Pelotas/RS, *Junho 2020*

Padrões de formatação para troca de dados

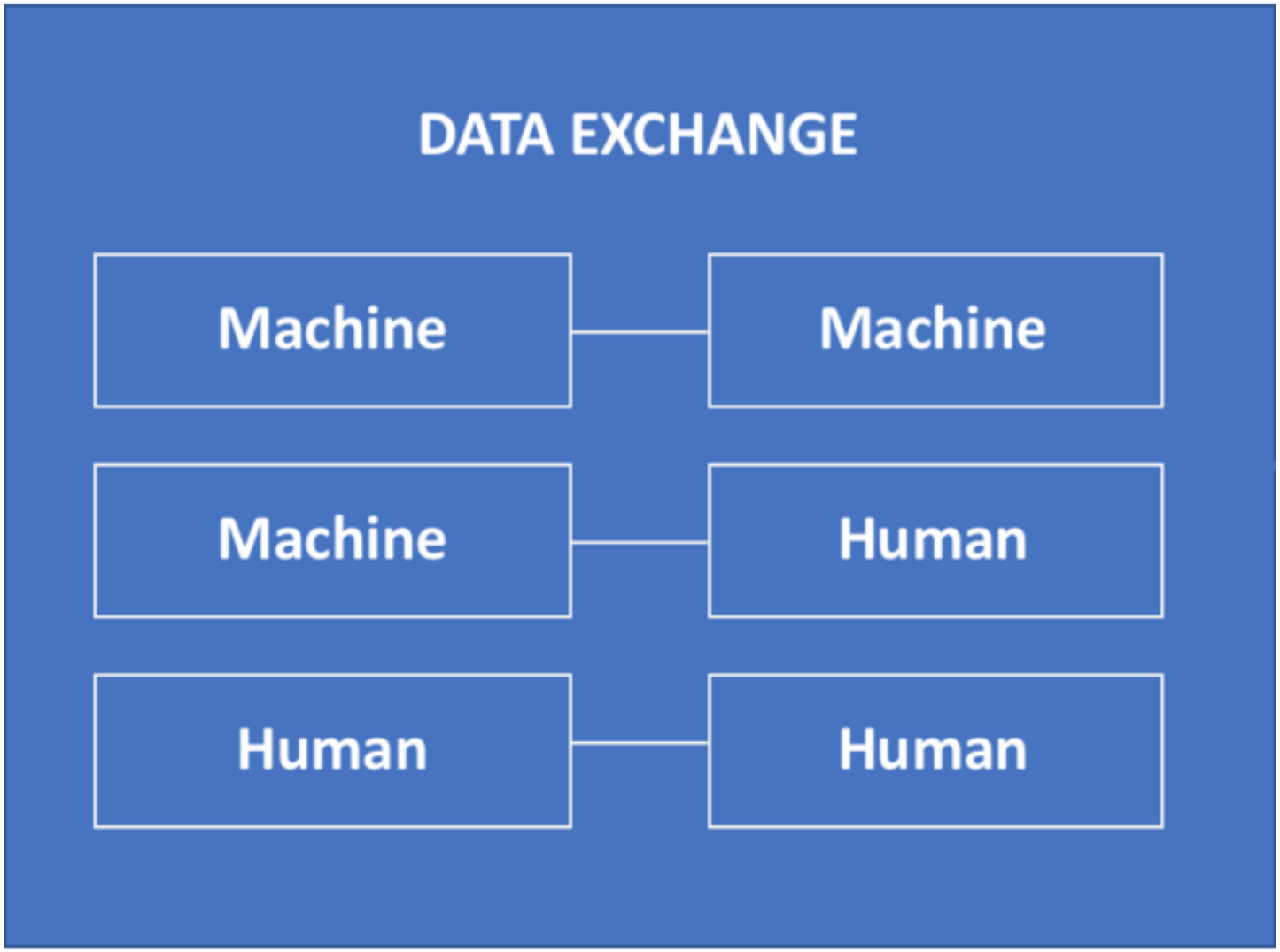
Cleber Peter

clebers.peter@sou.ucpel.edu.br

Universidade Católica de Pelotas (UCPEL)

Roteiro

- Introdução
- JavaScript Object Notation (JSON)
- Concise Binary Object Representation (CBOR)
- Javascript Object Signing and Encryption (JOSE)
- CBOR Object Signing and Encryption (COSE)

