Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής - Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Μελέτη ηλεκτρονικού πορτοφολιού και αξιολόγηση τεχνικής παρουσίασης ταυτοποιητικών δεδομένων** **Study on Wallet for personal identification and presentation techniques**
Ονοματεπώνυμο Φοιτητή	**Γεώργιος Σακαλής**
Πατρώνυμο	**Ιωάννης**
Αριθμός Μητρώου	**ΜΠΣΠ / 20045**
Επιβλέπων	**Ευάγγελος Σακκόπουλος, Αναπληρωτής Καθηγητής**

Ημερομηνία Παράδοσης **Δεκέμβριος 2023**

Τριμελής Εξεταστική Επιτροπή

Ευάγγελος Σακκόπουλος, Αναπληρωτής Καθηγητής	Ευθύμιος Αλέπης Αναπληρωτής Καθηγητής	Διονύσιος Σωτηρόπουλος Επίκουρος Καθηγητής

Περίληψη

Self-Sovereign Identity (SSI) είναι ο όρος για ένα νέο μοντέλο διαχείρισης ψηφιακών ταυτοτήτων (Digital Identities) στις ψηφιακές υπηρεσίες. Ο σκοπός του SSI είναι η αποκέντρωση της διαχείρισης των διαπιστευτηρίων ώστε τα φυσικά πρόσωπα να έχουν τη δυνατότητα να αποθηκεύουν τα δεδομένα τους σε ιδία μέσα, χωρίς να εμπλέκεται ένα κεντρικό αποθετήριο. Τα φυσικά πρόσωπα έχουν πλέον τον πλήρη έλεγχο της αποθήκευσης και της χρήσης των προσωπικών τους δεδομένων.

Στο πλαίσιο αυτού το μοντέλου, έννοιες και εργαλεία όπως τα Verifiable Credentials (VC), Verifiable Presentations (VP), Πορτοφόλι Ψηφιακής Ταυτότητας (Identity Wallet - IW) και Αποκεντρωμένα αναγνωριστικά (Decentralizes Identifiers - DID), έχουν αναπτυχθεί και τυποποιηθεί ή βρίσκονται σε διαδικασία τυποποίησης Επιπλέον διάφορα πρωτόκολλά επικοινωνίας για την έκδοση και την επαλήθευση των ψηφιακών ταυτοτήτων όπως DIDComm, OpenID κ.α, έχουν αναπτυχθεί ή βρίσκονται στη φάση του σχεδιασμού.

Συγκεκριμένα το OpenID foundation συμπεριέλαβε πρόσφατα τα παραπάνω μοντέλα και εργαλεία στο OpenID Connect πρότυπο, κάνοντας χρήση το ευρέως χρησιμοποιούμενο πρωτόκολλο αυθεντικοποίησης OAuth 2.0. Τα σχετικά πρότυπα είναι ακόμα στην κατάσταση πρόχειρο (draft), ωστόσο υπάρχουν κάποιές αρχικές υλοποιήσεις.

Η παρούσα εργασία επικεντρώνεται στη χρήση ταυτοποιητικών δεδομένων και συγκεκριμένα των Verifiable Credentials, Verifiable Presentations και των αποκεντρωμένων ταυτοτήτων (DIDs) με τη χρήση του OpenID Connect Στο σενάριο που έχει υλοποιηθεί το Relaying Party συνδέεται με το πορτοφόλι και γίνεται η αυθεντικοποίηση του χρήστη με την ηλεκτρονική ταυτότητα που είναι αποθηκευμένη σε αυτό. Για το λόγο αυτό έχει δημιουργεί μια νέα ροή, η “Verifiable Presentation Flow”, η οποία λειτουργεί συμπληρωματικά με τις άλλες ροές του OpenID Connect.

Abstract

Self-Sovereign Identity (SSI) is the term for a new model of Digital Identity management in digital services. It’s purpose is the decentralization of identity management, so the individuals will have the capacity to store their personal data in their own devices moving away from centralized repositories. In that way individuals fully control how their personal data are kept and used

In this context, new tools and models have been developed or are under development and standardization. Verifiable Credentials (VC), Verifiable Presentations (VP), Identity Wallets (IW) and Decentralized Identifiers (DIDs) are some of them that are being used to implement SSI. Additionally, new communication protocols have risen or existing ones are expanding in order to handle the issuance and the verification of digital identities, DIDComm, OpenID, etc.

For OpenID family of protocols specific, the OpenID Foundation has quite recently included the above models and tools in OpenID Connect protocol, making use of the widespread authentication protocol OAuth 2.0. The relative protocols are still under development in status draft. However some partial implementations have already started.

This work assessed the feasibility of integrating Verifiable Credentials, Verifiable Presentations and Decentralized Identities by extending OpenID Connect. For this reason, a new flow has been created. “Verifiable Presentation Flow”. The new flow can work complementary on other flows of OpenID Connect, for complex scenarios. In the current scenario, the user is interacting with a Relaying Party, and the latter communicates with the user’s wallet for the authentication by exchanging the verifiable credentials.

Πίνακας περιεχομένων

Πίνακας εικόνων

Ορολογία

Συντομογραφία	Όρος	Σημασία
URI	Uniform Resource Identifier	Αλφαριθμητικό αναγνωριστικό που ταυτοποιεί μοναδικά έναν διαδικτυακό πόρο.
IdP	Identity Provider	Γενικός όρος που αναφέρεται σε οποιοδήποτε σύστημα δημιουργεί, αποθηκεύει και διαχειρίζεται ψηφιακές ταυτότητες.
OAuth	Open Authorization	Πρωτόκολλο εξουσιοδότησης χρήσης που χρησιμοποιείται για να δώσει ένας χρήστης πρόσβαση σε μία εφαρμογή ή υπηρεσία ώστε εκείνη να έχει διακριτή πρόσβαση στα δεδομένα του.
OIDC	OpenID Connect	Πρότυπο που επεκτείνει το OAuth για να αποτελέσει μία καθαρή λύση αυθεντικοποίησης χρηστών.
RP	Relying Party	Είναι ο διακομιστής που δίνει πρόσβαση σε μία ασφαλή εφαρμογή.
SIOP	Self-Issued OpenID Provider	Πρότυπο αυθεντικοποίησης που δίνει την δυνατότητα στον ίδιο το χρήστη να παρέχει την ψηφιακή του ταυτότητα με χρήση αυτοεκδοθέντων διαπιστευτηρίων.
DID	Decentralized Identifier	Αποκεντρικοποιημένα Αναγνωριστικά - Ουσιαστικά πρόκειται για URIs που σώζονται πάνω σε blockchain για την ταυτοποίηση των εγγράφων DID.
JSON-LD	JSON Linked-Data	Επέκταση της μορφής JSON που ορίζει πλαίσιο αναφοράς για την μεταφορά συνδεδεμένων δεδομένων.
JWT	JSON Web Token	Αναγνωριστικό σε μορφή JSON που παράγεται κυρίως για σκοπούς αυθεντικοποίησης χρηστών και εξουσιοδότησης χρήσης.
JWS	JSON Web Signature	Μηχανισμός για την υπογραφή αυθαίρετων δεδομένων. Χρησιμοποιείται για την υπογραφή ενός JWT καθώς και κατά την δημιουργία μίας επαληθεύσιμης παρουσίασης (VP).
VC	Verifiable Credentials	Επαληθεύσιμα Διαπιστευτήρια – Πρότυπο για την ψηφιακή αναπαράσταση των διαπιστευτηρίων ενός χρήστη, η ορθότητα των οποίων μπορεί να επαληθευτεί εύκολα από έναν τρίτο.
VP	Verifiable Presentation	Επαληθεύσιμη Παρουσίαση – Πρότυπο για την εύκολη μεταφορά Επαληθεύσιμων Διαπιστευτηρίων μεταξύ του αιτούντος και του εκδοθούντος.
AM	Access Management	Είναι ο κλάδος της πληροφορικής που ασχολείται με την αναγνώριση, τον εντοπισμό, τον έλεγχο και την διαχείριση των χρηστών που έχουν πρόσβαση σε ένα σύστημα ή μία εφαρμογή.

IAM	Identity & Access Management	Η κατηγορία του κλάδου της πληροφορικής που ασχολείται με τα ζητήματα που περιβάλλουν τις ψηφιακές ταυτότητες και την διαχείριση πρόσβασης.
SSI	Self-Sovereign Identity	Αυτοκυριαρχική Ταυτότητα - Η ιδέα ότι η ψηφιακή ταυτότητα ενός χρήστη ανήκει στον ίδιο τον χρήστη, εξαρτάται από εκείνον και διαχειρίζεται αποκλειστικά από εκείνον. Είναι μία μορφή αποκεντρικοποιημένης Ταυτότητας (DI).
SSO	Single-Sign-On	Ενιαία πρόσβαση σε πολλαπλούς πόρους με ένα μόνο login. Επιτυγχάνεται συνήθως με χρήση ενός εκ των δύο παρακάτω κοινών πρωτοκόλλων: o SAML 2.O o OpenID Connect
FI	Federated Identity	Ομοσπονδιακή (ή κεντρικοποιημένη) Ταυτότητα – Αποτελεί τον αντίποδα μίας αποκεντρικοποιημένης ταυτότητας, όπου τον έλεγχο πάνω στις ταυτότητες των χρηστών, τον έχει μία συγκεκριμένη αρχή.
DI	Decentralized Identity	Αποκεντρικοποιημένη Ταυτότητα – Σε αυτό το μοντέλο, και οι χρήστες έχουν τον έλεγχο των ψηφιακών ταυτοτήτων τους.
PII	Personal Identifying Information	Δεδομένα προσωπικού χαρακτήρα τα οποία συνδυαζόμενα, είναι πιθανό να ταυτοποιήσουν έναν συγκεκριμένο χρήστη.
Dapp	Decentralized Application	Μια αποκεντρωμένη εφαρμογή είναι μια εφαρμογή που εκτελείται πάνω σε κάποιο κατανεμημένο καθολικό (π.χ. blockchain) ή σε κάποιο ομότιμο δίκτυο (P2P).

Εισαγωγή

Καθημερινά καλούμαστε να ταυτοποιούμαστε ή να αποδεικνύουμε κάποια ιδιότητα που έχουμε αποκτήσει:

- Στις δημόσιες υπηρεσίες ή στις τράπεζες συχνά μας ζητάνε την ταυτότητα μας.
- Για να ταξιδέψουμε σε κάποια άλλη χώρα χρειαζόμαστε διαβατήριο.
- Για να οδηγήσουμε αυτοκίνητο χρειαζόμαστε δίπλωμα οδήγησης για να πιστοποιήσουμε την ελάχιστη ικανότητα να οδηγούμε
- Στις συνεντεύξεις ζητείται πτυχίο πανεπιστημίου για να αποδείξουμε το επίπεδο γνώσεων μας
- Κάρτα εισόδου στον εργασιακό μας χώρο, η οποία επιτρέπει επιλεκτικά την είσοδο σε διάφορους τομείς ανάλογα με τον τύπο της εργασίας.

Η λίστα αυτή μεγαλώνει όλο και περισσότερο με το πέρασμα του χρόνου, καθώς αυξάνονται οι ανάγκες για ταυτοποίηση και επαλήθευση πιστοποίησης.

Παράλληλα με το φυσικό κόσμο, στον ψηφιακό, από τα πρώτα χρόνια υπήρξε η ανάγκη ταυτοποίησης των χρηστών στις διάφορες ηλεκτρονικές υπηρεσίες. Αρχικά η απλή χρήση username / password. Κατόπι προστέθηκε το 2FA με χρήση OTP device ή κωδικού μέσω κινητού. Πλέον Μιλάμε για MFA. Για απλοποίηση της εμπειρίας, είναι σύνηθες πλέον ο χρήστης να χρησιμοποιεί μια συγκεκριμένη υπηρεσία ταυτοποίησης, όπως κάποιο μέσω κοινωνικής δικτύωσης (Facebook, twitter, google) και να συνδέεται με τα ίδια στοιχεία ταυτοποίησης σε όλες τις εφαρμογές που υποστηρίζουν διασύνδεση με τις συγκεκριμένες υπηρεσίες (Federated Authentication).

Επίσης όσο συνεχίζεται η ψηφιακή μετάβαση του φυσικού κόσμου και όλο και περισσότερες υπηρεσίες γίνονται ψηφιακές, επιπλέον δεδομένα απαιτούνται σε ψηφιακή μορφή για την ταυτοποίηση των χρηστών. Για παράδειγμα οι τράπεζες χρειάζονται εκτεταμένες πληροφορίες για να γνωρίζουν τους χρήστες τους και να αποφεύγεται η διακίνηση παράνομου χρήματος. Ηλικία, διευθύνσεις και άλλα στοιχεία ζητούνται συνεχώς κατά την εγγραφή ενός χρήστη σε μια υπηρεσία.

Ωστόσο ο χρήστης δεν έχει κανένα έλεγχο στη χρήση των δεδομένων τους και ποιος τα προσπελαύνει. Πολλές φορές οι ίδιοι οι χρήστες όταν συνδέονται σε μια υπηρεσία με τη χρήση λογαριασμών μέσων κοινωνικής δικτύωσης αποκαλύπτουν περισσότερα δεδομένα από όσοι οι ίδιοι θέλουν. Επιπλέον ελλοχεύει ο κίνδυνος ανταλλαγής πληροφοριών χρηστών μεταξύ τρίτων εφαρμογών εν αγνοία του τελικού χρήστη

Γίνεται αντιληπτό ότι χρειάζεται μια διαφορετική προσέγγιση στην διαδικασία της ταυτοποίησης και επαλήθευσης στοιχείων ώστε ο χρήστης να ελέγχει ο ίδιος την πληροφορία που διαμοιράζεται, αλλά χωρίς να μειωθεί η εμπειρία του (πχ να χρειάζεται να διατηρεί πολλούς λογαριασμούς σε διαφορετικές εφαρμογές. Αυτό το κενό έρχεται να καλύψει μια νέα πρακτική με το όνομα Self-Sovereign Identity (SSI) χρησιμοποιόντας ταυτοποιητικά δεδομένα.

Ο χρήστης συνδέεται στις ίδιες εφαρμογές με πριν. Οι εφαρμογές αιτούνται τα ταυτοποιητικά δεδομένα και ο χρήστης μπορεί να αποδεχτεί ή να απορρίψει την αίτηση. Τα ταυτοποιητικά δεδομένα εκδιδονται από διάφορους εκδότες (Κρατικές οντότητες, πανεπιστήμια, διάφοροι οργανισμοί) συμπεριλαμβανομένου και του ιδίου του χρήστη. Είναι στην ευχέρεια της εφαρμογής το να εμπιστευτεί κάποιον εκδότη. Τα ταυτοποιητικά δεδομένα πλέον δε βρίσκονται στο χώρο του εκδότη, αλλά αποθηκεύονται σε κάποιο ηλεκτρονικό πορτοφόλι (digital wallet). Το ηλεκτρονικό πορτοφόλι μπορεί να είναι είτε μια εφαρμογή στο κινητό, είτε μια διαδικτυακή εφαρμογή στο cloud.

Οι υπηρεσίες αιτούνται ένα ή περισότερα στοιχεία από τα ταυτοποιητικά δεδομένα. Για παράδειγμα μια υπηρεσία που πουλάει αλκοόλ ζητάει ταυτοποίηση ότι ο χρήστης είναι ενήλικας, οπότε ζητάει την ημερομηνία γέννησης μόνο. Ή ακόμα καλύτερα η υπηρεσία μπορεί να ζητήσει από τα ταυτοποιητικά δεδομένα αν ο χρήστης είναι ενήλικας, χωρίς να αιτηθεί την ίδια την ημερομηνία γέννησης.

Κεφάλαιο 2
Επαληθεύσιμα διαπιστευτήρια

Όπως και στον φυσικό κόσμο, τα ταυτοποιητικά δεδομένα είναι μια συλλογή από ισχυρισμούς (claims). Για παράδειγμα στην ταυτότητα υπάρχει ως ισχυρισμός το όνομα, το επίθετο η ημερομηνία γέννησης κ.α. Αντίστοιχα και στον ψηφιακό κόσμο, τα ταυτοποιητικά είναι μια συλλογή από αντίστοιχους ισχυρισμούς για το αντικείμενο. Επιπλέον με τη χρήση τεχνολογιών όπως οι ψηφιακές υπογραφές τα ταυτοποιητικά δεδομένα γίνοται tamper-evident, δηλαδή αποκτά ένδειξη απαραβίαστου.

