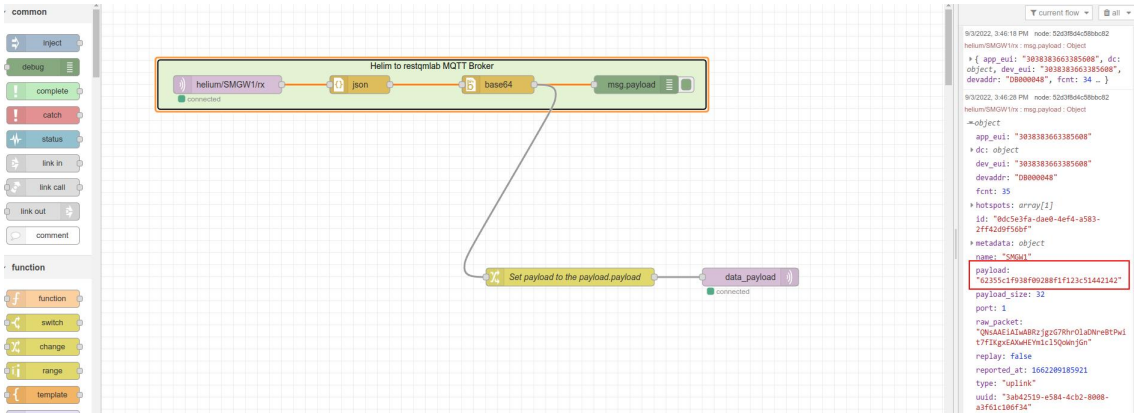
Για την ενσωμάτωση του MQTT Broker στην python θα χρησιμοποιηθεί το Paho Python Client. Το Paho Python Client παρέχει client class με υποστήριξη για MQTT v5.0, MQTT v3.1.1 και v3.1 σε Python 2.7 ή 3.x. Παρέχει επίσης ορισμένες βοηθητικές λειτουργίες για να κάνει τη δημοσίευση μεμονωμένων μηνυμάτων σε έναν διακομιστή MQTT πολύ απλή. Για να εγκαταστήσουμε το paho εκτελούμε την εντολή :

```
pip3 install paho-mqtt.
```

[110]

### 5.2.2 node-red

Ο server εκτελεί το node-red (like a service) και η εικόνα 91 δείχνει την σχεδίαση που ακολουθεί καθώς και το λαμβανόμενο κρυπτογραφημένο μήνυμα από την SMGW1. Στην ενότητα 7.3 του 1ου Παραρτήματος δίνονται οδηγίες για την εγκατάσταση του Node-Red στο Raspberry pi.




Εικόνα 91. Node-Red : Helium to restqmlab MQTT Broker

Το node helium/SMGW1/rx είναι ο MQTT Broker του restqmlab με topic helium/SMGW1/rx. Ακολουθούν οι ρυθμίσεις του node:

- Server : Ρυθμίζουμε τον MQTT Broker και συγκεκριμένα στο πεδίο server την ip του Broker 195.xxx.95.xx, port : 1883, Connect automatically και στο πεδίο security το username και password που έχουμε ορίσει.
- Topic : helium/SMGW1/rx

Το node jason μετατρέπει τα δεδομένα σε μορφή jason. Ακολουθούν οι ρυθμίσεις του node:

```
Action : Convert beetween JSON String & Object
property : msg.payload
```

Το node base64 μετατρέπει το ωφέλιμο φορτίο από τη μορφή base64 σε μορφή συμβολοσειράς (string). Ακολουθούν οι ρυθμίσεις του node:

```
Action : Convert Base64 to String
property : msg.payload.payload
```

Το node Set payload to payload.payload ορίζει ότι το ωφέλιμο φορτίο είναι απλώς το μήνυμα που έχει στείλει ο κόμβος χωρίς άλλες πληροφορίες. Ακολουθούν οι ρυθμίσεις του node:

```
Set : msg.payload
property : msg.payload.payload
```

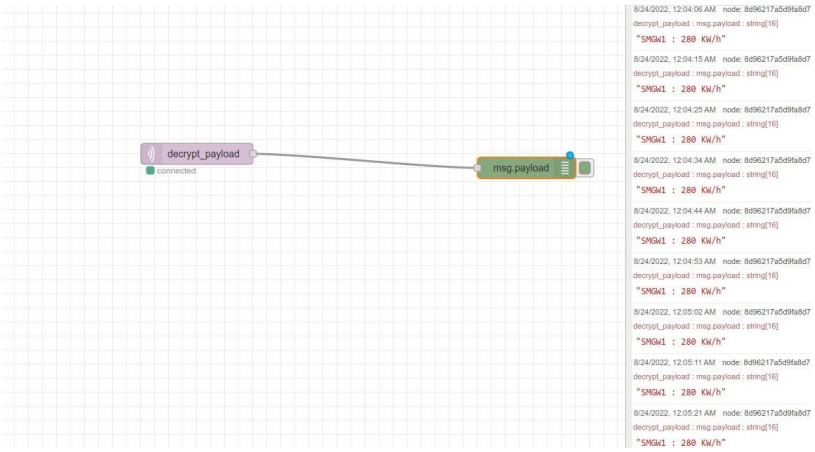
Στην περίπτωση που θέλουμε να χρησιμοποιήσουμε το Azure IoT Hub και όχι το MQTT Broker δίνονται οι ρυθμίσεις στην ενότητα 9.5.2 του 3ου Παραρτήματος

Στη συνέχεια, το ωφέλιμο φορτίο δημοσιεύεται στον restqmlab MQTT Broker στο θέμα (topic) data_payload μέσω του node data_payload. Στη συνέχεια, χρησιμοποιώντας το Paho κάνουμε εγγραφή (subscribe) στο θέμα data_payload έτσι ώστε το Multos να μπορεί να αποκρυπτογραφήσει το ωφέλιμο φορτίο και να δημοσιεύσει το αποκρυπτογραφημένο ωφέλιμο φορτίο στο θέμα decrypt_payload χρησιμοποιώντας τον κώδικα 5.3.1. Να σημειωθεί ότι το κλειδί της κρυπτογράφησης AES-CBC δεν μοιράζετε στον αέρα και εισάγετε στο Multos Trust Core του STRESQLab Smart Metering Server.

```
import paho.mqtt.client as mqtt
import time
import pymultosP11
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

# One time init

pkcs_handle = pymultosP11.init()

# 16 byte ICV needed for AES-CBC

icv = bytes(b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
    )

# ID of key slot containing the AES key to use

key_slot = 2

def on_message(client, userdata, msg):
   payload=str(msg.payload.decode("utf-8"))
   cipher_text = bytearray.fromhex((payload))
   plain_text = pymultosP11.aesDecCbc(pkcs_handle,cipher_text,icv,key_slot)
   print(plain_text.decode())
   client.publish("decrypt_payload", (plain_text.decode()))

mqttBroker ="195.xxx.95.xx" #PADA MQTT BROKER#

client = mqtt.Client("pi4")
client.connect(mqttBroker)

client.loop_start()

client.subscribe("data_payload")
client.on_message=on_message

time.sleep(1)
#client.loop_forever()
```

Κώδικας 8. Paho : αποκρυπτογράφηση ωφέλιμου φορτίου και publish στο θέμα decrypt_payload

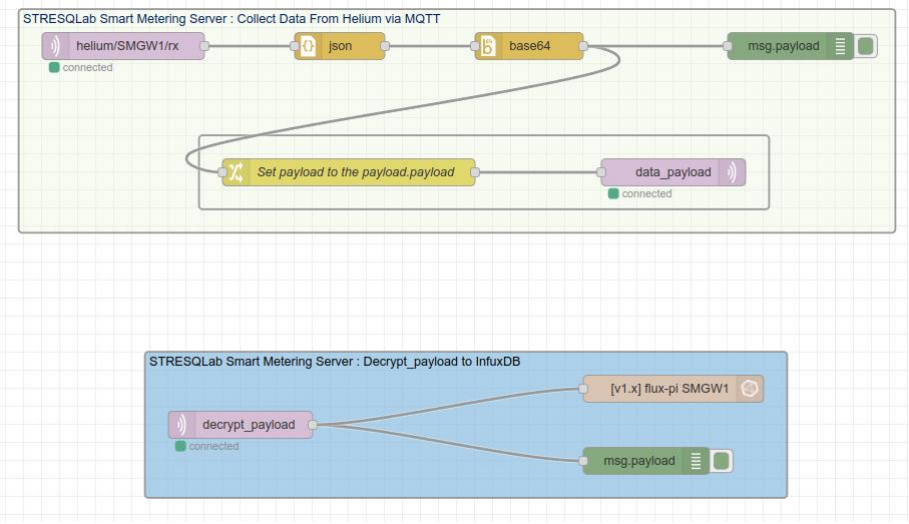
Στο θέμα decrypt_payload δημοσιεύετε το αποκρυπτογραφημένο ωφέλιμο φορτίο. Στην εικόνα 92 φαίνεται η αποκρυπτογράφηση του ωφέλιμου φορτίου της SGMW1.




Εικόνα 92. Node-Red

### 5.2.3 Οπτικοποίηση των δεδομένων

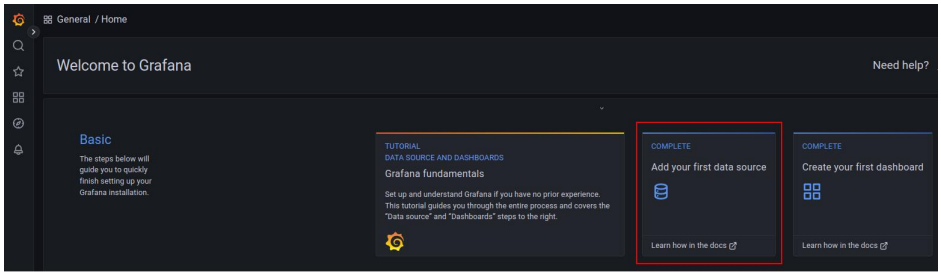
Για την οπτικοποίηση των δεδομένων θα χρησιμοποιηθεί η βάση δεδομένων InfluxDB και το λογισμικό Grafana. Στο Πέμπτο Παράρτημα δίνονται οδηγίες για την εγκατάσταση του InfluxDB την σύνδεση με το node-red, τις απαραίτητες ρυθμίσεις και την εγκατάσταση του Grafana. Έχει δημιουργηθεί μια βάση δεδομένων με το όνομα SMGW_MEASUREMENTS και μια κατηγορία με το όνομα SMGW1. Θα γίνει μια αλλαγή στον κώδικα 7 στην γραμμή 37 data_string = 'SGMW1 : 280 KW/h' σε data_string = 280.000000000000 . Το ωφέλιμο φορτίο θέλουμε να είναι σε μορφή integer και όχι string ώστε να γίνει η οπτικοποίηση στο Grafana. Επίσης χρησιμοποιώντας την βάση δεδομένων InfluxDB έχουμε την πληροφορία ότι το μήνυμα προέρχεται από τον κόμβο SMGW1. Στο ωφέλιμο φορτίο έγινε προσθήκη μηδενικών για να είναι multiple 16 λόγο του περιορισμού στην κωδικοποίηση AES-CBC. Το Grafana αφαιρεί τα μηδενικά. Στην εικόνα 93 δίνεται η σύνδεση του MQTT topic decrypt_payload με το InfluxDB μέσω του node-red. Η μόνη αλλαγή που χρειάζεται είναι με διπλό click στο node decrypt_payload και στο Output επιλέγουμε a parsed JSON object.




Εικόνα 93. Node-Red: Σύνδεση MQTT Broker με το InfluxDB

Εκτελούμε τον τροποποιημένο κώδικα 7 και μέσω browser ανοίγουμε το Grafana (http://192.168.2.23:3000/).

Επιλέγουμε το COMPLETE Add your first data source (εικόνα 94)

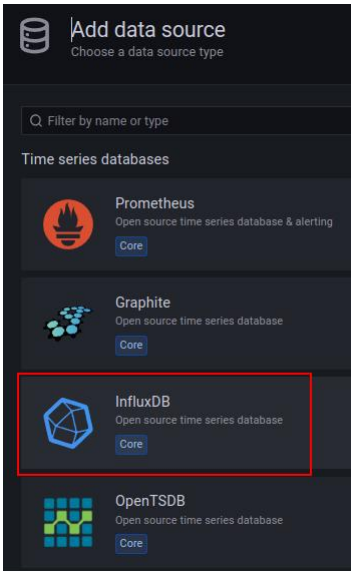



Εικόνα 94. Add your first data source

Και μετά επιλογή InfluxDB (εικόνα 95)

Στην συνέχεια γίνονται οι ακόλουθες ρυθμίσεις:

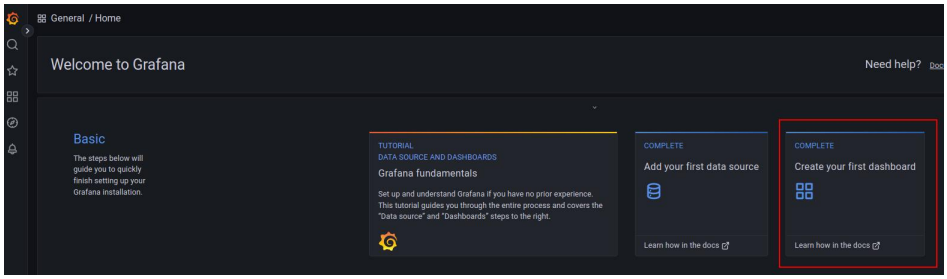
- name : Ορίζουμε όνομα (InfluxDB)




Εικόνα 95. data source - InfluxDB

- Query Language : InfluxQL
- URL : http://localhost:8086/
- database : Το όνομα της βάσης δεδομένων που έχει οριστεί (SMGW_MEASUREMENTS)
- username : Το username που έχει οριστεί
- passowrd : Ο κωδικός που έχει οριστεί

Αφού κάνουμε save πάμε μετά στο menu home και επιλέγουμε Complete Create your first dashboard (εικόνα 96)

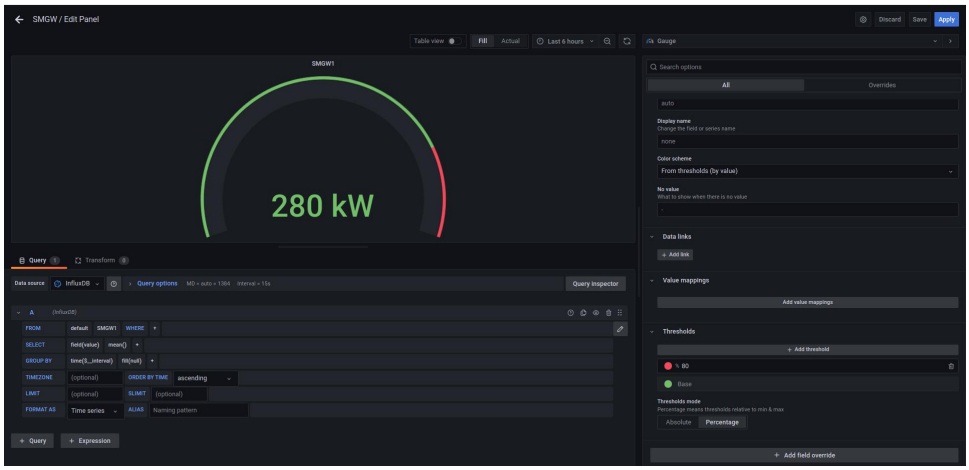



Εικόνα 96. Complete Create your first dashboard

Επιλέγουμε add a new panel και στην συνέχεια γίνονται οι ακόλουθες ρυθμίσεις (εικόνα 96):

- from
  - default
  - SMGW1
- Visualization : Επιλέγουμε το επιθυμητό Visualization πχ Gauge
- Panel Option - Title : Ορίζουμε όνομα (SMGW1)
- Standard options - Unit : KW

- Apply
- Save




Εικόνα 97. Ρυθμίσεις dashboard

Το Dashboard με τις μετρήσεις του κόμβου SMGW1 είναι έτοιμο όπως φαίνεται στην εικόνα 98


Εικόνα 98. Dashboard SMGW1

## 5.3 Επεκτάσεις STRESQLab Smart Metering Server

Σε αυτήν την ενότητα θα συζητηθούν επεκτάσεις του STRESQLab Smart Metering Server για την παροχή υπηρεσιών.

