La couche présentation : ASN.1 et BER

Abstract Syntax Notation #1 Et Basic Encoding Rules

Mickaël Hoerdt

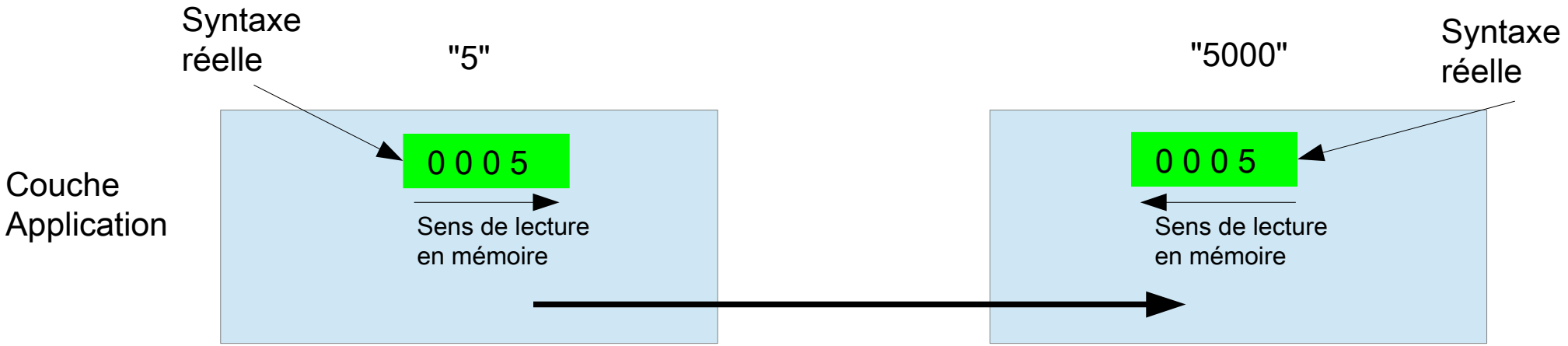
HEPIA

Contenu

- Introduction sur la couche présentation
- Le langage de description de syntaxe ASN.1
- Encodage de la syntaxe de représentation des données avec Basic Encoding Rule (BER)

Introduction à la couche de présentation (1)