Αυτή την ανάγκη, δηλαδή τη δημιουργία ενός προτύπου το οποίο θα περιγράφει ψηφιακά και επαληθεύσιμα οποιαδήποτε μορφής ταυτοποιητικά δεδομένα, με κρυπτογραφημένο και ασφαλή τρόπο για χρήση στο ψηφιακό κόσμο, έρχεται να καλύψει το «World Wide Web Consortium (W3C)» με τα Επαληθεύσιμα Διαπιστευτήρια / Verifiable Credentials (VC) [13]

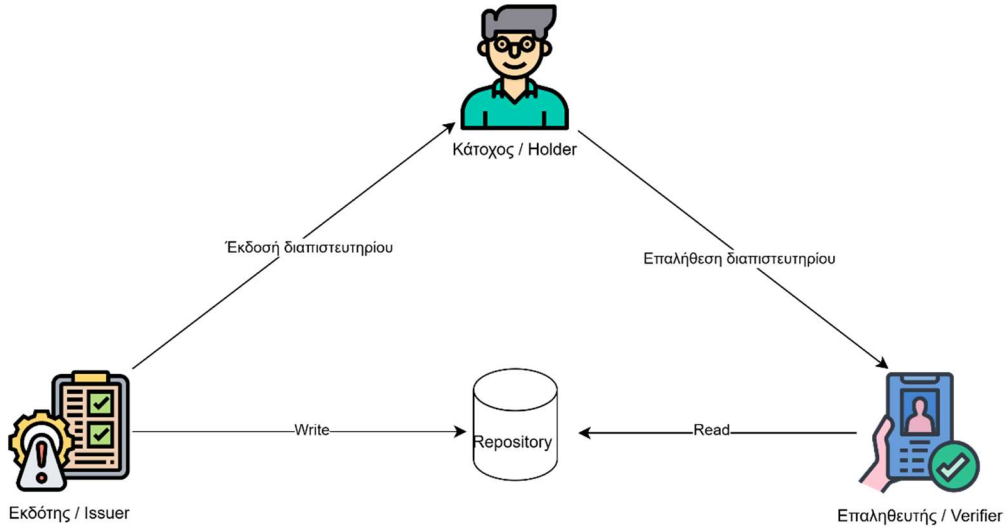
2.1 ΡΟΛΟΙ

Πριν αναλύσουμε την τεχνική μορφή των επαληθεύσιμων διαπιστευτηρίων πρέπει να ορίσουμε τους ρόλους που συμμετέχουν στην δημιουργία, την αποθήκευση τη διαβίβαση και την επαλήθευση των τους. Αυτοί είναι :

1. Ο «Εκδότης» (Issuer). Πρόκειται για την οντότητα που εκδίδει το διαπιστευτήριο για μια άλλη οντότητα «Αντικείμενο». Παράδειγμα εκδότη είναι η Ελληνική Αστυνομία για ψηφιακές ταυτότητες και διαβατήρια, ή κάποιο πανεπιστήμιο για τίτλους σπουδών
2. Ο «Ελεγκτής» (Verifier). Είναι η οντότητα που θέλει να επαληθεύσει την εγκυρότητα του διαπιστευτηρίου και να αντλήσει ορισμένα ή όλα τα στοιχεία του. Τέτοιο παράδειγμα είναι ο έλεγχος διαβατηρίων στα αεροδρόμια, ένα πανεπιστήμιο που θέλει να επιβεβαιώσει τον προπτυχιακό τίτλο σποδών ενός υποψηφίου φοιτητή, ακόμα και από το κοντινό μας παράδειγμα του covid, οι υπεύθυνοι χώρων για τους οποίους είναι απαιτηθεί πιστοποιητικό εμβολιασμού κατά την είσοδο.
3. Το «Αντικείμενο» (Subject). Πρόκειται για την οντότητα στην οποία αναφέρεται το διαπιστευτήριο. Για παράδειγμα ο κάτοχος μιας ταυτότητας ή ενός διαβατηρίου, ακόμα και ένα ζώο όταν πρόκειται για πιστοποιητικό εμβολιασμού.
4. Ο «Κάτοχος» (Holder). Πρόκειται για την οντότητα που έχει αποθηκεύσει και κάνει χρήση ενός η περισσότερων διαπιστευτηρίων. Ο κάτοχος συνήθως ταυτίζεται με τον αντικείμενο, όπως στην περίπτωση της ταυτότητας ή του διαβατηρίου. Ωστόσο αυτές οι έννοιες μπορεί να είναι διαφορετικές όπως στην περίπτωση ενός κατοικίδιου το οποίο είναι το αντικείμενο, ενώ ο ιδιοκτήτης είναι ο κάτοχος του διαπιστευτηρίου. Οι κάτοχοι αποθηκεύουν τα διαπιστευτήρια σε αποθετήρια όπως τα ψηφιακά πορτοφόλια.

Για τη συνέχεια του κειμένου ο κάτοχος και το αντικείμενο ταυτίζονται, εκτός και αν αναφέρεται αλλιώς. Επίσης για την υλοποίηση κάτοχος και αντικείμενο είναι η ίδια οντότητα. Οι οντότητες μπορεί να είναι φυσικά πρόσωπα, νομική οντότητα ή συσκευή

Οι παραπάνω οντότητες λειτουργούν με ένα τριγωνικό σχήμα πίστης όπως φαίνεται και στο παρακάτω σχήμα.



Εικόνα 1Ρόλοι και η μεταξύ τους ροή

Γίνεται αντιληπτό ότι:

- Ο εκδότης εμπιστεύεται τον κάτοχο
- Ο κάτοχος εμπιστεύεται το επαληθευτή
- Ο ελεγκτής εμπιστεύεται τον εκδότη

Ωστόσο, επειδή ο καθένας μπορεί να εκδώσει επαληθεύσιμα διαπιστευτήρια είναι απόφαση του επαληθευτή αν θα εμπιστευτεί τον εκδότη ή όχι.

2.2 ΑΝΑΠΑΡΑΣΤΑΣΗ

Η αναπαράσταση ενός επαληθεύσιμου διαπιστευτηρίου γίνεται με τη χρήση της σύνταξης JSON (JavaScript Object Notation) [14] και της επέκτασής JSON-LD[15][16]. Το JSON-LD μια μέθοδος για την κωδικοποίηση και τη διασύνδεση δομημένων δεδομένων με τη χρήση της σύνταξης JSON.

Η χρήση JSON και JSON-LD είναι η πιο διαδεδομένη μορφή αναπαράστασης των επαληθεύσιμων διαπιστευτηρίων. Ωστόσο στο πρότυπο δεν υπάρχει περιορισμός στην αποκλειστική τους χρήση. Οι διάφορες υπηρεσίες γύρω από τα επαλιθεύσιμα διαπιστευτήρια μπορούν να κάνουν χρήση άλλων μορφών αναπαράστασης δεδομένων, όπως XML, YAML ή CBOR. Στην παρούσα εργασία γίνεται χρήση των JSON/JSON-LD.

Παρακάτω βλέπουμε την αναπαράσταση ενός παραδείγματος επαληθεύσιμου διαπιστευτηρίου. Τα κύρια συστατικά του θα αναλυθούν στην επόμενη ενότητα.


```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "http://example.edu/credentials/1872",
  "type": [
    "VerifiableCredential",
    "AlumniCredential"
  ],
  "issuer": "https://example.edu/issuers/565049",
  "issuanceDate": "2010-01-01T19:23:24Z",
  "credentialSubject": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
      "name": [
        {
          "value": "Example University",
          "lang": "en"
        },
        {
          "value": "Exemple d'Université",
          "lang": "fr"
        }
      ]
    }
  },
  "proof": {
    "type": "RsaSignature2018",
    "created": "2017-06-18T21:19:10Z",
    "proofPurpose": "assertionMethod",
    "verificationMethod": "https://example.edu/issuers/565049#key-1",
    "jws":
    "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5XsITJX1CxPCT8yAV-
    TVkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-pQy7UJiN5mgRxD-WUcX16dUEMGlv50aqzpqh4Qktb3rk-
    BuQy72IFLOqV0G_zS245-kronKb78cPN25DGlcTwLtjPAYuNzVBAh4vGHSrQyHUdBBPM"
  }
}
```


Εικόνα 2 Παράδειγμα επαληθεύσιμου διαπιστευτηρίου

2.3 ΔΟΜΗ

@Context

Το πεδίο (property) **@context** καθορίζει τον τύπο του διαπιστευτιρίου και των πεδίων που αυτό μπορεί να έχει. Η τιμή του πεδίου αυτού είναι μια διατεταγμένη λίστα, όπου το πρώτο στοιχείο θα πρέπει να είναι ένα Uniform Resource Identifier (URI) και να έχει τιμή *https://www.w3.org/2018/credentials/v1*. Αυτή η τιμή οδηγεί σε ένα βασικό έγγραφο που περιγράφει τα πεδία που μπορεί να περιέχει ένα επαληθεύσιμο διαπιστευτήριο. Το έγγραφό αυτό δεν ανανεώνεται ποτέ και πρέπει να αποθηκεύεται τοπικά. Κάθε άλλη τιμή στη λίστα μπορεί να είναι είτε URI είτε ένα αντικείμενο JSON.

Στο παραπάνω παράδειγμα το δεύτερο στοιχείο της λίστας οδηγεί στη διεύθυνση https://www.w3.org/2018/credentials/examples/v1. Οποίος λάβει το διαπιστευτήριο με βάση αυτό το έγγραφο και με χρήση JSON-LD μπορεί να επιβεβαιώσει ότι το διαπιστευτήριο έχει τα απαιτούμενα πεδία.

Identifier - Id

Κάθε οντότητα για να περιγράφει η να υπάρξει μια αναφορά σε αυτή, είναι απαραίτητο ένα υπάρχει ένα μοναδικό αναγνωριστικό. Το πεδίο **id** είναι αυτό το αναγνωριστικό. Κάθε αναγνωριστικό πρέπει να είναι μοναδικό. Το πεδίο πρέπει να έχει μια και μόνο μόνο τιμή, και αυτή να είναι ένα URI. Η τιμή αυτή χαρακτηρίζει μοναδικά το τμήμα στο οποίο εμφανίζεται. Στο παραπάνω παράδειγμα το πεδίο id εμφανίζεται 2 φορές. Η πρώτη εμφάνιση συσχετίζεται με το διαπιστευτήριο, ενώ η δεύτερη με το αντικείμενο του διαπιστευτηρίου (credential subject). Στη δεύτερη εμφάνιση παρατηρούμε ότι η τιμή έχει τη μορφή ενός Decentralized Identifier – DID, το οποίο εξετάζεται παρακάτω.

Σε περιπτώσεις που χρήζουν προστασία προσωπικών δεδομένων τα Id θα πρέπει να είναι μιας χρήσης. Εναλλακτικά να αντικατασταθούν από μικρής διάρκειας token μιας χρήσης. Αυτές οι τεχνικές εμποδίζουν τη προσπάθεια συσχέτισης των ids πχ ένας εκδότης και ένας ελεγκτής να προσπαθήσουν να συσχετίσουν έναν κάτοχο.

Type

Το πεδίο type είναι υποχρεωτικό για τα επαληθεύσιμα διαπιστευτήρια. Χωρίς αυτό ένα διαπιστευτήριο δεν είναι έγκυρο. Η τιμή του πεδίου πρέπει να είναι ένα ή περισσότερα URIs ή στο πεδίο @context θα πρέπει να υπάρχει αντιστοίχιση για κάθε τύπο που περιέχεται. Στο παραπάνω παράδειγμα υπάρχουν 2 τύποι, οι VerifiableCredential and of AlumniCredential

Credential Subject

Το πεδίο **credentialSubject** περιέχει ένα ή περισσότερους ισχυρισμούς (claims) για το αντικείμενο του διαπιστευτηρίου. Επιπλέον μπορεί να έχει ισχυρισμούς περισσότερα από ένα αντικείμενα, Για παράδειγμα ένα διαπιστευτήριο γάμου μπορεί να περιέχει τα στοιχεία των συζύγων

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "http://example.edu/credentials/3732",
  "type": ["VerifiableCredential", "RelationshipCredential"],
  "issuer": "https://example.com/issuer/123",
  "issuanceDate": "2010-01-01T00:00:00Z",
  "credentialSubject": [{
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "name": "Jayden Doe",
    "spouse": "did:example:c276e12ec21ebfeb1f712ebc6f1"
  }, {
    "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
    "name": "Morgan Doe",
    "spouse": "did:example:ebfeb1f712ebc6f1c276e12ec21"
  }]
}
```

Εικόνα 3 Πολλαπλά αντικείμενα στο επαληθεύσιμο διαπιστευτήριο

Issuer

Το πεδίο **issuer** είναι υποχρεωτικό. Ο τύπος των τιμών που μπορεί να έχει είναι είτε URI είτε ένα αντικείμενο που να περιέχει ένα πεδίο **id**. Με τη χρήση του αντικειμένου μπορούμε παρέχουμε περισσότερες πληροφορίες για τον εκδότη, όπως το όνομά του όπως φαίνεται στο παρακάτω παράδειγμα.

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "http://example.edu/credentials/3732",
  "type": ["VerifiableCredential", "UniversityDegreeCredential"],
  "issuer": {
    "id": "did:example:76e12ec712ebc6f1c221ebfeb1f",
    "name": "Example University"
  },
  "issuanceDate": "2010-01-01T19:23:24Z",
  "credentialSubject": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "degree": {
      "type": "BachelorDegree",
      "name": "Bachelor of Science and Arts"
    }
  }
}
```

Εικόνα 4 Αντικείμενο ως εκδότης

Εναλλακτικές μορφές για την τιμή του πεδίου είναι η χρήση DID ή JSON Web Key – JWK

- Did - did:example:abfe13f712120431c276e12ecab
- JWK - https://example.com/keys/foo.jwk

Issuance Date

Το πεδίο **issuanceDate** είναι υποχρεωτικό πεδίο όπου αναφέρεται η χρονική στιγμή που το διαπιστευτήριο γίνεται έγκυρο. Η τιμή πρέπει να είναι UTC όπως ορίζεται στο XMLSCHEMA11-2 [18]. Σε μελλοντική έκδοση του πρωτοτύπου, το πεδίο θα αντικατασταθεί από το πεδίο **issued**. Επίσης θα προστεθεί ένα νέο πεδίο το validFrom**.** Και τα 2 πεδία θα έχουν τον ιδιο τύπο με το issuanceDate.

Proof

Το διαπιστευτήριο πρέπει να έχει τουλάχιστο ένα μηχανισμό επαλήθευσης καθώς και τα στοιχεία για να πραγματοποιήσει την επαλήθευση. Υπάρχουν δύο τύποι μηχανισμών επαλήθευσης, ο εξωτερικός και ο ενσωματωμένος. Παράδειγμα εξωτερικού μηχανισμού είναι το JSON Web Token - JWT [19], ενώ παράδειγμα ενσωματωμένου είναι η περίπτωση στο παράδειγμα του διαπιστευτηρίου (Κώδικας 1). Σε αυτή την πείπτωση το πεδίο **proof** απαιτείται.

Οι μηχανισμοί αυτοί είναι κατά κύριο λόγο ψηφιακές υπογραφές και δεν υπάρχει κάποιος συγκεκριμένος που να έχει προσδιοριστεί για τα επαληθεύσιμα διαπιστευτήρια. Το ίδιο το πρότυπο προτείνει τρεις μηχανισμούς:

- JSON Web Token - JWT [19] με χρήση JSON Web Signatures [20]
- Data Integrity Proofs [21]

- Camenisch-Lysyanskaya Zero-Knowledge Proofs [22]

Ειδική προσοχή χρειάζεται όταν απαιτούνται μέτρα αποφυγής συσχετίσεων.

Expiration

Το πεδίο **expirationDate** είναι προαιρετικό. Αν υπάρχει, θα πρέπει να είναι τύπου string με τη μορφή που ορίζεται στο XMLSCHEMA11-2 [18]. Δηλώνει τη στιγμή που το διαπιστευτήριο δεν θα είναι πλέον έγκυρο. Σε μελλοντική έκδοση του πρωτοτύπου θα προστεθεί η ιδιότητα **validUntil**.

Status

Το πεδίο **credentialStatus** είναι προαιρετικό. Αν υπάρχει, θα πρέπει να είναι ένα αντικείμενο με υποχρεωτικά τα εξής πεδία

- **id**, το οποίο θα είναι τύπου URI
- **type**, το οποίο θα έχει πληροφορία αναγνωρίσιμη από μηχανή σχετικά με την κατάσταση του διαπιστευτηρίου.

2.4 ΕΠΑΛΗΘΕΥΣΙΜΗ ΠΑΡΟΥΣΙΑΣΗ

Η επαληθεύσιμη παρουσίαση (verifiable presentation – vp) είναι μια συλλογή από ένα ή περισσότερα επαληθεύσιμα διαπιστευτήρια . Η δημιουργία της παρουσίασης πραγματοποιείται με τέτοιο τρόπο ώστε η κυριότητα να μπορεί να επαληθευτεί. Είναι σύνηθες μια επαληθεύσιμη παρουσίαση να είναι σύμπτυξη στοιχείων από πολλαπλά επαληθεύσιμα διαπιστευτήρια. Τα διαπιστευτήρια δεν χρειάζεται να είναι από την ίδιο εκδότη.

Οι επαληθεύσιμες παρουσιάσεις δομή παρόμοια με τα επαληθεύσιμα διαπιστευτήρια. Τα βασικά τους στοιχεία είναι:

- id, ένα μοναδικό αναγνωριστικό. Πρόκειται για προαιρετικό πεδίο που ακολουθεί τους κανόνες που αναφέρθηκαν παραπάνω.
- type, υποχρεωτικό πεδίο, όπου όριζεται ο τύπος της παρουσίας, όπως “VerifiablePresentation” Ακολουθεί τους ίδιους κανόνες που αναφέρθηκαν για τα συγκεκριμένα πεδία.
- verifiableCredential, το πεδίο περιέχει ένα ή περισσότερα επαληθεύσιμα διαπιστευτήρια ή δεδομένα που προέρχονται από αυτά με επαληθεύσιμη με κρυπτογραφικά μέσα μορφή.
- holder, προεραιτικό πεδίο, τύπου URI που για την οντότητα που δημιουργεί την αναπαράσταση
- proof, το οποίο αν υπάρχει, εξασφαλίζει ότι η παρουσίαση είναι επαληθεύσιμη. Ισχύουν όσα αναφέρθηκαν παραπάνω για τα επαληθεύσιμα διαπιστευτήρια.

