Comparison Between Different JSON Signature Schemes

JWS - Interoperability Concerns				
Parameter	**Predictive**	**Canonical**	**"Original"**	**Comment**
Data Serialization	Critical	Critical	*Data Format Independent (Base64Url)*	Predictive and Canonical modes depend on that JSON primitives are serialized according to ES6
Property Order	Critical			Predictive mode depends on that JSON properties are kept in their original order
ES6 "Quirk"	Minor Nuisance			Predictive mode depends on that numerical property names are dealt with in an ES6 compliant way

JWS - Implementation Considerations				
Parser Upgrade	Required[1,2]	Optional[2]	*Stand alone Solution*	Predictive mode depends on that property order is honored by the parser
Serializer Upgrade	Required[1,2]	Optional[2]		Predictive mode depends on that property order is honored by the serializer
Post Processor Option	N/A	Yes		A post processor option has the advantage that it can be introduced at an early stage

1. For non ES6 systems, this may be accomplished through existing options, additional code, or in some cases through upgraded JSON tools.
2. Signature support can ideally be integrated as a derived data type. This is the case for https://github.com/cyberphone/openkeystore

What does the JSON Canonicalization Scheme (**JCS**) https://cyberphone.github.io/doc/security/draft-rundgren-json-canonicalization-scheme.html bring to table which the abandoned I-D (**Staykov-Hu**) https://tools.ietf.org/html/draft-staykov-hu-json-canonical-form-00 did not?

- **JCS** *mandates* that a JSON Number is limited to IEEE-754 double including providing a rationale based on I-JSON/ES6, while **Staykov-Hu** builds on the *assumption* that JSON Number is an IEEE-754 double, although RFC8259 does in fact *not* require that.
- **JCS** builds on ES6 for number serialization rather than using XML's double (ECMAScript serialization wasn't viable before ES6).
- **JCS** builds on ES6 for the serialization of strings, while **StayKov-Hu** *does not specify string serialization* at all.
- **JCS** specifies a platform independent sorting algorithm. I'm not sure how to interpret the **StayKov-Hu** sorting scheme.
- The "market" for signed JSON is way bigger today than 2012.