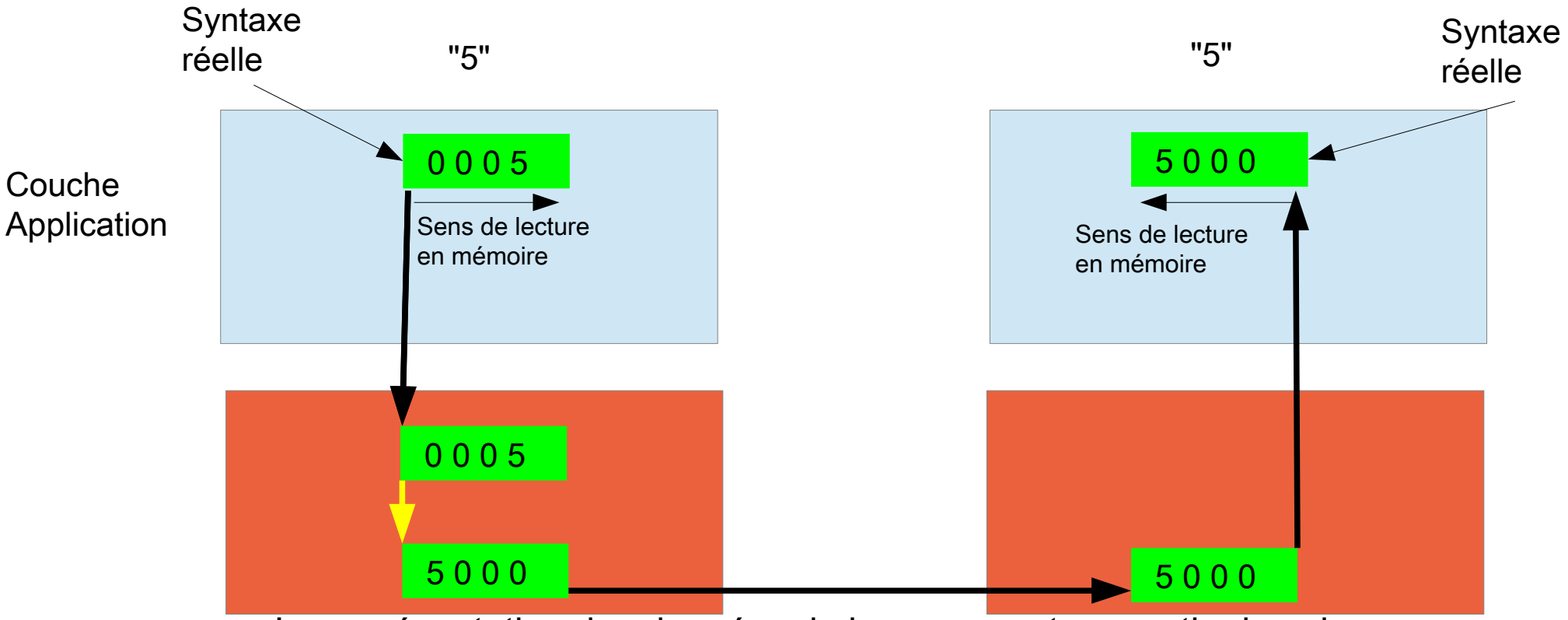
- Sans la couche de présentation



  - La représentation interne des données est différente
  - Ce qui fait que la présentation finale des données est différente aussi

Introduction à la couche de présentation (2)

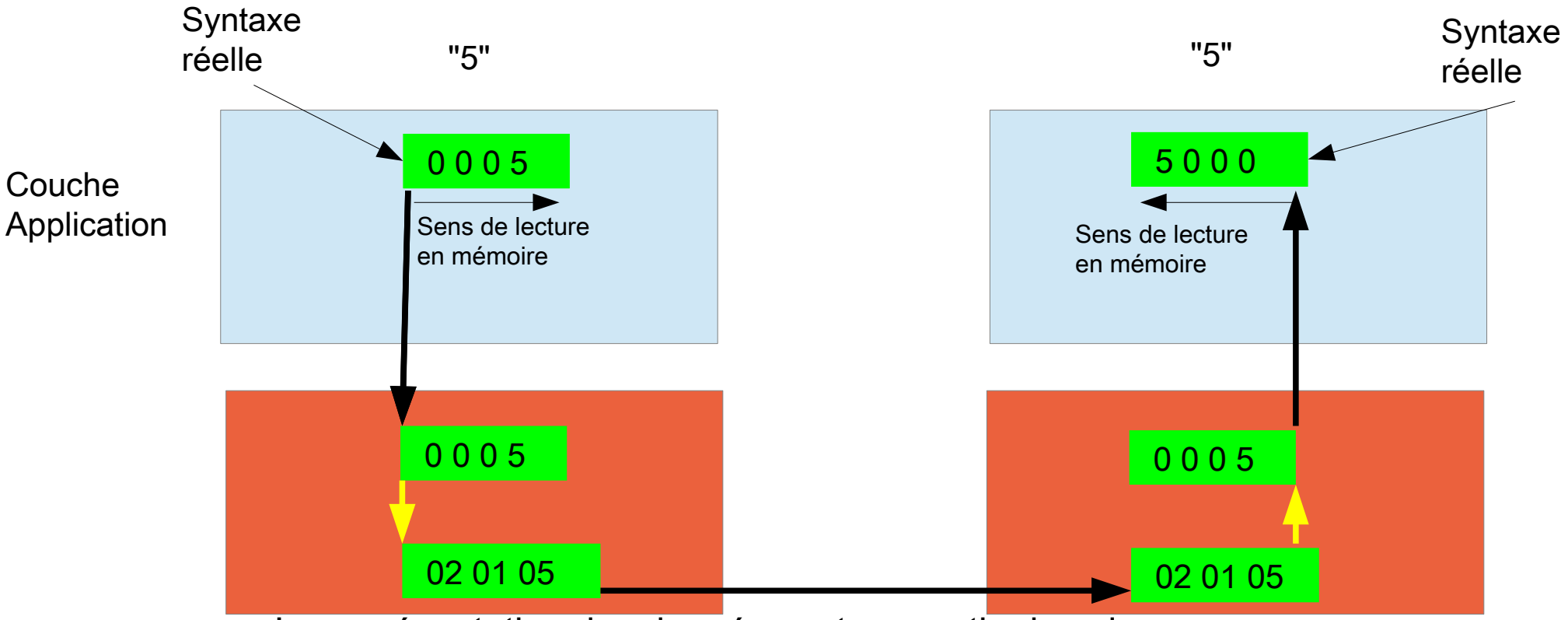
- Avec la couche de présentation



- La représentation des données de la source est convertie dans la représentation des données de la destination
- Les deux syntaxes de représentation doivent être connues de la source et de la destination : le nombre de syntaxes possible est **immense** (autant que d'applications différentes)

Introduction à la couche de présentation (3)

- Avec la couche de présentation et sérialisation



- La représentation des données est convertie dans la représentation des données de transfert (ex : htonl, htons)
- La représentation de transfert est décodée dans la syntaxe de destination (ex : ntohl, ntohs pour représenter les entiers en big-endian)

Introduction à la couche de présentation(4)

- Intérêt d'une syntaxe de représentation de transfert
  - Pas besoin de connaître toutes les syntaxes réelles utilisées par les applications qui communiquent
  - Joue le rôle d'un langage de représentation commun entre les différentes applications pouvant être exécutées sur :
    - Différentes architectures
    - Différents systèmes d'exploitations
    - Différents langages de programmation
  - Permet de décrire les messages d'un protocole de manière unifiée.