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "type": "VerifiablePresentation",

  "verifiableCredential": [{
    "@context": [
      "https://www.w3.org/2018/credentials/v1",
      "https://www.w3.org/2018/credentials/examples/v1"
    ],
    "id": "http://example.edu/credentials/1872",
    "type": ["VerifiableCredential", "AlumniCredential"],
    "issuer": "https://example.edu/issuers/565049",
    "issuanceDate": "2010-01-01T19:23:24Z",
    "credentialSubject": {
      "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
      "alumniOf": {
        "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
        "name": [{
          "value": "Example University",
          "lang": "en"
        }, {
          "value": "Exemple d'Université",
          "lang": "fr"
        }]
      }
    },
    "proof": {
      "type": "RsaSignature2018",
      "created": "2017-06-18T21:19:10Z",
      "proofPurpose": "assertionMethod",
      "verificationMethod": "https://example.edu/issuers/565049#key-1",
      "jws":
      "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5XsITJX1CxP
      CT8yAV-TVkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-pQy7UJiN5mgRxD-WUcX16dUEMGlv50a
      qzpqh4Qktb3rk-BuQy72IFLOqV0G_zS245-kronKb78cPN25DGlcTwLtjPAYuNzVBAh4vGHSrQ
      yHUdBBPM"
    }
  }],

  "proof": {
    "type": "RsaSignature2018",
    "created": "2018-09-14T21:19:10Z",
    "proofPurpose": "authentication",
    "verificationMethod": "did:example:ebfeb1f712ebc6f1c276e12ec21#keys-1",

    "challenge": "1f44d55f-f161-4938-a659-f8026467f126",
    "domain": "4jt78h47fh47",
    "jws":
    "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..kTCYt5XsITJX1CxPC
    T8yAV-TVIw5WEuts01mq-pQy7UJiN5mgREEMGlv50aqzpqh4Qq_PbChOMqsLfRoPsnsgxD-WUcX1
    6dUOqV0G_zS245-kronKb78cPktb3rk-BuQy72IFLN25DYuNzVBAh4vGHSrQyHUGlcTwLtjPAnKb
    78"
  }
}
```


Εικόνα 5 Παράδειγμα επαληθεύσιμης παρουσίασης

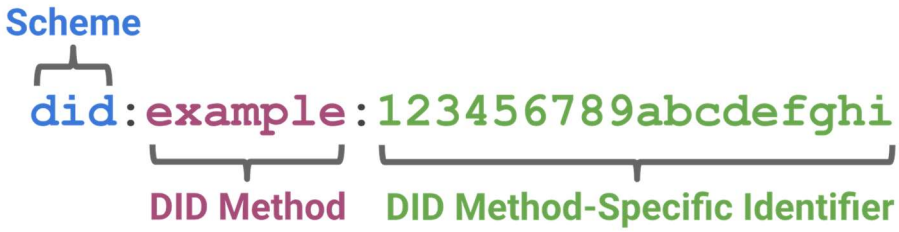
2.5 ΑΠΟΚΕΝΤΡΩΜΕΝΑ ΑΝΑΓΝΩΡΙΣΤΙΚΑ DIDS

Όλες οι οντότητες, φυσικά πρόσωπα και οργανισμοί, χρησιμοποιούν συγκεκριμένα αναγνωριστικά ανάλογα με το περιβάλλον την ενέργεια που εκτελούν. Έτσι υπάρχουν σε σειρά από αναγνωριστικά όπως αριθμοί ταυτότητας, διαβατηρίου, αφμ, διευθύνσεις, τηλέφωνα κ.α Τα αναγνωριστικά αυτά δημιουργούνται και διαχειρίζονται από συγκεκριμένες κεντρικές δομές (εκδότες), ανάλογα με τον τύπο της λειτουργίας που θέλουμε να εκτελέσουμε. Ωστόσο δεν υπάρχει έλεγχος στις δομές αν διαμοιράζονται τις πληροφορίες με ποιους και πότε.

Τα αποκεντρωμένα αναγνωριστικά όπως ορίζονται στο πρότυπο του W3C [23] είναι ένας νέος τύπος μοναδικών αναγνωριστικών, τα οποία ανήκουν μόνο στους δημιουργούς τους. Δεν υπάρχει κάποια κεντρική δομή που να ελέγχει τη δημιουργία και την ανάθεσή τους. Επιπλέον είναι επαληθεύσιμα με κρυπτογραφικά μέσα.

Εφόσον αποκεντρωμένα αναγνωριστικά ελέγχονται αποκλειστικά από την οντότητα που τα κατέχει, τότε αυτή μπορεί να εκδώσει όσα χρειάζεται για να καλύψει κάθε περίπτωση ανάλογα με το περιβάλλον που απευθύνεται.

Στην παρακάτω εικόνα βλέπουμε ένα παράδειγμα ενός DID.



Εικόνα 6 Απλό παράδειγμα αποκεντρωμένου αναγνωριστικού DID

Αποτελείται από τρία μέρη:

- Το did URI scheme αναγνωριστικό (Scheme)
- η DID μεθόδου (DID Method) [27]
- το αναγνωριστικό για τη συγκεκριμένη DID μέθοδο (DID Method-Specific Identifier

Κάθε έγκυρο DID αναγνωριστικό αντιστοιχεί σε ένα DID έγγραφο [23] το οποίο έχει ένα μοναδικό αναγνωριστικό. Το έγγραφο περιέχει πληροφορίες που σχετίζονται με το αναγνωριστικό όπως public keys.

```
{
    "@context": [
      "https://www.w3.org/ns/did/v1",
      "https://w3id.org/security/suites/ed25519-2020/v1"
    ],
    "id": "did:example:123456789abcdefghi",
    "authentication": [{

