Proposed Disposition of Comments on OBE JWS Profile v 0.2

Source	Section & para reference	Comment	Proposed Response
Detlef#1	1.1 3rd para	Clarify that TPP are also PSPs: "...... Account Servicing Payment Service Providers (ASPSPs) and Third Party Providers (TPPs)	Agree
Detlef#2	2.5	Suggest add summary or at the end: "Each signature compliant with requirements of this profile is also compliant with the requirements of JAdES according to ETSI TS 119 182-1 and with JSON Web Signature according to RFC 7515" Or similar summary useful for reader not familiar with all these specifications just to convince them that this is the right approach	Agree with changes This repeats last item 7 but emphasises point. Delete 7 but keep main summary statement.
Detlef#3	5.3.2	OK with requirements [for binding certificate to JWS]. But who is deciding about the options? As far as I understood this, if a TPP takes one of these options the Relying Party is not obliged to take these into account for verifying the signature. Correct?	Disagree If the TPP (or whoever is the signer) references a certificate to be used for validation, the relying party should not choose another for validating the signature otherwise the signature may be validated against an identity not as the TPP intended. However, if the relying party wants to use other sources as the basis for path validation that is OK provided it matches the identified certificate. Further clarification of conformance requirements is given in Annex B.

Source	**Section & para reference**	**Comment**	**Proposed Response**
Detlef#4	5.3.2 Option 13	If in future the algorithm SHA256 gets also deprecated how can we enforce the usage of another algorithm, i.e. SHA512.	Currently, section 7.2 recommendations 27 and 28 recommend use of ETSI TS 119 312 which is regularly updated. Should recommendation 28 be a requirement ? Note added referencing section 7. Also, section 7 updated to makes specific reference to x5t#S256 and x5t#o. Should there be a sunset clause mandating switch to x5t#o a certain time after SHA256 is declared not strong enough?
Detlef#5	3.3, 4.1, 5.6.2, 6.5.1	Minor editorials	Agree
Hervé	2.4	Add design Principle aiming that signature could also apply on HTTP responses.	Agree with changes Also, HTTP requests. Aimed at protecting HTTP requests and, where required, HTTP responses.
Kornél#1	2.1 (x2), 2.2, 2.5 item 6	Minor editorials	Agree

Source	Section & para reference	Comment	Proposed Response
Kornél#2	3.2	Proposal: change this requirement into a recommendation ("should" instead of "shall"). The JWS Compact Serialisation does not allow unsigned header fields to be added. This restricts the signature to JAdES-B-B level, it cannot be augmented, even later. A B-B level signature only protects the data until the signing certificate expires or is revoked. Therefore, a B-B level signature can be denied by the signer after revocation of his certificate, so it does not provide non-repudiation. Since from the OBE survey we know that non-repudiation can be a requirement in some cases (e.g. for some specific transactions), the profile should allow using at least JAdES-B-T level signatures too. When JAdES-B-T, JAdES-B-LT or JAdES-B-LTA level signatures are intended, the JWS JSON Serialization needs to be used, to allow the time-stamp, validation data, etc. to be placed in the unsigned header fields. The resulting JSON serialized JWS can then still be carried in the "x-jws-signature" header field using RFC 8187 ext-value encoding or RFC 2047 encoding	Disagree Using JWS compact serialisation and unsigned headers is not interoperable with use of compact serialisation. It also adds significantly to the complexity of the implementation. Augmentation is not appropriate to real-time TPP-ASPSP interactions addressed by the profile. Also, time-stamping is not necessary to assure the non-repudiability of the signatures. The signature includes a claimed signing time and, as this is a real time interaction, the relying party can verify this against the current time. In case of dispute the relying party can present its logs to confirm that the claimed signing time has be checked to be correct (within an acceptable window).

Source	Section & para reference	Comment	Proposed Response
Kornél#3	5.3.2 requirement 9 (x5t#256) And requirement 13 (x5t#o) 6.4.2 recommendation 025	This does not allow the "algorithm agile" alternative x5t#o, which is defined in JAdES as an alternative, which can be used instead of x5t#S256, but not together with it.	Agree with changes X5t#S256 is required to ensure interoperability between implementations. Propose change to JAdES to allow both to be present. Should there be a "sunset" clause on requiring X5t#S256 when x5t#o to be present?
Kornél#4	5.3.2 option 11, to 14 option 1	"may not" is ambiguous, as it could be understood as "shall not". Use "need not" as per ETSI terminology.	Agreed
Microsoft (FAPI) #1	5.3	I am a bit surprised by the requirement to use keys from X.509 certificates in Section 5.3, rather than keys from JWKs. But I understand that that may be the reality of the targeted deployment environments.	Disagree This is aligned with RFC 75151 and JWK (RFC 7517) both of which supported of the x5c and x5t#S256 In the EU the management of keys used for identification is required to follow eIDAS as be based on X.509.
Microsoft (FAPI) #2		I understand the reference to draft-cavage-http-signatures but everyone should be aware that this is a work in progress and is likely to change. If you want to keep the reference, the draft should probably explicitly say that the specification uses draft-cavage-http-signatures-10 – even though subsequent and potentially incompatible versions may be published.	Noted Document already reference the specific version. The aim is to have a technically stable requirements. It would be a nice-to-have a stable IETF RFC to follow but as yet there is nothing stable available. A possible approach is to get copyright release from the authors of Cavage for the one small section referenced.

