BUFR with ecCodes

Enrico Fucile

Enrico.Fucile@ecmwf.int

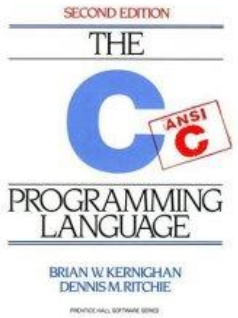
ECMWF

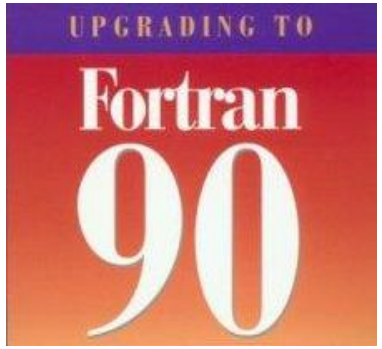
What is ecCodes?

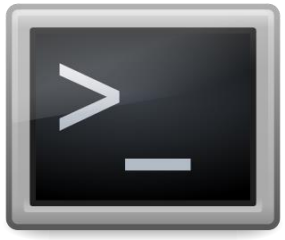
= GRIB-API (GRIB decoder) + BUFR decoder + GTS headers decoder

+ METAR decoder + ???? decoder

Languages and tools







bufr_dump
bufr_filter
bufr_ls
bufr_get
bufr_copy

WMO Binary Codes

Regulations. Setting the rules for encoding/decoding by using external tables.

Notes. Specifying and clarifying special cases.

Tables. External tables to be used by the decoder/encoder software.

