




Linked Data Signatures WG

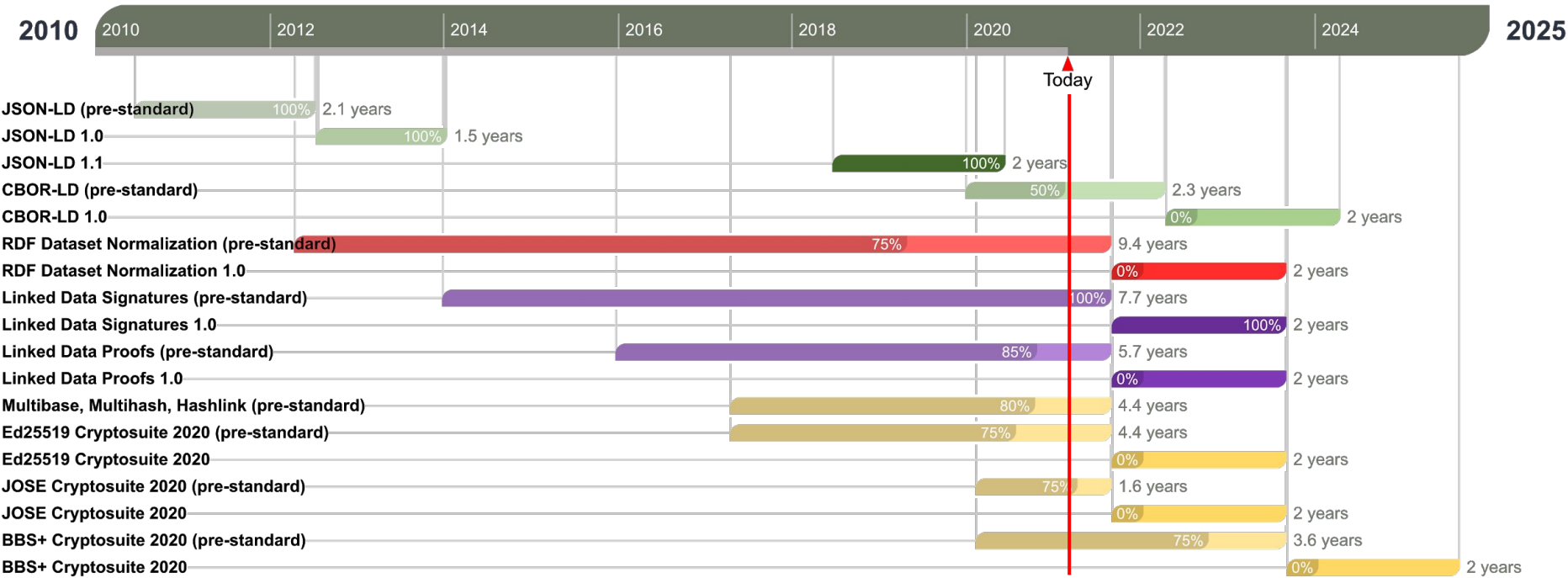
Adding digital signatures and digital proofs to existing linked data systems

In this session, we will discuss:

