BDI: embedded JWT's as Representation Evidence

Investigation

basic data infrastructure

Topsector Logistiek

Colophon

BDI: embedded JWT's as Representation Evidence
Investigation

Author
Huub van Thienen, CGI

May 2024

Management Summary

The BDI framework includes support for the authentication of a representative (human or machine) and verification of their mandate/representation, and (optionally) of their professional qualifications on behalf of their principal. This safeguards the non-repudiation of the liability their principal takes for their actions.

In case of humans, the assumption is that a human presents an ID and Representation Evidence.

The ID can be standard, or fitted with additional safeguards such as biometrics. The Representation Evidence should be able to show:

- The chain of subcontracting, up to a level that is suitable for security reasons;
  - It may be necessary to stop at an intermediate level, to protect the identity of the main principal on top of the chain from leaking.
- The confirmation of the identity of the human, as a representative.
- The professional qualifications of the human.
- Time limitations on the validity of the representation (not before, not after).
- Links to issuers of subcontracting orders, to validate real-time is the representation is still valid and not revoked.
- The non-repudiable evidence of the representation.

For real-life applications it is necessary to be able to operate (temporarily) offline: the check of the Representation Evidence without real-time validation at the issuers should be possible.

JSON Web Token (JWT, RFC 7519) is a well-known, compact, URL-safe means of representing claim sets to be transferred between two parties. As such it is a candidate for the Representation Evidence.

The specifications allow for embedding ("nesting") JWTs, mimicking the subcontracting chain.

There are two types of JWT claims:

- **Registered:**
  standard claims registered with the Internet Assigned Numbers Authority (IANA) and defined by the JWT specification to ensure interoperability with third-party, or external, applications.
- **Custom:**
  consists of non-registered public or private claims. Public claims are collision-resistant while private claims are subject to possible collisions.

Verifiable Credentials https://www.w3.org/TR/vc-data-model-2.0/ (Claim Name "vc") and Verifiable Presentations (Claim Name "vp") are registered claims, therefore nested JWTs support the future use of VCs and VP's.

A use case has been tested with a code example, showing the viability of using JWTs for this purpose. The M2M variant is simpler but equivalent. Extending the claim set creates even more possibilities, for instance for extra IT security or for eFTI/eCMR purposes.

Contents

1 BDI: Representation Register

The BDI framework includes support for the authentication of a representative (human or machine) and verification of their mandate, safeguarding the non-repudiation of the liability their principal (the entity who is represented) takes for their actions.

In the physical operation of our economy, this question of authentication of a representative and verification of their mandate is much more widespread and not limited to employees or contractors. The same applies to sub-contractors that perform business functions on someone else's premises. For example: a maintenance sub-contractor for a vendor of video security systems is commissioned to perform maintenance at a customer's site. The subcontractor is contracted to perform regular maintenance. At regular intervals a maintenance engineer shows up at the gate of the premises and claims access to the premises, to perform preventive maintenance on a security video system on behalf of the vendor that delivered the security system.

The security guards of the company where the security system is installed needs to verify: has he/she indeed been sent by the OEM? And can he/she indeed be authenticated and verified as being mandated by the sub-contractor, and does he/she have the required professional qualifications? And can that mandate be verified in a non-repudiable manner?

The BDI framework defines a Representation Register and Professional Qualification for this type use case. The Registers ae under the control of a Data Owner/Data Service Provider. These registers are accessible via a published endpoint. Authenticated third parties can verify:

- the representation mandate of authenticated natural persons acting of behalf of the data owner;
- the representation mandate of organizations (sub-contractors) acting on behalf of the data owner;
- the professional qualifications of the natural person;
- the standardized roles the data owner's organization supports.

https://bdinetwork.org/framework/representation/

2 Representation Evidence

2.1 Requirements

In the example of the engineer of the maintenance sub-contractor, the user journey starts when the engineer shows up to the gate. The security guard needs to be able to authenticate the person and verify the representation claim.