### 5.3.1 Χρήση Multos Trust Core Azure IoT Hub

Σε αυτήν την ενότητα θα δημιουργήσουμε ένα γεφύρωμα (bridge) με τα λαμβανόμενα δεδομένα του STRESQLab Smart Metering Server και αποστολή στο Azure IoT Hub μέσω του Multos Trust Core. Πιο συγκεκριμένα το bridge θα γίνει ανάμεσα στο MQTT Broker του restqmlab στο topic decrypt_payload και στο Azure IoT Hub.

Στην ενότητα 4.8.1 του 4ου Κεφαλαίου έχει γίνει εισαγωγή κλειδιού στο Multos (slot 0) για την χρήση στο IoT Hub και θα δωθεί ο κώδικας 9 του bridge:

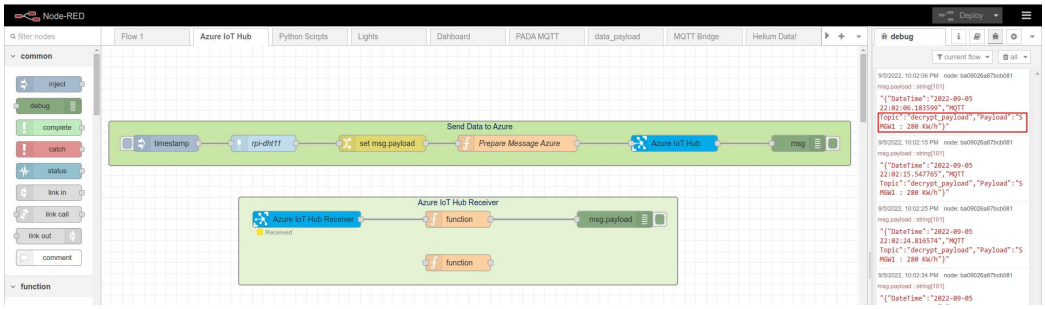
```
import paho.mqtt.client as mqtt
import os
import asyncio
import uuid
import pymultosAzure
from azure.iot.device.aio import IoTHubDeviceClient
from azure.iot.device import Message
from datetime import datetime

# Send a message using multos to enable mqtt
async def main():
    # Set up the MULTOS Trust Core HSM
    if pymultosAzure.init() <= 0:
        print("Failed to initialise MULTOS HSM")
        return

    # Key value is in the HSM slot 0
    connection_string = "HostName=msciot19009.azure-devices.net;DeviceId=mypi4-
    multos;SharedAccessKey=0"

    # Create instance of the device client using the connection string
    device_client = IoTHubDeviceClient.create_from_connection_string(
    connection_string)

    # Connect the device client.
    await device_client.connect()

    # Send a single message
    print("Sending message...")
    await device_client.send_message("Running MQTT")
    print("MQTT Deamon ready!")

    # Finally, shut down the client
    await device_client.shutdown()

if __name__ == "__main__":
    asyncio.run(main())

ScriptVersion = "1.0"
ModifiedDate = "Monday 15, November 2021"
MQTTBrokerIP = "195.251.95.14" #IP Address of your MQTT Broker
MQTTTopicSubscribe = "decrypt_payload" #MQTT Topic Filter
MQTTClientName = "STRESQLab Smart Metering Server" #Used to identify the device
     to your MQTT Broker

async def main():
 # Set up the MULTOS Trust Core HSM
    if pymultosAzure.init() <= 0:
        print("Failed to initialise MULTOS HSM")
        return
```

Κώδικας 9. Bridge MQTT - Azure IoT Hub

Τρέχοντας τα pyhton scipt γίνεται η αποκρυπτογράφηση του ωφέλιμου φορτίου και publish στο θέμα decrypt_payload και 9 γίνεται το Bridge και τα λαμβανόμενα δεδομένα στο θεμα decrypt_payload αποστέλνονται στο Azure IoT Hub μέσω ασφαλής σήραγγας. Στην ενότητα 9.5.1 του 3ου Παραρτήματος δίνονται οδηγίες για την εγκατάσταση του Azure IoT Hub στο node-red και οι ρυθμίσεις για το node Azure IoT Hub Receiver.

Στην εικόνα 99 φαίνονται τα μηνύματα που έχουν αποσταλεί στο IoT Hub απο το topic decrypt_payload.

Με την χρήση του Azure IoT Hub μπορούν να υλοποιηθούν διάφορες εφαρμογές με γραφή-




Εικόνα 99. Azure Results

ματα για χρήση προς υπηρεσίες κοινής ωφέλειας, καταναλωτές, κλπ που μπορούν να εκτελεστούν σε διάφορες πλατφόρμες όπως Android,iOS,windows,linux,web.

Σε αυτήν την ενότητα επιτύχαμε την δημιουργία ασφαλής επικοινωνίας ανάμεσα στον STRESQLab Smart Metering Server και στο Azure IoT Hub για μελλοντικές προεκτάσεις στο πεδίο των εφαρμογών.

### 5.3.2 Χρήση Ethereum και smart contracts

Η SMGW υποστηρίζει το metamask ώστε να μπορεί να εκτελέσει smart contracts. Στο 4ο Παραρτήμα δίνονται οδηγίες για την εγκατάσταση του metamask. Στον κώδικα 2 υλοποιείτε ένα smart contract για την αποστολή ενός απλού μήνυματος Hello World!

Θα είναι χρήσιμο να αναπτυχθούν smart contract’s όπου θα συγκεντρώνουν τα δεδομένα από τις SMGW’s, θα βοηθούν στην μοντελοποίηση της προσφοράς και ζήτησης της ηλεκτρικής ενέργειας και θα εκτελούν συναλλαγές πληρωμής προς τους παραγωγούς ηλεκτρικής ενέργειας από ΑΠΕ.

# ΚΕΦΑΛΑΙΟ 6

## Συμπεράσματα και Μελλοντικές Προεκτάσεις

Στην παρούσα διπλωματική εργασία πραγματοποιήθηκε η ανάπτυξη ευέλικτης και ασφαλούς πύλης δικτύου ανοικτού κώδικα για την υλοποίηση έξυπνου μετρητή ενέργειας. Έγινε μελέτη στις πιο διαδεδομένες τεχνολογίες LPWAN, το Sigfox, το NB-IoT και το LoRa. Έγινε σύγκριση και επιλογή δικτύου για την συνδεσιμότητα της SMGW και την υποστήριξη διαδικασιών τηλεμετρίας. Το Sigfox για την συγκεκριμένη εφαρμογή έχει κάποιους περιορισμούς σε σχέση με το LoRa και NB-IoT όπως το μήκος ωφέλιμου φορτίου για την ανοδική ζεύξη είναι 12 bytes, ένα τυπικό μέγεθος μηνύματος έξυπνου μετρητή είναι περίπου 100 byte άρα το LoRa και το NBIoT είναι πιο κατάλληλο για εφαρμογές έξυπνου μετρητή ενέργειας. Το LoRa είναι ελεύθερο δεν απαιτεί άδεια φάσματος και το κόστος εγκατάστασης είναι χαμηλό το μέγιστο μήκος ωφέλιμου φορτίουείναι 243 byte. Για τους αναφερθέν λόγους επιλέχθηκε το δίκτυο LoRa για την συνδεσιμότητα της SMGW. Το NB-IoT παρέχει μεγαλύτερη ασφάλεια σε σχέση με το LoRa αλλά σκοπός είναι να διερευνηθούν λύσεις που θα εξασφαλίζουν επιπλέον ασφάλεια στο LoRa.

Για ενσωματωμένο σύστημα επιλέχθηκε το Raspberry pi zero 2w. Το Raspberry pi zero 2w αποτελεί μια οικονομική λύση με κόστος που κυμαίνετε 15-20 €, διαθέτει τετραπύρηνο επεξεργαστή Arm Cortex-a53 όπου είναι επαρκής για την υλοποίηση της SMGW από μεριάς υπολογιστικής ισχύος. Επίσης το Raspberry είναι linux embedded system με πλεονέκτημα την επεκτασιμότητα.

Για την διασύνδεση με το δίκτυο LoRa επιλέχθηκε το module LoRa Node pHAT το οποίο υποστηρίζει την σειρά pi zero δίνοντας μεγάλο πλεονέκτημα στο συγκεκριμένο module καθώς μπορούν να υλοποιηθούν κόμβοι LoRa με χαμηλό κόστος. Στην συνέχεια έγινε σύγκριση των δυο πιο διαδεδομένων LoRa server, του TTN και του Chirpstack και επιλογή του Chirpstack καθώς αξιοποιεί όλο το εύρος στο ωφέλιμο φορτίο (243 bytes) του ανοδικού καναλιού και δεν περιορίζεται στα 51 bytes ανά μήνυμα όπως στο TTN, δεν υπάρχουν περιορισμοί στο κανάλι καθόδου όπως στο TTN που υπάρχει περιορισμός στα 10 μηνύματα την ημέρα ανά κόμβο.

Στην συνέχεια έγινε έρευνα τεχνολογιών Blockchain για διαδικασίες διασφάλισης της ακεραιότητας των δεδομένων. Πιο συγκεκριμένα αναλύθηκαν δυο τεχνολογίες Blockchain, το Ethereum και το Helium. Το Helium Blockchain είναι το μεγαλύτερο δημόσιο αποκεντρωμένο δίκτυο LoRaWAN, μπορούν να συνδεθούν οι κόμβοι LoRa, διασφαλίζει ότι όλες οι επικοινωνίες στο blockchain έχουν κρυπτογράφηση από άκρο σε άκρο καθιστώντας το πιο κατάλληλο για ευαίσθητες πληροφορίες, υπάρχει μεγάλη κάλυψη δικτύου από hotspot's όπου μπορούμε να συνδέσουμε τον κόμβο LoRa και αυτό οδηγεί σε μείωση κόστος υλικού αφού δεν απαιτεί να διαθέτουμε hotspot. Επίσης έχει χαμηλά τέλη συναλλαγής για κάθε 24 bytes είναι 0.00001 $. Το Ethereum Blockchain έχει υψηλά τέλη συναλλαγής για παράδειγμα για αποστολή ενός μηνύματος ανάλογα την τρέχουσα ισοτιμία μπορεί να κυμαίνεται στα 5 €. Το Ethereum δεν είναι ιδανικό για την αποστολή δεδομέ-

νων όμως δίνει την δυνατότητα για την δημιουργία smart contracts, dapps που δεν υποστηρίζονται ακόμα από το Helium Blockchain.

Το Blockchain από μόνο του δεν είναι σε θέση να εξασφαλίσει πλήρως μια συναλλαγή γιατί εγγυάται μόνο την αναλλοίωτη κατάσταση των δεδομένων, ενώ στις περισσότερες περιπτώσεις τα δεδομένα πρέπει να είναι ασφαλή στο σημείο παραγωγής. Για να καλυφθεί αυτό το κενό ασφαλείας, έγινε χρήση του Secure Element Multos Trust Core. Συγκεκριμένα με την χρήση του Multos Trust Core κρυπτογραφείται το ωφέλιμο φορτίο με κρυπτογράφηση AES-CBC πριν μεταδοθεί στο Helium και με αυτόν τον τρόπο αντιμετωπίστηκε η ευπάθεια αν κάποιος κακόβουλος χρήστης καταφέρει να ανακτήση τα κλειδια NwkSKey και AppSKey να μπορέσει να αποκρυπτογραφήσει το ωφέλιμο φορτίο.

Για μελλοντική επέκταση του συστήματος στο πεδίο των εφαρμογών προτάθηκε χρήση του Azure IoT Hub, που προσφέρει μια αξιόπιστη και ασφαλή αμφίδρομη επικοινωνία μεταξύ συσκευών και του cloud και λειτουργεί ως κεντρικός κόμβος μηνυμάτων για την επικοινωνία μεταξύ μιας εφαρμογής IoT και των συνδεδεμένων συσκευών της και δόθηκε λύση στο πεδίο της ασφάλειας με την χρήση του Multos Trust Core για την διασύνδεση του με το Azure IoT Hub.

Όταν θα ολοκληρωθεί η αναβάθμιση του Ethereum, στο Ethereum 2 όπου θα γίνει μετάβαση από τον αλγόριθμο proof-of-work στον αλγόριθμο proof-of-stake με σκοπό την μείωση της κατανάλωσης ηλεκτρικής ενέργειας, των τελών συναλλαγής, και επέκταση στη δημιουργία των smart contracts και αποκεντρωμένων εφαρμογών dapps, είναι σκόπιμο να γίνει μελέτη για την επεκτασιμότητα της SMGW στον τομέα των αποκεντρωμένων εφαρμογών, έξυπνων συμβολαίων με στόχο την υλοποίηση του Διαδικτύου της Ενέργειας.