No changes to the software are needed to use newly released tables

Implements

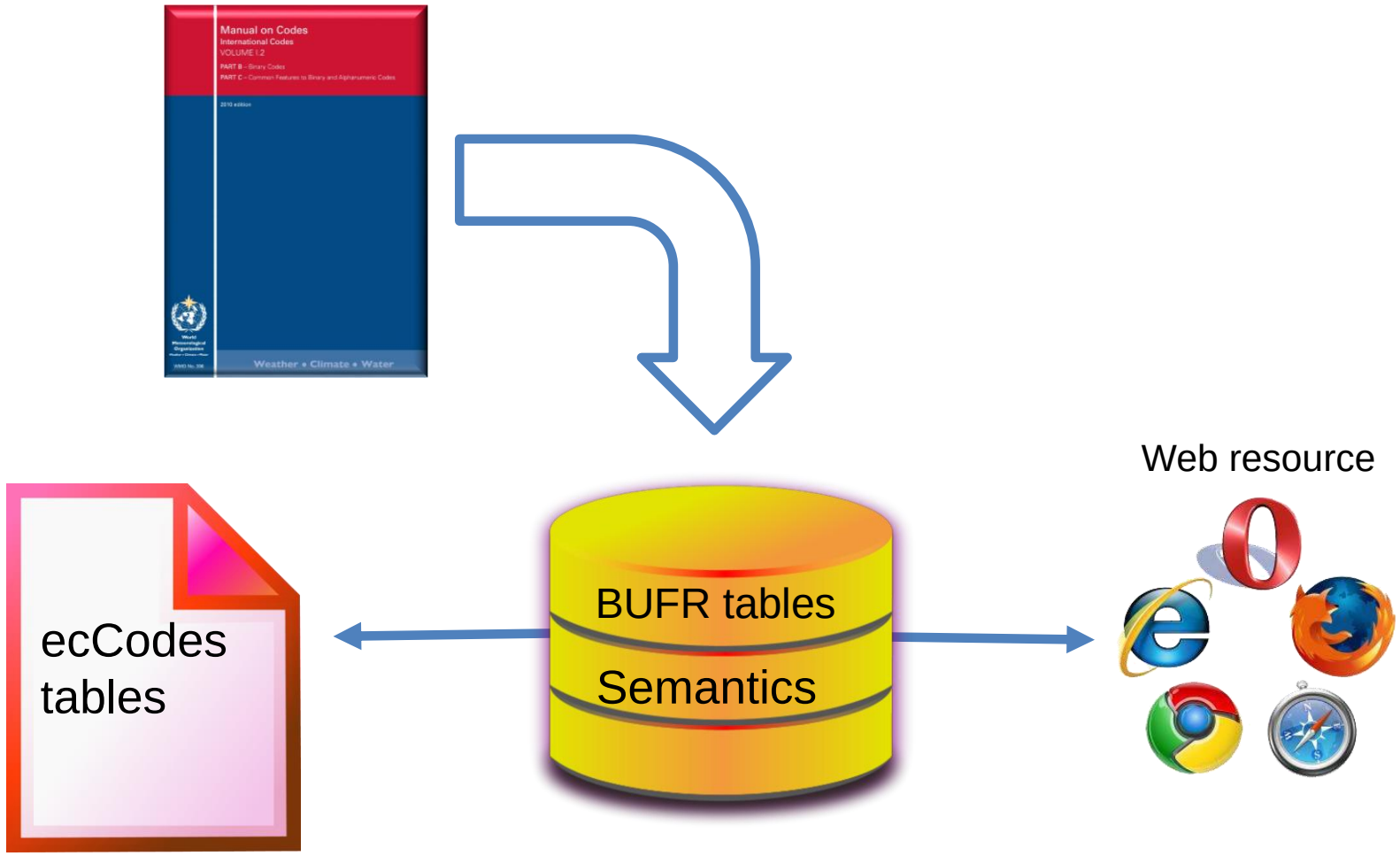
CODES

uses for decoding/encoding