The approach is that the engineer presents an ID (authentication) and a Representation Evidence.

The ID can be standard, or fitted with additional safeguards such as biometrics.

The Representation Evidence should be able to show:

- The chain of subcontracting, up to a level that is suitable for security reasons;
    - it may be necessary to stop at an intermediate level, to protect the identity of the main principal on top of the chain from leaking.
- The confirmation of the identity of the engineer, as a representative.
- Time limitations on the validity of the representation (not before, not after).
- Links to issuers of subcontracting orders, to validate real-time is the representation is still valid and not revoked.
- The non-repudiable evidence of the representation.

For real-life applications it is necessary to be able to operate (temporarily) offline: the check of the Representation Evidence with delayed validation at the issuers should be possible. If a delayed verification is acceptable is up to the parties involved: in many cases this might be acceptable, for instance if a pre-notification is done to the party that is checking the Representation Evidence. The pre-notification data can be matched with the data in de Evidence.

An equivalent (but less demanding) set of requirements can de defined for M2M interactions.

2.2 Use of Delegation Evidence (iSHARE) as Representation Evidence (BDI)?

The iSHARE specifications define the concept of Delegation Evidence in the form of a Jason Web Token (JWT). JWT's are ubiquitous in their use, with a lot of support available.

The Delegation Evidence is the mechanism to support subcontracting: a contractor can delegate the authorization to have access to data of the Data Owner to a chosen sub-contractor, several levels deep. This is a vital mechanism in the business arena.

At first glance this appears to be usable for Representation as well: however, this is a quite different concept and use case.

The Delegation Evidence is used when a Data User delegates the authorization to access data of the Data Owner to a third party (subcontractor). The subcontractor uses the Delegation Evidence when trying to access data of the Data Owner to prove the delegation. The subcontractor can delegate the authorization rights to a next party, creating a more complex JWT as Delegation Evidence.

The Data Owner maintains a hierarchical list of delegation's (in the form of a tree of EORI's). The list is used to evaluate the delegation claims in the JWT in the correct order, which prevents that mistakes in sub-delegation authorizations (more rights) propagate.

The iSHARE Delegation Evidence is focused upon access of data "at the source", which is quite different from the question of Representation by a representative. As such it is not advisable to try to combine the two purposes.

The idea to use a (nested/embedded) JWT as carrier of evidence has however inspired this investigation: is it possible to use nested/embedded JWT's as Representation Evidence?

3 Introduction JWT

JSON Web Token (JWT, RFC 7519) is a compact, URL-safe means of representing *claim sets* to be transferred between two parties. A claim set is just a set of *claims*, where each claim is a piece of information asserted about a subject. A claim is represented as a name/value pair consisting of a claim name (always a string) and a claim value (can be any JSON value).

The use of JWT is ubiquitous: the support in tools and knowhow is well developed.

Unsecured JWTs are just standardized claim sets. However, many if not most JWTs need to be signed and encrypted. The claim set in a JWT can be used as the payload of a JSON Web Signature (JWS, RFC 7515) structure or as the plaintext of a JSON Web Encryption (JWE, RFC 7516) structure, enabling the claims to be digitally signed or integrity protected with a Message Authentication Code (MAC) and/or encrypted.

The result nesting of a JWT in a JWS or in a JWE (or both!) is called a *Nested JWT*. (Note that this is *a different kind of nesting than the nesting/embedding* of representation evidences that are the topic of this note). The resulting JSON object is encoded using a Base64URL encoding, which produces a URL-safe plain ASCII string.

Because JWTs are in the end just ASCII strings, they can be used as the value of a claim in another JWT. We will designate this kind of nesting as embedded JWTs. This feature is the basis for representation hierarchies, where a representation is based on another representation.

A draft JWT specification https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/ defines how to implement selective disclosure of information. Selective disclosure would be a excellent feature in business environments.