      "id": "did:example:123456789abcdefghi#keys-1",
      "type": "Ed25519VerificationKey2020",
      "controller": "did:example:123456789abcdefghi",
      "publicKeyMultibase": "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
    }]
  }
```

Εικόνα 7 Παράδειγμα DID εγγράφου

Το scheme είναι ένα URI και ακολουθεί τη ονοματολογία όπως ορίζεται στο RFC3886[25] Στη γενική περίπτωση χρησιμοποιείται το λεκτικό did. Η μέθοδος ορίζει την αλληλεπίδραση με το DID έγγραφο. Υπάρχουν αρκετές μέθοδοι με ποικίλους μηχανισμούς και κανόνες για την αποθήκευση Είναι σύνηθες να γίνεται χρήση της τεχνολογίας blockchain ως αποθετήριο επαληθεύσιμων δεδομένων (Verifiable Data Registry – VDR).

Αυτή τη στιγμή υπάρχουν πάνω από 100 εγγεγραμμένες μέθοδοι [26] Μερικές από τις πιο γνωστές είναι:

- did:btcr, βασισμένο στο δημόσιο δίκτυο Bitcoin[28]
- did:ethr, βασισμένο στο Etherium blockchain[29]
- did:web, όπου τα DID έγγραφα αποθηκεύονται σε κάποιον web serve με μορφή JSON, κάτω από ένα **well-known** μονοπάτι

2.6 ΑΠΟΘΕΤΗΡΙΟ ΕΠΑΛΗΘΕΥΣΙΜΩΝ ΔΕΔΟΜΕΝΩΝ

Το αποθετήριο επαληθεύσιμων δεδομένων (Verifiable Data Registry – VDR) είναι το σύστημα αποθήκευσης των DID documents. Δεν υπάρχουν συγκεκριμένες προδιαγραφές για την υλοποίηση ενός αποθετηρίου. Μπορεί να είναι ένα δίκτυο blockchain, ένας ιστότοπος ή ένα οποιοδήποτε κατανεμημένο σύστημα, ακόμα και filesystem.

2.7 ΙΩΣ BLOCKCHAIN

Το blockchain είναι μια κατανεμημένη βάση δεδομένων που ονομάζεται αποκεντρωμένο ψηφιακό καθολικό. Η βάση αυτή λόγω της κατανεμημένης μορφής της στηρίζεται σε ένα δίκτυο πολλών υπολογιστών στο διαδίκτυο και καταγράφουν με ασφαλή τρόπο δεδομένα συναλλαγών. Τα δεδομένα του blockchain οργανώνονται σε block, τα οποία είναι χρονολογικά διατεταγμένα και ασφαλισμένα με κρυπτογράφηση.

Ενώ η πρώτη αναφορά για το blockchain έγινε το 1991[48], η εμφάνιση της τεχνολογίας γίνεται το 2008 από τον Satoshi Nakamoto με την εμφάνιση του κρυπτονομίσματος Bitcoin. Μέχρι σήμερα έχουν εμφανιστεί μια σειρά από κρυπτονομίσματα (πχ Etherium) και η τεχνολογία χρησιμοποείται και σε άλλυς χώρους όπως το gaming, τα έξυπνα συμβόλαια (smart contracts), εφοδιαστικές αλυσίδες (supply chains) κ.α.

Σε ένα αποκεντρωμένο δίκτυο blockchain, δεν υπάρχει κεντρική αρχή ή μεσάζων που να ελέγχει τη ροή δεδομένων ή συναλλαγών. Αντ' αυτού, οι συναλλαγές επαληθεύονται και καταγράφονται από ένα κατανεμημένο δίκτυο υπολογιστών που συνεργάζονται για να διατηρήσουν την ακεραιότητα του δικτύου.

Το blockchain δεν είναι τίποτα παραπάνω από μια διασυνδεδεμένη λίστα αντικειμένων που ονομάζονται blocks. Το block μπορεί να περιέχει οποιαδήποτε μορφή πληροφορίας. Κάθε block έχει οπωσδήποτε μια χρονική σήμανση και έχει κρυπτογραφηθεί από τον ιδιοκτήτη της πληροφορίας. Σε κάθε νέα εγγραφή σε αυτή τη λίστα η εγγραφή επαληθεύεται από το μηχανισμό ομοφωνίας. Υπάρχουν οι εξής τύποι μηχανισμών:

- Proof-of-work.: Οι συμμετέχοντες κόμβοι προσπαθούν να λύσουν ένα κρυπτογραφικό πρόβλημα. Με κάθε block που προστίθεται ή περισσότεροι κόμβοι εισέρχονται στο δίκτο, η δυσκολία του προβλήματος μεγαλώνει. Το Bitcoin χρησιμοποιεί αυτό το μηχανισμό.
- Proof-of-stake: Κάποιοι από τους συμμετέχοντες κόμβους ορίζονται ως επικύρωτές και αυτοί αποφασίζουν για την ομοφωνία.

Αυτοί είναι οι κύριοι μηχανισμοί. Ωστόσο υπάρχουν και άλλοι. Κάποιοι είναι υβρίδια των δύο παραπάνω μηχανισμών και κάποιοι ακολουθούν άλλο δρόμο.

Οφέλη του blockchain είναι :

- Αποκέντρωση : Δεν υπάρχει κανένα κεντρικό σημείο ελέγχου
- Διαφάνεια: Οι συναλλαγές είναι ορατές σε όλους ώστε να διευκολύνεται η επαλήθευση
- Αμετάβλητο: Μόλις μια εγγραφή λάβει μέρος στο δίκτυο, τότε αυτή δε μπορεί να μεταβληθεί ή να διαγραφεί. Αυτός είναι ο λόγος που θα πρέπει να αποφεύγονται προσωπικά στοιχεία να γράφονται στο blockchain.
- Μη αποκυρήξημο: Επειδη υπάρχει η αμεταβλητότητα, ο ιδιοκτήτης της εγγραφής δεν μπορεί να αρνηθεί την ύπαρξή της.

2.8 ΠΟΡΤΟΦΟΛΙ / WALLET

Το ψηφιακό πορτοφόλι (digital wallet) η αποθετήριο (repository) είναι λογισμικό στο οποίο αποθηκεύεται το επαληθεύσιμο διαπιστευτήριο. Το λογισμικό αυτό μπορεί να βρίσκεται πάνω σε κινητά, tablets και προσωπικούς υπολογιστές, αλλά να προσφέρεται και ως υπηρεσία ως SaaS Cloud εφαρμογής. Το πορτοφόλι είναι διαχειρίζεται με ασφάλεια το διαπιστευτήριο και αυτό δημιουργεί την επαληθεύσιμη παρουσίαση (vp) όταν ζητείται.

Κεφάλαιο 3
OAuth και OpenID

3.1 OPENID

Η ιστορία του OpenID ξεκινάει το 2005 ως προσπάθεια να αναπτυχθεί ένα κατανεμημένο σύστημα ταυτοποίησης, ώστε οι χρήστες να μπορούν να χρησιμοποιούν τα ίδια διαπιστευτήρια (login / password) σε διαφορετικούς ιστοτόπους και εφαρμογές, οι τελευταίοι να διατηρούν βάση δεδομένων με τα διαπιστευτήρια των χρηστών τους.

Το αρχικό πρωτόκολλο τράβηξέ τα βλέμματα πολλών εταιριών και μεγάλες εταιρίες συνέβαλαν στην περαιτέρω ανάπτυξή του. Για τη διακυβέρνηση του σχήματος και την ανάπτυξη και βελτίωση των πρωτοκόλλων έχει δημιουργηθεί το OpenID Foundation, ένα μη κερδοσκοπικό σχήμα.

Αυτή τη στιγμή το OpenID Connect αποτελεί το κύριο πρότυπο για την αυθεντικοποίηση σε διαδικτυακές εφαρμογές.

3.2 OAUTH

Το Oauth είναι πρότυπο για την εξουσιοδότηση χρηστών, επιτρέποντάς τους να διαμοιράζουν πληροφορίες για τους λογαριασμούς σε τρίτες εφαρμογές ή ιστοτόπους.

Ξεκίνησε ως υλοποίηση του OpenID στο Twitter το 2006 και το 2012 έγινε πρότυπο από το I.E.T.F [31]. Αυτή τη στιγμή βρίσκεται στην έκδοση 2.0.

3.2.1 Ρόλοι

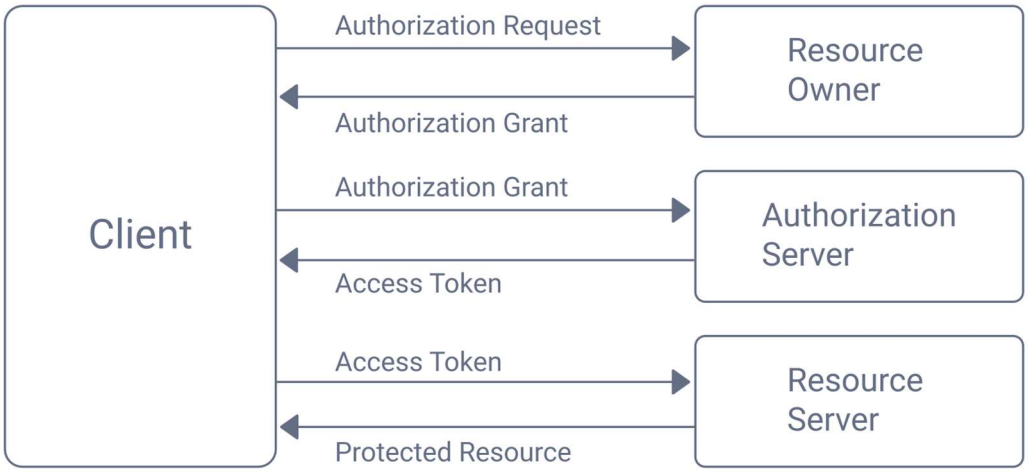
Στο πρωτόκολλο OAuth 2.0 διακρίνουμε τους εξής ρόλους

- Ιδιοκτήτης πόρων (resource owner) : Πρόκειται για την οντότητα που παρέχει πρόσβαση στους πόρους, συνήθως ο χρήστης
- Εξυπηρετητής πόρων (resource server): Η οντότητα που διανέμει τους προστατευόμενους πόρους που συνήθως είναι ένα API. Ο εξυπηρετητής λαμβάνει και επαληθεύει ένα access token όταν επεξεργάζεται τις εισερχόμενες κλίσεις.
- Πελάτης (client): Η εφαρμογή που θέλει πρόσβαση στους πόρους για λογαριασμό του ιδιοκτήτη. Ο πελάτης κατά την κλήση προς τον εξυπηρετητή, στέλνει και ένα access token.
- Εξυπηρετητής εξουσιοδότησης (authorization server) : Η οντότητα που εξουσιοδοτεί τον ιδιοκτήτη και στη συνέχεια εκδίδει ένα access token προς τον πελάτη.

3.2.2 Λειτουργία

Το OAuth 2.0 έχει σχεδιαστεί ώστε να επιτρέπει στον πελάτη την πρόσβαση σε πόρους για λογαριασμό του ιδιοκτήτη. Ο πελάτης λαμβάνει ένα access token το οποίο καθορίζει τον τύπο της επιτρεπόμενης πρόσβασης. Το OAuth 2.0 δεν περιλαμβάνει την αυθεντικοποίηση του χρήστη. Επίσης οι πελάτες δεν λαμβάνουν πληροφορίες για την ταυτότητα των χρηστών εκτός και αν το OAuth συνδυαστεί με το OpenID Connect, όπως θα δούμε παρακάτω.

Στην παρακάτω εικόνα βλέπουμε μια γενική μορφή της ροής στο OAuth 2.0 και την αλληλεπίδραση μεταξύ των ρόλων.



Εικόνα 8 Γενικό διάγραμμα ροής OAuth 2.0

1. Ο πελάτης ζητά εξουσιοδότηση από τον ιδιοκτήτη. Η αίτηση μπορεί να γίνει είτε απευθείας στον ιδιοκτήτη, είτε έμμεσα με ενδιάμεσο τον εξυπηρετητή αυθεντικοποίησης.
2. Ο πελάτης λαμβάνει το authorization grant, το οποίο ανήκει σε ένα από τους 4 τύπους που θα δούμε παρακάτω.
3. Ο πελάτης ζητά access token από τον εξυπηρετητή εξουσιοδότησης παρουσιάζοντας του το grant που έλαβε στο προηγούμενο βήμα.
4. Ο εξυπηρετητής εξουσιοδότησης αυθεντικοποιεί τον πελάτη, επαληθεύει το grant και εκδίδει το access token.
5. Ο πελάτης ζητάει από τον εξυπηρετητή πόρων τον προστατευόμενο πόρο και παρουσιάζει το access token.
6. Ο εξυπηρετητής πόρων επαληθεύει το access token και εξυπηρετεί την αίτηση.

3.2.3 Enpoints

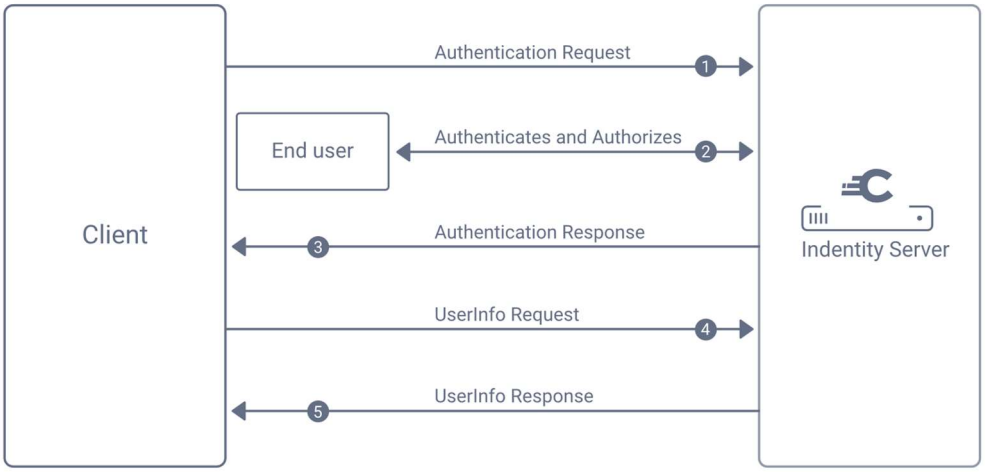
Το πρωτόκολλο ορίζει τρία endpoints που χρησιμοποιούνται για την αλληλεπίδραση μεταξύ των ρόλων:

- Authorization: χρησιμοποιείται από τον πελάτη για να λάβει την εξουσιοδότηση από τον ιδιοκτήτη του πόρου με ανακατεύθυνση στο user-agent
- Token: χρησιμοποιείται από τον πελάτη για την ανταλλαγή του authorization grant
- Redirection: χρησιμοποιείται από τον εξυπηρετητή εξουσιοδότησης για να επιστρέφει τα διαπιστευτήρια στον πελάτη μέσω του user-agent του ιδιοκτήτη.

3.3 OPENID CONNECT

Το OpenId Connect είναι ένα αλληλοσυμπληρωματικό προς το OAuth 2.0 πρωτόκολλο. Περιγράφει όλες τις απαραίτητες αλληλεπιδράσεις ώστε κάθε τύπος πελάτη να ζητάει και να λαμβάνει πληροφορίες αυθεντικοποίησης για τους τελικούς χρήστες καθώς και πρόσβαση στο backend API με τη χρήση OAuth 2.0 tokens.

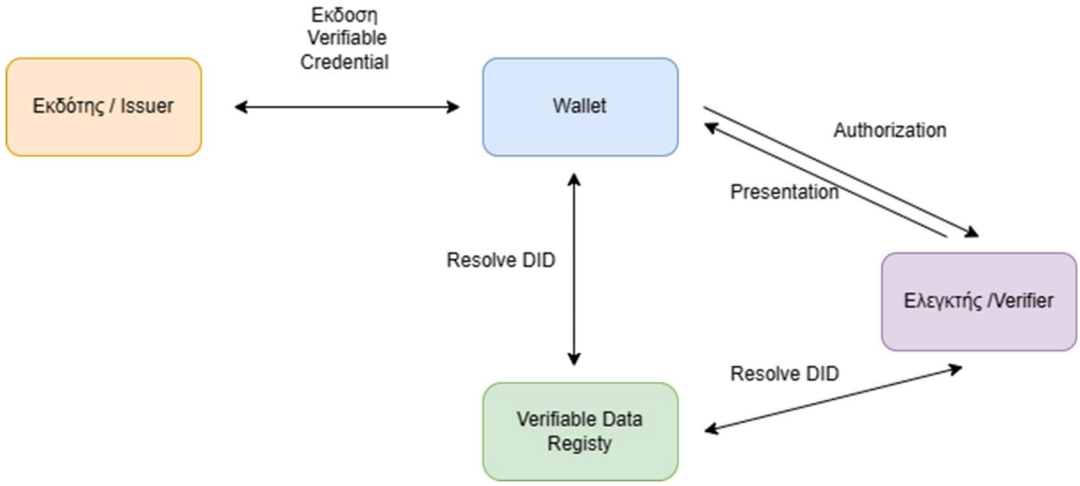
1. Ο πελάτης στέλνει αίτηση στον identity server για την αυθεντικοποίηση και εξουσιοδότηση του χρήστη.
2. Ο Identity server εξουσιοδοτεί το χρήστη
3. Ο Identity server επιστρέφει το αποτέλεσμα της αυθεντικοποίησης και εξουσιοδότησης στον πελάτη
4. Ο πελάτης προαιρετικά ζητάει πληροφορίες για τον χρήστη
5. Ο Identity Server επιστρέφει τις πληροφορίες του χρήστη



Εικόνα 9 Γενική ροή OpenID Connect

Κεφάλαιο 5
Υλοποίηση

Στο παρόν κεφάλαιο παρουσιάζονται οι επεκτάσεις που έχουν δημιουργηθεί στο OpenID Connect και OAuth 2.0 [34] ώστε να επιτυγχάνεται η αυθεντικοποίηση και εξουσιοδότηση με χρήση των επαληθεύσιμων διαπιστευτηρίων και των επαληθεύσιμων παρουσιάσεων. Η επικοινώνια των διαφορων εμπλεκομένων γίνεται με τη νέα ροή την οποία σε αυτή την εργασία αποκαλούμε **Verifiable Presentation Flow**.



Εικόνα 10 Verifiable Presentation Flow

Για το σκοπό της διατριβής αναπτυχθήκαν τρία διαφορετικές εφαρμογές

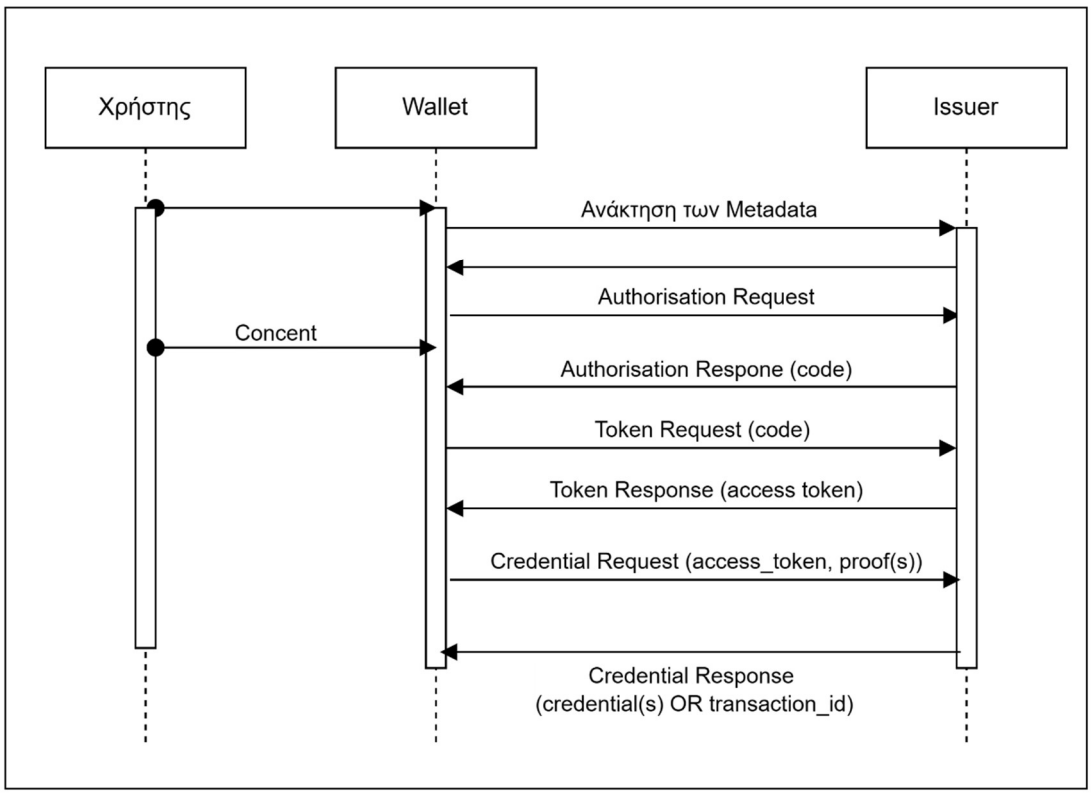
- Issuer, ο οποιος εκδίδει το επαληθεύσιμο διαπιστευτήριο
- Wallet, η εφαρμογή που αποθηκεύεται το διαπιστευτήριο και δημιουργεί την επαληθεύσιμη αναπαράσταση όταν ζητηθεί,
- Verifier: η εφαρμογή που ζητά τα επαληθεύσιμα διαπιστευτήρια.

Ο στόχος αυτών των εφαρμογών δεν είναι η πλήρης ανάπτυξη των επεκτάσεων ώστε να αποτελέσει ένα ολοκληρωμένο παραγωγικό σύστημα, αλλά να εξεταστούν οι νέες λειτουργικότητες. Για αυτό το λόγο ορισμένες λειτουργίες έγιναν με παραδοχές και ελλιπή χαρακτηριστικά.

4.1 ISSUER

Για την πληρότητα της εργασίας δημιουργήθηκε μια εφαρμογή η οποία εκδίδει επαληθεύσιμα διαπιστευτήρια. Πρόκειται για μια console εφαρμογή η οποία τυπώνει το διαπιστευτήριο σε signed JWT. Το συγκεκριμένο JWT θα μεταφορτωθεί χειροκίνητα στο Wallet σε επόμενο βήμα.

Όπως αναφέραμε και παραπάνω για την απλότητα της μελέτης χρησιμοποιήθηκαν πολλές παραδοχές. Μια από αυτή είναι ότι για Issuer χρησιμοποιούμε μια απλή εφαρμογή. Αντίθετα θα έπρεπε να είναι μια διαδικτυακή εφαρμογή με ισχυρή ασφάλεια. Αντί για τη χειροκίνητη μεταφόρτωση του διαπιστευτηρίου στο wallet, η επικοινωνία μεταξύ αυτών των 2 συστημάτων θα έπρεπε να γίνει αυτόματα ξεκινώντας από το πορτοφόλι. Η ροή που επιγραμματικά εικονίζεται στο παρακάτω σχήμα ορίζεται στο OpenID for Verifiable Credential Issuance [33].



Εικόνα 11 Βασική ροή για την έκδοση ενός διαπιστευτηρίου

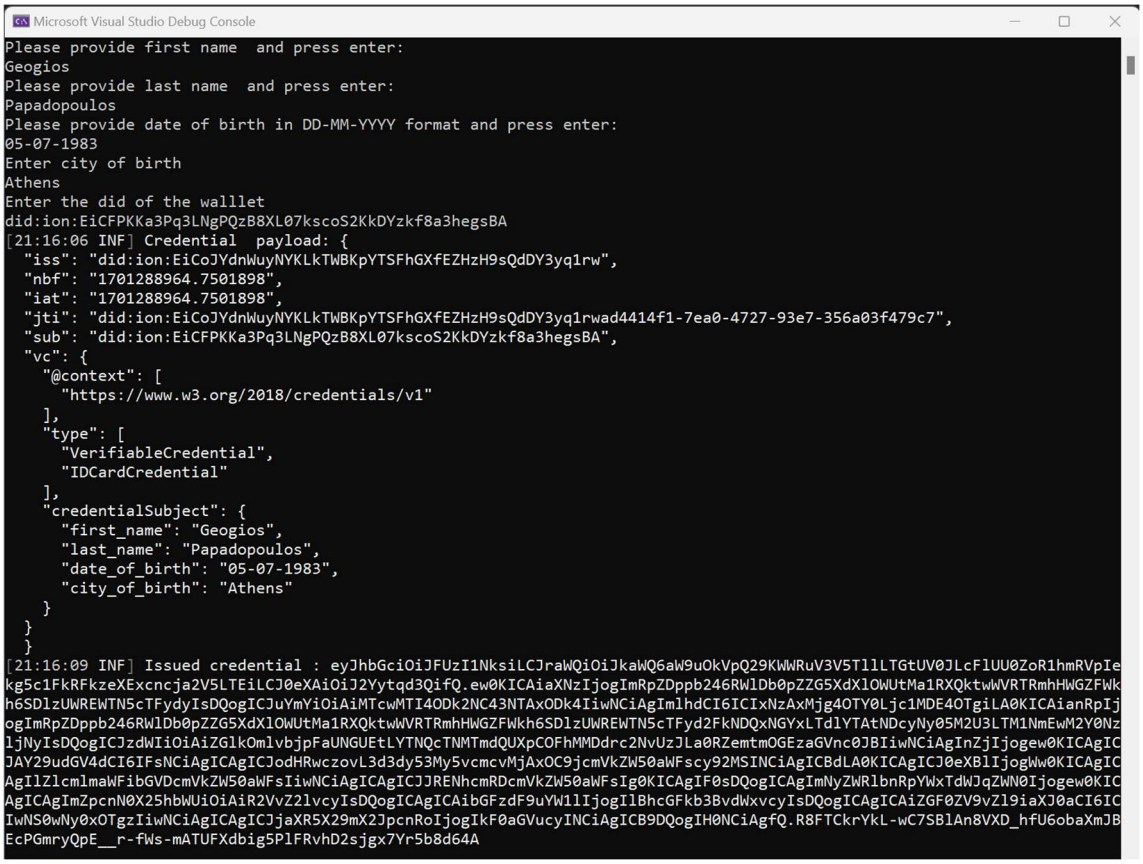
1. Το Wallet προσπελαύνει τα μεταδεδομένα του Isser από το */.well-known/openid-credential-issuer* Επίσης παρέχονται και τα δεδομένα του OAuth 2.0 εξυπηρετητή αυθεντικοποίησης
2. Το Wallet στέλνει ένα Authorisation request. Πριν σταλεί το response ο χρήστης καλείται να δώσει να συμπληρωσει Login/Password και να δώσει τη συγκατάθεσή του.
3. Ο Issuer στέλνει το response με το Authorization Code
4. Το Wallet στέλνει ένα Token Request με χρήση του Authorization Code. Ο Issuer επιστρέφει το Access Token
5. Το Wallet στέλνει ένα Credential Request με χρήση του Access Token και προαιρετικά, το proof της κατοχής του δημοσίου κλειδιού με το οποίο θα δεθεί το επαληθεύσιμο διαπιστευτήριο. Ο Issuer θα δημιουργήσει και θα επιστρέψει το διαπιστευτήριο.

Για το σκοπό της άσκησης δημιουργήσαμε ένα διαπιστευτήριο με αντικείμενο την ταυτότητα. Το ονομάσαμε IDCardCredential. Τα πεδία του διαπιστευτηρίου είναι :

- first_name
- last_name
- date_of_birth
- city_of_birth

Κατά την διαδικασία έκδοσης, ο Issuer ζητάει και το Wallet DID το οποίο χρησιμοποιούμε ως Subject στο διαπιστευτήριο. Κατά τη διαδικασία της εισαγωγής του επαληθεύσιμου διαπιστευτηρίου το Wallet DID διαφέρει από αυτό στο Wallet, τότε η εισαγωγή δεν γίνεται δεκτή.

Η υλοποίηση του Isser έγινε με c# και .net core 7.0. Παρακάτω βλέπουμε στιγμιότυπο από την εκτέλεση του Issuer



Εικόνα 12 Εκτέλεση του Issuer

Βλέπουμε το παραγόμενο διαπιστευτήριο καθώς και το υπογεγραμμένο JWT

```
{
 "iss": "did:ion:EiCoJYdnWuyNYKLkTWBKpYTSFhGXfEZHzH9sQdDY3yq1rw",
 "nbf": "1701288964.7501898",
 "iat": "1701288964.7501898",
 "jti":
 "did:ion:EiCoJYdnWuyNYKLkTWBKpYTSFhGXfEZHzH9sQdDY3yq1rwad4414f1-7ea0-4727-93e
 7-356a03f479c7",
 "sub": "did:ion:EiCFPKKa3Pq3LNgPQzB8XL07kscoS2KkDYzkf8a3hegsBA",
 "vc": {
   "@context": [
     "https://www.w3.org/2018/credentials/v1"
   ],
   "type": [
     "VerifiableCredential",
     "IDCardCredential"
   ],
   "credentialSubject": {
     "first_name": "Geogios",
     "last_name": "Papadopoulos",
     "date_of_birth": "05-07-1983",
     "city_of_birth": "Athens"
   }
 }
}
```


Εικόνα 13 Επαληθεύσιμο διαπιστευτήριο

HEADER: ALGORITHM & TOKEN TYPE


```
{
  "alg": "ES256K",
  "kid":
"did:ion:EiCoJYdnWuyNYKLkTWBKpYTSFhGXfEZHzH9sQdDY3yq1rw#
key-1",
  "typ": "vc+jwt"
}
```


Εικόνα 14 Η επικεφαλίδα από το JWT

PAYLOAD: DATA