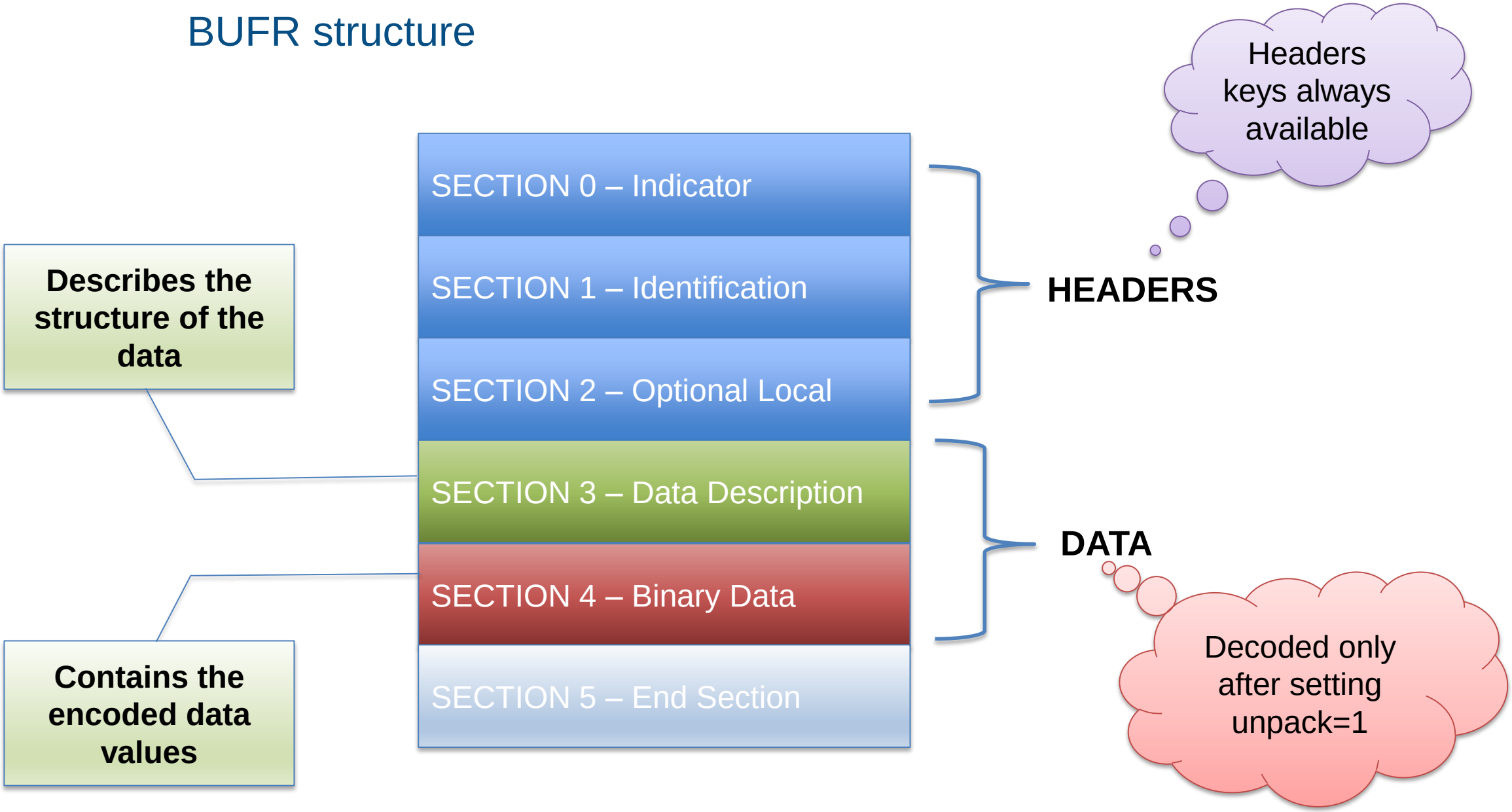
External text files

tables

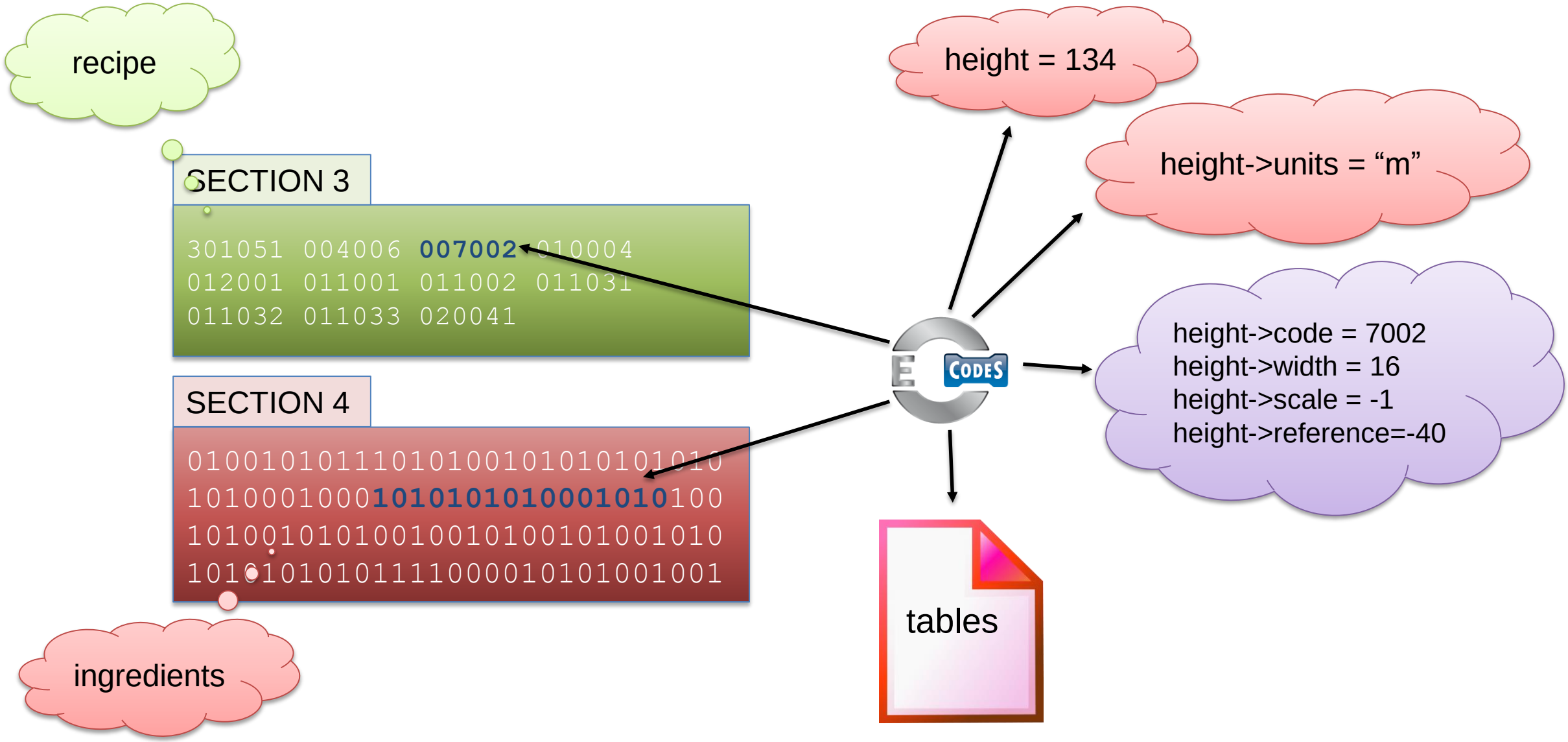
Vocabulary of key names from BUFR tables



