

JSON Web Proofs

JWTs with Superpowers*

JSON Web Proofs

What it is

- Data container for supporting “anonymous credentials” style use cases
- Features such as:
 - Selective Disclosure
 - Multi-use without linkability
 - Predicate Proofs

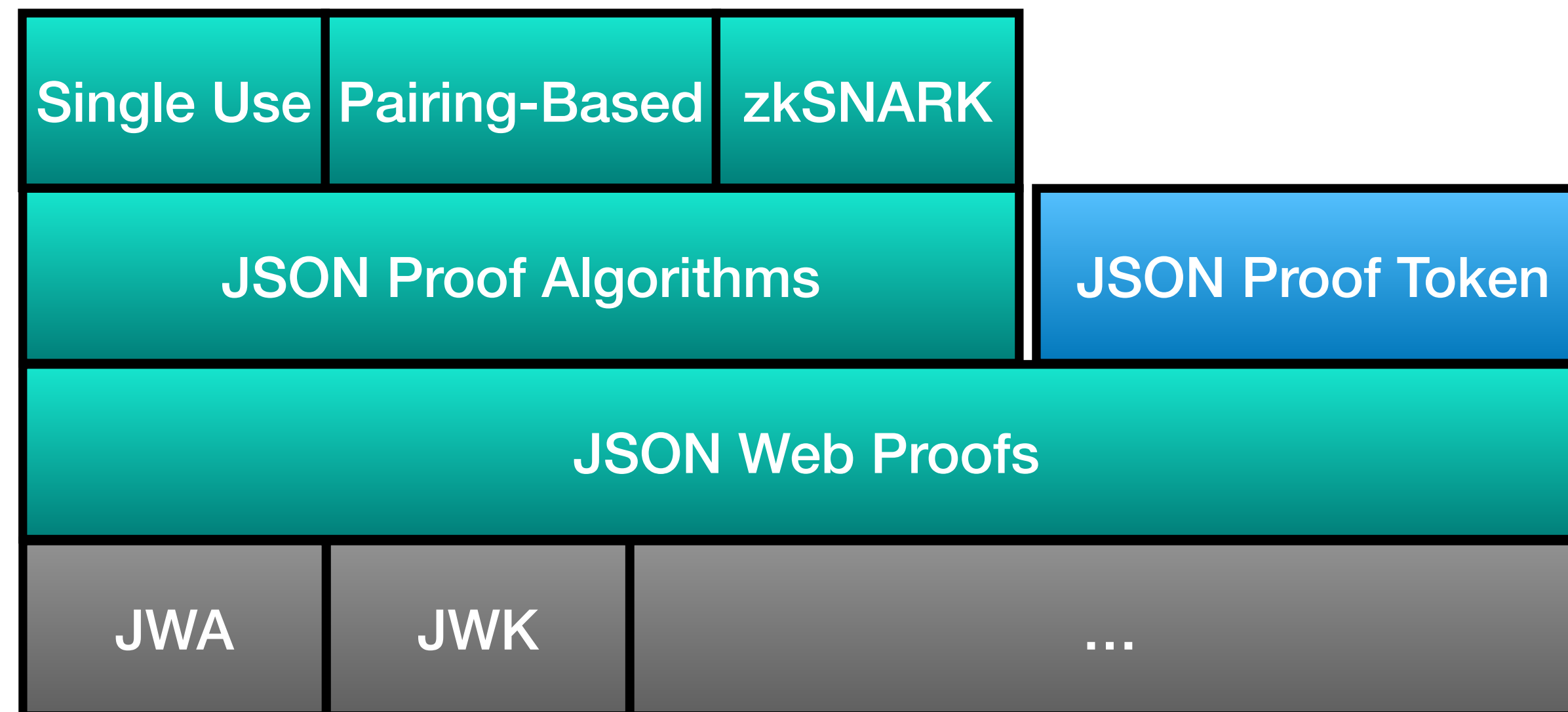
JSON Web Proofs

Standards Relations

- Incubated within Decentralized Identity Foundation Advanced Crypto WG
- Very Early!
- JOSE (JSON Object Signing and Encryption) inspired
 - Various JWA, JWK, JWT dependencies
- Would like to see it moved to IETF following incubation
- Also motivated to define an equivalent CBOR-format container