```
{
  "iss":
"did:ion:EiCoJYdnWuyNYKLkTWBKpYTSFhGXfEZHzH9sQdDY3yq1rw
",
  "nbf": "1701288964.7501898",
  "iat": "1701288964.7501898",
  "jti":
"did:ion:EiCoJYdnWuyNYKLkTWBKpYTSFhGXfEZHzH9sQdDY3yq1rw
ad4414f1-7ea0-4727-93e7-356a03f479c7",
  "sub":
"did:ion:EiCFPKKa3Pq3LNgPQzB8XL07kscoS2KkDYzkf8a3hegsBA
",
  "vc": {
    "@context": [
      "https://www.w3.org/2018/credentials/v1"
    ],
    "type": [
      "VerifiableCredential",
      "IDCardCredential"
    ],
    "credentialSubject": {
      "first_name": "Geogios",
      "last_name": "Papadopoulos",
      "date_of_birth": "05-07-1983",
      "city_of_birth": "Athens"
    }
  }
}
```


Εικόνα 15 Το σώμα από το JWT

4.2 WALLET

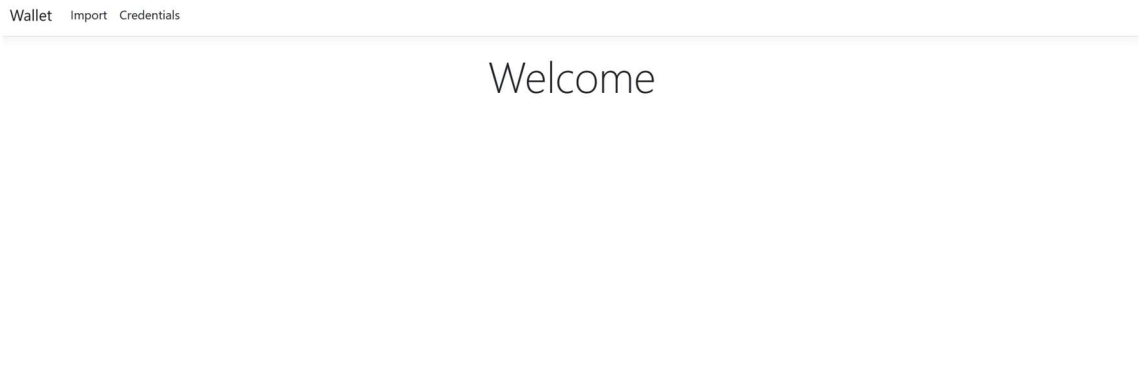
Η υλοποίηση του Wallet στα πλαίσια της συγκεκριμένης εργασίας είναι μια απλή διαδικτυακή εφαρμογή. Απέχει πολύ από το να χαρακτηριστεί πορτοφόλι αφού απουσιάζουν οι μηχανισμοί ασφαλείας που χαρακτηρίζουν μια τέτοια εφαρμογή, όπως authentication με login/password, biometrics ή οτιδήποτε άλλο.

Επιπλέον απουσιάζει οποιαδήποτε πρόβλεψη για κρυπτογράφηση στην αποθήκευση των επαληθεύσιμων διαπιστευτηρίων. Αντίθετα έχει χρησιμοποιηθεί μια απλή embedded NoSQL βάση δεδομένων.

Η λειτουργία της εφαρμογής επικεντρώνεται στην διαχείριση των επαληθεύσιμων διαπιστευτηρίων (εισαγωγή και διαγραφή), την αποδοχή αιτήσεων για επαληθεύσιμο διαπιστευτήριο και την δημιουργία και υπογραφή της επαληθεύσιμης παρουσίασης.

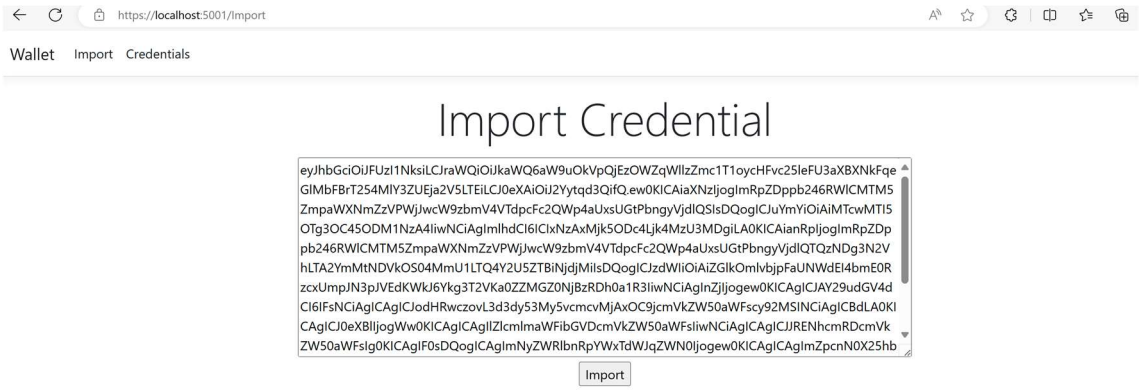
Η εφαρμογή έχει υλοποιηθεί ως Self Issued OpenID Provider [35], ωστόσο επικεντρώνεται μόνο στη σχετικά με τις επαληθεύσιμες παρουσιάσεις όπως ορίζονται στο OpenID for Verifiable Presentations [34]. Κάθε άλλη λειτουργία έχει παραλειφθεί. Στην επικοινωνία μεταξύ του Wallet και του Verifier ως clientId λαμβάνεται το DID του Verifier, όπως ορίζεται στις προδιαγραφές.

Η εφαρμογή του Wallet είναι μια διαδικτυακή εφαρμογή βασισμένη στο .net core και το asp .net core MVC.



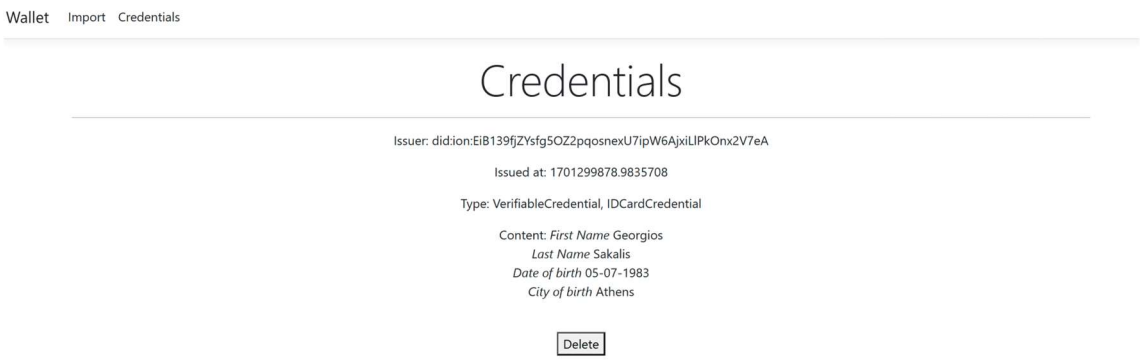
Από την αρχική οθόνη του wallet επιλέγουμε Import για να εισάγουμε το διαπιστευτήριο.

Εικόνα 16 Αρχική οθόνη Wallet



Εικόνα 17 Οθόνη εισαγωγής του διαπιστευτηρίου

Η επιτυχημένη προσθήκη του διαπιστευτηρίου μας οδηγεί στην οθόνη διαχείρισης των υπάρχοντών διαπιστευτηρίων



Εικόνα 18 Αποθηκευμένο διαπιστευτήριο

Όπως βλέπουμε στην παρακάτω εικόνα, το wallet επιβεβαιώνει την εγκυρότητα του JWT προτού προβεί στην αποθήκευση. Το wallet, αφού αποκωδικοποιήσει το εισερχόμενο JWT, από το πεδίο kid βρίσκει το DID id του issuer. Με αυτό ανασύρει το συγκεκριμένο DID document για τον issuer και εξάγει το public key του και πιστοποιεί την εγκυρότητα του JWT. Στο πεδίο vc βρίσκεται το διαπιστευτήριο.

```
.
[02:07:10 INF] Incoming JWT : eyJhbGciOiJFUzI1NksiLCJraWQiOiJkaWQ6aW9uOkVpQjEzOWZqWllzZmc1T1oycHFvc25leFU3aXBXNkFqeGlMbFBrT25
4MlY3ZUEja2V5LTEiLCJ0eXAiOiJ2Yytqd3QifQ.ew0KICAiaXNzIjogImRpZDppb246RWlCMTM5ZmpaWXNmZzVPWjJwcW9zbmV4VTdpcFc2QWp4aUxsUGtPbngyV
jdlQSIsDQogICJuYmYiOiAiMTcwMTI5OTg3OC45ODM1NzA4IiwNCiAgImlhdCI6ICIxNzAxMjk5ODc4Ljk4MzU3MDgiLA0KICAianRpIjogImRpZDppb246RWlCMT
M5ZmpaWXNmZzVPWjJwcW9zbmV4VTdpcFc2QWp4aUxsUGtPbngyVjdlQTQzNDg3N2VhLTA2YmMtNDVkOS04MmU1LTQ4Y2U5ZTBiNjdjMiIsDQogICJzdWIiOiAiZGl
kOmlvbjpFaUNWdEI4bmE0RzcxUmpJN3pJVEdKWkJ6Ykg3T2VKa0ZZMGZ0NjBzRDh0a1R3IiwNCiAgInZjIjogew0KICAgICJAY29udGV4dCI6IFsNCiAgICAgICJo
dHRwczovL3d3dy53My5vcmcvMjAxOC9jcmVkZW50aWFscy92MSINCiAgICBdLA0KICAgICJ0eXBlIjogWw0KICAgICAgIlZlcmlmaWFibGVDcmVkZW50aWFsIiwNC
iAgICAgICJJRENhcmRDcmVkZW50aWFsIg0KICAgIF0sDQogICAgImNyZWRlbnRpYWxTdWJqZWN0Ijogew0KICAgICAgImZpcnN0X25hbWUiOiAiR2Vvcmdpb3MiLA
0KICAgICAgImxhc3RfbmFtZSI6ICJTYWthbGlzIiwNCiAgICAgICJkYXRlX29mX2JpcnRoIjogIjA1LTA3LTE5ODMiLA0KICAgICAgImNpdHlfb2ZfYmlydGgiOiA
iQXRoZW5zIg0KICAgIH0NCiAgfQ0KICB9.R3oLNoLY_eNCUloOQfh-GQBV7n3oA1rmz0R8ZM14_sN-sAjLNf6G8nFz9VsXRgo1IlC4vvokxx2OYiThjhZFgg
[02:07:11 INF] Incoming JWT is Valid
[02:07:11 INF] Credential info: {
  "iss": "did:ion:EiB139fjZYsfg5OZ2pqosnexU7ipW6AjxiLlPkOnx2V7eA",
  "nbf": "1701299878.9835708",
  "iat": "1701299878.9835708",
  "jti": "did:ion:EiB139fjZYsfg5OZ2pqosnexU7ipW6AjxiLlPkOnx2V7eA434877ea-06bc-45d9-82e5-48ce9e0b67c2",
  "sub": "did:ion:EiCVtB8na4G71RjI7zITGJZBzbH7OeJkFY0ft60sD8tkTw",
  "vc": {
    "@context": [
      "https://www.w3.org/2018/credentials/v1"
    ],
    "type": [
      "VerifiableCredential",
      "IDCardCredential"
    ],
    "credentialSubject": {
      "first_name": "Georgios",
      "last_name": "Sakalis",
      "date_of_birth": "05-07-1983",
      "city_of_birth": "Athens"
    }
  }
  }
[02:07:11 INF] Credential to be imported : {@credentialObject}
[02:07:11 INF] Credential Saved: {"$oid":"6567e03fb06cd3098e53f70d"}
[02:07:11 INF] Executing RedirectResult, redirecting to /Credentials.
[02:07:11 INF] Executed action Wallet.Controllers.ImportController.Index (Wallet) in 443.779ms
[02:07:11 INF] Executed endpoint 'Wallet.Controllers.ImportController.Index (Wallet)'
[02:07:11 INF] HTTP POST /Import responded 302 in 461.7670 ms
```

Εικόνα 19 Επεξεργασία του εισερχομένου διαπιστευτηρίου

4.3 VERIFIER

Ο Verifier είναι ακόμα μια διαδικτυακή εφαρμογή η οποία παίζει το ρόλο του Relying Party (RP) όπως ορίζεται στην οικογένεια πρωτοκόλλων του OpenId. Γενικά ένα RP είναι μια οποιαδήποτε εφαρμογή η οποία απαιτεί αυθεντικοποίηση χρήστη. Σε επιτυχημένη αυθεντικοποίηση ο Verifier παρουσιάζει τα στοιχεία του χρήστη.

Στην παρούσα υλοποίηση ο Verifier επικοινωνεί κατευθείαν με το Wallet για να αντλήσει τα στοιχεία του χρήστη. Σε εναλλακτικές περιπτώσεις, όπως θα δούμε και στα συμπεράσματα, ο Verifier θα μπορεί να επικοινωνήσει μέσω OpenID Connect με ένα εξυπηρετητή αυθεντικοποίησης, και ο δεύτερος θα επικοινωνήσει με το Wallet μέσω των επεκτάσεων που αυτή τη στιγμή εξετάζουμε. Έτσι, η επέκταση για την υποστήριξη στο OpenID for Verifiable Presentations [34], μπορεί να γίνει στα κεντρικά συστήματα αυθεντικοποίησης χωρίς να αλλάξουν οι RP.

Η εφαρμογή του Verifier είναι μια διαδικτυακή εφαρμογή βασισμένη στο .net core και το asp .net core MVC.

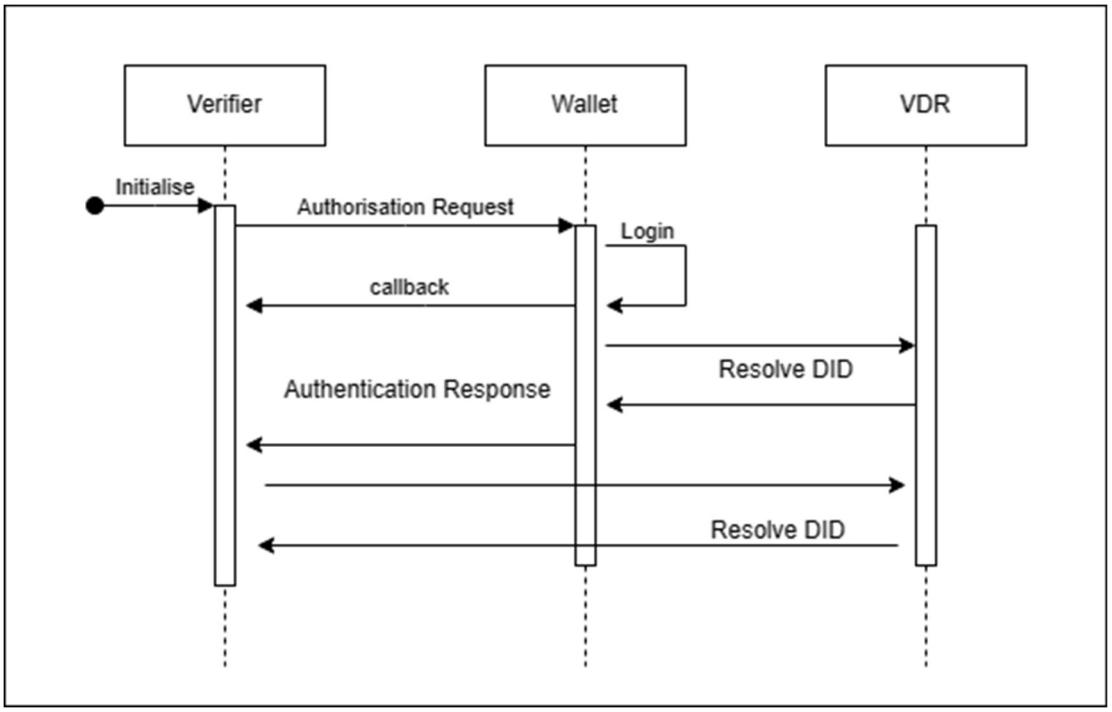
4.4 ΑΠΟΘΕΤΗΡΙΟ ΕΠΑΛΗΘΕΥΣΙΜΩΝ ΔΕΔΟΜΕΝΩΝ

Ως αποθετήριο επαληθεύσιμων δεδομένων (Verifiable Data Registry) θα μπορούσε να χρησιμοποιηθεί οπουδήποτε public blockchain δίκτυο. Στη συγκεκριμένη εργασία χρησιμοποιήσαμε το Identity Overlay Network (ION) το οποίο είναι Layer 2 open και βασίζεται στο bitcoin.

Χρησιμοποιήσαμε τα open source εργαλεία για να δημιουργηθούν DID documents για τον Issuer, Wallet και Verifier και τα public και Private keys. Καθώς το DID resolve δεν ήταν από τα κύρια σημεία αυτής της μελέτης, τα εξήχθησαν τα DID documents από το δίκτυο και αποθηκεύτηκαν τοπικά, ώστε να απλοποιηθεί η διαδικασία του resolve Και να μην υπάρχει εξάρτηση από το δίκτυο για την ολοκλήρωση της μελέτης. Τα υπόλοιπα στοιχεία επεξεργασίας κατά τη διαδικασία αυτή έχουν παραμείνει όπως περιεγραφήκαν.

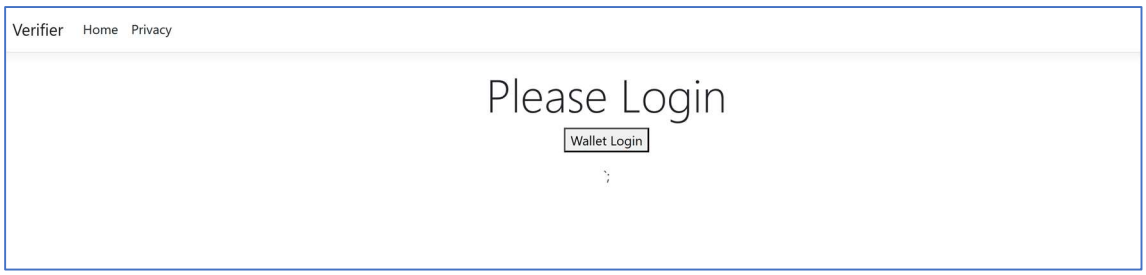
4.5 ΡΟΗ

Στην επόμενη εικόνα βλέπουμε σχηματικά τη ροή Verifiable Presentation Flow. Ο χρήστης αλληλεπιδρά με τον Verifier και καλείται να παρουσιάσει το επαληθεύσιμο διαπιστευτήριο του. Ο Verifier δημιουργεί ένα Authorization request προς το Wallet. Το Wallet επεξεργάζεται το request και κάνει resolve το DID του Verifier. Αφού επεξεργαστεί το request ζητά την έγκριση του χρήστη για την ολοκλήρωση της επαληθεύσιμης παρουσίασης. Αν ο χρήστης το απορρίψει, επιστρέφεται μήνυμα λάθους στο Wallet αλλιώς επιστρέφει πίσω την επαληθεύσιμη παρουσίαση. Κατά την επεξεργασία του response ο Verifier κάνει resolve το DID του Issuer για να ελέγξει την εγκυρότητα του.



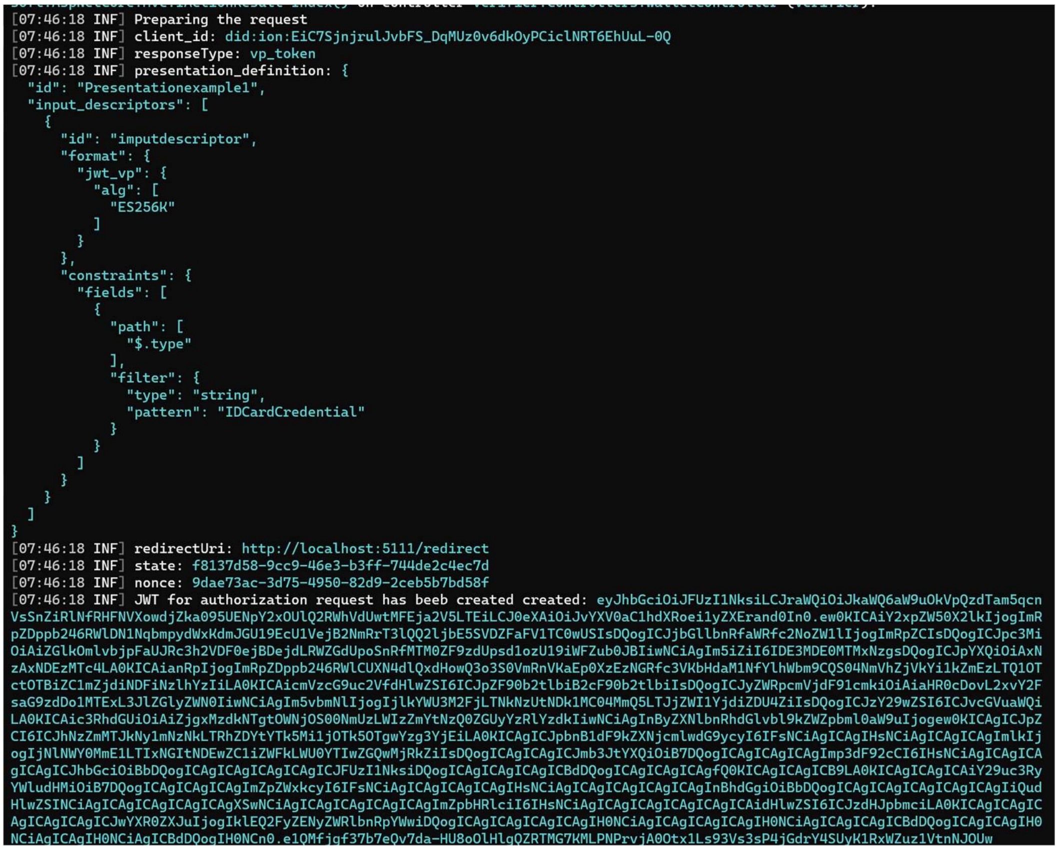
Εικόνα 20 Ροή Verifiable Presentation Flow

Ο χρήστης εισέρχεται στην αρχική οθόνη του Verifier



Εικόνα 21 Εισαγωγική οθόνη Verifier

Ο Verifier ως Relaying Party δημιουργεί ένα authorization request προς το Wallet. Συνήθως Η διαδικασία αυτή ξεκινάει με το σκανάρισμα κάποιου QR κωδικού που προβάλει το RP και σκανάρεται από το Wallet. Εδώ, όπως δηλώσαμε και παραπάνω η διαδικασία ξεκινάει ένα απλό σύνδεσμο προς το πορτοφόλι. Επιπλέον η διαδικασία αυθεντικοποίησης στο ίδιο το πορτοφόλι έχει απαλειφθεί. Σε παραγωγικό περιβάλλον ο χρήστης θα ενεργοποιούσε το πορτοφόλι με κάποιο βιομετρικό στοιχείο (π.χ δακτυλικό αποτύπωμα) ή username/password.