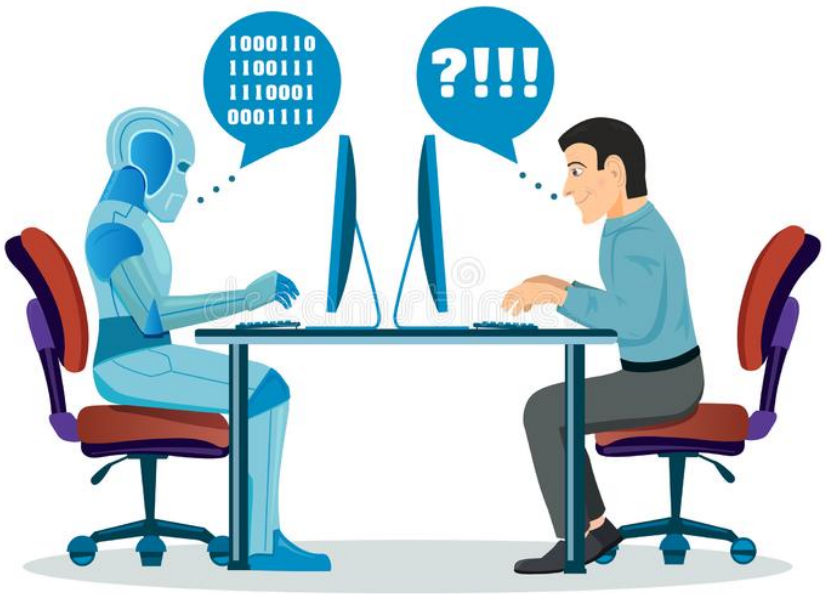

DATA EXCHANGE
Machine
Machine
Machine
Human
Human
Human

Máquina - Máquina (M2M)

Os dados devem ser formatados de tal forma que seja minimizada a quantidade de memória necessária para sua representação.

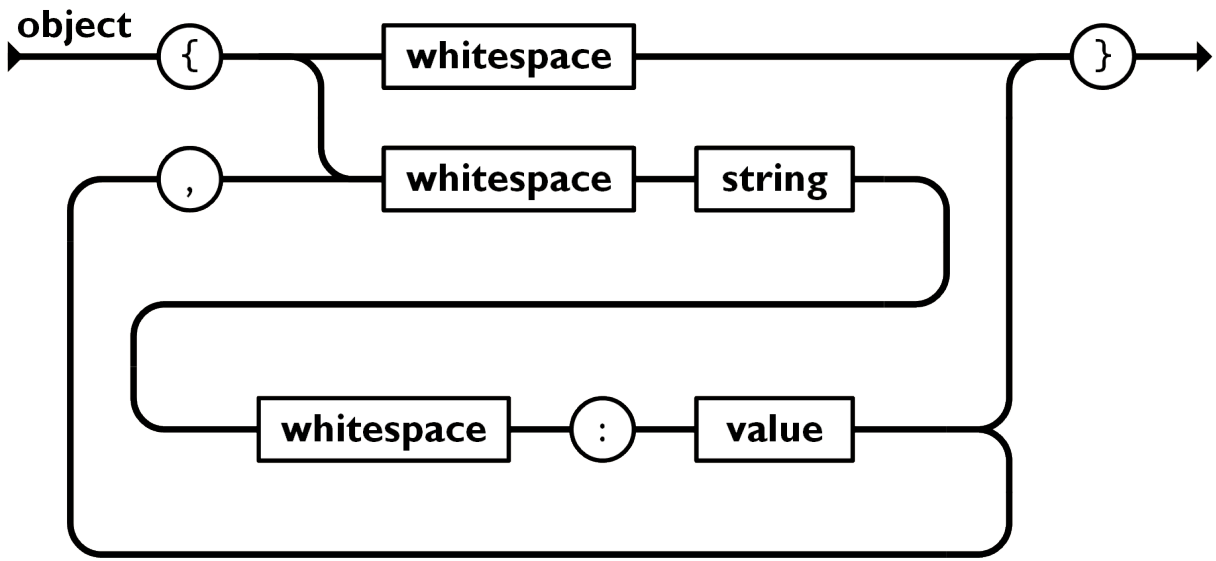
Homem - Máquina (H2M)

Os dados devem ser formatados de tal forma que possam ser facilmente interpretados por humanos.



JavaScript Object Notation (JSON)

Padronizado pela RFC 4627 é uma formatação amplamente adotada em aplicações H2M visto que é humanamente inteligível.



