



Railway Programming

La voie vers un code plus honnête

Qui sommes-nous?

Romain Berthon

 @RomainTrm

romainberthon.dev

romainberthon.blog



Sylvain Coudert

 @sylv_coud

punkindex.fr

Disclaimer!

Votre code vous ment!

- Effets de bord cachés

- ▢ Exception

```
var myValue = int.Parse(toto);
```

- ▢ Appel externe, mutations

```
public void PayReservation(Reservation reservation, decimal amount)
{
    _paymentProcess.Pay(amount);

    reservation.ReservationState = ReservationState.Validated;
}
```

- Signatures de méthodes malhonnêtes

- ▢ Fonctionnements cachés

- ▢ Types de sortie incomplets ou absent

Votre code vous ment!

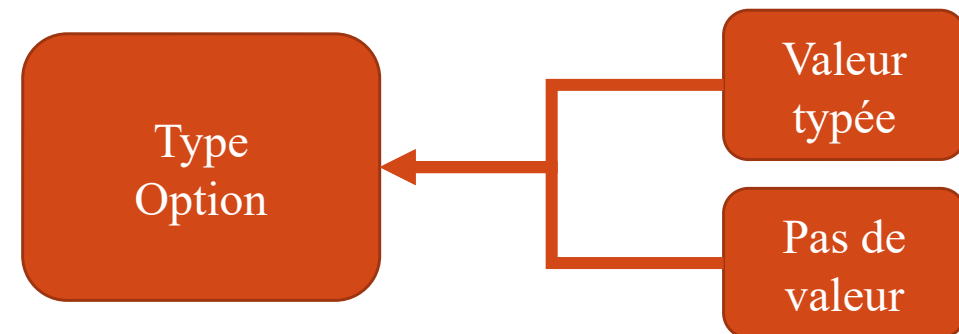
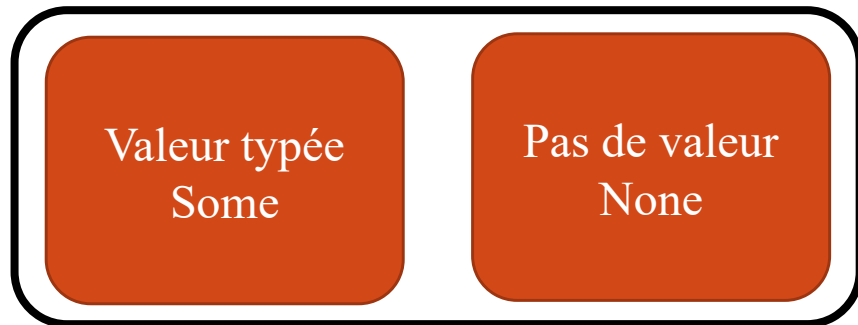
- Une vraie compréhension nécessite de tester ou de voir l'implem
- Les appels et signatures ne suffisent pas...

```
public void ProcessReservation(int userId, Reservation reservation, decimal reservationCost)
{
    if (reservation.ReservationState == ReservationState.Pending)
    {
        PayReservation(reservation, reservationCost);
    }
    if (reservation.ReservationState == ReservationState.Validated)
    {
        SendConfirmationMail(userId);
    }
}
```

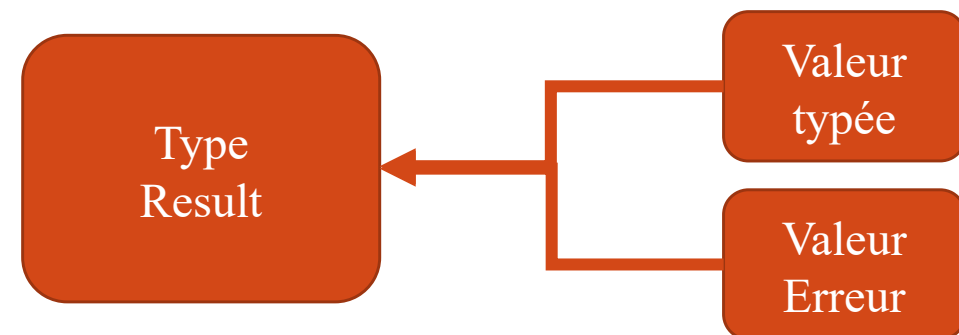
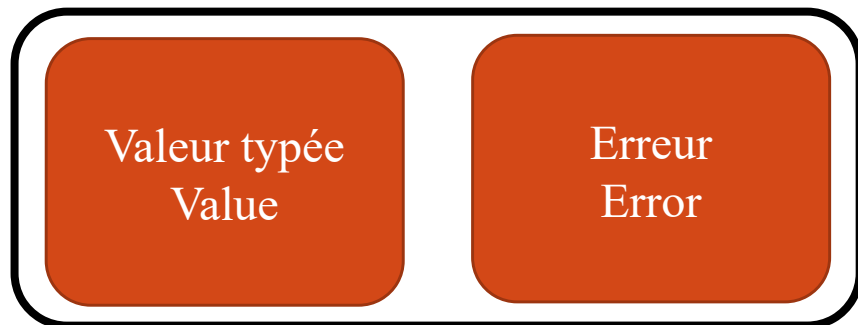
Live coding