JSON Web Proof Structure

(As envisioned)



Classic JSON Web Signature

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWUiOiIxMjM0NTY3ODkwIiwibmFtZSI6IjE2IiwiaWF0IjoxNTE2MzI1ODUyLCJ1aWQiOiJhZGQsInR5cCI6IkpXVCJ9.
SfIKxwRJSMeKKF2QT4fwpMeJf36POk6
yJV_adQssw5c

Classic JSON Web Signature

[illegible]

JSON Web Proof

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWliOiIxMjM0NTY3ODkwIiwibmFt
SI6IkpvaG4gRG9IiwiYWFWF0IjoxNTE2MM~
JhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9ey.
SfIKxwRJSMeKKF2QT4fwpMeJf36POk6
yJV_adQssw5c

JSON Web Proof

The diagram illustrates the structure of a zk-SNARK transaction, which is composed of three main parts:

- Protected Header:** The top section, highlighted in red, containing the text: `eyJhbGciOiJIUzE6IkpXVCJ9.`
- Payloads:** The middle section, highlighted in green, containing the text: `eyJzdWliOiIxMjM0NTY3ODkwIiwibmFtIjoiOXNTE2MM~`
- Proof:** The bottom section, highlighted in blue, containing the text: `SfIKxwRJSMeKKE2OT4fwpMeJf36POk6yJV_adQssw5c`

The transaction is represented as a stack of three blocks, with the Protected Header, Payloads, and Proof sections clearly demarcated by colored boxes.

JSON Web Proof

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZ
~~.SflKxwRJSMeKKF2QT4fwpMeJf36P
Ok6v..IV_adOssw5c

Two Omitted Payloads

JSON Proof Token

For Token/Credential-style Use-cases

«Protected Header»

```
{  
  "alg": "ES256",  
  "typ": "JWT"  
}
```

«Payload»

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "iat": 1516239022  
}
```

JWT Example

«Protected Header»

```
{  
  "alg": "ES256+SU",  
  "typ": "JPT",  
  "kid": "12345"  
}
```

«Payloads»

"1234567890"

"John Doe"

1516239022

JPT Example

JSON Proof Token

For Token/Credential-style Use-cases

```
{
  "jwks": [
    {
      "kid": "12345",
      ...
      "token-payloads": [
        { "claim": "sub" },
        { "claim": "name" },
        { "claim": "iat" }
      ]
    }
  ]
}
```

Issuer Metadata

```
«Protected Header»
{
  "alg": "ES256+SU",
  "typ": "JPT",
  "kid": "12345"
}
«Payloads»

"1234567890"

"John Doe"

1516239022
```

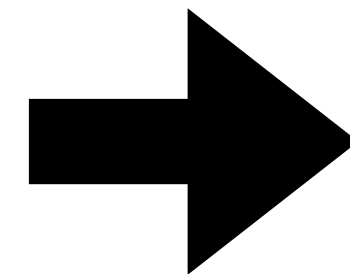
JPT Example

JSON Proof Token

Omitting a payload

```
{  
  "jwks": [  
    {  
      "kid": "12345",  
      .../  
      "token-payloads": [  
        { "claim": "sub" },  
        { "claim": "name" },  
        { "claim": "iat" }  
      ]  
    }  
  ]  
}
```

Issuer Metadata



```
«Protected Header»  
{  
  "alg": "ES256+SU",  
  "typ": "JPT",  
  "kid": "12345"  
}  
«Payloads»  
  
"1234567890"  
  
"John Doe"  
  
1516239022
```

JPT Example

Single Use Scheme

- Proof is simply a concatenation of signatures, e.g.
Sig(*header* || *Sig(payload₁)* || ... || *Sig(payload_n)*) || Sig(payload₁) || ... || Sig(payload_n)
- One could imagine a variety of other approaches (seeded Merkel tree, etc)
 - Multiple signatures is just easier to specify and implement
- Allows for use of NIST approved algorithms, out-of-box crypto support (including secure element usage)
- Does not expose some primitives needed for a subset of predicate algorithms
 - Other approaches available for selective disclosure, such as hash chains