ΚΕΦΑΛΑΙΟ 7

# Πρώτο παράρτημα

Στο πρώτο παράρτημα θα γίνει εγκατάσταση του λειτουργικού Raspbian OS, βασικών ρυθμίσεων, εγκατάσταση του Node-Red και βιβλιοθηκών της Python.

## 7.1 Εγκατάσταση Raspbian OS

Από το site www.raspberrypi.com/software/ κατεβάζουμε το πρόγραμμα Pi Imager. Ανοίγουμε το Pi Imager και από το μενού Operating System επιλέγουμε το Raspberry Pi OS 32 bit. Στο εικονίδιο γρανάζι ενεργοποιούμε το πρωτόκολλο SSH, ορίζουμε username και password για το login, γράφουμε το ssid και κωδικό για χρήση wifi του CPE. Στο μενού Storage ορίζουμε την διαδρομή της microSD και πατάμε στο πεδίο Write για να ξεκινήσει η εγγραφή (100).


Εικόνα 100. Εγκατάσταση Raspbian OS

### 7.1.1 Σύνδεση στο raspberry από άλλο υπολογιστή τοπικά

Η σύνδεση με το raspberry μπορεί επίσης να γίνει τοπικά από υπολογιστή. Για Windows OS μέσω του PuTTY και για linux κατευθείαν μέσω terminal με την μορφή της παρακάτω εντολής :

```
ssh pi@192.168.2.7
```

# Βάζουμε το password που έχουμε ορίσει κατά την εγκατάσταση (το default είναι raspberry).

### 7.1.2 Ορίζουμε static ip

Θα ορίσουμε static ip ξεκινώντας από την παρακάτω εντολή :

```
sudo nano /etc/dhcpcd.conf
```

Και μετά στο πεδίο # Example static IP configuration: γράφουμε τις παρακάτω γραμμές.

```
# Example static IP configuration:
interface wlan0
static ip_address=192.168.2.7/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.2.1
static domain_name_servers=192.168.2.1 #8.8.8.8 fd51:42f8:caae:d92e::1
```

Στη συνέχεια, πατάμε CTRL-X για έξοδο και αποθήκευση του αρχείου. Πατάμε Y και Enter.

### 7.1.3 Διαθέσιμες ενημερώσεις

Για να γίνουν οι διαθέσιμες ενημερώσεις στο σύστημα εκτελούμε την παρακάτω εντολή :

```
sudo apt-get update && sudo apt-get upgrade
```

## 7.2 Εγκατάσταση Mosquitto MQTT Broker

Για να γίνει εγκατάσταση του MQTT Broker εκτελούμε τις παρακάτω εντολές:

```
$ sudo apt-get install mosquitto
$ sudo apt-get install mosquitto-clients
```

Όταν ολοκληρωθεί η εγκατάσταση εκτελούμε την εντολή :

```
sudo nano /etc/mosquitto/mosquitto.conf
```

Και προσθέτουμε στο αρχείο τις παρακάτω γραμμές:

```
allow_anonymous true
listener 1883
```

Στη συνέχεια, πατάμε CTRL-X για έξοδο και αποθήκευση του αρχείου. Πατάμε Y και Enter.

Με την παρακάτω εντολή ορίζουμε το mosquitto σαν service και εκτελείτε αυτόματα με το που κάνει boot στο λειτουργικό σύστημα:

```
sudo systemctl enable mosquitto.service
```

## 7.3 Εγκατάσταση Node Red

Για να γίνει εγκατάσταση του node-red στο raspberry ακολουθούμε τα παρακάτω βήματα.

```
$ bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-
installers/master/deb/update-nodejs-and-nodered)
$ sudo apt install build-essential git curl
```

Με την παρακάτω εντολή ορίζουμε το node-red σαν service:

```
sudo systemctl enable nodered.service
```

### 7.3.1 Εγκατάσταση node Azure IoT Hub

Μέσω terminal εγκαθιστούμε τα απαραίτητα πακέτα με την παρακάτω εντολή:

```
npm install node-red-contrib-azure-iot-hub
```

## 7.4 Εγκατάσταση Βιβλιοθηκών για Python

Μέσω terminal εγκαθιστούμε τις βιβλιοθήκες python με τις παρακάτω εντολές:

```
$ sudo pip3 install azure-iot-device
$ sudo pip3 install azure-iot-hub

$ sudo pip3 install asyncio
```

# ΚΕΦΑΛΑΙΟ 8

## Δεύτερο παράρτημα

Στο δεύτερο παράρτημα θα γίνει εγκατάσταση του Chirpstack σύμφωνα με τις οδηγίες της επίσημης ιστιοσελίδας του. [111]

## 8.1 Εγκατάσταση απαιτούμενων εξαρτήσεων

Πρώτα θα γίνει εγκατάσταση των παρακάτω εξαρτήσεων:

- Mosquitto : MQTT Broker
- Redis - Μια βάση δεδομένων στη μνήμη που χρησιμοποιείται για την αποθήκευση σχετικά παροδικών δεδομένων.
- PostgreSQL - Η βάση δεδομένων μακροπρόθεσμης αποθήκευσης που χρησιμοποιείται από τα πακέτα ανοιχτού κώδικα.

Εκτελώντας από terminal την παρακάτω εντολή:

```
sudo apt install mosquitto mosquitto-clients redis-server
redis-tools postgresql
```

Στο πρώτο παράρτημα έχουμε κάνει εγκατάσταση στο Mosquitto οπότε αν έχουμε ακολουθήσει τα βήματα του πρώτου παραρτήματος από terminal θα έχουμε απάντηση ότι είναι εγκατεστημένο και θα κάνει εγκατάσταση στα υπόλοιπα.

Στην συνέχεια :

```
sudo -u postgres psql
```

Και μετά :

```
-- set up the users and the passwords
-- (note that it is important to use single quotes and a semicolon at the end!)
create role chirpstack_as with login password 'dbpassword';
create role chirpstack_ns with login password 'dbpassword';
```

```
-- create the database for the servers
create database chirpstack_as with owner chirpstack_as;
create database chirpstack_ns with owner chirpstack_ns;

-- change to the ChirpStack Application Server database
\c chirpstack_as

-- enable the pq_trgm and hstore extensions
-- (this is needed to facilitate the search feature)
create extension pg_trgm;
-- (this is needed to store additional k/v meta-data)
create extension hstore;

-- exit psql
\q
```

## 8.2 Εγκατάσταση ChirpStack software repository

Θα γίνει εγκατάσταση για το software repository του ChirpStack ακολουθώντας τις παρακάτω εντολές:

```
$ sudo apt install apt-transport-https dirmngr
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
1CE2AFD36DBCCA00
$ sudo echo "deb https://artifacts.chirpstack.io/packages/3.x/deb
stable main" | sudo tee /etc/apt/sources.list.d/chirpstack.list
$ sudo apt update
```

## 8.3 Εγκατάσταση ChirpStack Gateway Bridge

Θα γίνει εγκατάσταση του Chirpstack Gateway Bridge ακολουθώντας τις παρακάτω εντολές:

```
$ sudo apt install chirpstack-gateway-bridge
$ sudo systemctl start chirpstack-gateway-bridge
$ sudo systemctl enable chirpstack-gateway-bridge
```

## 8.4 Εγκατάσταση the ChirpStack Network Server

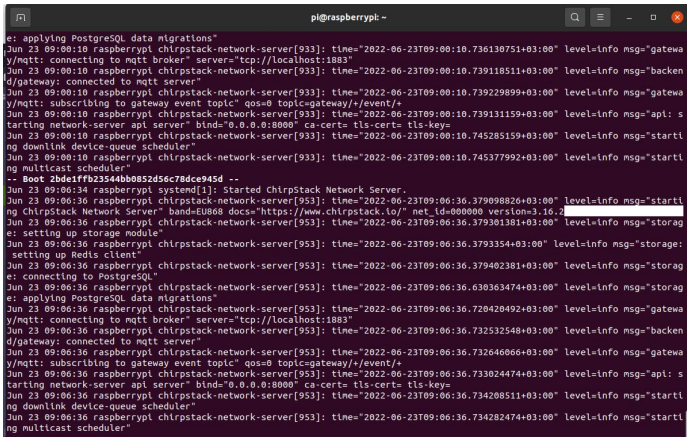
Θα γίνει εγκατάσταση του ChirpStack Network Server ακολουθώντας τις παρακάτω εντολές:

```
$ sudo apt install chirpstack-network-server
$ sudo systemctl start chirpstack-network-server
$ sudo systemctl enable chirpstack-gateway-bridge
```

Με την παρακάτω εντολή φαίνεται το log-output του Network Server

```
sudo journalctl -f -n 100 -u chirpstack-network-server
```

Αν έχουν γίνει σωστά τα βήματα στο terminal θα έχει το αποτέλεσμα όπως στην εικόνα 101.




Εικόνα 101. Log-output του Network Server

## 8.5 Εγκατάσταση ChirpStack Application Server

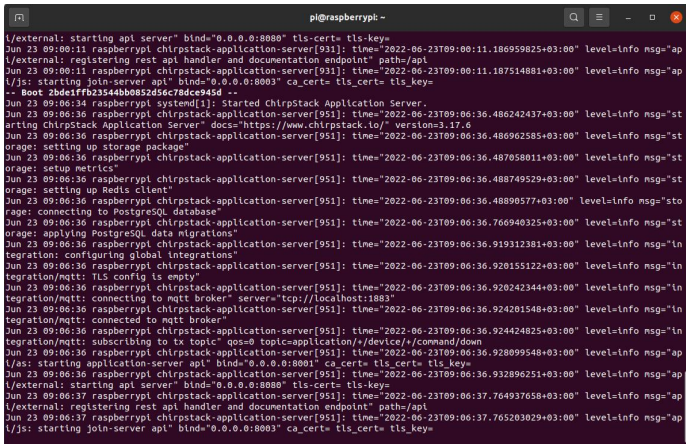
Θα γίνει εγκατάσταση του ChirpStack Application Server ακολουθώντας τις παρακάτω εντολές:

```
$ sudo apt install chirpstack-application-server
$ sudo systemctl start chirpstack-application-server
$ sudo systemctl enable chirpstack-application-server
```

Με την παρακάτω εντολή φαίνεται το log-output του Application Server

```
sudo journalctl -f -n 100 -u chirpstack-application-server
```

Αν έχουν γίνει σωστά τα βήματα στο terminal θα έχει το αποτέλεσμα όπως στην εικόνα 102.




Εικόνα 102. Log-output του Application Server

ΚΕΦΑΛΑΙΟ 9

# Τρίτο παράρτημα

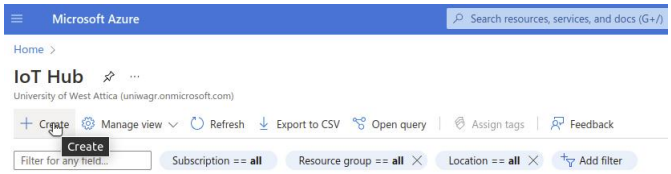
Στο τρίτο παράρτημα θα γίνει εγκατάσταση του Azure IoT Hub.

## 9.1 Εγκατάσταση IoT Hub

Μπορούμε να χρησιμοποιήσουμε διάφορους τρόπους για την δημιουργία ενος IoT Hub όπως για παράδειγμα Azure Portal, Azure CLI, ή PowerShell. Στην συγκεκριμένη ενότητα θα γίνει δημιουργία μέσω Azure Portal καθώς παρέχει ένα φιλικό γραφικό περιβάλλον.

Τα αρχικά βήματα που ακολουθούμε :

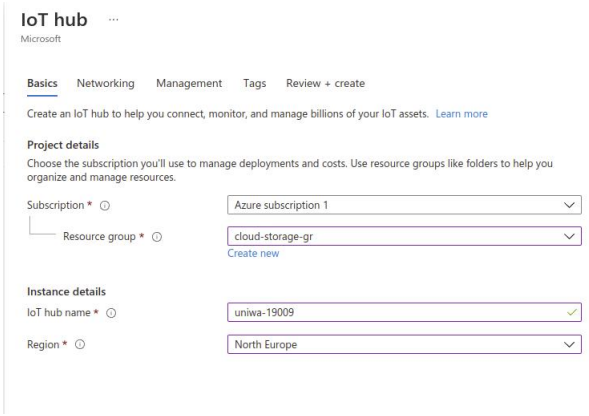
- Δημιουργούμε έναν λογαριασμό στο Microsoft Azure portal `https://azure.microsoft.com/en-us/free/search`
- Από το μενού Azure services επιλέγουμε το IoT Hub και μετά την επιλογή create όπως φαίνεται στην εικόνα 103.