JSON - Exemplo


```
{
   "widget":{
      "window":{
         "title":"Sample Konfabulator Widget",
         "name":"main_window",
         "width":500,
         "height":500
      },
      "image":{
         "src":"Images/Sun.png",
         "name":"sun1",
         "alignment":"center"
      },
      "text":{
         "data":"Click Here",
         "size":36,
         "style":"bold"
      }
   }
}
```

```
{
   "menu":{
      "header":"SVG Viewer",
      "items":[
         {
            "id":"Open"
         },
         {
            "id":"OpenNew",
            "label":"Open New"
         },
         {
            "id":"ZoomOut",
            "label":"Zoom Out"
         },
         {
            "id":"OriginalView",
            "label":"Original View"
         },
         null
      ]
   }
}
```

JSON x XML

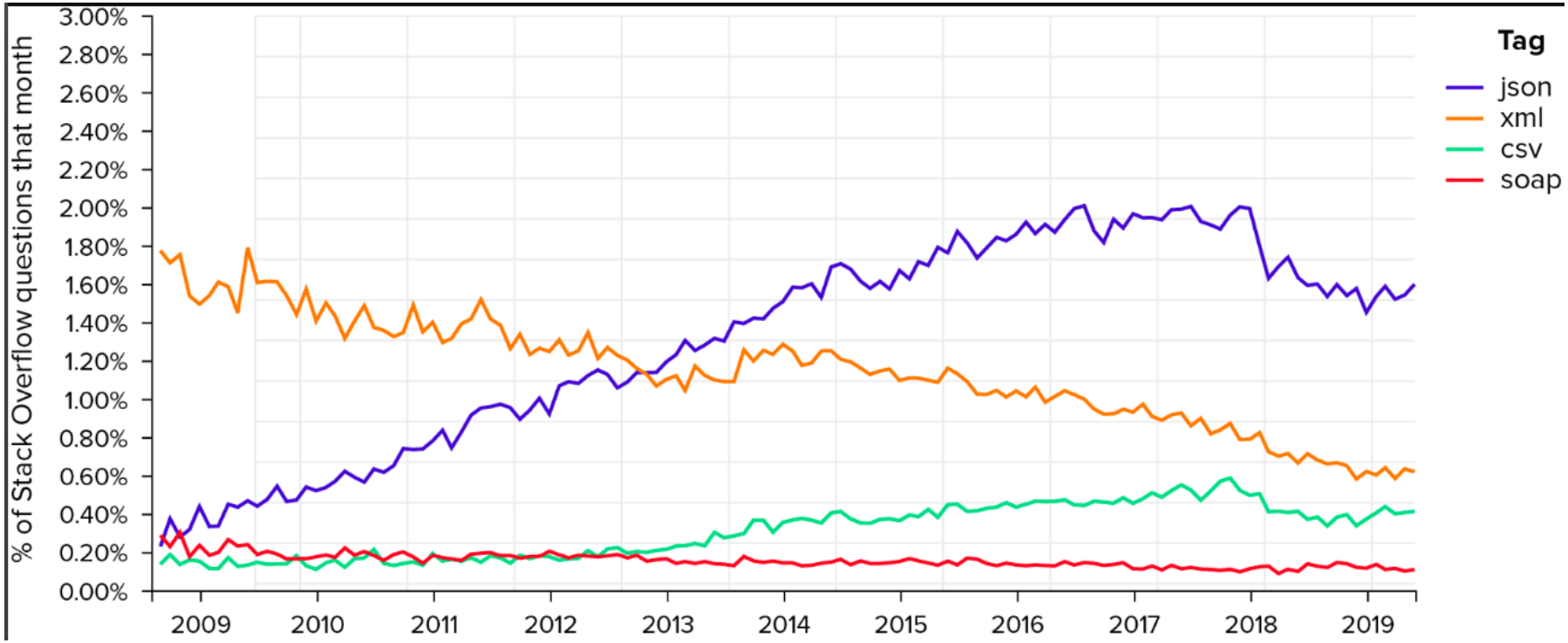
http://localhost:8080/Json/SyncReply/Contacts

```
{
  - Contacts: [
      - {
            FirstName: "Demis",
            LastName: "Bellot",
            Email: "demis.bellot@gmail.com"
        },
      - {
            FirstName: "Steve",
            LastName: "Jobs",
            Email: "steve@apple.com"
        },
      - {
            FirstName: "Steve",
            LastName: "Ballmer",
            Email: "steve@microsoft.com"
        },
      - {
            FirstName: "Eric",
            LastName: "Schmidt",
            Email: "eric@google.com"
        },
      - {
            FirstName: "Larry",
            LastName: "Ellison",
            Email: "larry@oracle.com"
        }
    ]
}
```

http://localhost:8080/Xml/SyncReply/Contacts

```
<ContactsResponse xmlns:i="http://www.w3.org/20
  <Contacts>
    <Contact>
      <Email>demis.bellot@gmail.com</Email>
      <FirstName>Demis</FirstName>
      <LastName>Bellot</LastName>
    </Contact>
    <Contact>
      <Email>steve@apple.com</Email>
      <FirstName>Steve</FirstName>
      <LastName>Jobs</LastName>
    </Contact>
    <Contact>
      <Email>steve@microsoft.com</Email>
      <FirstName>Steve</FirstName>
      <LastName>Ballmer</LastName>
    </Contact>
    <Contact>
      <Email>eric@google.com</Email>
      <FirstName>Eric</FirstName>
      <LastName>Schmidt</LastName>
    </Contact>
    <Contact>
      <Email>larry@oracle.com</Email>
      <FirstName>Larry</FirstName>
      <LastName>Ellison</LastName>
    </Contact>
  </Contacts>
</ContactsResponse>
```

JSON - Tendência



Concise Binary Object Representation (CBOR)

Padronizado pela RFC 7049 é uma formatação amplamente adotada em aplicações M2M visto sua representação minimalista para o tratamento de dados binários.