4 JWT’s support Verifiable Credentials and Verifiable Presentations

There are two types of JWT claims:

- **Registered:**
  standard claims registered with the Internet Assigned Numbers Authority (IANA) and defined by the JWT specification to ensure interoperability with third-party, or external, applications.
- **Custom:**
  consists of non-registered public or private claims. Public claims are collision-resistant while private claims are subject to possible collisions

The IANA "JSON Web Token Claims" registry https://www.iana.org/assignments/jwt/jwt.xhtml#claims lists the registered claims.

Verifiable Credentials https://www.w3.org/TR/vc-data-model-2.0/ (Claim Name “vc”) and Verifiable Presentations (Claim Name “vp”) are registered claims.

The use of nested JWTs supports the future use of VC’s and VP’s.

The JWT structure makes it therefore possible to include claims of Professional Qualifications.

5 Use case scenario

We outline a basic use case which the use of embedded JWT's is required:

1 Acne Inc. buys a batch of hydrofluoric acid from its supplier Lets-B-chemical.

2 They agree that Acne will send a truck and driver to a Lets-B-chemical location to pick it up.

3 Lets-B-chemical will only deliver the order if the driver can prove:
a She works on behalf of Acne
b She is qualified to handle the dangerous materials

4 Unfortunately, Acne does not have a truck nor a qualified driver. Therefore, Acne commissions Van Gend & Loos to pick up the batch.

5 To comply with their environmental objectives, Van Gend & Loos is part of a network of transportation firms to reduce the number of trips.

6 Through this network, they learn that De Snelle Visser has already planned a trip to Lets-B-Chemical. Van Gend & Loos decides to share the ride.

7 The parties draw up a contract between them.

8 Dolly Driver is an employee of De Snelle Visser.

9 Dolly Driver gets the order to pick up the bath and transport it to the next destination. She receives proof of representation for Snelle Visser and order information.

10 Arriving at Lets-B-Chemical Dolly identifies herself and proves that she is indeed entitled to transport the chemicals on behalf of Acne Inc.

6 Using embedded JWT's as Representation Evidence

1 Acne Inc. generates a JWT for Van Gend & Loos so the latter can prove its representation on behalf of the former, for a specific order: the Acne order to Lets-B reference, and the transport order/ freightbill (including link for eFTI) to van Gend & Loos are included. This JWT is signed by Acne to prove both its integrity and its authenticity.

2 Van Gend & Loos generates a JWT for De Snelle Visser. This JWT contains the transport order/ freightbill reference between Van Gend & Loos en De Snelle Visser, as well as the embedded JWT from Acne.

3 After accepting the order from Van Gend & Loos, and receiving their JWT, De Snelle Visser generates another JWT with their internal order number, their freightbill information, embedding the JWTs under 2 and 3. The JWT includes information about Dolly Driver, asserting her ID, her employment (and her Professional Qualifications) . This JWT is signed by De Snelle Visser.

4 As she arrives at LetsB-chemical Dolly offers the security guard both her JWT (in the simplest form a QR code) and her driver's license. The security guard verifies her ID, and evaluates the JWT's, for example with a software tool that reads the QR code en unpacks the JWT's.
   - The ID matches with the data in the JWT, De Snelle Visser confirms her representation and qualifications.
   - The Van Gend & Loos JWT confirms the representation of De Snelle Visser on their behalf.
   - The Acne JWT confirms the representation of Van Gend & Loos on their behalf, including the order reference of the customer.
   - The guard now has the proof that Dolly Driver is the right representative to pick up the batch. The JWT is stored.

The cryptographic signing of each (embedded) JWT creates proof of representation and the liability assumed by the principal.
Appendix 4 lists a simple code example, written in Python.

Both Python and Java have packages/libraries for JWT management:
- **Python:** pyjwt
- **Java:** JJWT

In a more advanced implementation, links to Representation Registers of each company are included in the JWT and used to verify real-time the validity.