Εικόνα 103. Δημιουργία ενός IoT Hub

Όταν επιλέξουμε create θα εμφανίσει ένα μενού όπως φαίνεται στην εικόνα 104
Ρυθμίζουμε στο μενού Basic τα υπομενού

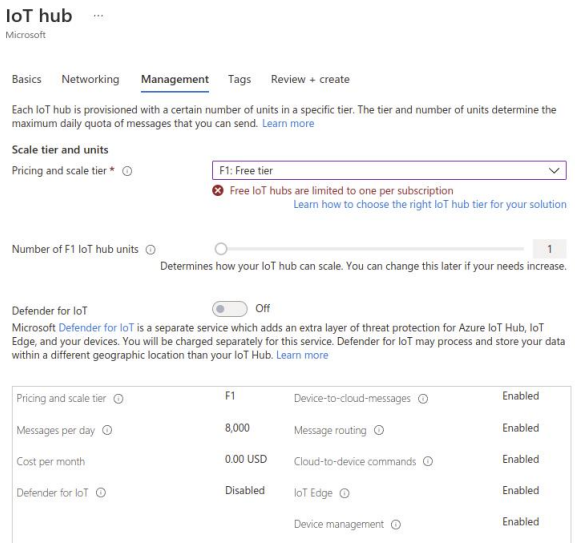
- Subscription : Επιλέγουμε την συνδρομή που έχουμε δημιουργήσει (Azure Subscription 1)
- Resource group : Δημιουργούμε ένα group που θα μοιράζεται τα ίδια δικαιώματα, πολιτικές και τον ίδιο κύκλο ζωής. Επιλέγουμε create δίνουμε ένα όνομα (cloud-storage-gr)
- IoT hub name : Το όνομα του IoT Hub (uniwa19009)
- Region : Επιλέγουμε την πιο κοντινή περιοχή (North Europe)




Εικόνα 104. Ρυθμίσεις IoT Hub

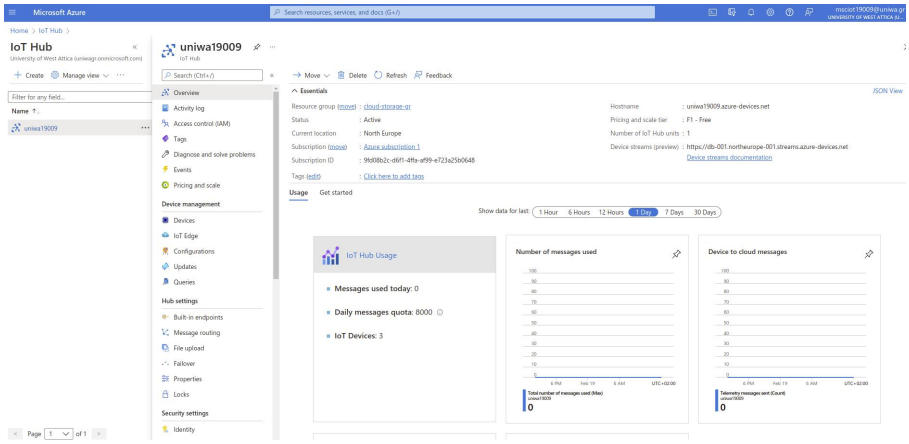
Μετά στο μενού Management στην επιλογή Scale tier and units (εικόνα 105) διαλέγουμε την συνδρομή σύμφωνα με τις απαιτήσεις που θέλουμε. Υπάρχει η δωρεάν επιλογή που παρέχει :

- 8000 μηνύματα ανά ημέρα
- Device-to-cloud-μηνύματα
- Δρομολόγηση μηνυμάτων
- Cloud-to-device εντολές
- IoT Edge
- Διαχείριση συσκευών




Εικόνα 105. Scale tier and units

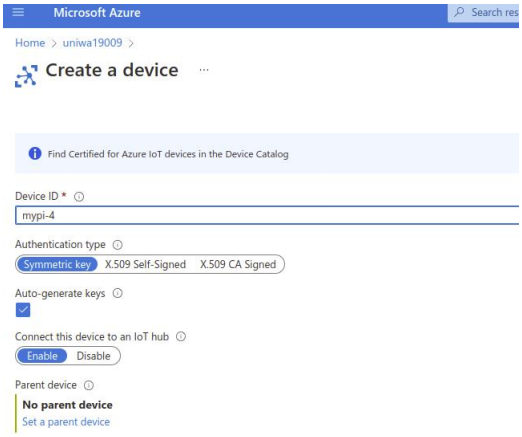
Στην εικόνα 105 εμφανίζει μήνυμα λάθος επειδή έχω δημιουργήσει ήδη ένα IoT Hub και η δωρεαν συνδρομή είναι μέχρι ένα IoT Hub. Μετά επιλέγουμε το Review + create και δημιουργούμε το IoT Hub. Όπως φαίνεται στην εικόνα 106 έχουμε δημιουργήσει το IoT Hub με το όνομα uniwa19009.




Εικόνα 106. IoT Hub - uniwa19009

## 9.2 Δημιουργία συσκευής στο IoT Hub

Στο περιβάλλον του IoT Hub μέσω του μενού Device management και add device μπορούμε να προσθεσουμε συσκευές. Στην εικόνα 107 φαίνεται η δημιουργία μιας νέα συσκευής.



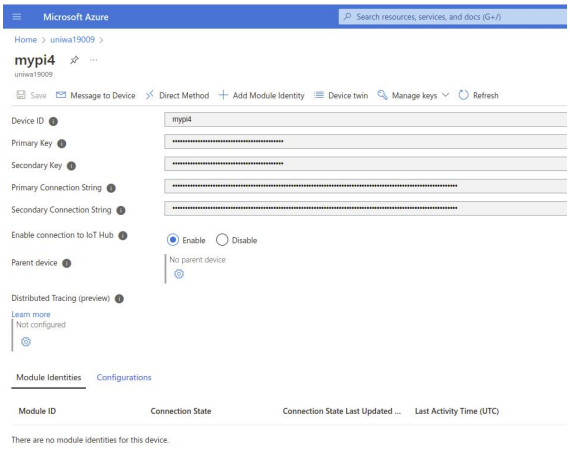

Εικόνα 107. δημιουργία μιας νέα συσκευής

Στις επιλογές :

- Device ID : Επιλέγουμε το όνομα της συσκευής (mypi4)
- Authentication type : Τον τύπο ελέγχου ταυτότητας (Symmetric key). Επίσης παρέχει την δυνατότητα X.509 certificates.
- Connect this device to an IoT hub : Ενεργοποιούμε ή απενεργοποιουμε την αλληλεπίδραση της συσκευής με το IoT Hub (Enable)
- Save : Δημιουργούμε την συσκευή.

Έχουμε δημιουργήσει την συσκευή mypi4 και στο μενού Device management → Devices εμφανίζεται η συσκευή mypi4 και κάνοντας click πάνω στην συσκευή εμφανίζονται πληροφορίες όπως φαίνεται στην εικόνα 108.

Συνοπτικά οι πληροφορίες είναι :




Εικόνα 108. mypi4

- Device ID : Ταυτότητα συσκευής που χρησιμοποιείται για τον έλεγχο ταυτότητας της συσκευής και τον έλεγχο πρόσβασης
- Primary Key : Το πρωτεύον συμμετρικό κοινόχρηστο κλειδί πρόσβασης είναι αποθηκευμένο σε μορφή base64
- Secondary Key : Το δευτερεύον συμμετρικό κοινόχρηστο κλειδί πρόσβασης είναι αποθηκευμένο σε μορφή base64 Primary Connection String : Συμβολοσειρά σύνδεσης που βασίζεται στο πρωτεύον κλειδί που χρησιμοποιείται σε κλήσεις API, το οποίο επιτρέπει στη συσκευή να επικοινωνεί με το IoT Hub
- Secondary Connection String : Συμβολοσειρά σύνδεσης που βασίζεται σε δευτερεύον κλειδί που χρησιμοποιείται σε κλήσεις API που επιτρέπει στη συσκευή να επικοινωνεί με το IoT Hub
- Enable connection to IoT Hub : Ενεργοποιούμε ή απενεργοποιουμε την αλληλεπίδραση της συσκευής με το IoT Hub

## 9.3 Σύνδεση raspberry pi4 με το Azure IoT Hub

Στην συνέχεια μέσω terminal κάνουμε εγκατάσταση βιβλιοθηκών που παρέχουν λειτουργικότητα για επικοινωνία με το Azure IoT Hub τόσο για συσκευές όσο και για μονάδες.

```
$ sudo pip3 install azure-iot-device
$ sudo pip3 install azure-iot-hub
```

Οπως θα δούμε παρακάτω θα μπορούμε να χρησιμοποιήσουμε την εντολή μέσω python :

```
from azure.iot.device import IoTHubDeviceClient
```

Στην συνέχεια κατεβάζουμε ένα repository από το github που περιέχει κώδικα python για να συνδεθούν συσκευές IoT με το Azure IoT Hub με την παρακάτω εντολή:

```
$ git clone https://github.com/Azure/azure-iot-sdk-python.git
```

Κάνουμε unzip τον φάκελο.

## 9.4 Αποστολή ενός απλού μηνύματος

Θα γίνει αποστολή ενός απλού μηνύματος Hello World!. Μέσω terminal πάμε στον φάκελο :

```
$ cd /home/pi/azure-iot-sdk-python/azure-iot-device/samples
```

Και κάνουμε edit το αρχείο simple send message.py με την εντολή:

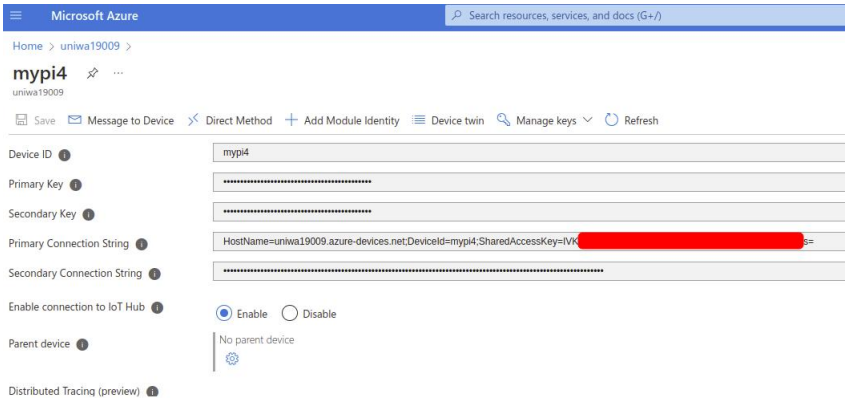
```
$ sudo nano simple\send\message.py
```

Ο κώδικας 10 δίνεται παρακάτω

```
# --------------------------------------------------------------------------
# Copyright (c) Microsoft Corporation. All rights reserved.
# Licensed under the MIT License. See License.txt in the project root for
# license information.
# --------------------------------------------------------------------------

import os
import asyncio
from azure.iot.device.aio import IoTHubDeviceClient

CONNECTION_STRING = "HostName=uniwa19009.azure-devices.net;DeviceId=mypi4;
    SharedAccessKey=
ThisIsnoTrue/TestonlyN/EEVYMkpGHCykox9FSJJuHs="

async def main():
    # Fetch the connection string from an enviornment variable
    conn_str = os.getenv("IOTHUB_DEVICE_CONNECTION_STRING")

    # Create instance of the device client using the connection string
    device_client = IoTHubDeviceClient.create_from_connection_string(
    CONNECTION_STRING)

    # Connect the device client.
    await device_client.connect()

    # Send a single message
    print("Sending message...")
    await device_client.send_message("Hello World!")
    print("Message successfully sent!")

    # Finally, shut down the client
    await device_client.shutdown()


if __name__ == "__main__":
    asyncio.run(main())

    # If using Python 3.6 or below, use the following code instead of asyncio.
    run(main()):
    # loop = asyncio.get_event_loop()
    # loop.run_until_complete(main())
    # loop.close()
```

Κώδικας 10. Αποστολή Hello World στο IoT Hub

Από το Azure IoT Hub κάνουμε click στην συσκευή mypi4 και κάνουμε copy το Primary Connection String (εικόνα 109) και paste στο CONNECTION STRING (στο αρχείο simple send message.py). Μετά στο send a single message και συγκεκριμένα στο await device client.send message γράφουμε το μήνυμα που θέλουμε να στείλουμε πχ Hello World! await device client.send message(”Hello World!”)




Εικόνα 109. Copy the Primary Connection String

Πατάμε Ctl + O για να κάνουμε save το αρχείο και ctl + X για να κάνουμε exit.

Πριν εκτελούμε το αρχείο simple_send_message.py από το Azure Portal κάνουμε click στην συσκευή mypi4 και πάνω δεξιά click cloud shell όπως φαίνεται στην εικόνα 110


Εικόνα 110. Azure Cloud Shell

Και στο terminal που εκτελεί το azure γράφουμε την εντολή :

```
$ az iot hub monitor-events --hub-name uniwa19009 --device-id mypi4
```

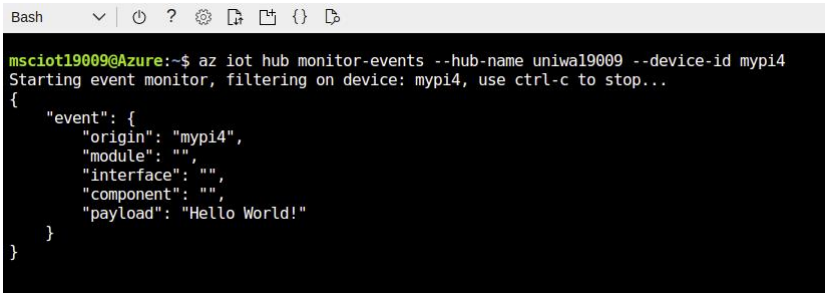
Και ξεκινάει η παρακολούθηση συμβάντων για την συσκευή mypi4. Μετά πάμε στο terminal του raspberry και με την παρακάτω εντολή εκτελούμε το αρχείο simple_send_message.py:

```
$ az iot hub monitor-events --hub-name uniwa19009 --device-id mypi4
```

Και εμφανίζει το μήνυμα :

```
Sending message...
Message successfully sent!
```

Και αν πάμε στο terminal του azure εμφανίζει το μήνυμα Hello World! όπως φαίνεται στην εικόνα 111




Εικόνα 111. Hello World!

**Στο IoT σύστημα μας θα χρησιμοποιήσουμε το node-red για την λήψη των μηνυμάτων και όχι το terminal του Azure. Με αυτόν τον τρόπο δεν δεν υπάρχει καθόλου χρέωση και εκμεταλλευόμαστε το Azure IoT Hub σαν ένα "remote mqtt broker" .** Η οπτικοποίηση (dashboard) μπορεί να γίνει μέσω του node-red.

## 9.5 Node-Red και Azure IoT HuB

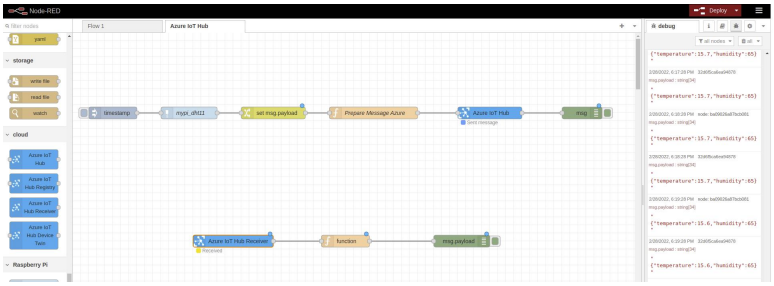
Σε αυτήν την ενότητα θα συνδέσουμε το Azure IoT Hub στο Node-Red. Θα δοθούν 2 μεθόδοι , με την μέθοδο node Azure IoT Hub και node mqtt.