```
[07:46:18 INF] Preparing the request
[07:46:18 INF] client_id: did:ion:EiC7SjnjrulJvbFS_DqMUz0v6dkOyPCiclNRT6EhUuL-0Q
[07:46:18 INF] responseType: vp_token
[07:46:18 INF] presentation_definition: {
  "id": "Presentationexample1",
  "input_descriptors": [
    {
      "id": "imputdescriptor",
      "format": {
        "jwt_vp": {
          "alg": [
            "ES256K"
          ]
        }
      },
      "constraints": {
        "fields": [
          {
            "path": [
              "$.type"
            ],
            "filter": {
              "type": "string",
              "pattern": "IDCardCredential"
            }
          }
        ]
      }
    }
  ]
}
[07:46:18 INF] redirectUri: http://localhost:5111/redirect
[07:46:18 INF] state: f8137d58-9cc9-46e3-b3ff-744de2c4ec7d
[07:46:18 INF] nonce: 9dae73ac-3d75-4950-82d9-2ceb5b7bd58f
[07:46:18 INF] JWT for authorization request has beeb created created: eyJhbGciOiJFUzI1NksiLCJraWQiOiJkaWQ6aW9uOkVpQzdTam5qcn
VsSnZiRlNfRHFNVXowdjZka095UENpY2xOUlQ2RWhVdUwtMFEja2V5LTEiLCJ0eXAiOiJvYXV0aC1hdXRoei1yZXErand0In0.ew0KICAiY2xpZW50X2lkIjogImR
pZDppb246RWlDN1NqbmpydWxKdmJGU19EcU1VejB2NmRrT3lQQ2ljbE5SVDZFaFV1TC0wUSIsDQogICJjbGllbnRfaWRfc2NoZW1lIjogImRpZCIsDQogICJpc3Mi
OiAiZGlkOmlvbjpFaUJRc3h2VDF0ejBDejdLRWZGdUpoSnRfMTM0ZF9zdUpsd1ozU19iWFZub0JBIiwNCiAgIm5iZiI6IDE3MDE0MTMxNzgsDQogICJpYXQiOiAxN
zAxNDEzMTc4LA0KICAianRpIjogImRpZDppb246RWlCUXN4dlQxdHowQ3o3S0VmRnVKaEp0XzEzNGRfc3VKbHdaM1NfYlhWbm9CQS04NmVhZjVkYi1kZmEzLTQ1OT
ctOTBiZC1mZjdiNDFiNzlhYzIiLA0KICAicmVzcG9uc2VfdHlwZSI6ICJpZF90b2tlbiB2cF90b2tlbiIsDQogICJyZWRpcmVjdF91cmkiOiAiaHR0cDovL2xvY2F
saG9zdDo1MTExL3JlZGlyZWN0IiwNCiAgIm5vbmNlIjogIjlkYWU3M2FjLTNkNzUtNDk1MC04MmQ5LTJjZWI1YjdiZDU4ZiIsDQogICJzY29wZSI6ICJvcGVuaWQi
LA0KICAic3RhdGUiOiAiZjgxMzdkNTgtOWNjOS00NmUzLWIzZmYtNzQ0ZGUyYzRlYzdkIiwNCiAgInByZXNlbnRhdGlvbl9kZWZpbml0aW9uIjogew0KICAgICJpZ
CI6ICJhNzZmMTJkNy1mNzNkLTRhZDYtYTk5Mi1jOTk5OTgwYzg3YjEiLA0KICAgICJpbnB1dF9kZXNjcmlwdG9ycyI6IFsNCiAgICAgIHsNCiAgICAgICAgImlkIj
ogIjNlNWY0MmE1LTIxNGItNDEwZC1iZWFkLWU0YTIwZGQwMjRkZiIsDQogICAgICAgICJmb3JtYXQiOiB7DQogICAgICAgICAgImp3dF92cCI6IHsNCiAgICAgICA
gICAgICJhbGciOiBbDQogICAgICAgICAgICAgICJFUzI1NksiDQogICAgICAgICAgICBdDQogICAgICAgICAgfQ0KICAgICAgICB9LA0KICAgICAgICAiY29uc3Ry
YWludHMiOiB7DQogICAgICAgICAgImZpZWxkcyI6IFsNCiAgICAgICAgICAgIHsNCiAgICAgICAgICAgICAgInBhdGgiOiBbDQogICAgICAgICAgICAgICAgIiQud
HlwZSINCiAgICAgICAgICAgICAgXSwNCiAgICAgICAgICAgICAgImZpbHRlciI6IHsNCiAgICAgICAgICAgICAgICAidHlwZSI6ICJzdHJpbmciLA0KICAgICAgIC
AgICAgICAgICJwYXR0ZXJuIjogIklEQ2FyZENyZWRlbnRpYWwiDQogICAgICAgICAgICAgIH0NCiAgICAgICAgICAgIH0NCiAgICAgICAgICBdDQogICAgICAgIH0
NCiAgICAgIH0NCiAgICBdDQogIH0NCn0.e1QMfjgf37b7eQv7da-HU8oOlHlgQZRTMG7KMLPNPrvjA0Otx1Ls93Vs3sP4jGdrY4SUyK1RxWZuz1VtnNJOUw
```

Εικόνα 22 Στιγμιότυπο δημιουργίας authorization requst

Το request έχει τις εξής ελάχιστες παραμέτρους

- request_type: έχει την τιμή vp_token, ώστε να ενημερώσει το Wallet ότι θέλει ως απάντηση ένα επαληθεύσιμο διαπιστευτήριο. Επιπλέον το Wallet θα επιστρέψει και ένα id_token.
- client_id : εδώ δηλώνεται το DID του Verifier.
- redirectUrl: το endpoint στο οποιο θα δεχτει ο Verifier την απάντηση από το Wallet.
- state: τυχαία τιμή που χρησιμοποιείται για την αποφυγή επιθέσεων Cross Site Request Forgery (CSRF) attacks
- nonce: χρησιμοποιείται για να συνδεθεί η επαληθεύσιμη παρουσίαση και να αποφευχθεί ένα vp_token injection. Τυχαία τιμή που διαφοροποιείται με κάθε request.

```
//Header
{
  "alg": "ES256K",
  "kid": "did:ion:EiC7SjnjrulJvbFS_DqMUz0v6dkOyPCiclNRT6EhUuL-0Q#key-1",
  "typ": "oauth-authz-req+jwt"
}

//Payload
{
  "client_id": "did:ion:EiC7SjnjrulJvbFS_DqMUz0v6dkOyPCiclNRT6EhUuL-0Q",
```

```
  "client_id_scheme": "did",
  "iss": "did:ion:EiBQsxvT1tz0Cz7KEfFuJhJt_134d_suJlwZ3S_bXVnoBA",
  "nbf": 1701453690,
  "iat": 1701453690,
  "jti": "did:ion:EiC7SjnjrulJvbFS_DqMUz0v6dkOyPCiclNRT6EhUuL-0Q-834ec7b9-
44c8-4fcf-99e9-6ee65b0500f2",
  "response_type": "id_token vp_token",
  "redirect_uri": "http://localhost:5111/redirect",
  "nonce": "56b13733-b364-47d0-ac10-fcc2f63d2dfe",
  "scope": "openid",
  "state": "f01fd319-ba21-4c33-ba87-16abb34f6f06",
  "presentation_definition": {
    "id": "3b4138d0-db7c-4d03-8cec-d8d61f488c4e",
    "input_descriptors": [
      {
        "id": "e2ed3c1a-0fa5-4ef3-8295-d17ed6a8590b",
        "format": {
          "jwt_vp": {
            "alg": [
              "ES256K"
            ]
          }
        },
        "constraints": {
          "fields": [
            {
              "path": [
                "$.type"
              ],
              "filter": {
                "type": "string",
                "pattern": "IDCardCredential"
              }
            }
          ]
        }
      }
    ]
  }
}
```


Εικόνα 23 JWT Token Request Object


```
[07:46:18 INF] Executing RedirectResult, redirecting to https://localhost:5001/authorize?request=eyJhbGciOiJFUzI1NksiLCJraWQi
OiJkaWQ6aW9uOkVpQzdTam5qcnVsSnZiRlNfRHFNVXowdjZka095UENpY2xOUlQ2RWhVdUwtMFEja2V5LTEiLCJ0eXAiOiJvYXV0aC1hdXRoei1yZXErand0In0.e
w0KICAiY2xpZW50X2lkIjogImRpZDppb246RWlDN1NqbmpydWxKdmJGU19EcU1VejB2NmRrT3lQQ2ljbE5SVDZFaFV1TC0wUSIsDQogICJjbGllbnRfaWRfc2NoZW
llIjogImRpZCIsDQogICJpc3MiOiAiZGlkOmlvbjpFaUJRc3h2VDF0ejBDejdLRWZGdUpoSnRfMTM0ZF9zdUpsd1ozU19iWFZub0JBIiwNCiAgIm5iZiI6IDE3MDE
0MTMxNzgsDQogICJpYXQiOiAxNzAxNDEzMTc4LA0KICAianRpIjogImRpZDppb246RWlCUXN4dlQxdHowQ3o3S0VmRnVKaEp0XzEzNGRfc3VkbHdaM1NfYlhWbm9C
QS04NmVhZjVkYi1kZmEzLTQ1OTctOTBiZC1mZjdiNDFiNzlhYzIiLA0KICAicmVzcG9uc2VfdHlwZSI6ICJpZF90b2tlbiB2cF90b2tlbiIsDQogICJyZWRpcmVjd
F91cmkiOiAiaHR0cDovL2xvY2FsaG9zdDo1MTExL3JlZGlyZWN0IiwNCiAgIm5vbmNlIjogIjlkYWU3M2FjLTNkNzUtNDk1MC04MmQ5LTJjZWI1YjdiZDU4ZiIsDQ
obml0aW9uIjogew0KICAgICJpZCI6ICJhNzZmMTJkNy1mNzNkLTRhZDYtYTk5Mi1jOTk5OTgwYzg3YjEiLA0KICAgICJpbnB1dF9kZXNjcmlwdG9ycyI6IFsNCiAg
ICAgIHsNCiAgICAgICAgImlkIjogIjNlNWY0MmE1LTIxNGItNDEwZC1iZWFkLWU0YTIwZGQwMjRkZiIsDQogICAgICAgICJmb3JtYXQiOiB7DQogICAgICAgI
mp3dF92cCI6IHsNCiAgICAgICAgICAgICJhbGciOiBbDQogICAgICAgICAgICAgICJFUzI1NksiDQogICAgICAgICAgICBdDQogICAgICAgICAgfQ0KICAgICAgI
B9LA0KICAgICAgICAiY29uc3RyYWludHMiOiB7DQogICAgICAgICAgImZpZWxkcyI6IFsNCiAgICAgICAgICAgICAgICAgIHsNCiAgICAgICAgICAgICAgICAgICAgInBhdGgiOiBbDQo
gICAgICAgICAgICAgICAgICAgICAgICAgXSwNCiAgICAgICAgICAgICAgICAgICAgICAgImZpbHRlciI6IHsNCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAidHlwZSI6
ICJzdHJpbmciLA0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICBdDQogICAgICAgIH0NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
SUyK1RxWZuz1VtnNJOUw&state=f8137d58-9cc9-46e3-b3ff-744de2c4ec7d&nonce=9dae73ac-3d75-4950-82d9-2ceb5b7bd58f.
```


Εικόνα 24 Authorization request από τον Verifier προς στο Wallet

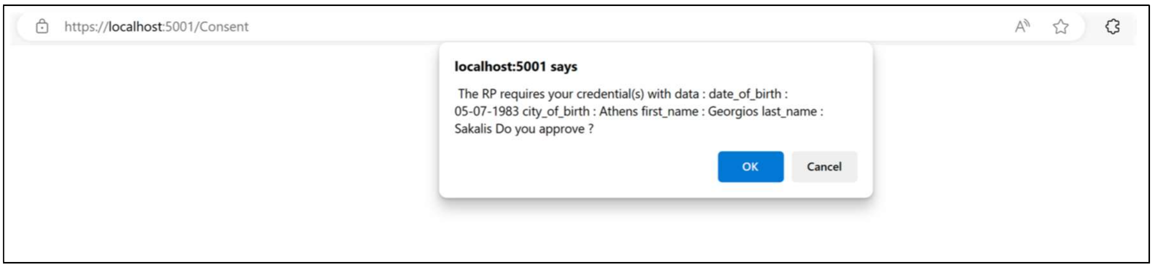
Επιπλέον υπάρχει και η παράμετρος presentation_definition.και περιγράφει τη δομή της επαλιθεύσιμης παρουσίασης καθώς και το επαλιθεύσιμο διαπιστευτήριο που αυτή περιέχει όπως φαίνεται στην παρακάτω εικόνα. Το πεδίο “Format” που περιέχει είναι τύποτυ jwt_vp, το οποίο σημαίνει πως η παροτσίαση θα επιστραφεί ως JSON Web Token και το πεδίο proof θα δημιουργηθεί με αλγόριθμο ES256K. Επίσης ορίζεται ότι το διαπιστευτήριο θα είναι τύπου IDCardCredential