CBOR data	Data item 1				Data item 2				Data item X...
Byte count	1 byte (CBOR data item header)		Variable	Variable	1 byte (CBOR data item header)		Variable	Variable	etc...
Structure	Major type	Additional information	Payload length (optional)	Data payload (optional)	Major type	Additional information	Payload length (optional)	Data payload (optional)	etc...
Bit count	3 Bits	5 Bits	8 Bits × variable	8 Bits × variable	3 Bits	5 Bits	8 Bits × variable	8 Bits × variable	etc..

CBOR - Exemplo

```
name: "Strawberry Pie"
data: <00 01 02 03 04 05 06 07 08 09>
```

```
{"name":"Strawberry Pie","data":"AAECAwQFBgcICQ=="}
```

Length: 51 bytes.

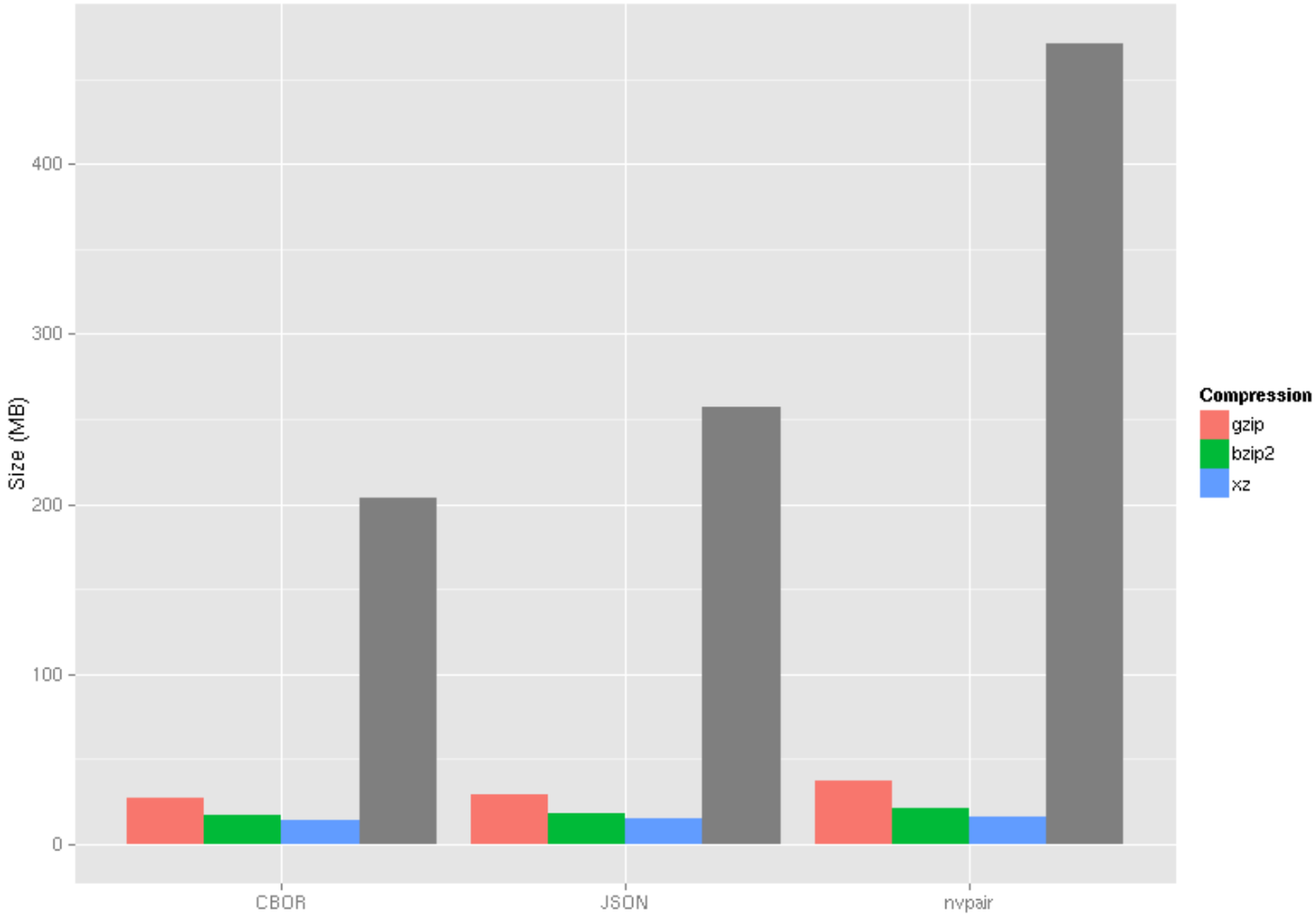
```
a2646e616d656e537472617762657272792050696965696a7065675f646174614a00010203040506070809
```

```
a2                  -- Map, 2 pairs
  64                -- String, length: 4
    6e616d65        -- {Key:0}, "name"
  6e                -- String, length: 14
    537472617762657272792050696965 -- {Val:0}, "Strawberry Pie"
  64                -- String, length: 4
    64617461 -- {Key:1}, "data"
  4a                -- Bytes, length: 10
    00010203040506070809 -- {Val:1}, 00010203040506070809
```

