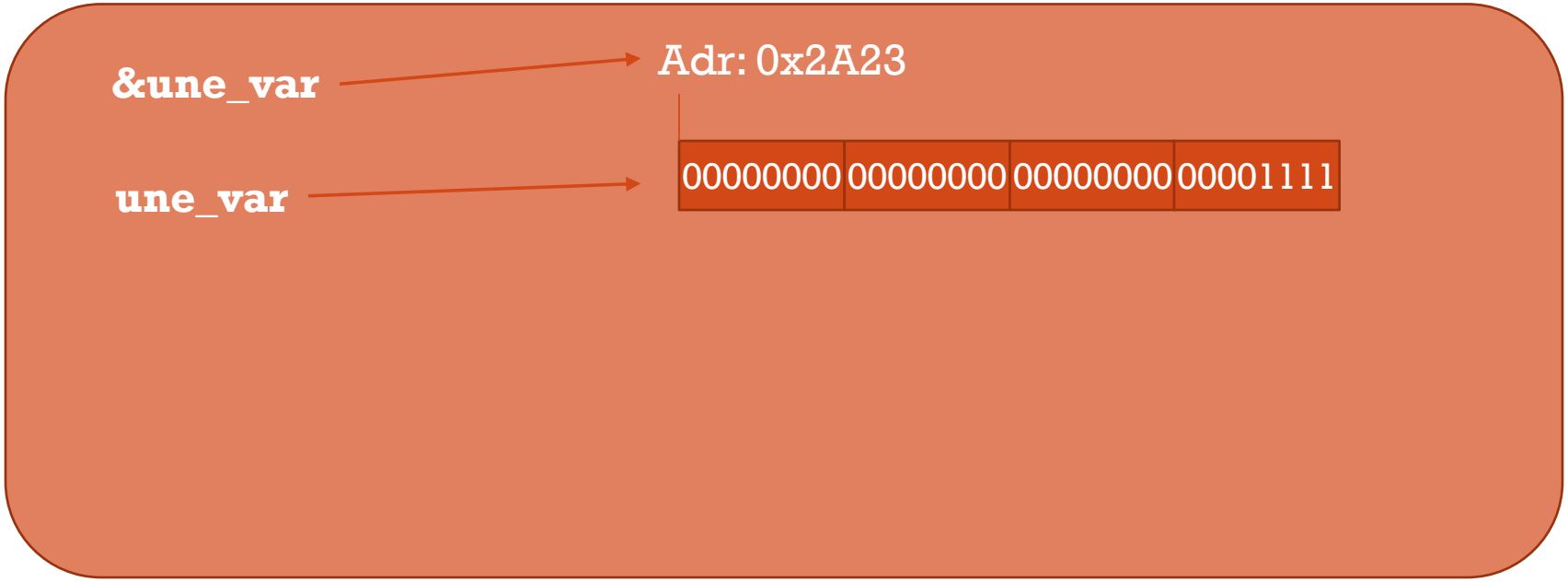
INF155 – RAPPELS
LES POINTEURS

Anis Boubaker, Ph.D.
Maître d'enseignement
École de Technologie Supérieure

ADRESSE D'UNE VARIABLE

int une_var = 15;



La mémoire vive

L'opérateur (unaire!) "&" permet d'obtenir l'adresse d'une variable.

PRINTF ET LES ADRESSES MÉMOIRE

- La fonction ***printf*** prévoit un code de formatage dédié à l'affichage des adresses mémoire en hexadécimal:

%p

- **Exemple:**

printf("L'adresse de la variable *ma_var* est : **%p**", **&**ma_var);

QU'EST-CE QU'UN POINTEUR?