```
{
  "id": "Presentationexample1",
  "input_descriptors": [
    {
      "id": "imputdescriptor",
      "format": {
        "jwt_vp": {
          "alg": [
            "ES256K"
          ]
        }
      },
      "constraints": {
        "fields": [
          {
            "path": [
              "$.type"
            ],
            "filter": {
              "type": "string",
              "pattern": "IDCardCredential"
            }
          }
        ]
      }
    }
  ]
}
```

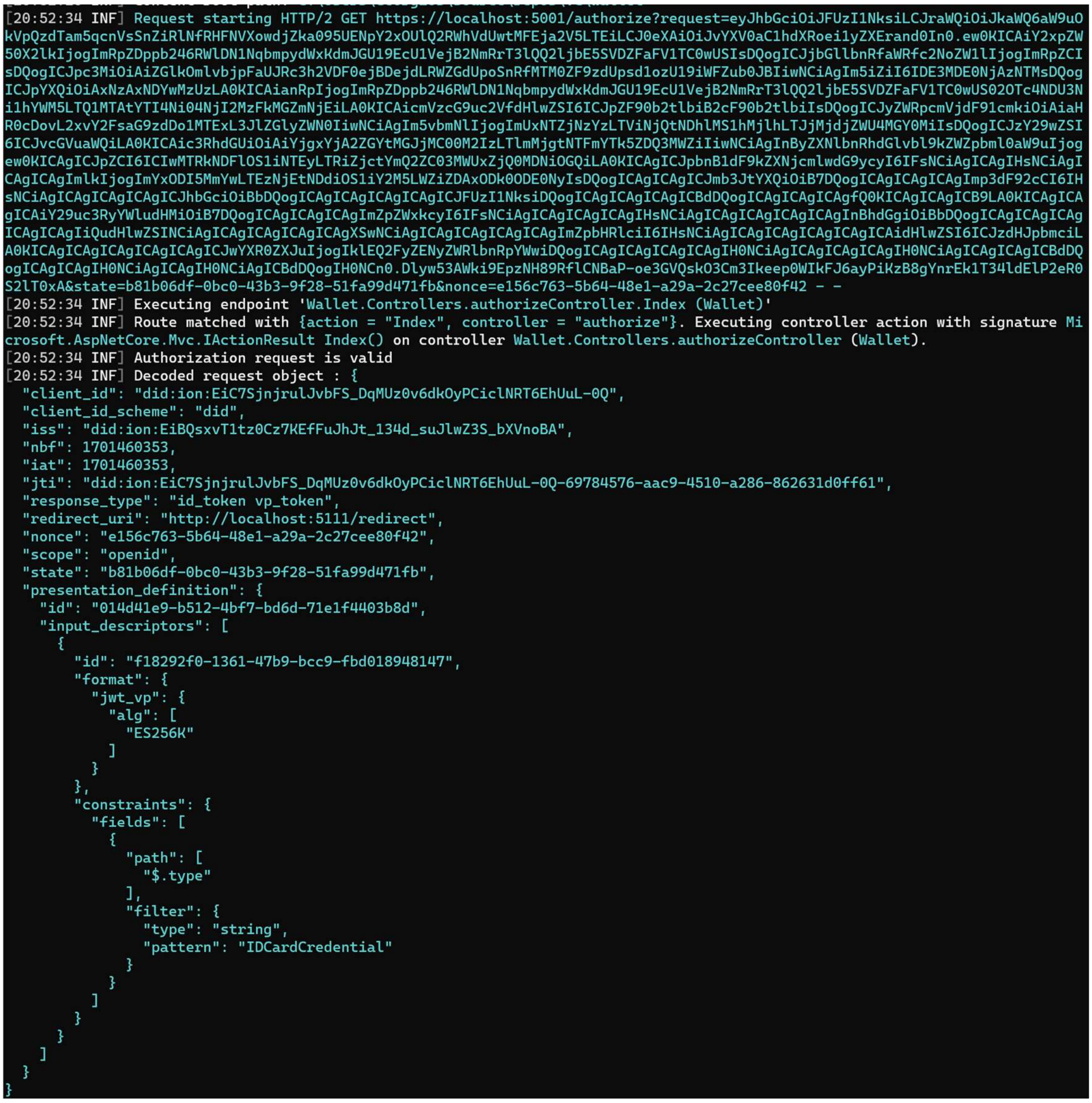
Εικόνα 25 Η τιμή του πεδίου presentation_definition

Το Wallet λαμβάνει το request. Όπως προαναφέραμε, σε ένα κανονικό πορτοφόλι ο χρήστης θα αυθεντικοποιούσε τον εαυτό του πριν προχωρήσει η επεξεργασία. Το επόμενο βήμα είναι ο έλεγχος εγκυρότητας του εισερχόμενου JWT. Όπως και στην περίπτωση εισαγωγής του επαληθεύσιμου διαπιστευτηρίου, εξάγεται το DID του Verifier από το kid και η εφαρμογή προελαύνει το DID αρχείο Verifier. Από εκεί θα λάβει το public key του και με αυτό θα κάνει την επαλήθευσή του JWT όπως φαίνεται και στη Εικόνα 27 Λήψη request στο Wallet.

Εφόσον το JWT είναι έγκυρο, το Wallet εμφανίζει στο χρήστη πληροφορίες για την αίτηση και τον προτρέπει να την εγκρίνει ή όχι.



Εικόνα 26 Έγκριση της αίτησης



```
[20:52:34 INF] Request starting HTTP/2 GET https://localhost:5001/authorize?request=eyJhbGciOiJFUzI1NksiLCJraWQiOiJkaWQ6aW9uO
kVpQzdTam5qcnVsSnZiRlNfRHFNVXowdjZka095UENpY2xOUlQ2RWhVdUwtMFEja2V5LTEiLCJ0eXAiOiJvYXV0aC1hdXRoeilyZXErand0In0.ew0KICAiY2xpZW
50X2lkIjogImRpZDppb246RWlDN1NqbmpydWxKdmJGU19EcU1VejB2NmRrT3lQQ2ljbE5SVDZFaFV1TC0wUSIsDQogICJjbGllbnRfaWRfc2NoZW1lIjogImRpZCI
sDQogICJpc3MiOiAiZGlkOmlvbjpFaUJRc3h2VDF0ejBDejdLRWZGdUpoSnRfMTM0ZF9zdUpsd1ozU19iWFZub0JBIiwNCiAgIm5iZiI6IDE3MDE0NjAzNTMsDQog
ICJpYXQiOiAxNzAxNDYwMzUzLA0KICAianRpIjogImRpZDppb246RWlDN1NqbmpydWxKdmJGU19EcU1VejB2NmRrT3lQQ2ljbE5SVDZFaFV1TC0wUS02OTc4NDU3N
i1hYWM5LTQ1MTAtYTI4Ni04NjI2MzFkMGZmNjEiLA0KICAicmVzcG9uc2VfdHlwZSI6ICJpZF90b2tlbiB2cF90b2tlbiIsDQogICJyZWRpcmVjdF91cmkiOiAiaH
R0cDovL2xvY2FsaG9zdDo1MTExL3JlZGlyZWN0IiwNCiAgIm5vbmNlIjogImUxNTZjNzYzLTViNjQtNDhlMS1hMjlhLTJjMjdjZWU4MGY0MiIsDQogICJzY29wZSI
6ICJvcGVuaWQiLA0KICAic3RhdGUiOiAiYjgxYjA2ZGYtMGJjMC00M2IzLTlmMjgtNTFmYTk5ZDQ3MWZiIiwNCiAgInByZXNlbnRhdGlvbl9kZWZpbml0aW9uIjog
ew0KICAgICJpZCI6ICIwMTRkNDFlOS1iNTEyLTRiZjctYmQ2ZC03MWUxZjQ0MDNiOGQiLA0KICAgICJpbnB1dF9kZXNjcmlwdG9ycyI6IFsNCiAgICAgIHsNCiAgI
CAgICAgImlkIjogImYxODI5MmYwLTEzNjEtNDdiOS1iY2M5LWZiZDAxODk0ODE0NyIsDQogICAgICAgICJmb3JtYXQiOiB7DQogICAgICAgICAgImp3dF92cCI6IH
sNCiAgICAgICAgICAgICJhbGciOiBbDQogICAgICAgICAgICAgICJFUzI1NksiDQogICAgICAgICBdDQogICAgICAgICAgfQ0KICAgICAgICB9LA0KICAgICA
gICAiY29uc3RyYWludHMiOiB7DQogICAgICAgICAgImZpZWxkcyI6IFsNCiAgICAgICAgICAgIHsNCiAgICAgICAgICAgICAgInBhdGgiOiBbDQogICAgICAgICAg
ICAgICAgIiQudHlwZSINCiAgICAgICAgICAgICAgXSwNCiAgICAgICAgICAgICAgImZpbHRlciI6IHsNCiAgICAgICAgICAgICAgICAidHlwZSI6ICJzdHJpbmciL
A0KICAgICAgICAgICAgICAgICJwYXR0ZXJuIjogIklEQ2FyZENyZWRlbnRpYWwiDQogICAgICAgICAgICAgIH0NCiAgICAgICAgICAgIH0NCiAgICAgICAgICBdDQ
ogICAgICAgIH0NCiAgICAgIH0NCiAgICBdDQogIH0NCn0.Dlyw53AWki9EpzNH89RflCNBaP-oe3GVQskO3Cm3Ikeep0WIkFJ6ayPiKzB8gYnrEk1T34ldElP2eR0
S2lT0xA&state=b81b06df-0bc0-43b3-9f28-51fa99d471fb&nonce=e156c763-5b64-48e1-a29a-2c27cee80f42 - -
[20:52:34 INF] Executing endpoint 'Wallet.Controllers.authorizeController.Index (Wallet)'
[20:52:34 INF] Route matched with {action = "Index", controller = "authorize"}. Executing controller action with signature Mi
crosoft.AspNetCore.Mvc.IActionResult Index() on controller Wallet.Controllers.authorizeController (Wallet).
[20:52:34 INF] Authorization request is valid
[20:52:34 INF] Decoded request object : {
  "client_id": "did:ion:EiC7SjnjrulJvbFS_DqMUz0v6dkOyPCiclNRT6EhUuL-0Q",
  "client_id_scheme": "did",
  "iss": "did:ion:EiBQsxvT1tz0Cz7KEfFuJhJt_134d_suJlwZ3S_bXVnoBA",
  "nbf": 1701460353,
  "iat": 1701460353,
  "jti": "did:ion:EiC7SjnjrulJvbFS_DqMUz0v6dkOyPCiclNRT6EhUuL-0Q-69784576-aac9-4510-a286-862631d0ff61",
  "response_type": "id_token vp_token",
  "redirect_uri": "http://localhost:5111/redirect",
  "nonce": "e156c763-5b64-48e1-a29a-2c27cee80f42",
  "scope": "openid",
  "state": "b81b06df-0bc0-43b3-9f28-51fa99d471fb",
  "presentation_definition": {
    "id": "014d41e9-b512-4bf7-bd6d-71e1f4403b8d",
    "input_descriptors": [
      {
        "id": "f18292f0-1361-47b9-bcc9-fbd018948147",
        "format": {
          "jwt_vp": {
            "alg": [
              "ES256K"
            ]
          }
        },
        "constraints": {
          "fields": [
            {
              "path": [
                "$.type"
              ],
              "filter": {
                "type": "string",
                "pattern": "IDCardCredential"
              }
            }
          ]
        }
      }
    ]
  }
}
```


Εικόνα 27 Λήψη request στο Wallet

Αφού ο χρήστης εγκρίνει την αίτηση, το Wallet αρχίζει και ετοιμάζει την παρουσίαση του διαπιστευτηρίου καθώς και το id_token.

```
//header
{
  "alg": "ES256K",
  "kid": "did:ion:EiCVtB8na4G71RjI7zITGJZBzbH7OeJkFY0ft60sD8tkTw#key-1",
                                "typ": "JWT"
}

//payload

{
  "iss": "did:ion:EiCVtB8na4G71RjI7zITGJZBzbH7OeJkFY0ft60sD8tkTw",
  "sub": "did:ion:EiCVtB8na4G71RjI7zITGJZBzbH7OeJkFY0ft60sD8tkTw",
  "nbf": 1701461297,
  "iat": 1701464897,
  "jti": "88fb5d01-35a6-4e12-b510-397079540512",
  "exp": 1701464897
}
```

Εικόνα 28 id_token σε JWT

```
//header
{
  "alg": "ES256K",
  "kid": "did:ion:EiCVtB8na4G71RjI7zITGJZBzbH7OeJkFY0ft60sD8tkTw#key-1",
  "typ": "vp+jwt"
}

//payload

{
  "iss": "did:ion:EiCVtB8na4G71RjI7zITGJZBzbH7OeJkFY0ft60sD8tkTw",
  "nbf": 1701461297,
  "iat": 1701461297,
  "jti": "did:ion:EiCVtB8na4G71RjI7zITGJZBzbH7OeJkFY0ft60sD8tkTw-7d630bfc-
588e-4d90-9ec9-91fd65f4bdb0",
  "vp": {
    "@content": [
      "https://www.w3.org/2018/credentials/v1",
      "https://www.w3.org/2018/credentials/examples/v1"
    ],
    "type": [
      "VerifiablePresentation"
    ],
    "verifiableCredential": [

"eyJhbGciOiJFUzI1NksiLCJraWQiOiJkaWQ6aW9uOkVpQjEzOWZqWllzZmc1T1oycHFvc25leFU3
aXBXNkFqeGlMbFBrT254MlY3ZUEja2V5LTEiLCJ0eXAiOiJ2Yytqd3QifQ.ew0KICAiaXNzIjogIm
RpZDppb246RWlCMTM5ZmpaWXNmZzVPWjJwcW9zbmV4VTdpcFc2QWp4aUxsUGtPbngyVjdlQSIsDQo
gICJuYmYiOiAiMTcwMTI5OTg3OC45ODM1NzA4IiwNCiAgImlhdCI6ICIxNzAxMjk5ODc4Ljk4MzU3
MDgiLA0KICAianRpIjogImRpZDppb246RWlCMTM5ZmpaWXNmZzVPWjJwcW9zbmV4VTdpcFc2QWp4a
UxsUGtPbngyVjdlQTQzNDg3N2VhLTA2YmMtNDVkOS04MmU1LTQ4Y2U5ZTBiNjdjMiIsDQogICJzdW
IiOiAiZGlkOmlvbjpFaUNWdEI4bmE0RzcxUmpJN3pJVEdKWkJ6Ykg3T2VKa0ZZMGZ0NjBzRDh0a1R
```

```
3IiwNCiAgInZjIjogew0KICAgICJAY29udGV4dCI6IFsNCiAgICAgICJodHRwczovL3d3dy53My5v
cmcvMjAxOC9jcmVkZW50aWFscy92MSINCiAgICBdLA0KICAgICJ0eXBlIjogWw0KICAgICAgIlZlc
mlmaWFibGVDcmVkZW50aWFsIiwNCiAgICAgICJJRENhcmRDcmVkZW50aWFsIg0KICAgIF0sDQogIC
AgImNyZWRlbnRpYWxTdWJqZWN0Ijogew0KICAgICAgImZpcnN0X25hbWUiOiAiR2Vvcmdpb3MiLA0
KICAgICAgImxhc3RfbmFtZSI6ICJTYWthbGlzIiwNCiAgICAgICJkYXRlX29mX2JpcnRoIjogIjA1
LTA3LTE5ODMiLA0KICAgICAgImNpdHlfb2ZfYmlydGgiOiAiQXRoZW5zIg0KICAgIH0NCiAgfQ0KI
CB9.R3oLNoLY_eNCUloOQfh-GQBV7n3oA1rmz0R8ZM14_sN-
sAjLNf6G8nFz9VsXRgo1IlC4vvokxx2OYiThjhZFgg"
    ]
  },
  "exp": 1701464897
}
```


Εικόνα 29 Επαληθεύσιμη παρουσίαση σε JWT

Συμπεράσματα

Σκοπός αυτής της εργασίας είναι η μελέτη και αξιολόγηση τεχνικών για την παρουσίαση ταυτοποιητικών δεδομένων που είναι αποθηκευμένα σε ένα ηλεκτρονικό πορτοφόλι. Πιο συγκεκριμένα επιλέκτηκε η εξέταση των προεκτάσεων που προτείνει το OpenID για τις επαληθεύσιμες παρουσιάσεις.

Για το σκοπό αυτό αναπτύχθηκαν τρεις εφαρμογές: Οι δύο διαδικτιακές είναι ο Verifier και το Wallet μεταξύ των οποίων γίνεται η επικοινωνία και η ανταλλαγή των διαπιστευτηρίων και ο Issuer που χρησιμοποιείται βοηθητικά για την έκδοση ενός διαπιστευτηρίου και την αποθήκευσή του στο πορτοφόλι. Υλοποιήθηκε με βάση τα πρότυπα μια νέα ροή βασισμένη στο OpenID και OAuth 2.0, η Verifiable Presentations.

Οι παραπάνω εφαρμογές είναι απλοϊκές και υλοποιήθηκαν για την μελέτη των συγκεκριμένων προτύπων. Για παράδειγμα το Wallet θα έπρεπε να υποστηρίζει ασφάλεια και αυθεντικοποίση καθώς και δεν υποστηρίζεται απευθείας σύνδεση με τον Issuer, ο οποίος είναι και αυτός μια απλοποιημένη εφαρμογή. Από τις επεκτάσεις του OpenID[33] υλοποιήθηκαν μόνο οι απαραίτητες για το βασικό σενάριο. Επιπλέον και η χρήση των επαληθεύσιμων διαπιστευτηρίων έγινε με την απλούστερη μορφή τους, χωρίς να εξεταστεί μια πιο εκλεπτυσμένη μορφή (πχ. Αντί να επιστρέψουμε την ημερομηνία γέννησης να επιστρέψουμε ότι ο χρήστης είναι ενήλικας).

Το OpenID Connect είναι ίσως το ποιο διαδεδομένο πρωτόκολλο αυθεντικοποίησης Η προέκταση του ώστε να υποστηρίζει τα επαληθεύσιμα διαπιστευτήρια αποτελεί μια λογική εξέλιξη, καθώς δεν απαιτεί νέα τεχνογνωσία ή ανάπτυξη νέων πρωτοκόλλων. Κάτω από την ίδια υλοποίηση οι διάφορες διαδικτυακές υπηρεσίες μπορούν να υποστηρίξουν τόσο το νέο τρόπο αυθεντικοποίησης όσο και τον «παραδοσιακό» federated Authentication. Υπάρχουν πολλές υλοποιήσεις και τεχνικές λύσεις, τόσο εμπορικές όσο και ανοικτού λογισμικού, που εύκολα εξελίσσονται. Αλώστε για το σκοπό της εργασίας έγινε επέκταση της βιβλιοθήκης IdentityModel [45].

Ήδη αρκετοί providers που παρέχουν υπηρεσίες αυθεντικοποίησης ως υπηρεσία έχουν αρχίσει σταδιακά να υλοποιούνε μέρος των προδιαγραφών που εξετάστηκαν και να το προσφέρουν ως υπηρεσία [37]. Η Ευρωπαϊκή Επιτροπή μέσω του EUDI Wallet [50] εξετάζει μια σειρά από σενάρια που τα ταυτοποιητικά δεδομένα μπορούν να χρησιμοποιηθούν.

Τα πρότυπα όπως το OpenID for Verifiable Pesentations είναι ακόμα σε εξέλιξη, με αρκετά ακόμα ανοικτά θέματα, Ωστόσο τα δεδομένα από τις πρώτες υλοποιήσεις δίνουν θετικά αποτελέσματα και φαίνεται ότι σύντομα θα αποτελέσει το νέο μοντέλο αυθεντικοποίησης.

Αξιοσημείωτα μέρη υλοποίησης

6.1 DID REGISTRATION

Παρακάτω είναι ο κώδικας που χρησιμοποιήθηκε για την δημιουργία των DID documents και η δημοσίευσή του σε blockchain network και συγκεκριμένα στο ION Network[42] καθώς και τη δημιουργία των public και private key για τον Issuer, το Verifier και το Wallet. Ο κώδικας αυτός βασίζεται στα ION TOOLS [41]. Η μέθοδος δημιουργεί το private και το public key κα τα αποθηκεύσει στα αρχεία privateKey.json και publicKey.json . Επίσης, κατά τη δημιουργία του DID document περικλείει το public key, το οποίο θα χρησιμοποιηθεί από τους ενδιαφερόμενους για να επαληθεύσουν τα JWT. Για τη δημιουργία των κλειδιών χρησιμοποιείται ως αλγόριθμος υπογραφής ο ECDSA με secp256k1 καμπύλη[47].