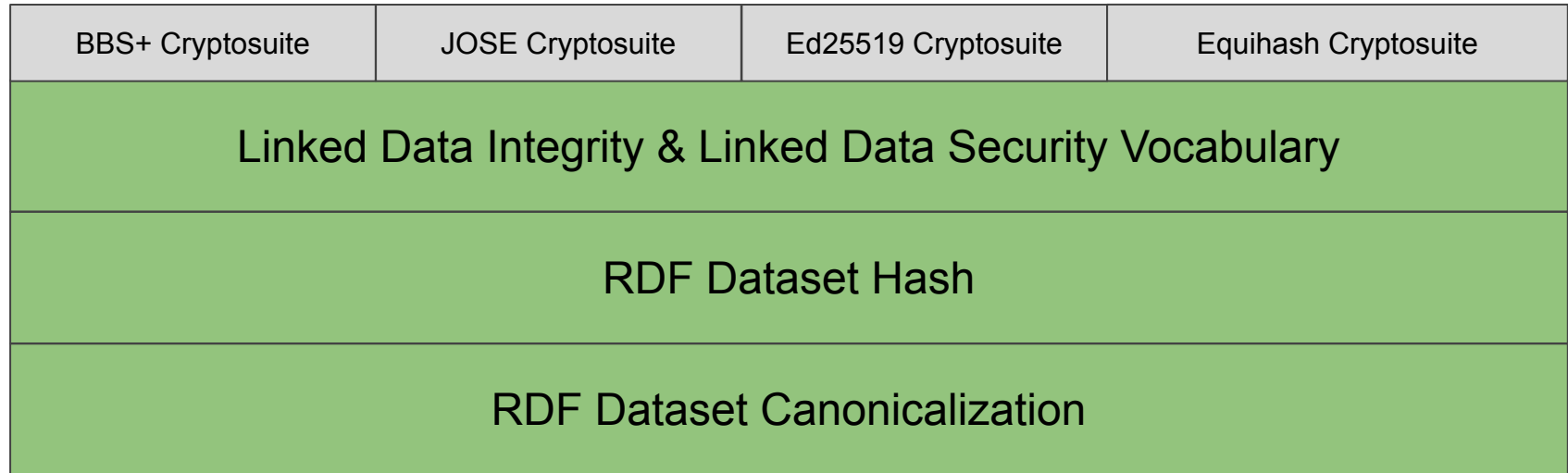


- 
- A short horizontal line with a teal-to-orange gradient.
- A Roadmap and the LDS Charter
 - RDF Dataset Canonicalization (RDC)
 - RDF Dataset Hash (RDH)
 - Linked Data Integrity (LDI)
 - Linked Data Security Vocabulary (LDSV)
 - The current (final?) Charter state...

Linked Data Security Roadmap



The Linked Data Security Stack (excerpt)



§ 5. Use Cases and Requirements

Some typical use cases for RDF Dataset Canonicalization and/or signatures are:

Detecting changes in Datasets

When processing RDF Datasets over a period of time, determining if information has changed is helpful. For example, knowing if information has changed helps with data cache invalidation, detecting if expected data has been tampered with or modified, or when debugging unexpected changes in source RDF Datasets.

Requirement: RDF Dataset Canonicalization and Hash algorithms.

Space-efficient verification of the contents of Datasets

If unique identification of RDF Datasets is possible, one can cryptographically hash the information to establish a storage-efficient way to verify that the information has not changed over time. One property of cryptographic digests is that one can verify data integrity. For example, a small device sending an RDF Dataset to a remote storage location can compute a cryptographic digest for later use in verifying that all the data arrived intact and has not been tampered with.

Requirement: RDF Dataset Canonicalization and Hash algorithms.

(Contributed by Alan Karp.)

RDF Dataset Canonicalization (RDC)

This specification defines an algorithm to produce a [canonicalization function](#) of an arbitrary RDF Dataset.

Reference: [RDF Dataset Canonicalization](#), eds. Dave Longley, Manu Sporny, W3C Draft Community Group Report, 2019.

Expected completion: WG-START + 24 months.

See also the mathematical basis for [RDF Dataset Canonicalization](#):

1. [RDF Dataset Canonicalization](#), Rachel Arnold, Dave Longley, W3C Credentials Community Group, 2020.
2. [Canonical Forms for Isomorphic and Equivalent RDF Graphs: Algorithms for Learning and Labelling Blank Nodes](#), Aidan Hogan, ACM Trans. Web, vol. 11, no. 4, p. 22:1-22:62, 2017.

RDF Dataset Hash (RDH)

This specification details the processing steps of a hash function for an arbitrary RDF Dataset. These step include (1) the generation of a [canonical form](#) using the algorithm specified in the “RDF Dataset Canonicalization” deliverable, (2) sorting the [N-Quads](#) serialization of the canonical form generated in the previous step, and (3) application of a cryptographic hash function on the sorted [N-Quads](#) representation.

Reference: [Linked Data Proofs 1.0](#), eds. Dave Longley, Manu Sporny, W3C Draft Community Group Report, 2020.

Expected completion: WG-START + 24 months.

Linked Data Integrity (LDI)

This specification defines a framework for expressing, in an RDF Graph, proofs of integrity of RDF Datasets. The group defines that framework to work with RDH, although the hashing algorithm, and other constituents of proofs of integrity, are identified as assertions, allowing the same framework to be used with other algorithms. Beyond the generic (normative) framework, this deliverable also provides a normative definition for Linked Data Signatures, as a prominent approach used as a proof of integrity. The specification enables a 3rd party to verify the integrity of the data.

Reference: [Linked Data Proofs 1.0](#), eds. Dave Longley, Manu Sporny, W3C Draft Community Group Report, 2020.


Expected completion: WG-START + 24 months.

Linked Data Security Vocabulary (LDSV)

This specification defines an RDF Vocabulary for the terms defined in the [Linked Data Integrity](#) deliverable. The specification may also define one or more [JSON-LD Context](#) documents to be used by a JSON-LD serialization.