Length: 35 bytes.

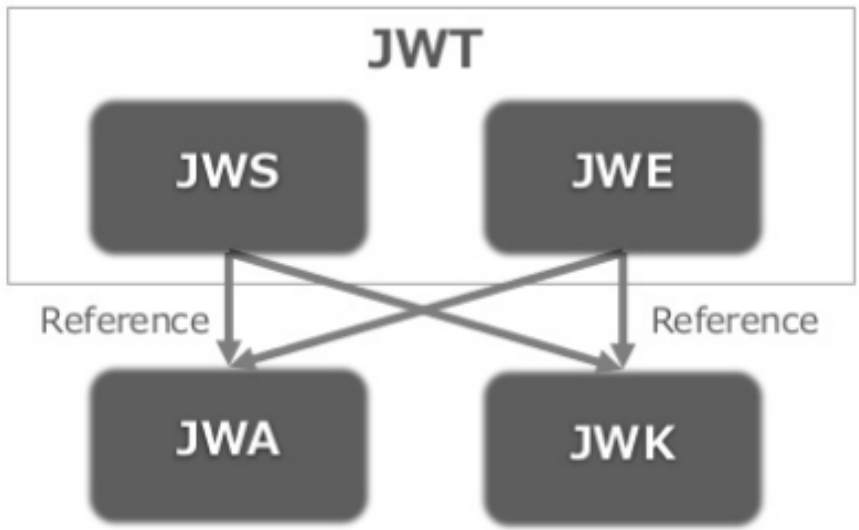
	JSON	CBOR
Median encoding time (ms)	3.983	0.538
Median decoding time (ms)	3.151	0.006
Encoded size (bytes)	1,213,676	910,262

CBOR x JSON


Size (MB)
400
300
200
100
0
CBOR
JSON
nvpair
Compression
gzip
bzip2
xz

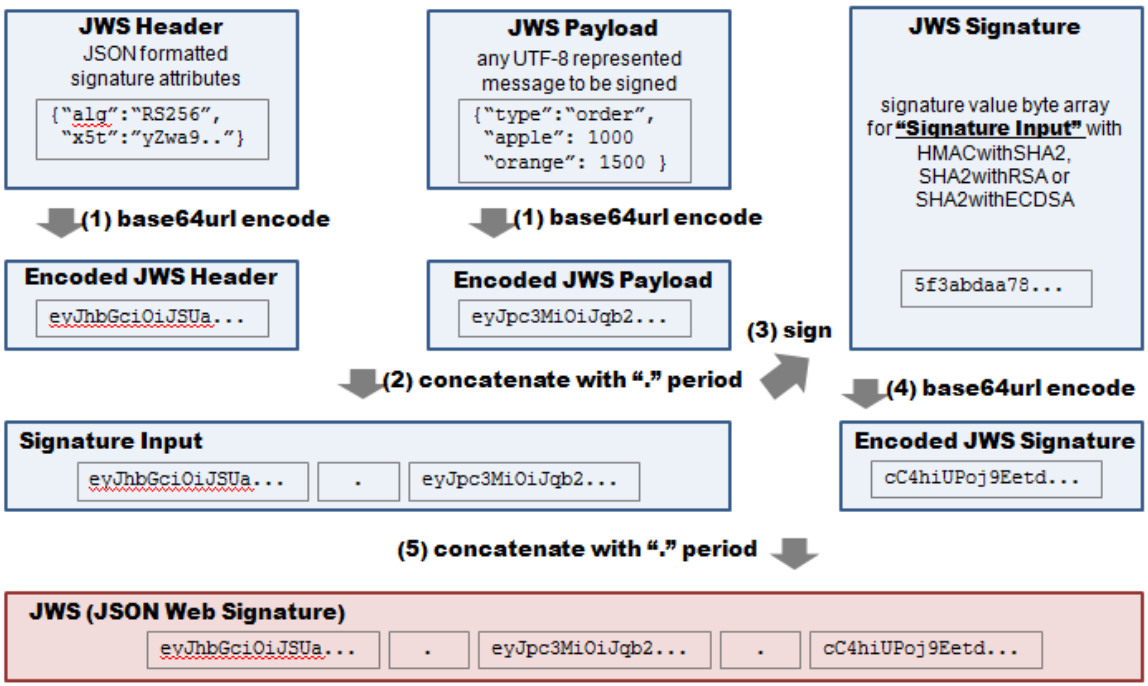
Javascript Object Signing and Encryption (JOSE)