```
const { anchor, DID, generateKeyPair, sign, verify } =
require('@decentralized-identity/ion-tools')
const fs = require('fs').promises
const main = async () => {
  const authnKeys = await generateKeyPair('secp256k1')
  console.log("Created private/public key pair")
  console.log("Public key:", authnKeys.publicJwk)
  // Write private and public key to files
  await fs.writeFile(
    'publicKey.json',
    JSON.stringify(authnKeys.publicJwk)
  )
  await fs.writeFile(
    'privateKey.json',
    JSON.stringify(authnKeys.privateJwk)
  )
  console.log("Wrote public key to publicKey.json")
  console.log("Wrote private key to privateKey.json")

  const did = new DID({
    content: {
      // Register the public key for authentication
      publicKeys: [
        {
          id: 'auth-key',
          type: 'EcdsaSecp256k1VerificationKey2019',
          publicKeyJwk: authnKeys.publicJwk,
          purposes: ['authentication']
        }
      ],
      // Register an IdentityHub as a service
      services: [
        {
          id: 'domain-1',
          type: 'LinkedDomains',
          serviceEndpoint: 'https://example.com:3008'
        }
      ]
```

```
    }
  });
  console.log("DID Document 1", JSON.stringify(did));
  const didUri = await did.getURI('short');
  console.log("Generated DID:", didUri)
  console.log("DID Document", JSON.stringify(did));
  const anchorRequestBody = await did.generateRequest(0)
  console.log("Anchor Request", JSON.stringify(anchorRequestBody));
  const anchorResponse = await anchor(anchorRequestBody);
  //      const anchorRequest = new ION. anchorRequest(anchorRequestBody)
  //      const anchorResponse = await anchorRequest.submit()
  console.log(JSON.stringify(anchorResponse))


}
main()

// Create private/public key pair
```

Εικόνα 30 Κώδικας δημιουργίας DID document

6.2 JWT

Για τη υπογραφή και την επαλήθευση των JWT, χρησιμοποιήθηκαν οι Microsoft.IdentityModel JWT βιβλιοθήκες που συνοδεύουν το .net core. Ωστόσο ο ES256K αλγόριθμος που χρησιμοποιείται για την υπογραφή και την επαλήθευση των JWT δεν υποστηρίζεται από τις συγκεκριμένες βιβλιοθήκες. Για αυτό το λόγο δημιουργήσαμε την υποδομή [48] ώστε να χρησιμοποιηθούν βιβλιοθήκες τρίτων που υλοποιούν τον αλγόριθμο

```
public string CreateToken(string payload, string kid, string typ)
{

   var privateKeyString =
System.IO.File.ReadAllText("ExternalFiles\\privateKey.json");
   dynamic privateKey = JObject.Parse(privateKeyString);

   var curve = ECNamedCurveTable.GetByName(privateKey.crv.ToString());

   var bouncyCastleEcdsaSecurityKey =
GetBouncyCastleEcdsaSecurityKey(privateKey,  kid);

   var signingCredentials = new
SigningCredentials(bouncyCastleEcdsaSecurityKey, "ES256K");

   var handler = new JsonWebTokenHandler();

   var headerClaims = new Dictionary<string, object>();
   headerClaims.Add("typ", typ);

   var token = handler.CreateToken(payload, signingCredentials, headerClaims
);
```

```
   return token;
}
```

Εικόνα 31 Δημιουργία JWT Token

```
public TokenValidationResult ValidateToken(string token)
{
    TokenValidationResult result;
    var handler = new JsonWebTokenHandler();
    try
    {
        var jsonToken = new JsonWebToken(token);

        var kid = jsonToken.Kid;
        var issuer = jsonToken.Issuer;


        var publicKey = GetPublicKeyFromDIDDocument(kid);
        IdentityModelEventSource.ShowPII = true;
        var validationResult = handler.ValidateToken(token, new
TokenValidationParameters()
        {
            IssuerSigningKey = GetBouncyCastleEcdsaSecurityKey (publicKey,
kid),
            ValidateIssuerSigningKey = false,
                                ValidateLifetime =false,
                                ValidateIssuer=true,
                                ValidIssuer=issuer,
                                ValidateAudience= false,

        });; ;
        result = validationResult;
    }
    catch (Exception ex)
    {

        result = new TokenValidationResult()
        {
            IsValid = false,
            Exception = ex

        };
    }
    return result;
}
```

Εικόνα 32 Επαλήθευση JWT

```
private BouncyCastleEcdsaSecurityKey GetBouncyCastleEcdsaSecurityKey(dynamic
cryptographicKey, string kid)
{
    BouncyCastleEcdsaSecurityKey bouncyCastleEcdsaSecurityKey;
    var curve = ECNamedCurveTable.GetByName(cryptographicKey.crv.ToString());
```

```
    var domainParamenters = new ECDomainParameters(curve.Curve, curve.G,
curve.N, curve.H, curve.GetSeed());

    if (cryptographicKey.d!=null)
    {
        var d = Base64UrlEncoder.DecodeBytes(cryptographicKey.d.ToString());
        var ecPrivateKeyParameters = new ECPrivateKeyParameters(new
BigInteger(1, d), domainParamenters);
        bouncyCastleEcdsaSecurityKey = new
BouncyCastleEcdsaSecurityKey(ecPrivateKeyParameters);
    }
    else
    {
        var x = Base64UrlEncoder.DecodeBytes(cryptographicKey.x.ToString());
        var y = Base64UrlEncoder.DecodeBytes(cryptographicKey.y.ToString());
        var point = curve.Curve.CreatePoint(
            new BigInteger(1, x),
            new BigInteger(1, y));
        var ecPpubicKeyParameters = new ECPublicKeyParameters(point,
domainParamenters);
        bouncyCastleEcdsaSecurityKey = new
BouncyCastleEcdsaSecurityKey(ecPpubicKeyParameters);
    }

    bouncyCastleEcdsaSecurityKey.KeyId= kid;
    return bouncyCastleEcdsaSecurityKey;

}
```

Εικόνα 33 Βοηθητική συνάρτηση

```
public class BouncyCastleEcdsaSecurityKey : AsymmetricSecurityKey
 {
     public BouncyCastleEcdsaSecurityKey(ECKeyParameters keyParameters)
     {
         KeyParameters = keyParameters;
         CryptoProviderFactory.CustomCryptoProvider = new
CustomCryptoProvider();
     }

     public ECKeyParameters KeyParameters { get; }
     public override int KeySize => throw new NotImplementedException();

     [Obsolete("HasPrivateKey method is deprecated, please use
PrivateKeyStatus.")]
     public override bool HasPrivateKey => KeyParameters.IsPrivate;

     public override PrivateKeyStatus PrivateKeyStatus =>
KeyParameters.IsPrivate ? PrivateKeyStatus.Exists :
PrivateKeyStatus.DoesNotExist;
 }
```

Εικόνα 34 Κλάση BouncyCastleEcdsaSecurityKey

```
public class CustomCryptoProvider : ICryptoProvider
```

```
{
    public bool IsSupportedAlgorithm(string algorithm, params object[] args)
=> algorithm == "ES256K";

    public object Create(string algorithm, params object[] args)
    {
        if (algorithm == "ES256K"
            && args[0] is BouncyCastleEcdsaSecurityKey key)
        {
            return new CustomSignatureProvider(key, algorithm);
        }

        throw new NotSupportedException();
    }

    public void Release(object cryptoInstance)
    {
        if (cryptoInstance is IDisposable disposableObject)
            disposableObject.Dispose();
    }
}
```

Εικόνα 35 Κλάση CustomCryptoProvider

```
public class CustomSignatureProvider : SignatureProvider
{
    public CustomSignatureProvider(BouncyCastleEcdsaSecurityKey key, string
algorithm)
        : base(key, algorithm) { }

    protected override void Dispose(bool disposing) { }

    public override byte[] Sign(byte[] input)
    {
        var ecDsaSigner = new ECDsaSigner();
        BouncyCastleEcdsaSecurityKey key = Key as
BouncyCastleEcdsaSecurityKey;

        ecDsaSigner.Init(true, key.KeyParameters);

        byte[] hashedInput;
        using (var hasher = SHA256.Create())
        {
            hashedInput = hasher.ComputeHash(input);
        }

        var output = ecDsaSigner.GenerateSignature(hashedInput);

        var r = output[0].ToByteArrayUnsigned();
        var s = output[1].ToByteArrayUnsigned();

        var signature = new byte[r.Length + s.Length];
        r.CopyTo(signature, 0);
        s.CopyTo(signature, r.Length);
```

```
        return signature;
    }

    public override bool Verify(byte[] input, byte[] signature)
    {
        var ecDsaSigner = new ECDsaSigner();
        BouncyCastleEcdsaSecurityKey key = Key as
BouncyCastleEcdsaSecurityKey;

        ecDsaSigner.Init(false, key.KeyParameters);

        byte[] hashedInput;
        using (var hasher = SHA256.Create())
        {
            hashedInput = hasher.ComputeHash(input);
        }

        var r = new BigInteger(1, signature.Take(32).ToArray());
        var s = new BigInteger(1, signature.Skip(32).ToArray());

        return ecDsaSigner.VerifySignature(hashedInput, r, s);
    }
}
```

Εικόνα 36 Κλάση CustomSignatureProvider

6.3 IDENTITY MODEL

Το IdentityModel [48] είναι μια συλλογή από βιβλιοθήκες για .net core και Asp.net core που χειρίζονται αρκετά θέματα για επικοινωνία με τα πρωτόκολλα OpenID Connect και OAuth 2.0. Οι βιβλιοθήκες αυτές χρησιμοποιήθηκαν ή επεκτάθηκαν για την εύκολη υλοποίηση της εργασίας.

Βιβλιογραφία

1. Eleni Papadopoulou, Evangelos Sakkopoulos: Common Public Service Vocabulary for Semantic Interoperability in Greek Voucher Services. IISA 2022: 1-8
2. George Kastanas, Evangelos Sakkopoulos: Mobile student information services: A case study in Greek Open University. IISA 2022: 1-4
3. Rafael Torres Moreno, Jorge Bernal Bernabé, Jesús García Rodríguez, Tore Kasper Frederiksen, Michael Stausholm, Noelia Martínez, Evangelos Sakkopoulos, Nuno Ponte, Antonio F. Skarmeta: The OLYMPUS Architecture - Oblivious Identity Management for Private User-Friendly Services. Sensors 20(3): 945 (2020)
4. Emmanouil Viennas, Zafeiria-Marina Ioannou, Georgios Pavlidis, Giannis Tzimas, Evangelos Sakkopoulos: HappyCruise: An architecture for Personalized Secure Boarding on Cruises. IISA 2020: 1-8
5. Stavros Piotopoulos, Evangelos Sakkopoulos: Smart eGov Services for Citizenship: Improving Personalized Services. IISA 2020: 1-8
6. Evangelos Sakkopoulos, Zafeiria-Marina Ioannou, Emmanouil Viennas: Mobile Personal Information Exchange Over BLE. IISA 2018: 1-8
7. Evangelos Sakkopoulos, Emmanouil Viennas, Mersini Paschou, Zafeiria-Marina Ioannou, Vassiliki Gkantouna, Efrosini Sourla, Giannis Tzimas, Spyros Sioutas, Athanasios K. Tsakalidis: Mobile Data Fusion from Multiple Tracking Sensors to Augment Maritime Safety: Mobile Detection, Early Identification, and Tracking of Moving Objects. IEEE MS 2015: 112-119
8. Mersini Paschou, Christos Papadimitiriou, Nikolaos Nodarakis, Konstantinos Korezelidis, Evangelos Sakkopoulos, Athanasios K. Tsakalidis: Enhanced healthcare personnel rostering solution using mobile technologies. J. Syst. Softw. 100: 44-53 (2015)
9. Web service discovery mechanisms: Looking for a needle in a haystack. J Garofalakis, Y Panagis, E Sakkopoulos, A Tsakalidis - International Workshop on Web Engineering, 2004
10. easyHealthApps: e-Health Apps dynamic generation for smartphones & tablets. M Paschou, E Sakkopoulos, A Tsakalidis Journal of medical systems 37, 1-12
11. Adaptive mobile web services facilitate communication and learning Internet technologies. E Sakkopoulos, M Lytras, A Tsakalidis, IEEE Transactions on Education 49 (2), 208-215
12. World Wide Web Consortium (W3C). Verifiable Credentials Data Model v1.1. 2022. url: https://www.w3.org/TR/vc-data-model/
13. Verifiable Credentials Use Cases (w3.org)
14. I. E. T. Force. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 8259. December. 2014. url: https://www.rfc-editor.org/rfc/rfc8259
15. World Wide Web Consortium (W3C). JSON-LD 1.1, A JSON-based Serialization for Linked Data. 2020. url: https://www.w3.org/TR/json-ld11
16. Linked JSON Community Group, JSON-LD – JSON for Linking Data, https://json-ld.org
17. Credentials Community Group. W3C Editor's Draft. Verifiable Credentials Implementation Guidelines 1.0. url: https://w3c.github.io/vc-imp-guide/
18. World Wide Web Consortium (W3C). W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes. 2012. url: https://www.w3.org/TR/xmlschema11- 2/#dateTime
19. I. E. T. Force. JSON Web Token (JWT). RFC 7519. Microsoft, Ping Identity, NRI, May 2015, pp. 1–29. url: https://www.rfc-editor.org/rfc/rfc7519.

20. I. E. T. Force. JSON Web Signature (JWS). RFC 7515. Microsoft, Ping Identity, NRI, May 2015, pp. 1–59. url: https://www.rfc-editor.org/rfc/rfc7515
21. World Wide Web Consortium (W3C). Verifiable Credential Data Integrity 1.0 - Securing the Integrity of Verifiable Credential Data. 2022. url: https ://w3c.github.io/vc-data-integrity/
22. J. Camenisch and A. Lysyanskaya. “A signature scheme with efficient protocols”. In:International Conference on Security in Communication Networks. Springer. 2002,pp. 268–289
23. World Wide Web Consortium (W3C). Decentralized Identifiers (DIDs) v1.0. 2022. url: https://www.w3.org/TR/did-core/
24. World Wide Web Consortium (W3C) Use Cases and Requirements for Decentralized Identifiers. 17 March 2021.url: https://www.w3.org/TR/did-use-cases/
25. I. E. T. Force. Uniform Resource Identifier (URI): Generic Syntax. January 2005. Internet Standard. url: https://www.rfc-editor.org/rfc/rfc3986
26. World Wide Web Consortium (W3C). DID Specification Registries. 2023. url: https://www.w3.org/TR/did-spec-registries/
27. World Wide Web Consortium (W3C). The did:key Method v0.7. 2022. url: https://w3c-ccg.github.io/did-method-key
28. W3C Credentials Community Group. BTCR DID Method, url: https://w3c-ccg.github.io/didm-btcr/
29. Decentralized Identity Foundation, Veramo core team. ETHR DID Method Specification. url: https://github.com/decentralized-identity/ethr-did-resolver/blob/master/doc/did-method-spec.md
30. W3C Credentials Community Group. did:web Decentralized Identifiers Method Spec. ur: https://github.com/w3c-ccg/did-method-web
31. I. E. T. Force. The OAuth 2.0 Authorization Framework. url: https://www.rfc-editor.org/rfc/rfc6749
32. F. N. Sakimura, NRI, J. Bradley, P. Identity, M. Jones, Microsoft, B. de Medeiros, Google, C. Mortimore, and Salesforce. OpenID Connect Core 1.0 incorporating errata set 1. 2014. url: https://openid.net/specs/openid-connect-core-1_0.html
33. T. Lodderstedt, K. Yasuda, T. Looker. OpenID for Verifiable Credential Issuance. 2022. url: https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html
34. O. Terbu, T. Lodderstedt, K. Yasuda, A. Lemmon, T. Looker. OpenID for Verifiable Presentations. 2022. url https://openid.net/specs/openid-4-verifiable-presentations-1_0.html
35. K. Yasuda, M. Jones, T. Lodderstedt. Self-Issued OpenID Provider v2. 2023. url: https://openid.net/specs/openid-connect-self-issued-v2-1_0.html
36. M. S. Ferdous, F. Chowdhury, and M. O. Alassafi. “In Search of Self-Sovereign Identity Leveraging Blockchain Technology”. In: IEEE Access 7 (2019), pp. 103059– 103079. url : https://ieeexplore.ieee.org/document/8776589
37. M. Security. Microsoft Entra Verified ID. 2023. url: https://www.microsoft.com/ en-us/security/business/identity-access/microsoft-entra-verified-id
38. M. Security. Decentralized identity and verifiable credentials. 2023. url: https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE5cxkr?culture=en-us&country=us
39. European Commission. European Blockchain Services Infrastructure (EBSI) Blockchain. 2023. url: https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/ Home
40. European Commission. eIDAS Expert Group. European Digital Identity Architecture and Reference Framework. 2023. url: https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-wallet-architecture-and-reference-framework

41. Decentralized Identity Foundation. ION Tools. 2023. url: https://github.com/decentralized-identity/ion-tools
42. Decentralized Identity Foundation. ION Tools. 2023. url: https://identity.foundation/ion/
43. World Wide Web Consortium (W3C). Securing Verifiable Credentials using JSON Web Tokens. 2023. url: https://w3c.github.io/vc-jwt/
44. Decentralized Identity Foundation. DIF Universal Resolver. 2023. url: https://dev.uniresolver.io/
45. Duende Software. IndentityModel. url : https://github.com/IdentityModel
46. Scott Brady. Supporting Custom JWT Signing Algorithms (ES256K) in .NET Core. url : https://www.scottbrady91.com/c-sharp/supporting-custom-jwt-signing-algorithms-in-dotnet-core
47. National Institute of Standards and Technology (NIST). Federal Information Processing Standards Publication 186-5, Digital Signature Standard (DSS). url : https://csrc.nist.gov/pubs/fips/186-5/final
48. Haber, Stuart; Stornetta, W. Scott (January 1991). "How to time-stamp a digital document". Journal of Cryptology. 3 (2): 99–111. doi:10.1007/bf00196791.
49. Nakamoto, Satoshi (24 May 2009). url: Bitcoin: A Peer-to-Peer Electronic Cash System
50. EU Digital Identity Wallet Pilot implementation. 2023. url: EU Digital Identity Wallet Pilot implementation | Shaping Europe's digital future (europa.eu)