Introduction à la couche de présentation(5)

- Fonction de la couche présentation :
  - Fournir à la couche supérieure (la couche application) des données où :
    - La sémantique est conservée
    - La syntaxe de cette sémantique est connue
  - Moyens utilisés :
    - Description des type de données indépendante de l'implémentation (Par exemple ASN.1)
    - Sérialisation des différents type des données utilisés (Par exemple BER)
  - (Faire le lien entre la couche session et la couche application)

Contenu

- Introduction sur la couche présentation
- Le langage de description de syntaxe ASN.1
- Encodage de la syntaxe de représentation des données avec Basic Encoding Rule (BER)

Abstract Syntax Notation (ASN.1)

- Langage utilisé pour décrire de manière abstraite, des données qui sont destinées à être communiquées sur un réseau.
- Est défini par une norme de l'ITU* (X.680, X.681, X.682, X.683)
- Par définition, ce langage est indépendant du
  - Processeur
  - Système d'exploitation
  - Langage de programmation
- Utilisé pour définir les messages de beaucoup de normes de protocoles. Par ex : SMTP, MAP (Mobile Application Part : GSM, UMTS), H323 (VOIP), LDAP, …
- Utilisé pour décrire le format des certificats X509 (RFC2459, RFC3280, RFC5280)*

* https://nostdahl.com/2017/08/11/x-509-certificates-explained/
* http://fr.wikipedia.org/wiki/Union_internationale_des_t%C3%A9l%C3%A9communications

Abstract Syntax Notation (ASN.1)

- Le langage permet de décrire :
  - Les types fondamentaux/universaux
    - INTEGER (entiers)
    - BOOLEAN (booléens)
    - Chaînes de caractères
    - Etc
  - Des types composés et construire des nouveaux types (comme typedef en C):
    - SEQUENCE
    - SET
    - CHOICE
    - Etc

Abstract Syntax Notation (ASN.1)

- Syntaxe de définition de nouveau types :
  - Badge ::= INTEGER
  - Name ::= Printable String
  - Wage ::= REAL
  - Deceased ::= BOOLEAN
  - First-name ::= [APPLICATION 1] Printable String

Abstract Syntax Notation (ASN.1)

- Un exemple un peu plus compliqué

```
Person ::= SEQUENCE {
  familyname Printable String,
  firstname SEQUENCE OF Printable String,
  birthday GENERALIZED TIME,
  gender ENUMERATED {male(0), female(1),
    unknown(2)},
  children [0] SEQUENCE OF Person
    OPTIONAL,
  spouse [1] Person OPTIONAL,
  ...
}
```

Pour éviter l'ambiguité lors de l'encodage des données

Contenu

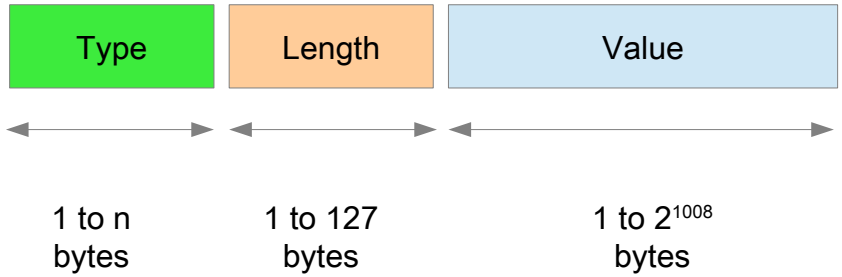
- Introduction sur la couche présentation
- Le langage de description de syntaxe ASN.1
- Encodage de la syntaxe de représentation des données avec Basic Encoding Rule (BER)

Basic Encoding Rule (BER)

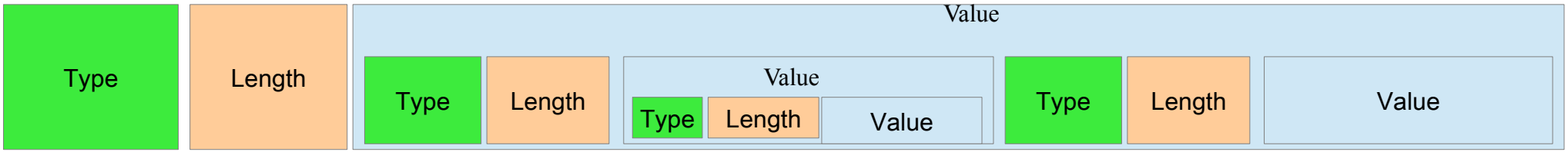
- Est une norme d'encodage de syntaxe de transfert recommandée par l'ITU (X.690)
- Utilisée par beaucoup de protocoles réseaux (SNMP, SET, LDAP, ...)
- Est décrit par un langage standard de description de la syntaxe : ASN.1 et peut donc être encodé/décodé facilement avec un « compilateur/décompilateur » ASN.1 <-> BER