Conjunto de RFC's que padronizam a troca de informações de forma segura através de objetos JSON.



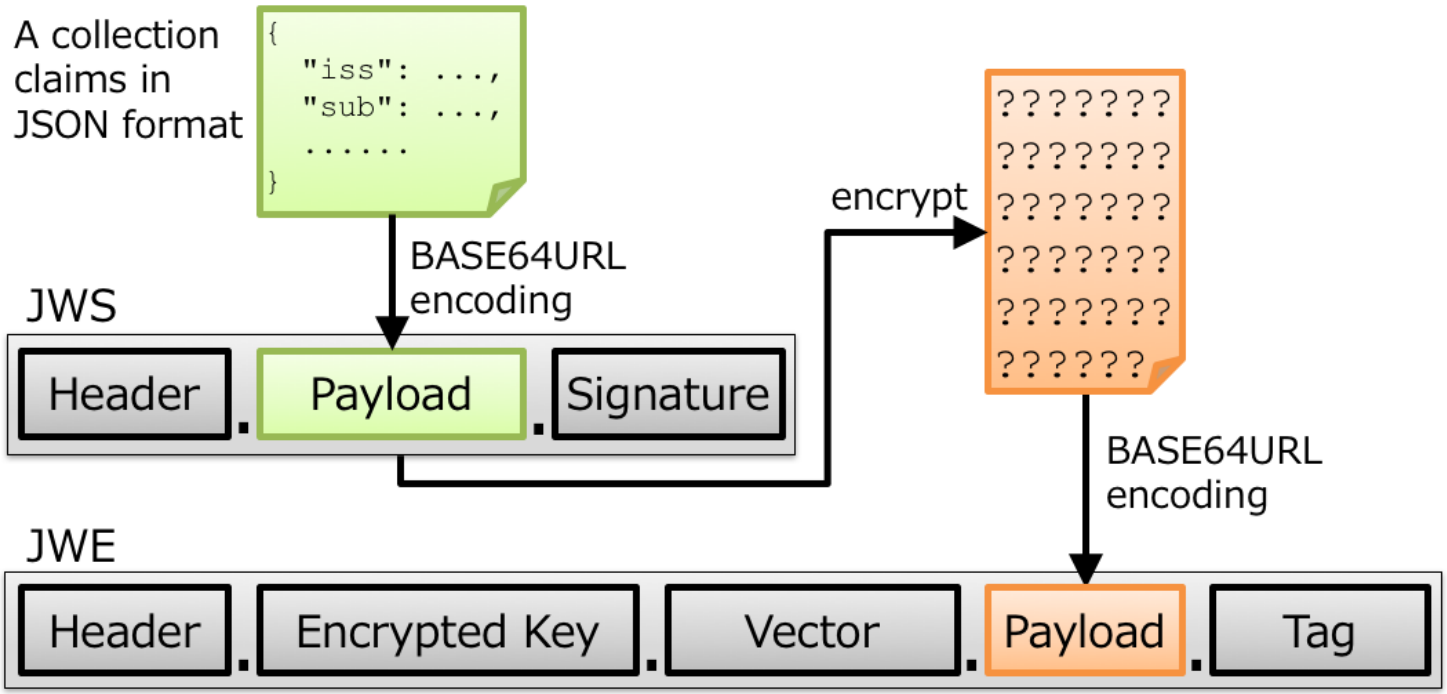
JSON Web Signature (JWS)

Padronizado pela RFC 7515 estabelece o padrão para assinatura digital de objetos JSON.