### 9.5.1 Μέθοδος node Azure IoT Hub

Μέσω terminal εγκαθιστούμε τα απαραίτητα πακέτα:

```
npm install node-red-contrib-azure-iot-hub
```

Στην εικόνα 112 δίνεται το flow του node-red χρησιμοποιώντας τα node του Azure IoT Hub



Εικόνα 112. Flow Azure IoT Hub (Send Data and monitoring)

Ας δούμε τις ρυθμίσεις στα nodes: Στο node set msg.payload εισάγουμε:

```
{
  "data": {
  "temp" : $number(payload),
  "humidity" : $number(humidity)
}
```

Στο node Prepare Message Azure εισάγουμε:

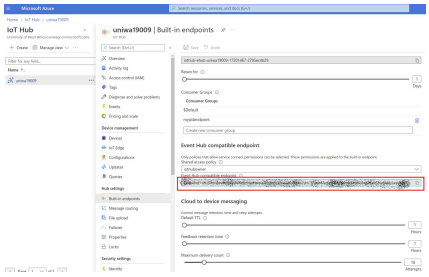
```
msg.payload = {
    'deviceId' : "node-red",
    'key' : "To primary key
    tis siskevis mas",
    'protocol' : "mqtt",
    'data' : msg.payload
}
return msg;
```

Στο node Azure IoT Hub επιλέγομε:

- Protocol : MQTT
- Hostname : uniwa19009.azure-devices.net

Τα κάτω nodes είναι για να κάνουμε monitoring to iot hub :
Στο node Azure IoT Hub Receiver επιλέγομε:

- connectionString : Την τιμή Event Hub compatible endpoint. Την συγκεκριμένη τιμή την βρίσκουμε απο το azure portal → IoT Hub → uniwa19009 → Build-in endpoints → Event Hub compatible endpoint (εικόνα 113).



Εικόνα 113. end-point

Στο node function πάμε στο menu On Message και γράφουμε:

```
msg.payload = Buffer.from
(JSON.stringify(msg.payload));
msg.payload = msg.payload.toString();
return msg;
```

Πατώντας το timestamp στέλνουμε τα δεδομένα από τον DHT11 στο Azure. Επίσης εμφανίζονται όλα τα απεσταλμένα μηνύματα προς το IoT Hub, τρέχοντας και ξεχωριστά αρχεία python εκτός του node-red.

### Με την μέθοδο node mqtt

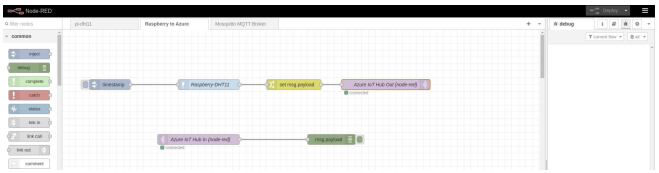
Στο IoT Hub έχει γίνει δημιουργία μια νέα συσκευή με το όνομα node-red. Στο terminal του azure εκτελούμε την εντολή :

```
az iot hub generate-sas-token -d node-red -n uniwa19009 --du 172800
```

και εμφανίζει το terminal :

```
{
  "sas": "SharedAccessSignature
  sr=uniwa19009.azure-devices.net\%2Fdevices\%2Fnode-
  red&sig=NA7zY8THISISnottruLikeThisjnvp\%
  2FHur2z\%2FBSgs0\%3D&se=*********"
}
```

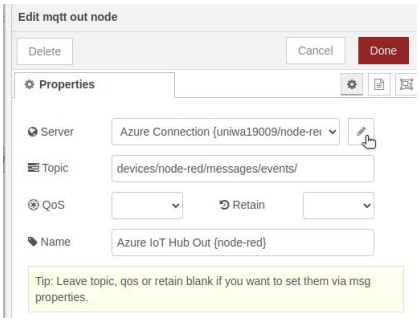
Κάνουμε copy από το SharedAccessSignature εώς το τέλος (χωρίς τα ”). Πάμε στο περιβάλλον του Node-Red να συνδέσουμε το Azure IoT Hub για την συσκευή node-red όπως φαίνεται στην εικόνα 114.



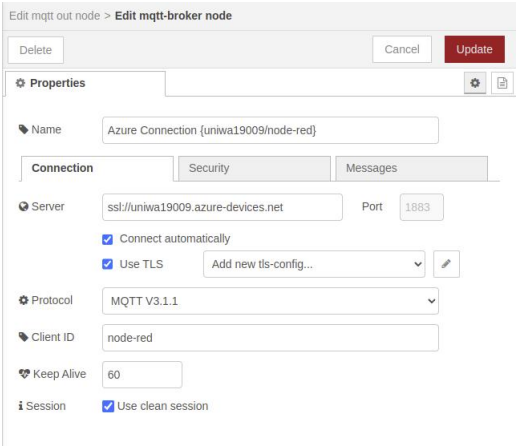
Εικόνα 114. Node-Red – Azure IoT Hub

Στην ουσία έχουμε δημιουργήσει ένα node MQTT OuT και ένα flow node In με τις ακόλουθες ρυθμίσεις. Στο πεδίο topic γράφουμε devices/node-red/messages/events/ και στην συνέχεια στο πεδίο server πατάμε το εικονίδιο για τις ρυθμίσεις όπως φαίνεται την εικόνα 115

Μετά ρυθμίζουμε όπως στην εικόνα 116




Εικόνα 115. Node-Red – Azure IoT Hub




Εικόνα 116. Node-Red – Azure IoT Hub

- Στο server : ssl://uniwa19009.azure-devices.net δηλαδή [ssl://το όνομα του IoT Hub.azure-devices.net]
- Protocol : MQTT 3.1
- Client ID : Το όνομα της συσκεύης (node-red)

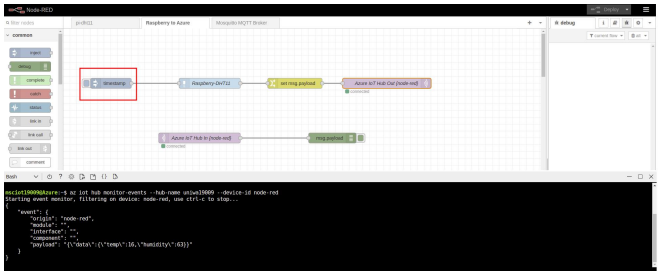
Μετά πάμε στο πεδίο security και κάνουμε τις ακόλουθες ρυθμίσεις:

- username : uniwa19009.azure-devices.net/node-red
- password : Κάνουμε copy paste το sas token που δημιουργήσαμε από το terminal του azure με την εντολή az iot hub generate-sas-token -d node-red -n uniwa19009 –du 172800 SharedAccessSignature sr=uniwa19009.azure-devices.net%2Fdevices%2Fnode-redsig=NA7zY8THISISnottruLikeThisjnvp%2FHur2z%2FBSgs0%3Dse=*********

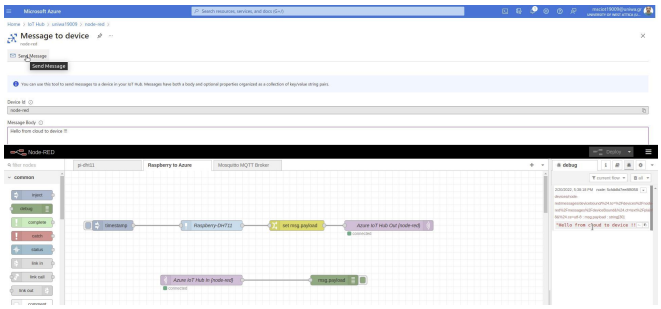
Με τον ίδιο τρόπο ρυθμίζουμε και το node MQTT In αλλά στο πεδίο topic γράφουμε devices/node-red/messages/#.

Κάνοντας click στο timestamp θα δούμε από το terninal του azure ότι στάλθηκαν επιτυχώς τα δεδομένα από τον αισθητήρα DHT όπως φαίνεται στην εικόνα 117

Τώρα θα στείλουμε μήνυμα από το Azure στο Node-red. Πάμε στην συσκευή node-red γράφουμε το μήνυμα Hello from cloud to device και send. Το μήνυμα στάλθηκε επιτυχώς όπως φαίνεται στην εικόνα 118



Εικόνα 117. Message sent



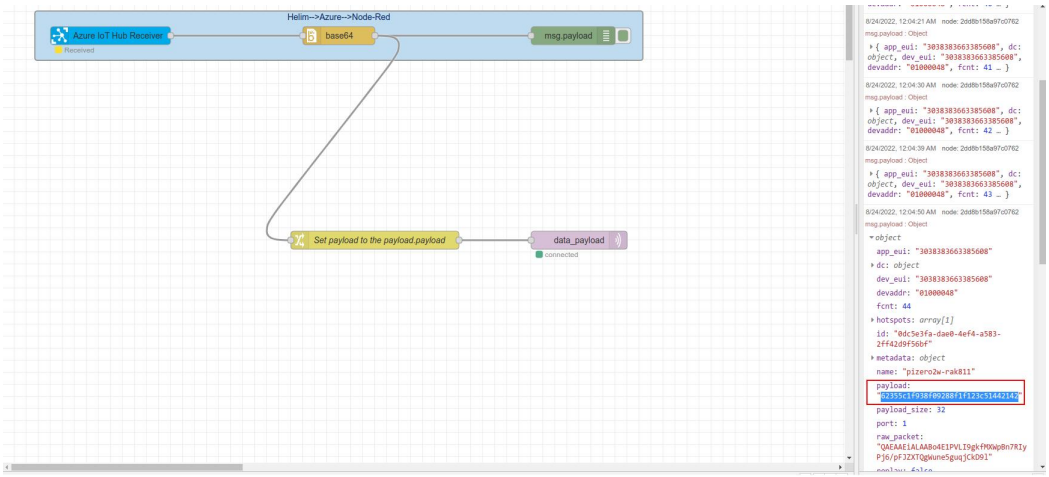
Εικόνα 118. Hello from cloud to device!

Πατώντας το timestamp στέλνουμε τα δεδομένα από τον DHT11 στο Azure. Επίσης εμφανίζονται όλα τα απεσταλμένα μηνύματα προς το IoT Hub, τρέχοντας και ξεχωριστά αρχεία python εκτός του node-red.

### 9.5.2 Azure IoT Receiver - Helium

Σε αυτήν την ενότητα δίνεται σαν επιπλέον πληροφορία αν θέλουμε τα δεδομένα από το Helium Blockchain να τα λαμβάνουμε στο Azure.

Μέσω του node Azure IoT Hub Receiver λαμβάνουμε τα δεδομένα από το Helium Blockchain, δηλαδή τα δεδομένα που έχουμε στείλει από τον(τους) κόμβους.




Εικόνα 119. Node-Red

Στο node Azure IoT Hub Receiver επιλέγομε:

- connectionString : Την τιμή Event Hub compatible endpoint. Την συγκεκριμένη τιμή την

βρίσκουμε απο το azure portal → IoT Hub → uniwa19009 → Build-in endpoints → Event Hub compatible endpoint (εικόνα 113).

Το node base64 μετατρέπει το ωφέλιμο φορτίο από τη μορφή base64 σε μορφή συμβολοσειράς (string). Ακολουθούν οι ρυθμίσεις του node:

```
Action : Convert Base64 to String
property : msg.payload.payload
```

Το node Set payload to payload.payload ορίζει ότι το ωφέλιμο φορτίο είναι απλώς το μήνυμα που έχει στείλει ο κόμβος χωρίς άλλες πληροφορίες. Ακολουθούν οι ρυθμίσεις του node:

```
Set : msg.payload
property : msg.payload.payload
```

Οι ρυθμίσεις που αφορούν στο Helium Console για το integration Azure IoT Hub δίνονται στην ενότητα 5.1.1 του 5ου Κεφαλαίου

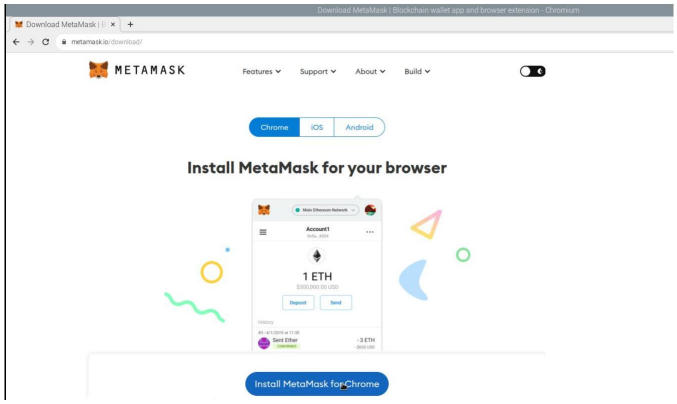
ΚΕΦΑΛΑΙΟ 10

# Τέταρτο παράρτημα

Στο τέταρτο παράρτημα θα γίνει εγκατάσταση εργαλείων που θα χρησιμοποιήσουμε για το Ethereum όπως το Metamask και το Ganache.

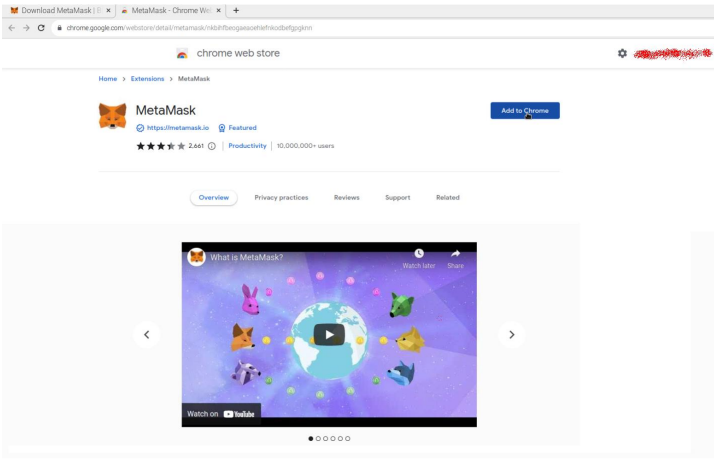
## 10.1 Εγκατάσταση Metamask

Από το link `https://metamask.io/download/` εγκαθιστούμε το metamask στο chronium. Κάνουμε click στο Install MetaMask for Chrome (εικόνα 120)

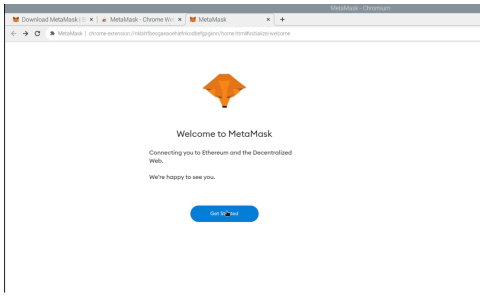



Εικόνα 120. Install MetaMask for Chrome

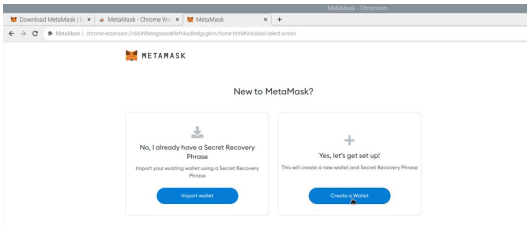
Στην συνέχεια επιλέγουμε add to chrome (εικόνα 121).
Στην συνέχεια επιλέγουμε Get Started (εικόνα 122).
Στην συνέχεια επιλέγουμε Create a wallet (εικόνα 123).
Στην συνέχεια επιλέγουμε I Agree (εικόνα 124).
Στην συνέχεια ορίζουμε κωδικό (εικόνα 125).
Στην συνέχεια εμφανίζει το Secret Recovery Phase (εικόνα 126). Το password καθώς και το Secret Recovery Phase πρέπει να τα αποθηκεύσουμε και να μην τα χάσουμε γιατί δεν θα μπορούμε να επαναφέρουμε το πορτοφόλι μας.
Στην συνέχεια θα μας ζητηθεί να εισάγουμε το Secret Recovery Phase κάνοντας click με την σωστή σειρά τα κουτιά και μετά confirm (εικόνα 127).
Ακολουθώντας τα παραπάνω βήματα η εγκατάσταση έχει γίνει επιτυχώς (εικόνα 128).