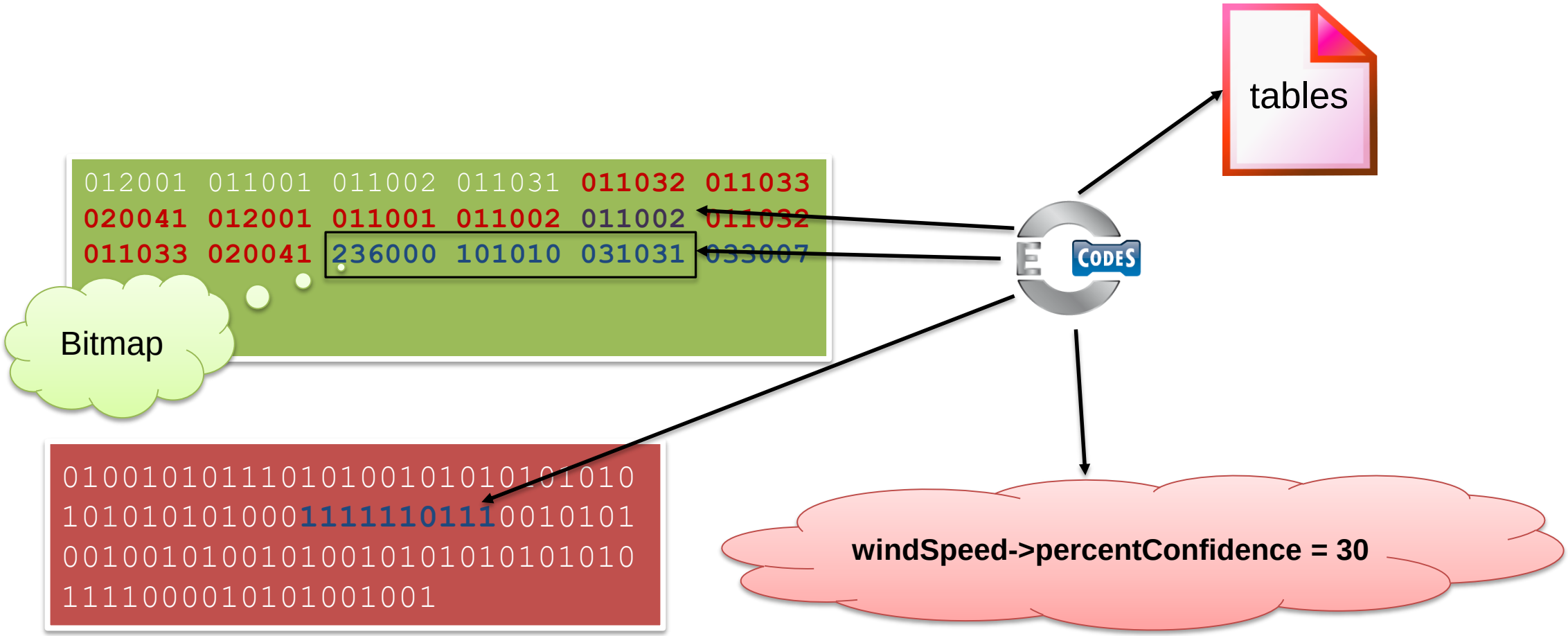
BUFR structure



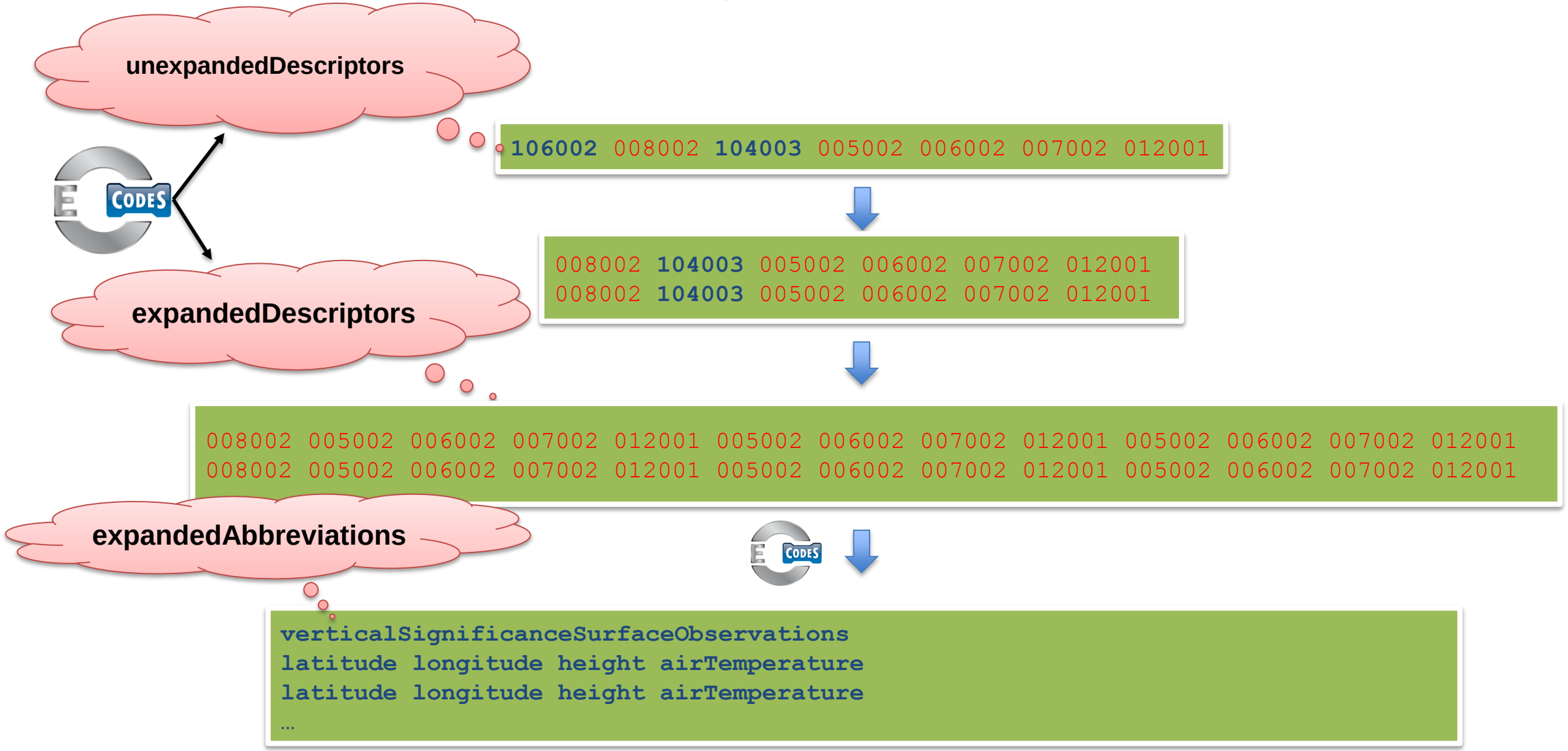
BUFR data


recipe
SECTION 3
301051 004006 007002 010004
012001 011001 011002 011031
011032 011033 020041
SECTION 4
01001010111010100101010101010
10100010001010101010001010100
10100101010010010100101001010
10101010101111000010101001001
ingredients
CODES
height = 134
height->units = “m”
height->code = 7002
height->width = 16
height->scale = -1
height->reference=-40
tables

BUFR bitmap and quality information



BUFR replication (descriptors starting with 1)



BUFR uncompressed data and subsets

SECTION 3

- numberOfSubsets
- observedData
- compressedData

```
unexpandedDescriptors = {301051 004006 007002
010004 012001 011001 011002 011031 011032
011033 020041}
```

numberOfSubsets= 2
compressedData=0

SECTION 4

```
01001010111010100101010101010100010001010
01010100010101001010010101001001010010100 1
01010101010101111000010101001001010001101 0
01010100010101010100101010101001010101101 0
10101010101010101010100101010101010101010 1
01010101010101010101010101010101010101010 1
00101000101001001101010110101010100101010 1
01010101010001010010100 1
```