Note that the embedded JWT, which represents a claim (qualification) can be checked on it's validity, but that in particular for professional qualifications this is 'only' a digital copy of the actual claim. The digital twin would need to be available under a governed process. In VC world this is intended to be the issuer (Trusted Issuer). For BDI this should be the same: Or the Issuing party is the certification body itself (ideal scenario) and a BDI participant (role: Trusted Issuer) or a Service Provider (today scenario) does the checks-and-balances and digitized the claim for usage by the subject.

This is similar to who Secure Logistics (Rotterdam) and Smartlox (Schiphol) now 'load' the Identities with the verified attributes on the respective secure Cargo Cards.

7 Using the JWT for eFTI purposes

The same JWT could also be used by authorities that need to inspect (roadside or remote) the vehicle and cargo.

Unpacking the JWT will show the reference and the link to the eFTI-pplatform of the transport order, allowing authorities to inspect the details of the transport at the source (after authentication).

This requires more investigation.

Appendix 1

BDI Convention JWT Default Registered Claim Names

The following Claim Names are defined in the RFC https://datatracker.ietf.org/doc/html/rfc7519 and registered in the IANA "JSON Web Token Claims" registry. The BDI convention describes how to use these claims.

"iss" (Issuer) Claim
The "iss" (issuer) claim identifies the principal that issued the JWT. The processing of this claim is generally application specific. The "iss" value is a case-sensitive string containing a StringOrURI value. Use of this claim is OPTIONAL.
The BDI convention is to identify the issuer by the URI of the bdi subdomain of the issuer. The following examples are valid:

```
https://_bdi.acme.com
https://_bdi.connekt.nl
```

The _bdi subdomain has registered endpoint names. One of the registered endpoints is designed to verify the validity of the JWT. The following examples are valid:

```
https://_bdi.acme.com/repr
https://_bdi.connekt.nl/repr
```

This endpoint will generate a "valid" (not revoked) or no response if the "jti" is offered for a validity check.

"sub" (Subject) Claim[1]
The "sub" (subject) claim identifies the principal that is the subject of the JWT. The claims in a JWT are normally statements about the subject. The subject value MUST either be scoped to be locally unique in the context of the issuer or be globally unique. The processing of this claim is generally application specific. The "sub" value is a case-sensitive string containing a StringOrURI value. Use of this claim is OPTIONAL. The BDI convention is to identify the subject by the URI of the bdi subdomain of the subject. The following examples are valid:

```
https://_bdi.acme.com
https://_bdi.connekt.nl
```

The _bdi subdomain has registered endpoint names. One of the registered endpoints is designed to respond with the legal identifier of the subject (EORI, LEI, DUNS etc.), plus the URI of the Association that is the home of the subject.

The following examples of the endpoints are valid:

```
https://_bdi.acme.com/ID_Assoc
https://_bdi.connekt.nl/ID_Assoc
```

Valid responses are in the form of "NL65080246@_bdi.containerassoc.nl".

1 https://datatracker.ietf.org/doc/rfc9493/ defines more granular sub_id claims that can be usefull

Appendix 1

BDI Convention JWT Default Registered Claim Names

"aud" (Audience) Claim
The "aud" (audience) claim identifies the recipients that the JWT is intended for. Each principal intended to process the JWT MUST identify itself with a value in the audience claim. If the principalprocessing the claim does not identify itself with a value in the "aud" claim when this claim is present, then the JWT MUST be rejected. In the general case, the "aud" value is an array of case-sensitive strings, each containing a StringOrURI value. In the special case when the JWT has one audience, the "aud" value MAY be single case-sensitive string containing a StringOrURI value. The interpretation of audience values is generally application specific. Use of this claim is OPTIONAL.
The BDI convention is to identify the audience by the URI of the bdi subdomain of the audience. The following examples are valid:

```
https://_bdi.acme.com
https://_bdi.connekt.nl
```

"exp" (Expiration Time) Claim
The "exp" (expiration time) claim identifies the expiration time on or after which the JWT MUST NOT be accepted for processing. The processing of the "exp" claim requires that the current date/time MUST be containing a NumericDate value. Use of this claim is OPTIONAL.
The BDI convention is to allow for a much wider time window, fitting to the physical reality context.