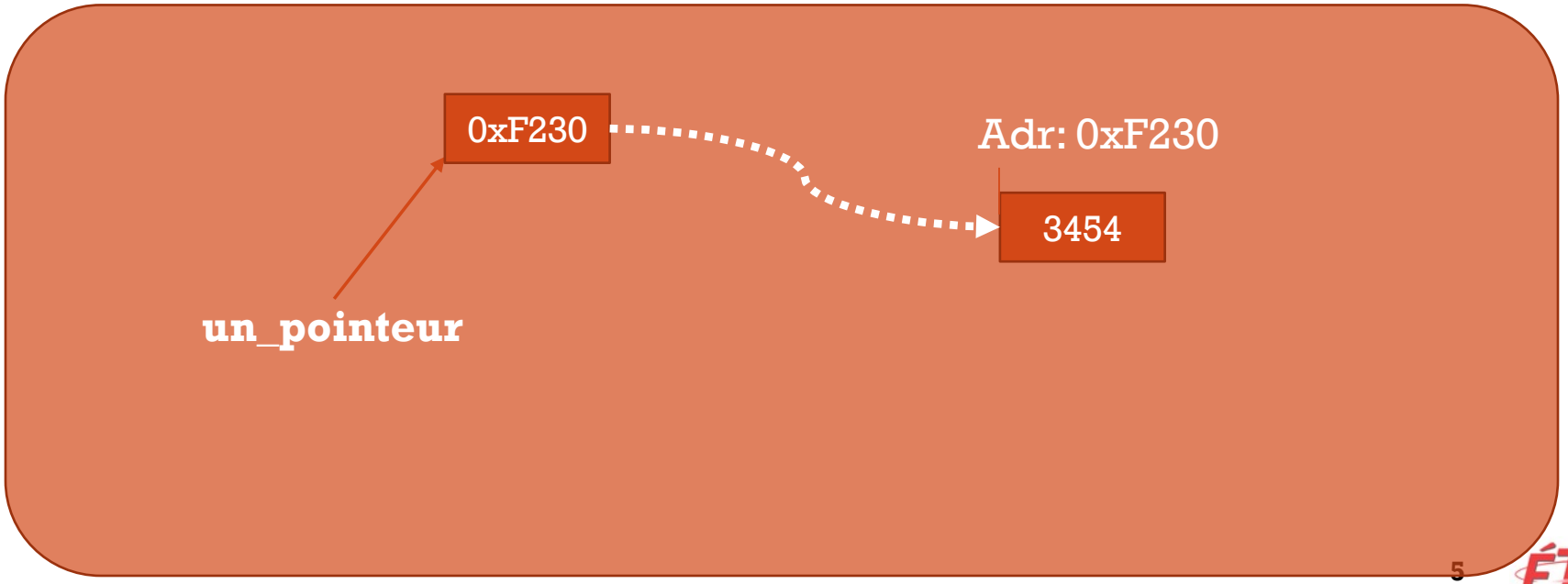
- Un pointeur **est une variable** qui permet de **stocker une adresse mémoire**.

- Comme pour une variable, on peut lui assigner une valeur et on peut modifier cette valeur (mais avec précautions).

- Comme pour une variable, un pointeur doit-être déclaré

POURQUOI UN POINTEUR?

- Un pointeur **est une variable** qui permet de **stocker une adresse mémoire**.
- On utilise les pointeurs dans le but d'accéder "indirectement" à une information stockée en mémoire.



DÉCLARATION D'UN POINTEUR

Syntaxe:

```
type *identifiant[= une_adresse];
```

- **Exemple:**

```
int *pointeur_entier; //Déclare un pointeur sur un entier
```

- La déclaration comprend:
  - Le type de données référencées par le pointeur;
  - Un identifiant, précédé d'un astérisque (*) pour signifier que c'est un pointeur.
  - Une valeur initiale, facultative mais très recommandée! (voir la suite)

VALEUR D'UN POINTEUR

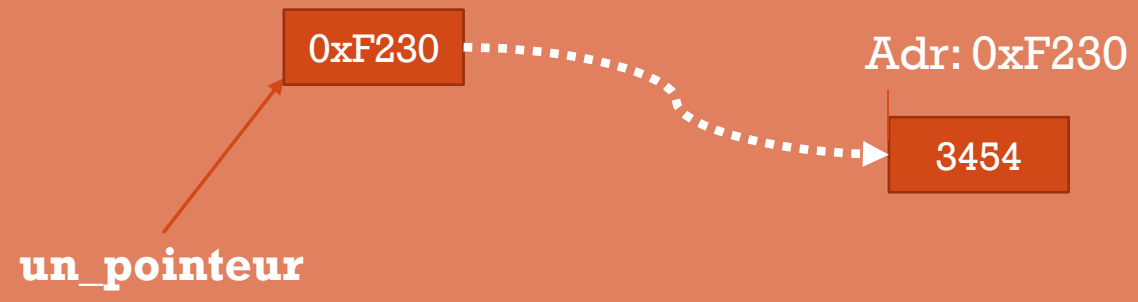
- Un pointeur contient **une adresse mémoire**.
- En règle générale, **on n'affecte pas de valeurs numériques à un pointeur, mais on le calcule à partir de l'adresse d' autres variables.**
- Exemple:

```
int entier = 10 ;
int *pointeur_entier = NULL; //Déclare un pointeur sur un entier

pointeur_entier = &entier; //On affecte l'adresse de entier.
```

DÉRÉFÉRENCEMENT D'UN POINTEUR

- **Déréférencement**: mot compliqué pour dire "accéder à la valeur pointée par le pointeur".