subsetNumber=1
…
airTemperature=301
…

subsetNumber=2
…
airTemperature=305.2
…

BUFR compressed data and subsets

SECTION 3

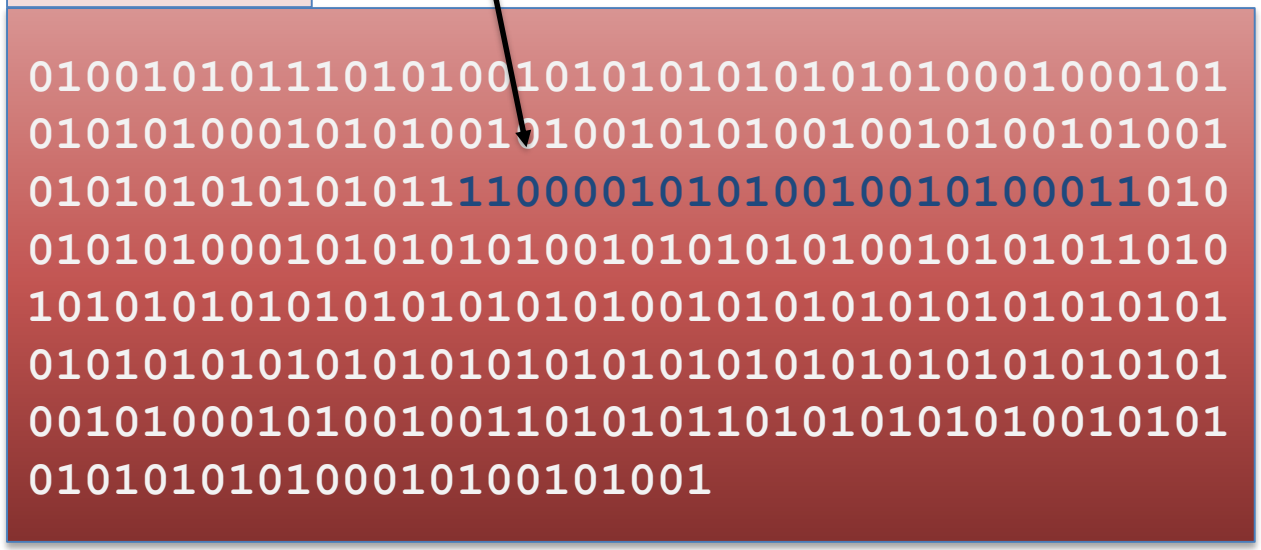
- numberOfSubsets
- observedData
- compressedData

unexpandedDescriptors = {301051 004006 007002 010004 012001 011001 011002 011031 011032 011033 020041}