Reference: [The Security Vocabulary](#), ed. Manu Sporny, Orie Steele, Tobias Looker, W3C Draft Community Group Report, 2020.

Expected completion: WG-START + 24 months.



2.2 Other Deliverables

Other non-normative documents may be created such as:

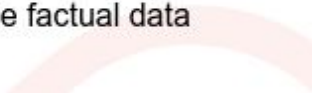
- A Linked Data Security Registry, containing Linked Data related cryptographic terms, including, but not limited to, the terms defined in the Linked Data Security Vocabulary (LDSV), as well as terms for the definition of other Linked Data Integrity related proof methods.

If, during the chartered period of the Working Group, the [next version of the W3C Process](#) is adopted, incorporating the concept of a [Registry Track](#), the Working Group will consider publishing the Linked Data Security Vocabulary Recommendation with an additional [Registry Section](#) incorporating the terms planned in this deliverable.

- Note on additional Linked Data Integrity techniques that are not necessarily relying, or only partially, on the specifications developed by this Working Group. (See also [the explainer for some more details](#).)
- Test suite and implementation report for the specification.
- Primer or Best Practice documents to support web developers when designing applications.

1.1 Out of Scope

The following items are out of scope, and will not be addressed by this Working group:

- Definition of new cryptographic signature or encryption algorithms such as RSA, ECDSA, EdDSA, and AES. This Working Group will only define usage of, and suitable terms to *identify*, such algorithms.
 - Authenticity and trust issues of Web (Data) content that go beyond the exchange and the integration of simple factual data expressed in RDF. (See also the [some further considerations](#) in the explainer document.)
- 



Appendix

What can we do with this stack?



```
{
  "@context": "http://schema.org/",
  "@type": "Person",
  "name": "Jane Doe",
  "jobTitle": "Professor",
  "telephone": "(425) 123-4567",
  "url": "http://www.janedoe.com"
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2019-09-16T15:22:02Z",
    "verificationMethod": "did:example:123#key1",
    "domain": "website.example",
    "nonce": "fpcwi43io"
    "proofPurpose": "authentication",
    "proofValue": "zj902k...lY7dhH",
  }
}
```

Information

Proof
(aka: digital signature)



- Transforms input to deterministic output (useful for digital signatures)
- 2 mathematical proofs with peer review

RDF Dataset Canonicalization Example



```
{
  "@context": "http://schema.org/",
  "@type": "Person",
  "name": "Jane Doe",
  "jobTitle": "Professor",
  "telephone": "(425) 123-4567",
  "url": "http://www.janedoe.com"
}
```



```
{"@context":"http://schema.org/","@type":
:"Person","name":"Jane Doe","jobTitle":
:"Professor","telephone":"(425) 123-456
7","url":"http://www.janedoe.com"}
```



```
_:c14n0 <http://schema.org/jobTitle> "Professor" .
_:c14n0 <http://schema.org/name> "Jane Doe" .
_:c14n0 <http://schema.org/telephone> "(425) 123-4567" .
_:c14n0 <http://schema.org/url> <http://www.janedoe.com> .
_:c14n0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/Person> .
```



Used to express various types of digital proofs such as:

- Proof of Work
- Proof of Existence
- Proof of Elapsed Time
- Digital Signatures

LD Proof Example




```
"proof": {  
  "type": "EquihashProofOfWork2018",  
  "created": "2020-09-16T15:22:02Z",  
  "domain": "blockchain.example",  
  "proofPurpose": "capabilityInvocation",  
  "proofValue": "z8je...0a19E",  
  "nonce": "fdfeld4d"  
}
```



Linked Data Signatures

- A type of Linked Data Proof
- Used to express Digital Signatures

LD Signature Example

A horizontal bar with a teal segment on the left and an orange segment on the right.

```
"proof": {  
  "type": "Ed25519Signature2020",  
  "created": "2021-03-16T15:22:02Z",  
  "verificationMethod": "did:example:123#key1",  
  "domain": "website.example",  
  "nonce": "fjpmqi43io"  
  "proofPurpose": "authentication",  
  "proofValue": "zj902k...lY7dhH",  
}
```



- Provide approved / pre-packaged cryptography suites.
- Typically bundle Canonicalization algorithm, hashing algorithm, and signature algorithm

Linked Data Cryptosuite



Cryptosuites define:

- Public key formats
- Canonicalization mechanism
- Hashing mechanism
- Signature formats