Source	**Section & para reference**	**Comment**	**Proposed Response**
Microsoft (FAPI) #3		The x5t#o header parameter is pretty strange. If new thumbprint algorithms are needed, it would be better to register new values, like x5t#S256 was, rather than to introduce a level of indirection to determine the digest algorithm. It's not the end of the world, but it's certainly not how the JOSE working group would have added additional digest algorithms.	The approach generally recommended is that implementations and their specifications should be algorithm agile so that when an algorithm is found to be compromised a new one can be easily adopted. I would consider it preferable to not fix cryptographic algorithms in protocols but to define the protocol in a way that it can be easily change.
Roberto #1	2.3 definition of UTC	I won't mention GMT, it's not an industry standard	Agree with changes As GMT is more generally understood than UTC is suggested to keep this as a Note "Note: This is equivalent to Greenwich Mean Time"
Roberto #2	3.1	I would provide some security considerations in using JWS in http headers. See: • https://github.com/WICG/webpackage/issues/237#issuecomment-401191859 • https://tools.ietf.org/html/draft-ietf-oauth-jwt-bcp-07 • Why not json: https://httpwg.org/http-extensions/draft-ietf-httpbis-header-structure.html#why-not-json • See Signed exchanges for some security considerations: https://wicg.github.io/webpackage/draft-yasskin-http-origin-signed-responses.html#name-signing-oracles-are-permane	Noted Seems to continue discussion on JWS vs something else

Source	Section & para reference	Comment	Proposed Response
Roberto #3	4.1	If we standardize this header, we should remove the X- and file a registration request to the IANA. https://tools.ietf.org/html/rfc6648 deprecates X-headers. Together with this specification you should file a IANA registration for eg. the JWS-Signature header. This document should include this filing registration.	Noted For moment this is not to be standardised at IETF level to keep as is.
Roberto #4	5.3.1 x5c definition	or certificate chain, represented as a JSON array(s)	Agree with changes Adopt JWS description: "X.509 public key certificate or certificate chain corresponding to the key used to digitally sign the JWS.
Roberto #5	5.5.2	This may prevent the ability to nest a JWT inside the outer one. See https://tools.ietf.org/html/rfc7519#appendix-A.2	No change There is no requirement for nesting JWT in JWS to create a nested JWT.
Roberto #6	6.2.1 2nd para	iirc this is RFC3339. Not sure JSON has a datetime format, though Javascript may process RFC3339 / ISO8601 dates correctly. Why not using the UNIX-Timestamp format like the other iat,exp,nbf claims?	Partially Agree This is aligned with RFC 3339. Which is generally internationally (ISO 8601) recognised format for time. Text to be updated to reference RFC 3339.
Roberto #7	6.2.1 3rd para &	The HTTP Date header has specific constraints and a weird format. I would dis-entangle a set of signature metadata from the Date header. The signature validity should be checked using the nbf, iat, exp claims	Partially agree Delete checks vs HTTP header. See next comment on nbf, iat, exp claims.

Source	Section & para reference	Comment	Proposed Response
Roberto #8	6.2.2	IMHO setting a time window in a spec is not correct. The owner of the key should be in charge of deciding the interval of validity of its own signature using the exp,iat,nbf claims. Any other decision is in my opinion arbitrary and does not protect the right of the signer to limit the validity of its signature. If we want a signature to last for 4 hours, we should ask the signer to explicit this in the claim.	Disagree The aim of this time window is not to have expiry of the signature. Rather to ensure that there are checks on the claimed signing time to ensure that the validation checks are carried out around the right time and the signature is "live". This should not be under the control of the Signer.
Roberto #9	6.3.1	This is not correct. Digest conveys the checksum of the selected representation. Here is the clarification spec on digest we're writing in the ietf https://httpwg.org/http-extensions/draft-ietf-httpbis-digest-headers.html	Agree with changes The scope of the digest needs to be defined. Also, should refer to HTTP Message Body not payload. Propose to adopt text based on Berlin group implementation guidelines section 12.1. " The "Digest" Header shall contain a Hash of the message body, if the message does not contain a body, the "Digest" header must contain the hash of an empty bytelist. Note: In case of a multipart message the same method is used to calculate the digest. I.e. a hash of the (whole) message body is calculated including all parts of the multipart message as well as the separators." See; https://www.di-mgt.com.au/sha_testvectors.html for SHA hash of empty byte (bit) string.

Source	Section & para reference	Comment	Proposed Response
			Also, delete the words " if the HTTP Payload [Body] is present". Change terminology for HTTP Payload to HTTP Body.
Roberto #10	6.2.2 Recommendation 22	add Content-Encoding" if present" Please, see https://httpwg.org/http-extensions/draft-ietf-httpbis-digest-headers.html#usage-in-signatures	Agree
Editor	5.5.2	Alingment with JAdES	cty "shall not" be present.
Juan Carlos	5.3.2	Would it not be worth to also add a mention to x5t#o? This would leave the first sentence as follows: The "x5u" header parameter may be used to further identify a certificate if "x5t#S256" or "x5t#o" are used.	Agreed
Juan Carlos	5.3.2	It is not completely clear the concrete meaning of the second sentence: "If present, this may not be checked against the certificate used to validate the signature by the relying party	Already changed "may not" to "need not"