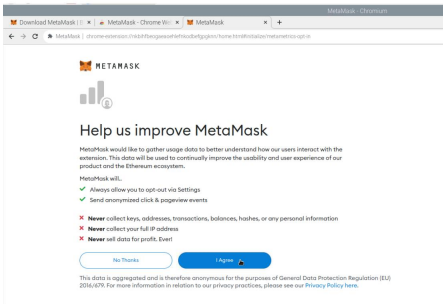

Εικόνα 121. Add to Chrome



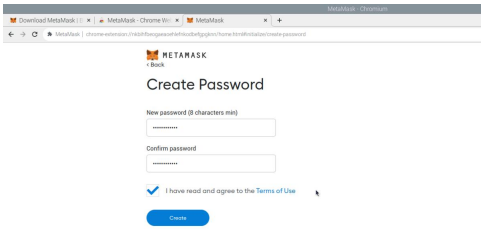

Εικόνα 122. Get Started



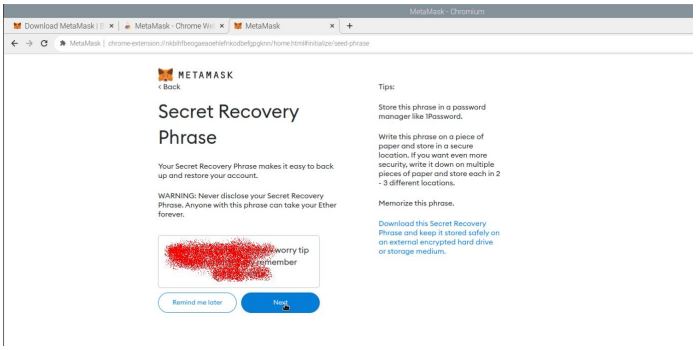

Εικόνα 123. Create a wallet



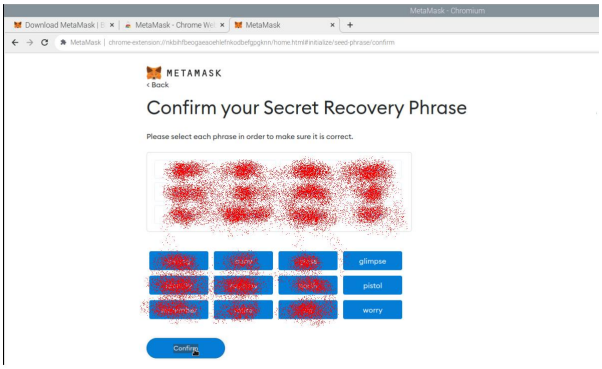

Εικόνα 124. I Agree



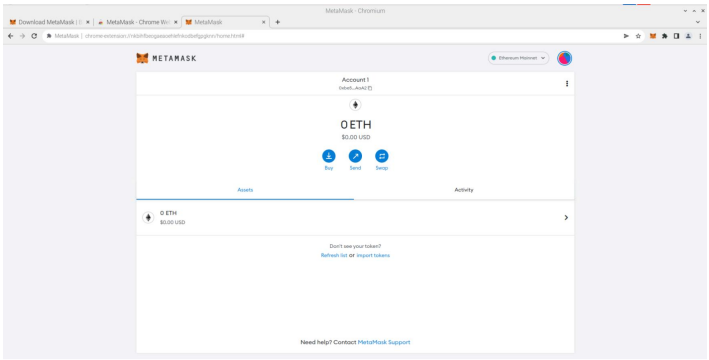

Εικόνα 125. Create Password




Εικόνα 126. Secret Recovery Phase




Εικόνα 127. Confirm



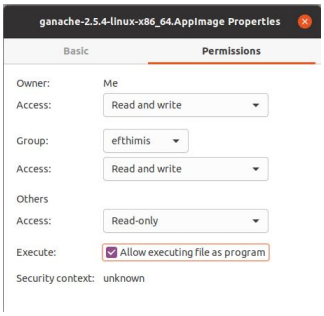

Εικόνα 128. Metamask Ready

## 10.2 Εγκατάσταση Ganache

Στο παρακάτω link κατέβουμε το Ganache για έκδοση linux (υπάρχουν εκδόσεις για Windows, Mac OS) https://github.com/trufflesuite/ganache-ui/releases/tag/v2.5.4

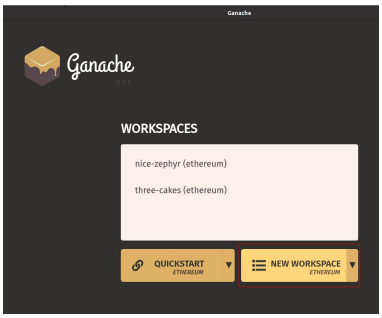
Το ganache δεν υποστιρίζει ARM CPU συνεπώς δεν τρέχει σε raspberry pi. Αυτό δεν οδηγεί σε περιορισμό επειδή το ganache το χρησιμοποιούμε για εικονικό blockchain με εικονικά χρήματα για δοκιμές σε συναλλαγές, smart contracts, dapps.

Κάνουμε δεξί click στο αρχείο ganache που κατεβάσαμε επιλέγουμε properties και στην συνέχεια στο tab Permissions επιλέγουμε Allow executing file as program (εικόνα 129).



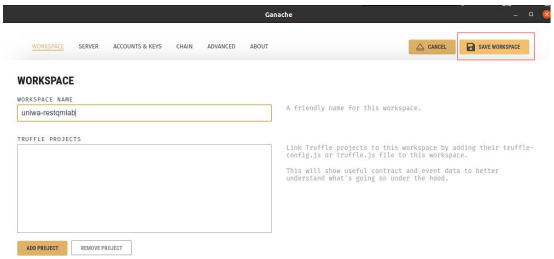

Εικόνα 129. Allow executing file as program

Μετά με διπλό click ανοίγουμε το πρόγραμμα. Επιλέγουμε το tap NEW WORKSPACE ETHEREUM (εικόνα 130)



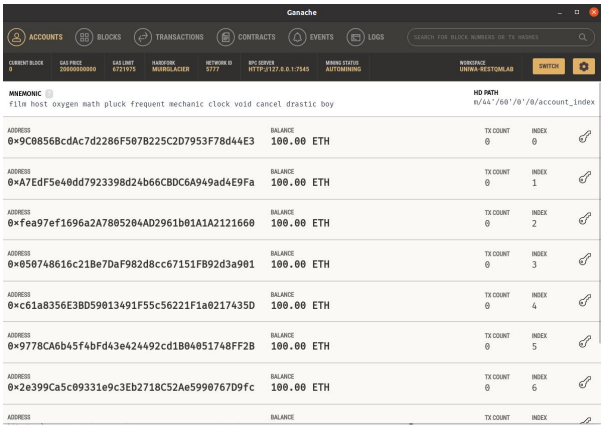

Εικόνα 130. NEW WORKSPACE ETHEREUM

Στην συνέχεια δίνουμε ένα όνομα για το παράδειγμα μας το όνομα uniwa-restqmlab και SAVE WORKSPACE (εικόνα 131).




Εικόνα 131. NEW WORKSPACE : uniwa-restqmlab

Τώρα διαθέτουμε 10 ψηφιακά πορτοφόλια από 100 εικονικά ether το καθένα (εικόνα 132).




Εικόνα 132. Ganache

ΚΕΦΑΛΑΙΟ 11

# Πέμπτο παράρτημα

Στο πέμπτο παράρτημα θα γίνει εγκατάσταση της βάσης δεδομένων InfluxDB, σύνδεση με το node-red και εγκατάσταση του λογισμικού Grafana για την οπτικοποίηση των δεδομένων.

## 11.1 Εγκατάσταση InfluxDB

Αρχικά αναβαθμίσουμε όλα τα εγκατεστημένα πακέτα μέσω terminal με τις εντολές :

```
$ sudo apt update
$ sudo apt upgrade
```

Μετά θα γίνει προσθήκη του InfluxDB repository key με την εντολή:

```
$ curl https://repos.influxdata.com/influxdb.key | gpg --dearmor |
sudo tee /usr/share/keyrings/influxdb-archive-keyring.gpg >/dev/null
```

Και στην συνέχεια προσθήκη του repository στην λίστα πηγών :

```
$ echo "deb [signed-by=/usr/share/keyrings/influxdb-archive-keyring.gpg]
https://repos.influxdata.com/debian $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/influxdb.list
```

Ενημερώσουμε ξανά τη λίστα πακέτων :

```
$ sudo apt update
```

Τώρα θα γίνει εγκατάσταση του InfluxDB :

```
$ sudo apt install influxdb
```

Και ενεργοποίηση του InfluxDB κατά την εκκίνηση του pi:

```
$ sudo systemctl unmask influxdb
$ sudo systemctl enable influxdb
$ sudo systemctl start influxdb
```

### 11.1.1 Ρυθμίσεις InfluxDB

Μέσω terminal εκτελούμαι την εντολή :

```
$ influxdb
```

Και ορίζουμε username και password με την μορφή της παρακάτω εντολής :

```
$ influx -username admin -password
```

Δημιουργούμε μια βάση δεδομένων :

```
$ CREATE DATABASE SMGW_MEASUREMENTS
```

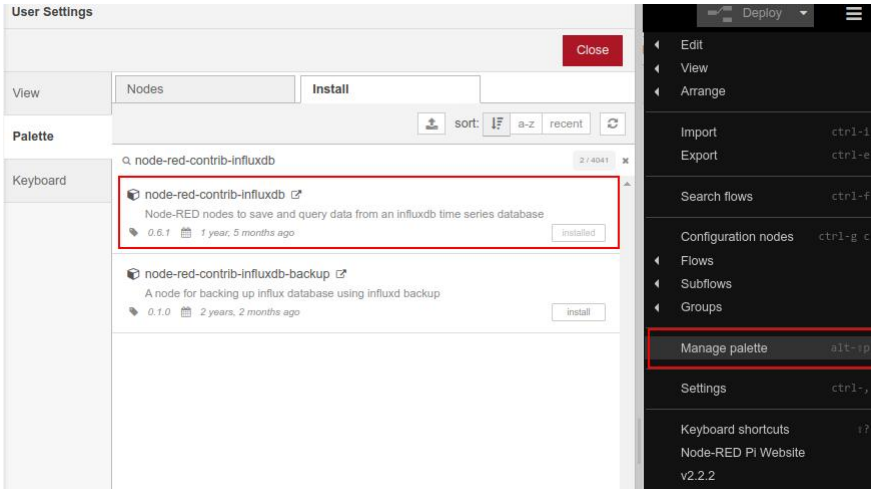
Και κάνουμε χρήση της βάσης δεδομένων με την εντολή:

```
$ USE SMGW_MEASUREMENTS
```

[112]

### 11.1.2 Σύνδεση InfluxDB με το node-red

Μέσω του μενού Burger (πάνω δεξιά) → επιλογή manage palette → και εγκατάσταση του πακέτου node-red-contrib-influxdb όπως φαίνεται στην εικόνα 133

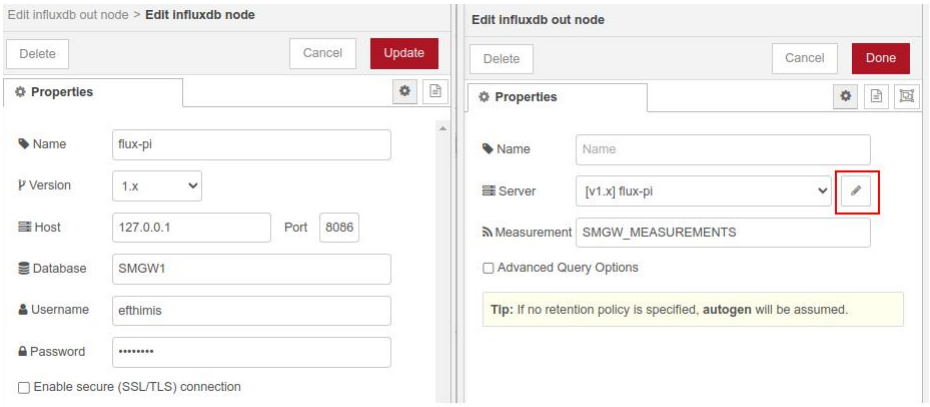



Εικόνα 133. node-red-contrib-influxdb

Στην συνέχεια μέσω του node InfluxDB out κάνουμε τις ακόλουθες ρυθμίσεις (εικόνα 134) :

- Server : flux-pi και κάνουμε τις ακόλουθες ρυθμίσεις :
  - Version : 1x
  - Host : 127.0.0.1 και port : 8086
  - Database : SMGW_MEASUREMENTS
  - Username και password που ορίστηκε
- Measurement : SMGW1

Το node InfluxDB out είναι έτοιμο για χρήση. Εύκολα μπορεί να συνδεθεί με ένα επιθυμητό topic ενός MQTT Broker.




Εικόνα 134. Ρυθμίσεις node-red influxdb

## 11.2 Εγκατάσταση Grafana

Προσθήκη κλειδιού APT που χρησιμοποιείται για τον έλεγχο ταυτότητας πακέτων με την εντολή:

```
$ wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
```

Προσθήκη repository του Grafana APT:

```
$ echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee
-a /etc/apt/sources.list.d/grafana.list
```

Στην συνέχεια εγκατάσταση του Grafana :

```
$ sudo apt-get update
$ sudo apt-get install -y grafana
```

Ενεργοποίηση και έναρξη του Grafana ως service:

```
$ sudo /bin/systemctl enable grafana-server
$ sudo /bin/systemctl start grafana-server
```

Μέσω browser http://<διεύθυνση IP του pi>:3000. Προεπιλεγμένο όνομα χρήστη admin και τον προεπιλεγμένος κωδικός πρόσβασης admin [113].