"nbf" (Not $_{\text{Before}}$) Claim
The "nbf" (not before) claim identifies the time before which the JWT MUST NOT be accepted for processing. The processing of the "nbf"claim requires that the current date/time MUST be after or equal to the not-before date/time listed in the "nbf" claim. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew. Its value MUST be a number containing a,NumericDate value. Use of this claim is OPTIONAL.
The BDI convention is to allow for a much wider time window, fitting to the physical reality context.

“iat" (Issued At) Claim
The "iat" (issued at) claim identifies the time at which the JWT was issued. This claim can be used to determine the age of the JWT. Its value MUST be a number containing a NumericDate value. Use of thisclaim is OPTIONAL.
The BDI convention is to allow for a much wider time window, fitting to the physical reality context.

"jti" (JWT ID) Claim
The "jti" (JWT ID) claim provides a unique identifier for the JWT. The identifier value MUST be assigned in a manner that ensures that there is a negligible probability that the same value will be accidentally assigned to a different data object; if the application uses multiple issuers, collisions MUST be prevented among values produced by different issuers as well. The "jti" claim can be used to prevent the JWT from being replayed. The "jti" value is a case-sensitive string. Use of this claim is OPTIONAL.
The BDI convention is to use a truncated hash of the content for the “jti” value. The “jti” value is used to query issuers if the JWT is not revoked.

Appendix 2

Other relevant JWT Registered Claim Names

The following Claim Names are registered in the IANA "JSON Web Token Claims" registry as to be used global, and useful in the BDI context. This is a subset of the complete list.

Claim Name	Claim Description
name	Full name
given_name	Given name(s) or first name(s)
family_name	Surname(s) or last name(s)
middle_name	Middle name(s)
gender	Gender
birthdate	Birthday
zoneinfo	Time zone
phone_number	Preferred telephone number
phone_number_verified	True if the phone number has been verified
address	Preferred postal address
vc	Verifiable Credential as specified in the W3C Recommendation Verifiable Credentials Data Model 1.0 - Expressing verifiable information on the Web (19 November 2019)
vp	Verifiable Presentation as specified in the W3C Recommendation Verifiable Credentials Data Model 1.0 - Expressing verifiable information on the Web (19 November 2019)
msisdn	End-User's mobile phone number formatted according to ITU-T recommendation [E.164]
also_known_as	Stage name, religious name or any other type of alias/pseudonym with which a person is known in a specific context besides its legal name. This must be part of the applicable legislation and thus the trust framework (e.g., be an attribute on the identity card).
ath	The base64url-encoded SHA-256 hash of the ASCII encoding of the associated access token's value
sub_id	Subject Identifier
msgi	Message Integrity Information

Appendix 3

BDI Convention JWT Custom Claim Names

The following Claim Names are not registered/custom, defined within the namespace of a set of parties.

IT security claim

“ipa” (IP address) claim

The "ipa" (IP address) claim identifies the IP address of the client computer (Data Consumer) that would like access to data. This claim allows the receiver to verify this address against white/allow lists, and implement a firewall that opens a port only for a specific IPadress and for a limited amount of time. The processing of this claim is generally application specific. Use of this claim is OPTIONAL.

Logistics claim

"fbl" (Freightbill) Claim

The "fbl" (freightbill) claim identifies the freightbill issued by the principal that issued the JWT. The processing of this claim is generally application specific. The "fbl" value is a case-sensitive string containing a StringOrURI value. Use of this claim is OPTIONAL.