JSON Web Encryption (JWE)

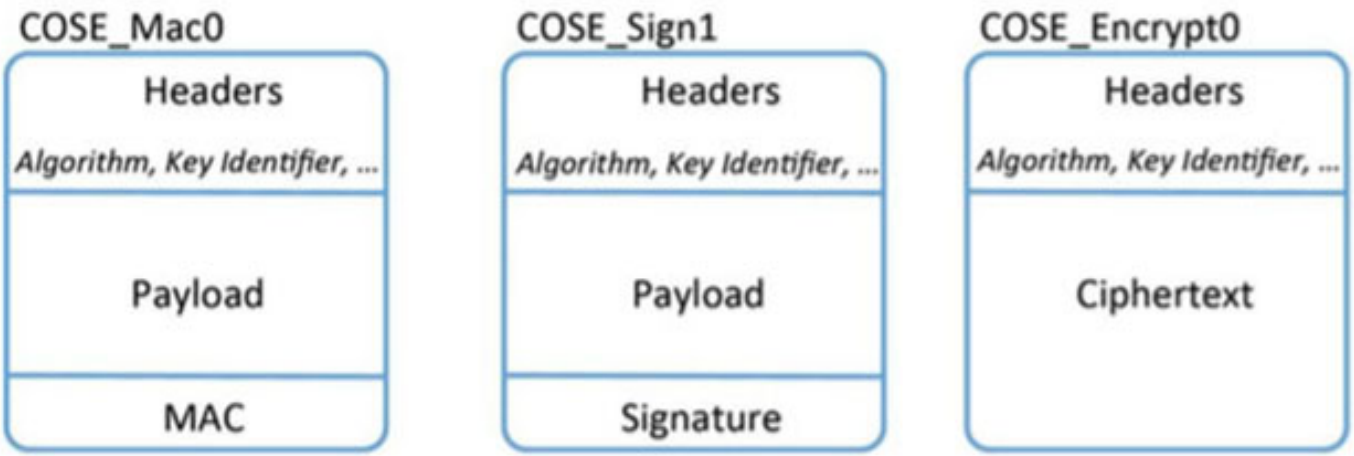
Padronizado pela RFC 7516 estabelece o padrão para a encriptação de conteúdo em objetos JSON.



JSON Web Tokens (JWT) - RFC 7519

CBOR Object Signing and Encryption (COSE)

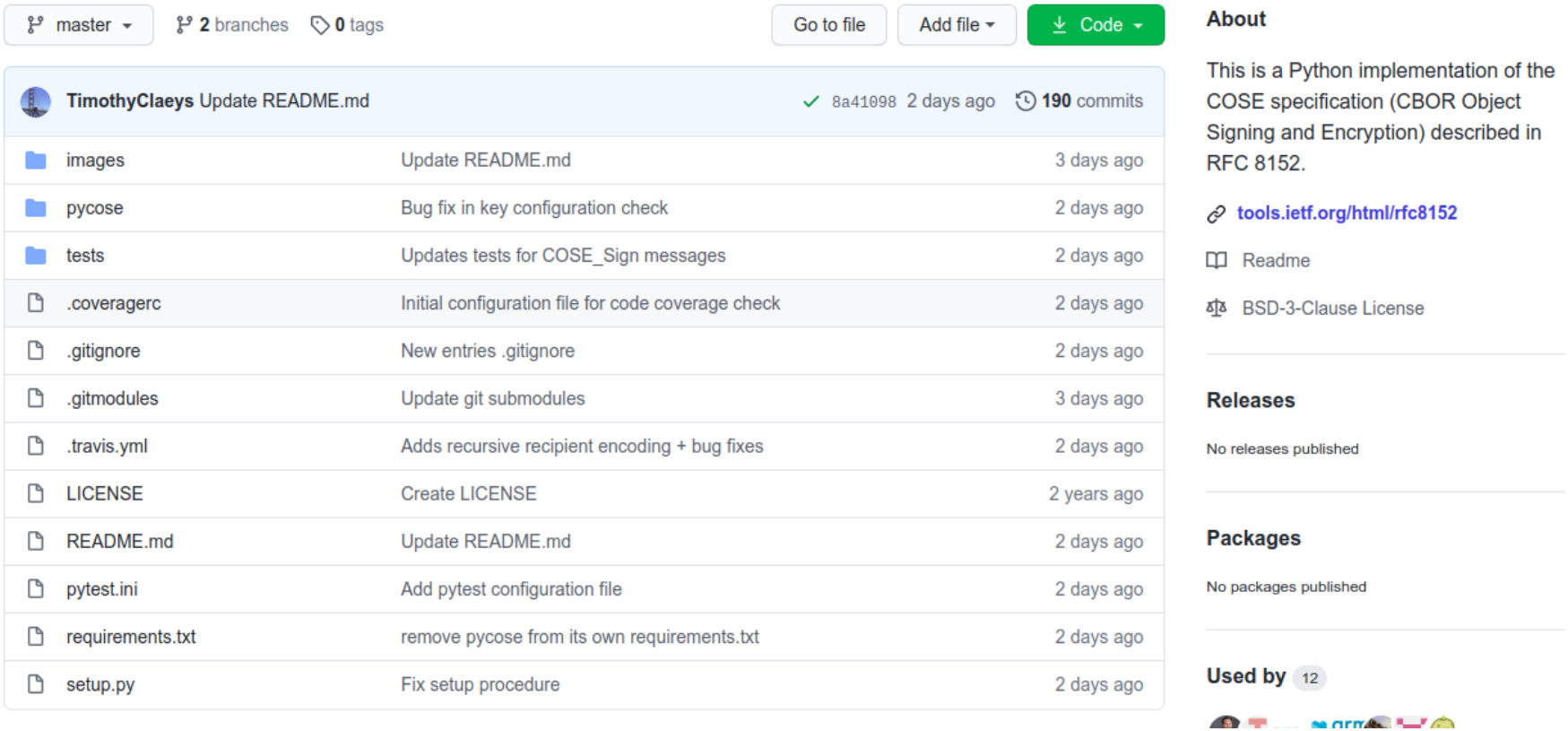
Padronizado pela RFC 8152 estabelece os padrões para encriptar e assinar digitalmente objetos CBOR.