Déréférencer "un_pointeur" veut dire: obtenir la case mémoire référencée par un_pointeur (0xF230) pour lire sa valeur (3454) ou la modifier.

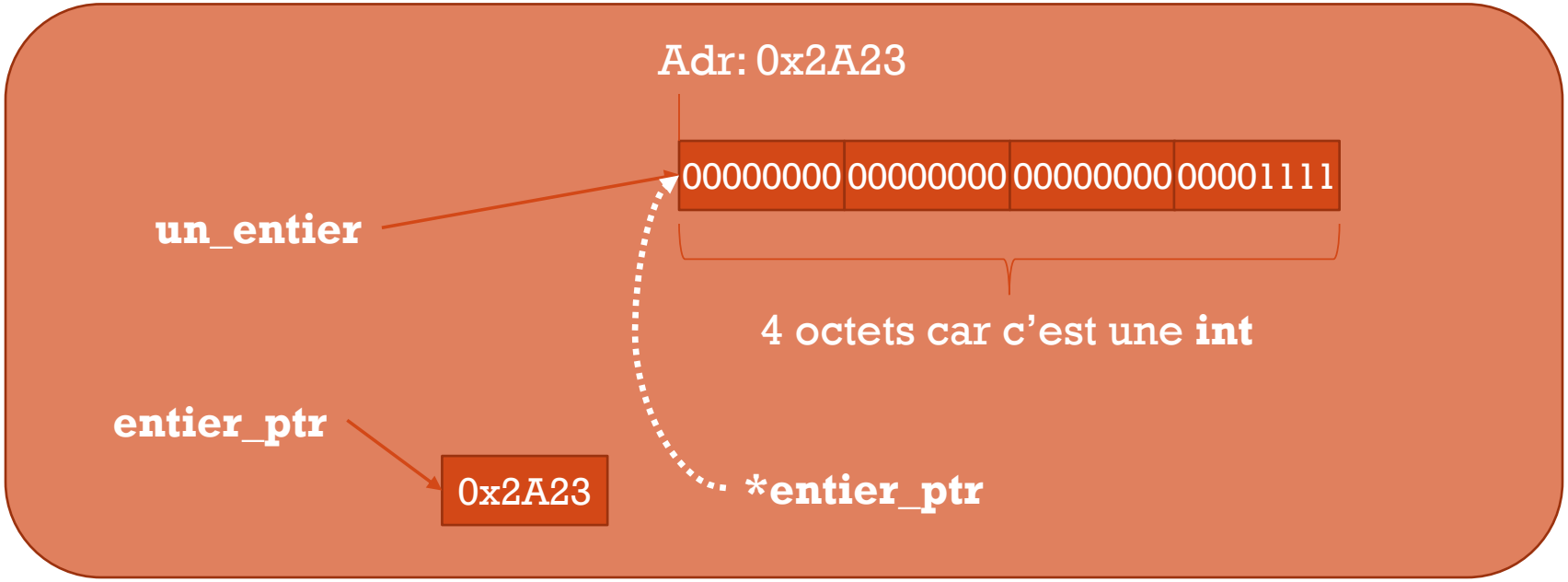
DÉRÉFÉRENCEMENT D'UN POINTEUR

- Pour déréférencer un pointeur, nous utilisons l'opérateur unaire "*" (astérisque).
- Exemple:

```
int un_entier=15;
int *entier_ptr = &un_entier;

//Déréférencement de entier_ptr,
//Affiche "Valeur = 15"
printf("Valeur = %d" , *entier_ptr);
```

DÉRÉFÉRENCEMENT D'UN POINTEUR



- Le déréférencement de entier_ptr (*entier_ptr) permet d'accéder à la valeur stockée à l'adresse référencée par entier_ptr (0x2A23)

- **C'est ici que le type de pointeur intervient:** entier_ptr est un pointeur vers un int ➔ Nous devons lire une valeur de type int (i.e. 4 octets)