# Βιβλιογραφικές Αναφορές


[1] T. Alam, «A Reliable Communication Framework and Its Use in Internet of Things (IoT),» Αύγ. 2020. DOI: 10.36227/techrxiv.12657158. διεύθν.: https://www.techrxiv.org/articles/preprint/A_Reliable_Communication_Framework_and_Its_Use_in_Internet_of_Things_IoT_/12657158.

[2] *Internet of Things (IoT): Technologies, Applications, Challenges and Solutions*. διεύθν.: https://www.routledge.com/Internet-of-Things-IoT-Technologies-Applications-Challenges-and-Solutions/Tripathy-Anuradha/p/book/9780367572921.

[3] M. A. Razzaque, M. Milojevic-Jevric, A. Palade και S. Clarke, «Middleware for Internet of Things: A Survey,» *IEEE Internet of Things Journal*, τόμ. 3, αρθμ. 1, σσ. 70–95, Φεβ. 2016, ISSN: 2327-4662. DOI: 10.1109/JIOT.2015.2498900. διεύθν.: http://ieeexplore.ieee.org/document/7322178/.

[4] M. Burhan, R. Rehman, B. Khan και B.-S. Kim, «IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey,» *Sensors*, τόμ. 18, αρθμ. 9, σ. 2796, Αύγ. 2018, ISSN: 1424-8220. DOI: 10.3390/s18092796. διεύθν.: http://www.mdpi.com/1424-8220/18/9/2796.

[5] T. F. Mabrouk, «An Agent Based Approach to Create an Intelligent and Autonomous Operational Concept for the Internet of Things,» σ. 5, 2018.

[6] P. Sethi και S. R. Sarangi, «Internet of Things: Architectures, Protocols, and Applications,» *Journal of Electrical and Computer Engineering*, τόμ. 2017, σσ. 1–25, 2017, ISSN: 2090-0147, 2090-0155. DOI: 10.1155/2017/9324035. διεύθν.: https://www.hindawi.com/journals/jece/2017/9324035/.

[7] R. W. World, *What is CoAP protocol IoT | CoAP Architecture, message format*, https://www.rfwireless-world.com/IoT/CoAP-protocol.html, 2016.

[8] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari και M. Ayyash, «Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,» *IEEE Commun. Surv. Tutorials*, τόμ. 17, αρθμ. 4, σσ. 2347–2376, 2015, ISSN: 1553-877X, 2373-745X. DOI: 10.1109/COMST.2015.2444095. διεύθν.: https://ieeexplore.ieee.org/document/7123563/.

[9] M. Orlando, A. Estebsari, E. Pons, M. Pau, S. Quer, M. Poncino, L. Bottaccioli και E. Patti, «A Smart Meter Infrastructure for Smart Grid IoT Applications,» *IEEE Internet of Things Journal*, τόμ. 9, αρθμ. 14, σσ. 12 529–12 541, Ιούλ. 2022, ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2021.3137596. διεύθν.: https://ieeexplore.ieee.org/document/9659809/.

[10] N. Kroener, K. Förderer, M. Lösch και H. Schmeck, «State-of-the-Art Integration of Decentralized Energy Management Systems into the German Smart Meter Gateway Infrastructure,» *Applied Sciences*, τόμ. 10, αρθμ. 11, σ. 3665, Μάι. 2020, ISSN: 2076-3417. DOI: 10.3390/app10113665. διεύθν.: https://www.mdpi.com/2076-3417/10/11/3665.

[11] A. Ghasempour, «Internet of Things in Smart Grid: Architecture, Applications, Services, Key Technologies, and Challenges,» *Inventions*, τόμ. 4, αρθμ. 1, σ. 22, Μαρ. 2019, ISSN: 2411-5134. DOI: 10.3390/inventions4010022. διεύθν.: https://www.mdpi.com/2411-5134/4/1/22.

[12] Y. Zhou, W. Ni και Z. Zhu, «Architecture of Energy Internet and Its Technologies in Application Reviewed,» *Journal of Clean Energy Technologies*, τόμ. 5, αρθμ. 4, σσ. 320–327, Ιούλ. 2017, ISSN: 1793821X. DOI: 10.18178/JOCET.2017.5.4.391. διεύθν.: http://www.jocet.org/index.php?m=content&c=index&a=show&catid=56&id=717.

[13] K. Förderer, M. Lösch, R. Növer, M. Ronczka και H. Schmeck, «Smart Meter Gateways: Options for a BSI-Compliant Integration of Energy Management Systems,» *Applied Sciences*, τόμ. 9, αρθμ. 8, σ. 1634, Απρ. 2019, ISSN: 2076-3417. DOI: 10.3390/app9081634. διεύθν.: https://www.mdpi.com/2076-3417/9/8/1634.

[14] LinkLabs, *The Past, Present, Future of LPWAN*, https://www.link-labs.com/blog/past-present-future-lpwan, 2017.

[15] A. I. Petrariu και A. Lavric, «SigFox Wireless Communication Enhancement for Internet of Things: A study,» στο *2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, Bucharest, Romania: IEEE, Μαρ. 2021, σσ. 1–4, ISBN: 978-1-66541-878-2. DOI: 10.1109/ATEE52255.2021.9425213.

[16] K. Mekki, E. Bajic, F. Chaxel και F. Meyer, «Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT,» *ICT Express*, τόμ. 5, αρθμ. 1, σσ. 1–7, Μαρ. 2019, ISSN: 24059595. DOI: 10.1016/j.icte.2017.12.005.

[17] Y. Song, J. Lin, M. Tang και S. Dong, «An Internet of Energy Things Based on Wireless LPWAN,» *Engineering*, τόμ. 3, αρθμ. 4, σσ. 460–466, Αύγ. 2017, ISSN: 20958099. DOI: 10.1016/J.ENG.2017.04.011.

[18] R. S. Sinha, Y. Wei και S.-H. Hwang, «A survey on LPWA technology: LoRa and NB-IoT,» *ICT Express*, τόμ. 3, αρθμ. 1, σσ. 14–21, Μαρ. 2017, ISSN: 24059595. DOI: 10.1016/j.icte.2017.03.004.

[19] Qualcomm, «Leading the LTE IoT evolution to connect the massive Internet of Things,» Ιούλιος 2017.

[20] L. Wang, G. Adhikary, B. Sui και Bergam, «A Primer on 3GPP Narrowband Internet of Things (NB-IoT),» τόμ. 11, αρθμ. 55, 2017. DOI: 10.1109/MCOM.2017.1600510CM.

[21] H. Fattah, *5G LTE Narrowband Internet of Things (NB-IoT)*. Φλόριντα: CRC Press, 2019.

[22] R. Schwarz, *Narrowband Internet of Things, whitepaper*, https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1ma266/1MA266_0e_NB_IoT.pdf/, 2017.

[23] R. S. Sinha, Y. Wei και S.-H. Hwang, «A survey on LPWA technology: LoRa and NB-IoT,» *ICT Express*, τόμ. 3, αρθμ. 1, σσ. 14–21, Μαρ. 2017, ISSN: 24059595. DOI: 10.1016/j.icte.2017.03.004. διεύθν.: https://linkinghub.elsevier.com/retrieve/pii/S2405959517300061.

[24] W. S. C. Beard, *Ασύρματες επικοινωνίες δίκτυα Συστήματα*. Θεσσαλονίκη: Τζιόλα, 2017.

[25] 3GPP(a), *TS 36.101: User Equipment (UE) radio transmission and reception, version 13.4.0*, https://www.etsi.org/deliver/etsi_ts/136100_136199/136101/13.04.00_60/ts_136101v130400p.pdf/, 2016.

[26] 3GPP(b), *TS 36.101: User Equipment (UE) radio transmission and reception, version 14.5.0*, https://www.etsi.org/deliver/etsi_ts/136100_136199/136101/14.05.00_60/ts_136101v140500p.pdf/, 2017.

[27] C. Gomez, J. C. Veras, R. Vidal, L. Casals και J. Paradells, «A Sigfox Energy Consumption Model,» *Sensors*, τόμ. 19, αρθμ. 3, σ. 681, 7 Φεβ. 2019, ISSN: 1424-8220. DOI: 10.3390/s19030681.

[28] A. I. Petrariu και A. Lavric, «SigFox Wireless Communication Enhancement for Internet of Things: A study,» σσ. 1–4, 25 Μαρ. 2021. DOI: 10.1109/ATEE52255.2021.9425213. διεύθν.: https://ieeexplore.ieee.org/document/9425213/.

[29] D. Ismail, M. Rahman και A. Saifullah, «Low-power wide-area networks: opportunities, challenges, and directions,» σσ. 1–6, 4 Ιαν. 2018. DOI: 10.1145/3170521.3170529. διεύθν.: https://dl.acm.org/doi/10.1145/3170521.3170529.

[30] SigFox, *Understand the strategic business advantage of IoT*, https://build.sigfox.com/study, 2020.

[31] Vodafone, *Narrowband-IoT: pushing the boundaries of IoT*, https://www.vodafone.com/business/news-and-insights/white-paper/narrowband-iot-pushing-the-boundaries-of-iot, 2017.

[32] Sigfox, «Sigfox Device ETSI Mode White Paper,» σσ. 1–12, 19 Νοέ. 2018. διεύθν.: https://support.sigfox.com/docs/sigfox-device-etsi-mode-white-paper.

[33] N. Naik, «LPWAN Technologies for IoT Systems: Choice Between Ultra Narrow Band and Spread Spectrum,» στο *2018 IEEE International Systems Engineering Symposium (ISSE)*, Ιούλ. 2018, σσ. 1–8. DOI: 10.1109/SysEng.2018.8544414.

[34] SIGFOX, *Radio Configurations*, https://build.sigfox.com/sigfox-radio-configurations-rc/, 2020.

[35] L. Alliance, *Lora Alliance*, https://lora-alliance.org/, 2022.

[36] L. Alliance, «A technical overview of LoRa and LoRaWAN,» Νοέμβριος 2015. διεύθν.: https://cdn.everythingrf.com/live/LoRaWAN101-technical-overview.pdf.

[37] A. Augustin, J. Yi, T. Clausen και W. Townsley, «A Study of LoRa: Long Range & Low Power Networks for the Internet of Things,» *Sensors*, τόμ. 16, αρθμ. 9, σ. 1466, 9 Σεπτ. 2016, ISSN: 1424-8220. DOI: 10.3390/s16091466. διεύθν.: http://www.mdpi.com/1424-8220/16/9/1466.

[38] TTI, *What is a LoRaWAN Network Server*, https://www.thethingsindustries.com/news/what-lorawan-network-server/, 2020.

[39] D. Poluektov, M. Polovov, P. Kharin, M. Stusek, K. Zeman, P. Masek, I. Gudkova, J. Hosek και K. Samouylov, «On the Performance of LoRaWAN in Smart City: End-Device Design and Communication Coverage,» τόμ. 11965, V. M. Vishnevskiy, K. E. Samouylov και D. V. Kozyrev, επιμελητές, σσ. 15–29, 2019, Series Title: Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-36614-8_2. διεύθν.: http://link.springer.com/10.1007/978-3-030-36614-8_2.

[40] N. Andreadou, E. Kotsakis και M. Masera, «Smart Meter Traffic in a Real LV Distribution Network,» *Energies*, τόμ. 11, αρθμ. 5, σ. 1156, Μάι. 2018, ISSN: 1996-1073. DOI: 10.3390/en11051156. διεύθν.: https://www.mdpi.com/1996-1073/11/5/1156.

[41] R. W. World, *LoRa Modulation Basics | CSS Modulation | Advantages, Properties*, https://www.rfwireless-world.com/Terminology/LoRa-modulation-vs-CSS-modulation.html, 2017.

[42] C. Gu, L. Jiang και R. Tan, «LoRa-Based Localization: Opportunities and Challenges,» σ. 6, 2019.

[43] B. Reynders και S. Pollin, «Chirp spread spectrum as a modulation technique for long range communication,» στο *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, Mons, Belgium: IEEE, Νοέ. 2016, σσ. 1–5, ISBN: 978-1-5090-4361-3. DOI: 10.1109/SCVT.2016.7797659. διεύθν.: http://ieeexplore.ieee.org/document/7797659/.

[44] G. Ferre και A. Giremus, «LoRa Physical Layer Principle and Performance Analysis,» στο *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Bordeaux: IEEE, Δεκ. 2018, σσ. 65–68, ISBN: 978-1-5386-9562-3. DOI: 10.1109/ICECS.2018.8617880. διεύθν.: https://ieeexplore.ieee.org/document/8617880/.

[45] N. Gonzalez, A. Van Den Bossche και T. Val, «Specificities of the LoRa™ Physical Layer for the Development of New Ad Hoc MAC Layers,» στο *Ad-hoc, Mobile, and Wireless Networks*, N. Montavont και G. Z. Papadopoulos, επιμελητές, τόμ. 11104, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, σσ. 163–174, ISBN: 978-3-030-00246-6 978-3-030-00247-3. DOI: 10.1007/978-3-030-00247-3_16. διεύθν.: http://link.springer.com/10.1007/978-3-030-00247-3_16.

[46] L. Davoli, E. Pagliari και G. Ferrari, «Hybrid LoRa-IEEE 802.11s Opportunistic Mesh Networking for Flexible UAV Swarming,» *Drones*, τόμ. 5, αρθμ. 2, σ. 26, 15 Απρ. 2021, ISSN: 2504-446X. DOI: 10.3390/drones5020026. διεύθν.: https://www.mdpi.com/2504-446X/5/2/26.

[47] L. Alliance, «LoRaWAN Regional Parameters,» 2020. διεύθν.: https://lora-alliance.org/wp-content/uploads/2021/05/RP002-1.0.3-FINAL-1.pdf.

[48] S. Kim, H. Lee και S. Jeon, «An Adaptive Spreading Factor Selection Scheme for a Single Channel LoRa Modem,» *Sensors*, τόμ. 20, αρθμ. 4, σ. 1008, 13 Φεβ. 2020, ISSN: 1424-8220. DOI: 10.3390/s20041008. διεύθν.: https://www.mdpi.com/1424-8220/20/4/1008.