Basic Encoding Rule (BER)

- Principe :
  - Tout les types (fondamentaux ou composés) sont encodés sous forme de message Type, Length, Value (TLV) de longueur variable :



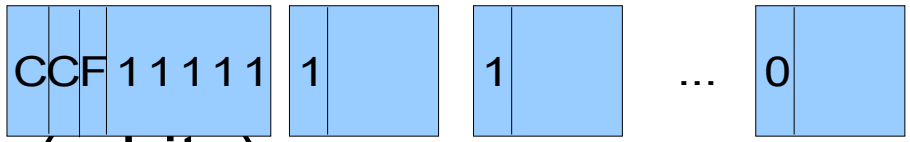
  - Un type composé encapsule d'autres types de la manière suivante :



Basic Encoding Rule : champs type

- Le champs type contient :
  - Un identifiant de classe (2 bits)
    - Universal (00)
    - Application (01)
    - Private (10)
    - Context-specific (11)
  - Un identifiant de forme (1 bit)
    - Atomic (0)
    - Structured (1)
  - Un tag identifiant le type (n bits)

one-byte type field; 0 ≤ tag ≤ 30



n-byte type field; tag ≥ 31

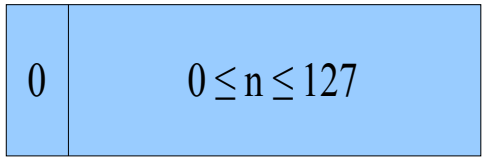
BER : les tags de classe « universal »

```
0   reserved for BER
1   BOOLEAN
2   INTEGER
3   BIT STRING
4   OCTET STRING
5   NULL
6   OBJECT IDENTIFIER
7   ObjectDescriptor
8   INSTANCE OF, EXTERNAL
9   REAL
10  ENUMERATED
11  EMBEDDED PDV
12  UTF8String
13  RELATIVE-OID
16  SEQUENCE, SEQUENCE OF
17  SET, SET OF
18  NumericString
19  PrintableString
20  TeletexString, T61String
21  VideotexString
22  IA5String
23  UTCTime
24  GeneralizedTime
25  GraphicString
26  VisibleString, ISO646String
27  GeneralString
28  UniversalString
29  CHARACTER STRING
30  BMPString
```

IA5String = Chaîne de caractère au format ascii
Pour les autres formats de chaînes de caractères voir
http://www.obj-sys.com/asn1tutorial/node128.html

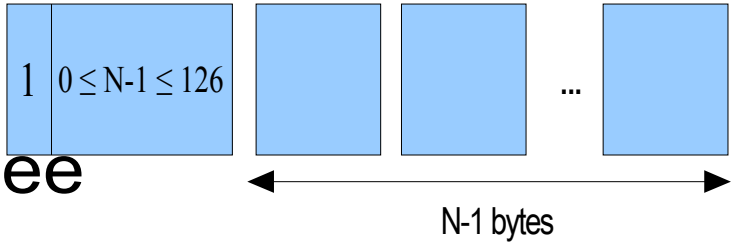
Basic Encoding Rule : champs length

- Si longueur < 127
  - Champs codé sur 1 octet



one-byte field

- Si longueur > 127
  - Champs codé sur n octets
    - N < 127
  - La longueur réelle est codée sur 126 octets au plus
    - i.e longueur max = 2^1008



N-byte field

BER : Exemple

```
Person ::= SET {
        name      IA5String,
        age       INTEGER,
        female    BOOLEAN
}
```

Supposons qu'on veuille encoder une instance du type `Person` avec les valeurs `Maggie`, `4` et `True`

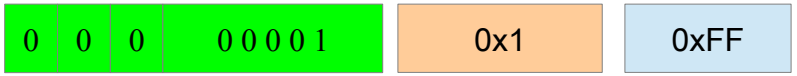
Encodage de Maggie en IA5String, de type Universal, Atomic, Tag 22

= 0x16 0x6 077 0x65 0x71 0x71 0x73 0x69

Encodage de 4 en Integer, de type Universal, Atomic, Tag 2

= 0x02 0x01 0x04

Encodage de True en Boolean, de type Universal, Atomic, Tag 1



= 0x01 0x01 0xFF

Encodage de SET avec les valeurs précédentes, de type Universal, Structured, Tag 17