numberOfSubsets= 2
compressedData=1

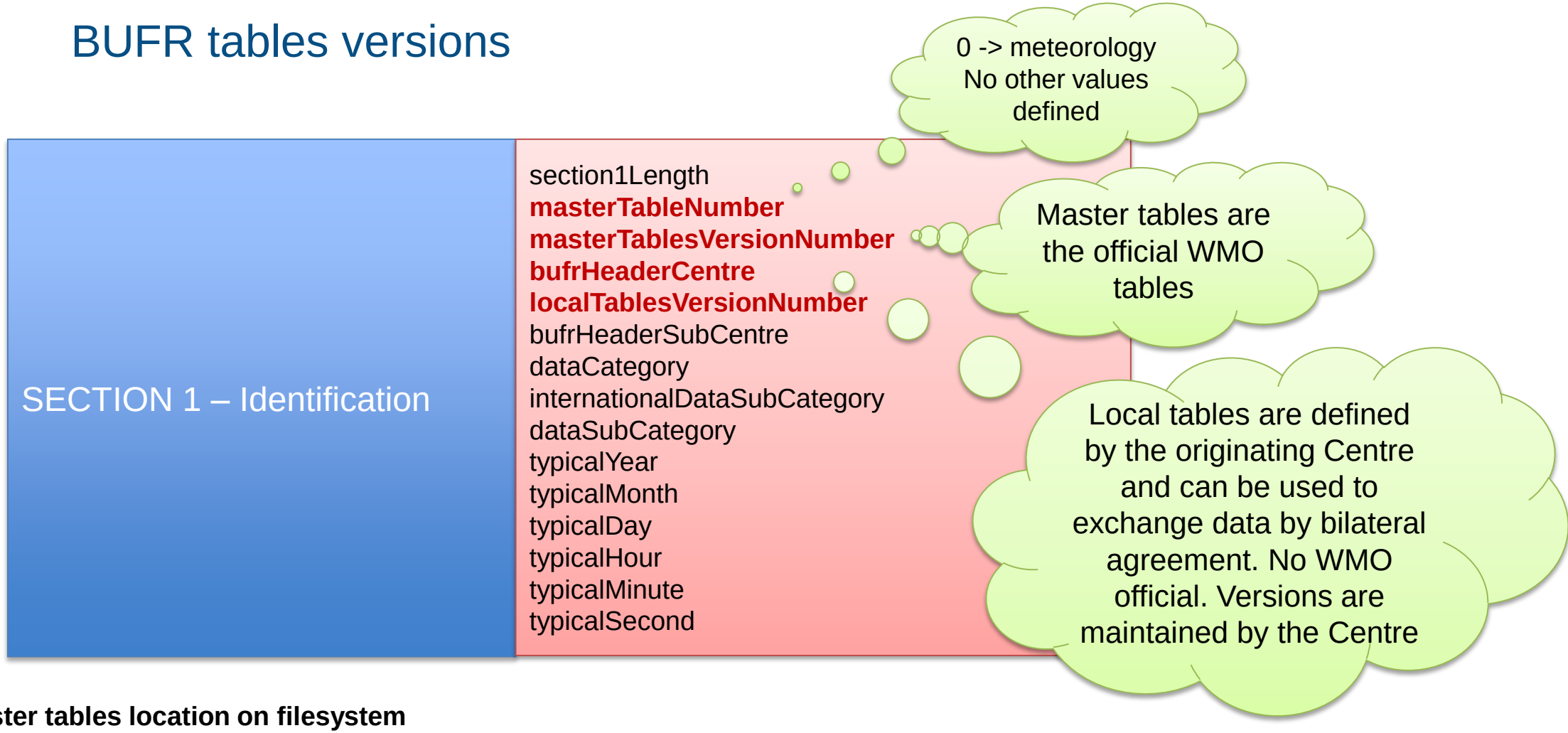
Subset 1 and 2

SECTION 4



…
airTemperature={301, 305.3}
…

BUFR tables versions



WMO master tables location on filesystem
definitions/bufr/tables/[masterTableNumber]/wmo/[masterTablesVersionNumber]

Local tables location on filesystem
definitions/bufr/tables/[masterTableNumber]/local/[localTablesVersionNumber]/[bufrHeaderCentre:l]/[bufrHeaderSubCentre]

Special BUFR Table B descriptors

Element descriptors corresponding to the following classes in Table B shall remain in effect until superseded by redefinition:

X (class)

01	Identification
02	Instrumentation
03	Reserved
04	Location (time)
05	Location (horizontal – 1)
06	Location (horizontal – 2)
07	Location (vertical)
08	Significance qualifiers
09	Reserved

```
[
      {
       "key" : "latitude",
       “code" : “005002”
      },
      [
       {
        "key" : "longitude",
        “code" : “006002”
       },
       [
        {
         "key" : "height",
         “code" : “007002”
        },
        {
         "key" : "airTemperature",
         “code" : “12001”
        }
       ]
      ]
 ],
 [
      {
       "key" : "latitude",
       “code" : “005002”
      },
      …
 ]
```

JSON output from bufr_dump


```
[  {"key" : "beamIdentifier",
    "value" : 1,
    "units" : "CODE TABLE" },
   [ {"key" : "radarIncidenceAngle",
      "value" : [...],
      "units" : "deg"},
     [ { "key" : "antennaBeamAzimuth",
        "value" : [...],
        "units" : "deg"},
       {"key" : "backscatter",
        "value" : [...],
        "units" : "dB"},
        ...
   ],
  [  {"key" : "beamIdentifier",
       "value" : 2,
       "units" : "CODE TABLE" },
       ...
```

Access by rank (python example)

```
x=codes_get(bufr,'#2#backscatter')

xu=codes_get(bufr,'#2#backscatter->units')
```

Access by condition (python example)

```
xc=codes_get(bufr,'/beamIdentifier=2/backscatter')
```

```
xc=codes_get(bufr,'/beamIdentifier=2/backscatter->units')
```

BUFR tools in ecCodes

Similar tools as for GRIB. Very few changes in behaviour.

- **bufr_dump** (different from GRIB as the output is JSON, different options)
- **bufr_filter**
- **bufr_compare**
- **bufr_copy**
- **bufr_get**
- **bufr_set**
- **bufr_ls** (same as for GRIB, but less effective due to the complexity of the message)

ecCodes release

- Starting with IFS cycle 43r1 is used in the operations for GRIB processing.
- Migration of the BUFR encoding decoding software is in progress.
- Version 2.6.0 is the latest release
https://software.ecmwf.int/wiki/display/ECC/Releases
- Documentation available from the project home page
https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home