[49] M. Saari, A. M. bin Baharudin, P. Sillberg, S. Hyrynsalmi και W. Yan, «LoRa — A survey of recent research trends,» σσ. 0872–0877, Μάι. 2018. DOI: 10.23919/MIPRO.2018.8400161. διεύθν.: https://ieeexplore.ieee.org/document/8400161/.

[50] E. B., *Lora Documentation*, https://lora.readthedocs.io/en/latest/#duty-cycle-time-on-air-toa, 2018.

[51] avbentem.github, *Airtime calculator for LoRaWAN*, https://avbentem.github.io/airtime-calculator/ttn/eu868/100, 2021.

[52] L. Alliance, «LoRaWAN Security Whitepaper,» Φεβρουάριος 2017. διεύθν.: https://lora-alliance.org/wp-content/uploads/2020/11/lorawan_security_whitepaper.pdf.

[53] T. T. Industries, *ABP vs OTAA*, https://www.thethingsindustries.com/docs/devices/abp-vs-otaa/, 2021.

[54] I. Butun, N. Pereira και M. Gidlund, «Security Risk Analysis of LoRaWAN and Future Directions,» *Future Internet*, τόμ. 11, αρθμ. 1, σ. 3, Δεκ. 2018, ISSN: 1999-5903. DOI: 10.3390/fi11010003. διεύθν.: http://www.mdpi.com/1999-5903/11/1/3.

[55] T. T. Industries, *Security*, https://www.thethingsnetwork.org/docs/lorawan/security/, 2021.

[56] Techplayon, *LoRa – Device Activation Call Flow (Join Procedure) using OTAA and ABP*, https://www.techplayon.com/lora-device-activation-call-flow-join-procedure-using-otaa-and-abp/, 2018.

[57] Dragino1, *LG02 Dual Channels LoRa IoT Gateway*, https://www.dragino.com/products/lora-lorawan-gateway/item/135-lg02.html, 2019.

[58] Dragino2, *LG02 LoRa Gateway User Manual 63 / 63*, https://www.dragino.com/downloads/downloads/LoRa_Gateway/LG02-OLG02/LG02_LoRa_Gateway_User_Manual_v1.6.0.pdf, 2019.

[59] Dragino3, *Raspberry Pi HAT featuring GPS and LoRa® technology*, https://www.dragino.com/products/lora/item/106-lora-gps-hat.html, 2019.

[60] Pycom, *LoPy4*, https://pycom.io/product/lopy4/, 2019.

[61] Dragino, *Arduino Shield featuring LoRa® technology*, https://www.dragino.com/products/lora/item/102-lora-shield.html, 2019.

[62] T. T. Network, *The Things Network*, https://www.thethingsnetwork.org/, 2015.

[63] T. T. N. TTN, *LoRaWAN Architecture*, https://www.thethingsnetwork.org/docs/lorawan/architecture/, 2015.

[64] B. Mendes, «Framework for the Integration of Transmission Optimization Components into LoRaWAN Stack,» σ. 12,

[65] ChirpStack, *The ChirpStack project*, https://www.chirpstack.io/project/, 2020.

[66] Y. Himeur, A. Sayed, A. Alsalemi, F. Bensaali, A. Amira, I. Varlamis, M. Eirinaki, C. Sardianos και G. Dimitrakopoulos, «Blockchain-based recommender systems: Applications, challenges and future opportunities,» *Computer Science Review*, τόμ. 43, σ. 100 439, Φεβ. 2022, ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2021.100439. διεύθν.: https://www.sciencedirect.com/science/article/pii/S1574013721000769.

[67] pisupply, *Getting Started with the Raspberry Pi LoRa Node pHAT*, https://learn.pi-supply.com/make/getting-started-with-the-raspberry-pi-lora-node-phat/, 2020.

[68] G. : jezmck, *IoT LoRa pHAT Node*, https://https://github.com/PiSupply/IoTLoRaRange/tree/master/IoT%20LoRa%20Raspberry%20Pi%20Node%20pHAT, 2021.

[69] pinout, *IoT LoRa Node pHAT*, https://pinout.xyz/pinout/iot_lora_node_phat, 2020.

[70] jezmck, *IoT LoRa pHAT Node*, https://github.com/PiSupply/IoTLoRaRange/tree/master/IoT%20LoRa%20Raspberry%20Pi%20Node%20pHAT, 2021.

[71] V. Deshpande, L. George και H. Badis, «SaFe: A Blockchain and Secure Element Based Framework for Safeguarding Smart Vehicles,» σσ. 181–188, Σεπτ. 2019. DOI: 10.23919/WMNC.2019.8881408. διεύθν.: https://ieeexplore.ieee.org/document/8881408/.

[72] F. Lombardi, L. Aniello, S. De Angelis, A. Margheri και V. Sassone, «A Blockchain-based Infrastructure for Reliable and Cost-effective IoT-aided Smart Grids,» 42 (6 pp.)–42 (6 pp.) 2018. DOI: 10.1049/cp.2018.0042. διεύθν.: https://digital-library.theiet.org/content/conferences/10.1049/cp.2018.0042.

[73] investopedia, *Proof of Work (PoW)*, https://www.investopedia.com/terms/p/proof-work.asp, 2022.

[74] Q. Wang, X. Zhu, Y. Ni, L. Gu και H. Zhu, «Blockchain for the IoT and industrial IoT: A review,» *Internet of Things*, τόμ. 10, σ. 100 081, Ιούν. 2020, ISSN: 25426605. DOI: 10.1016/j.iot.2019.100081. διεύθν.: https://linkinghub.elsevier.com/retrieve/pii/S254266051930085X.

[75] K. R. Ozyilmaz και A. Yurdakul, «Designing a Blockchain-Based IoT With Ethereum, Swarm, and LoRa: The Software Solution to Create High Availability With Minimal Security Risks,» *IEEE Consumer Electronics Magazine*, τόμ. 8, αρθμ. 2, σσ. 28–34, Μαρ. 2019, ISSN: 2162-2248, 2162-2256. DOI: 10.1109/MCE.2018.2880806. διεύθν.: https://ieeexplore.ieee.org/document/8634961/.

[76] AWS, *What is Decentralization in Blockchain?* https://aws.amazon.com/blockchain/decentralization-in-blockchain/, 2021.

[77] investopedia, *What Is a Cryptocurrency Public Ledger?* https://www.investopedia.com/tech/what-cryptocurrency-public-ledger/, 2021.

[78] Simplilearn, *What is Blockchain Wallet*, https://www.simplilearn.com/tutorials/blockchain-tutorial/blockchain-wallet, 2022.

[79] R. Financial, *Byzantine Fault Tolerance (BFT)*, https://river.com/learn/terms/b/byzantine-fault-tolerance-bft/, 2021.

[80] Β. Παζόπουλος, *Το επενδυτικό εγχειρίδιο του bitcoin*. Παπανικολή 50, 15232, Χαλάνδρι, Αθήνα: Media2day, 2022.

[81] V. Buterin, «Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.,» σ. 36,

[82] S. Cui, M. R. Asghar, S. D. Galbraith και G. Russello, «ObliviousDB: Practical and Efficient Searchable Encryption with Controllable Leakage,» στο *Foundations and Practice of Security*, A. Imine, J. M. Fernandez, J.-Y. Marion, L. Logrippo και J. Garcia-Alfaro, επιμελητές, τόμ. 10723, Cham: Springer International Publishing, 2018, ISBN: 978-3-319-75649-3 978-3-319-75650-9. DOI: 10.1007/978-3-319-75650-9_13. διεύθν.: http://link.springer.com/10.1007/978-3-319-75650-9_13.

[83] AWS, *What is Ethereum?* https://aws.amazon.com/blockchain/what-is-ethereum/, 2021.

[84] Ethereum, *INTRODUCTION TO DAPPS*, https://ethereum.org/en/dapps/#what-are-dapps, 2022.

[85] Cryptopedia, *Metamask: An ETH-First Crypto Wallet for Swaps and dApps*, https://www.gemini.com/cryptopedia/what-is-metamask-how-to-use-metamask-extension#section-what-is-meta-mask, 2021.

[86] S. Beyer, *What is Ethereum Ganache?* https://www.mycryptopedia.com/what-is-ethereum-ganache/, 2010.

[87] M. Blog, *Remix Explained – What is Remix?* https://moralis.io/remix-explained-what-is-remix/, 2021.

[88] Remix, *Welcome to Remix's documentation!* https://remix-ide.readthedocs.io/en/latest/, 2019.

[89] A. Haleem, A. Allen, A. Thompson, M. Nijdam και R. Garg, «A Decentralized Wireless Network,» σ. 20,

[90] Coinspeaker, *What Is Helium Network and HNT Token*, https://www.coinspeaker.com/guides/what-is-helium-network-hnt-token/, 2021.

[91] Helium, *The Helium Consensus Protocol*, https://docs.helium.com/blockchain/consensus-protocol/, 2018.

[92] V. Deshpande, L. George και H. Badis, «SaFe: A Blockchain and Secure Element Based Framework for Safeguarding Smart Vehicles,» στο *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*, Paris, France: IEEE, Σεπτ. 2019, σσ. 181–188, ISBN: 978-3-903176-18-8. DOI: 10.23919/WMNC.2019.8881408.

[93] V. Deshpande, L. George και H. Badis, «PulSec: Secure Element based framework for sensors anomaly detection in Industry 4.0,» *IFAC Proceedings Volumes*, τόμ. 52, σσ. 1204–1209, Αύγ. 2019. DOI: 10.1016/j.ifacol.2019.11.362.

[94] P. Urien, «Blockchain IoT (BIoT): A New Direction for Solving Internet of Things Security and Trust Issues,» στο *2018 3rd Cloudification of the Internet of Things (CIoT)*, Paris, France: IEEE, Ιούλ. 2018, σσ. 1–4, ISBN: 978-1-5386-4629-8. DOI: 10.1109/CIOT.2018.8627112. διεύθν.: https://ieeexplore.ieee.org/document/8627112/.

[95] Kaspersky, *Secure Element*, https://encyclopedia.kaspersky.com/glossary/secure-element/, 2022.

[96] Multos(a), *TRUST CORE DETAILS*, https://multos.com/support/multos-trust-anchor/developer-boards/trust-core-details/, 2021.

[97] Multos(b), *MULTOS TRUST CORE Development Kit*, https://www.elfa.se/Web/Downloads/_t/ds/BD-PLUGIN-RevA-001-0001_eng_tds.pdf, 2021.

[98] Multos(c), *PKCS#11*, https://multos.com/wp-content/uploads/2021/06/MULTOS-PKCS11-Application-Notes.pdf, 2021.

[99] M. Vaidehi και B. J. Rabi, «Design and analysis of AES-CBC mode for high security applications,» στο *Second International Conference on Current Trends In Engineering and Technology - ICCTET 2014*, Coimbatore, India: IEEE, Ιούλ. 2014, σσ. 499–502, ISBN: 978-1-4799-7987-5 978-1-4799-7986-8. DOI: 10.1109/ICCTET.2014.6966347. διεύθν.: https://ieeexplore.ieee.org/document/6966347.

[100] H. T. Assafli και I. A. Hashim, «Security Enhancement of AES-CBC and its Performance Evaluation Using the Avalanche Effect,» στο *2020 3rd International Conference on Engineering Technology and its Applications (IICETA)*, Najaf, Iraq: IEEE, Σεπτ. 2020, σσ. 7–11, ISBN: 978-1-72818-231-5. DOI: 10.1109/IICETA50496.2020.9318803. διεύθν.: https://ieeexplore.ieee.org/document/9318803/.

[101] A. Z. Mustafeez, *What is CBC?* https://www.educative.io/answers/what-is-cbc, 2022.

[102] J. Clulow, «On the Security of PKCS #11,» στο *Cryptographic Hardware and Embedded Systems - CHES 2003*, G. Goos, J. Hartmanis, J. van Leeuwen, C. D. Walter, Ç. K. Koç και C. Paar, επιμελητές, τόμ. 2779, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, σσ. 411–425, ISBN: 978-3-540-40833-8 978-3-540-45238-6. DOI: 10.1007/978-3-540-45238-6_32. διεύθν.: http://link.springer.com/10.1007/978-3-540-45238-6_32.

[103] S. Delaune, S. Kremer και G. Steel, «Formal Analysis of PKCS#11,» στο *2008 21st IEEE Computer Security Foundations Symposium*, Pittsburgh, PA, USA: IEEE, 2008, σσ. 331–344, ISBN: 978-0-7695-3182-3. DOI: 10.1109/CSF.2008.16.

[104] S. Forsstrom και U. Jennehag, «A performance and cost evaluation of combining OPC-UA and Microsoft Azure IoT Hub into an industrial Internet-of-Things system,» στο *2017 Global Internet of Things Summit (GIoTS)*, Geneva, Switzerland: IEEE, Ιούν. 2017, σσ. 1–6, ISBN: 978-1-5090-5873-0. DOI: 10.1109/GIOTS.2017.8016265. διεύθν.: http://ieeexplore.ieee.org/document/8016265/.

[105] R. Stackowiak, *Azure Internet of Things Revealed: Architecture and Fundamentals*. Berkeley, CA: Apress, 2019, ISBN: 978-1-4842-5469-1 978-1-4842-5470-7. DOI: 10.1007/978-1-4842-5470-7. διεύθν.: http://link.springer.com/10.1007/978-1-4842-5470-7.

[106] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran και M. Guizani, «Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges,» *IEEE Wireless Communications*, τόμ. 24, αρθμ. 3, σσ. 10–16, Ιούν. 2017, ISSN: 1536-1284. DOI: 10.1109/MWC.2017.1600421. διεύθν.: http://ieeexplore.ieee.org/document/7955906/.

[107] Microsoft, *Encryption of Azure Iot Hub data at rest using customer-managed keys*, https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-customer-managed-keys, 2021.

[108] Microsoft, *Control access to IoT Hub using Shared Access Signatures and security tokens*, https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-dev-guide-sas?tabs=node, 2022.

[109] Eclipse_Foundation(a), *Mosquitto An open source MQTT broker*, https://mosquitto.org, 2006.

[110] Eclipse_Foundation(b), *Python Client*, https://www.eclipse.org/paho/clients/python/, 2020.

[111] ChirpStack, *Quickstart Debian or Ubuntu*, https://www.chirpstack.io/project/guides/debian-ubuntu/, 2020.

[112] Emmet, *Installing InfluxDB to the Raspberry Pi*, https://pimylifeup.com/raspberry-pi-influxdb/, 2022.

[113] G. L. Team, *Install Grafana on Raspberry Pi*, https://grafana.com/tutorials/install-grafana-on-raspberry-pi/, 2021.