The BDI convention is to identify the freightbill by its ID number and the endpoint of the issuer of the freightbill. The endpoint may be the URI of the bdi subdomain of the issuer, of the URI of the service provider.. The following examples are valid:

```
: 9864734@_bdi.acme.com
: 9874734@transfollow.nl
```

"efti" (eFTI) Claim

The "efti" (eFTI) claim identifies the eFTI information source issued by the principal that issued the JWT. The processing of this claim is generally application specific. The "fbl" value is a case-sensitive string containing a StringOrURI value. Use of this claim is OPTIONAL.

The BDI convention is to identify the ID of the freight bill, and the eFTI platform endpoint of the issuer of the freightbill. The endpoint may be the URI of the bdi subdomain of the issuer, of the URI of the service provider. The following examples are valid:

```
: 9864734@_bdi.acme.com/efti
: 9874734@efti.transfollow.nl
```

Appendix 4

Code example

```
from jwt import encode, decode
from typing import Dict
STRINT = str | int
STROPTION = str | None

# WARNING: This code is for demonstration purposes only.
# DO NOT USE IN PRODUCTION!

# There must be an entry in the keystore before any party can participate:
ks: Dict[str, str] = {}
ks["Acne_inc"] = "welkom123"
ks["Van Gend en Loos"] = "secret"
ks["De Snelle Visser"] = "qwerty"

def getkey(company_name: str) -> str:
    return ks.get(company_name, "")

# Acne commissions Van Gend en Loos to get a package from supplier Lets-B-Chemical.
# Acne creates a jwt containing the contract. The jwt is signed using Acne's key.

contract_A: str = "get a a batch of chemicals from Lets-B-Chemical"
jwt_A: str = encode({"contr": contract_A}, getkey("Acne_inc"), algorithm="HS256")

# Van Gend en Loos shares a ride with De Snelle Visser.
# Van Gend en Loos creates a jwt containing both the contract between itself and De
Snelle Visser
# and the token provided by Acne:

payload_B: Dict[str, str] = {"contr": "provide transport capacity", "embedded":
jwt_A}
jwt_B: str = encode(payload_B, getkey("Van Gend en Loos"), algorithm="HS256")

# To get contract_A from this token, do as follows:

contr_a_jwt: str = decode(jwt_B, getkey("Van Gend en Loos"), algorithms=["HS256"]).
get("embedded", "")
contr_a: str = decode(contr_a_jwt, getkey("Acne_inc"), algorithms=["HS256"]).
get("contr", "")
print(f'contract A from jwt_B: {contr_a}')
```

Appendix 4

Code example

```
# To get contract_B from this token, do as follows:

contr_b: str = decode(jwt_B, getkey("Van Gend en Loos"), algorithms=["HS256"]).
get("contr", "")
print(f'contract B from jwt_B: {contr_b}')

# De Snelle Visser sends Dolly Driver to do the pick up (someone has to do it...)
# To enable identification, De Snelle Visser has already given a token to Dolly:

employee_data: Dict[str, STRINT] = {"name": "Dolly Driver", "role": "driver",
"employee_id": 101}
employee_jwt: str = encode(employee_data, getkey("De Snelle Visser"), algo-
rithm="HS256")

# For this particular job, De Snelle Visser also creates a job_jwt, comprising both
the employee
# jwt and the contract_B:

job_data: Dict[str, str] = {"contract": jwt_B, "employee": employee_jwt}
job_jwt: str = encode(job_data, getkey("De Snelle Visser"), algorithm="HS256")

# To get te name of the driver from this jwt, do as follows:

job_jwt_payload: Dict[str, str] = decode(job_jwt, getkey("De Snelle Visser"),
algorithms=["HS256"])
empl_data_jwt: str = job_jwt_payload.get("employee", "")
driver_name = decode(empl_data_jwt, getkey("De Snelle Visser"), algo-
rithms=["HS256"]).get("name", "")
print(f'{driver_name=}')
```

Topsector Logistiek
Ezelsveldlaan 59
2611 RV Delft
+31 15 251 65 65
www.topsectorlogistiek.nl

Topsector
Logistiek