COSE - Definições

Name	Value	L	M	k	Description
AES-CCM-16-64-128	10	16	64	128	AES-CCM mode 128-bit key, 64-bit tag, 13-byte nonce
AES-CCM-16-64-256	11	16	64	256	AES-CCM mode 256-bit key, 64-bit tag, 13-byte nonce
AES-CCM-64-64-128	12	64	64	128	AES-CCM mode 128-bit key, 64-bit tag, 7-byte nonce
AES-CCM-64-64-256	13	64	64	256	AES-CCM mode 256-bit key, 64-bit tag, 7-byte nonce
AES-CCM-16-128-128	30	16	128	128	AES-CCM mode 128-bit key, 128-bit tag, 13-byte nonce
AES-CCM-16-128-256	31	16	128	256	AES-CCM mode 256-bit key, 128-bit tag, 13-byte nonce
AES-CCM-64-128-128	32	64	128	128	AES-CCM mode 128-bit key, 128-bit tag, 7-byte nonce
AES-CCM-64-128-256	33	64	128	256	AES-CCM mode 256-bit key, 128-bit tag, 7-byte nonce

PYCOSE - Biblioteca Python



master · 2 branches · 0 tags · Go to file · Add file · Code

TimothyClaeys Update README.md — 8a41098 2 days ago — 190 commits

images	Update README.md	3 days ago
pycose	Bug fix in key configuration check	2 days ago
tests	Updates tests for COSE_Sign messages	2 days ago
.coveragerc	Initial configuration file for code coverage check	2 days ago
.gitignore	New entries .gitignore	2 days ago
.gitmodules	Update git submodules	3 days ago
.travis.yml	Adds recursive recipient encoding + bug fixes	2 days ago
LICENSE	Create LICENSE	2 years ago
README.md	Update README.md	2 days ago
pytest.ini	Add pytest configuration file	2 days ago
requirements.txt	remove pycose from its own requirements.txt	2 days ago
setup.py	Fix setup procedure	2 days ago

About

This is a Python implementation of the COSE specification (CBOR Object Signing and Encryption) described in RFC 8152.

tools.ietf.org/html/rfc8152

Readme

BSD-3-Clause License

Releases

No releases published

Packages

No packages published

Used by 12

Referências

RFC 4627. 2006. **The application/json Media Type for JavaScript Object Notation (JSON)** *IETF Network Working Group.*

JSON.org. *Acessado em Ago 2020. Disponível em: https://www.json.org/*

Seva Safris. 2020. **A Deep Look at JSON vs. XML, Part 1: The History of Each Standard** *Disponível em: https://www.toptal.com/web/json-vs-xml-part-1*

RFC 7049. 2013. **Concise Binary Object Representation (CBOR)** *Internet Engineering Task Force (IETF).*

CBOR.io. *Acessado em Ago 2020. Disponível em: https://cbor.io/*

Matt Vollrath. 2019. **Extensible Binary Encoding with CBOR** *Disponível em: https://www.endpoint.com/blog/2019/03/18/extensible-binary-encoding-with-cbor*

+ Referências

Josef Sipek. 2017. **CBOR vs. JSON vs. libnvpair** *Disponível em: https://blahg.josefsipek.net/?p=579*

RFC 7515. 2015. **JSON Web Signature (JWS)** *IETF Network Working Group.*

RFC 7516. 2015. **JSON Web Encryption (JWE)** *IETF Network Working Group.*

RFC 7519. 2015. **JSON Web Token (JWT)** *IETF Network Working Group.*

JPrabath Siriwardena. 2016. **JWT, JWS and JWE for Not So Dummies! (Part I)** *Disponível em: https://medium.facilelogin.com/jwt-jws-and-jwe-for-not-so-dummies-b63310d201a3*

JWT.io. *Acessado em Ago 2020. Disponível em: https://jwt.io/*