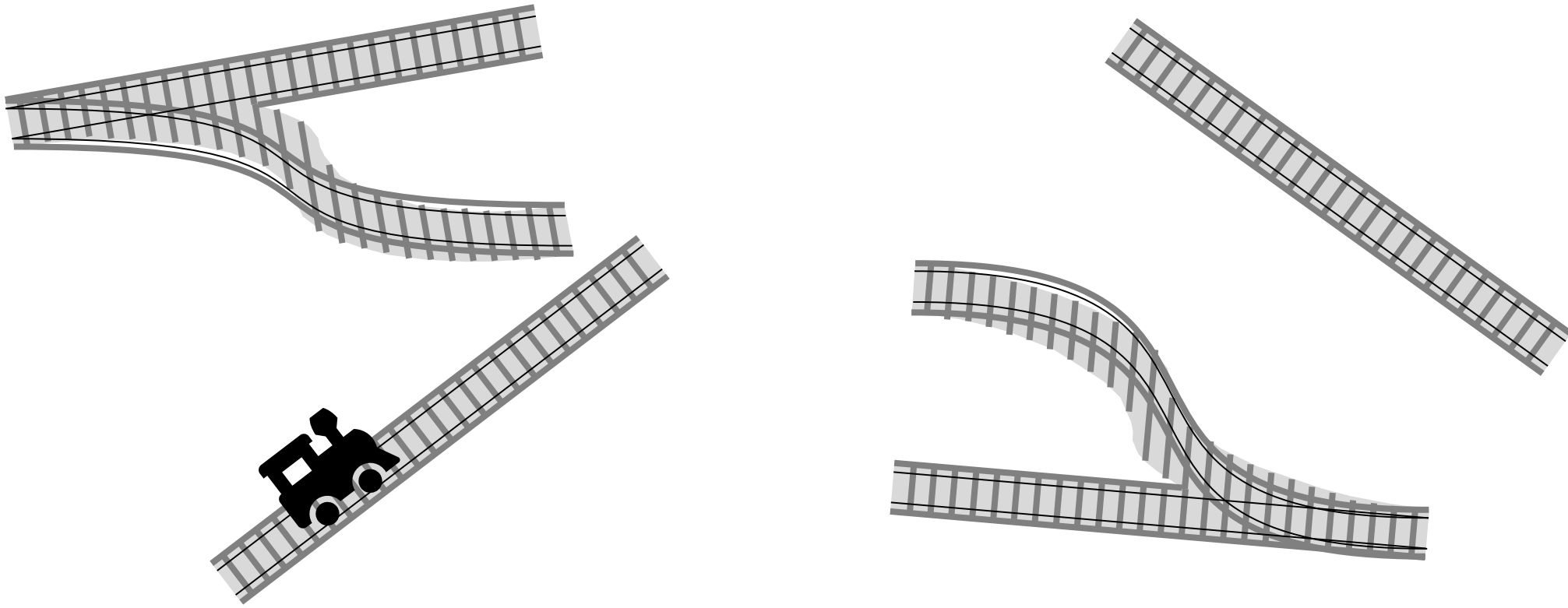
Make the implicit explicit

- Rendre des informations explicites par les types:
 - ▢ Identifier les valeurs optionnelles, le type Option/Maybe



- ▢ Identifier les erreurs potentielles, le type Result



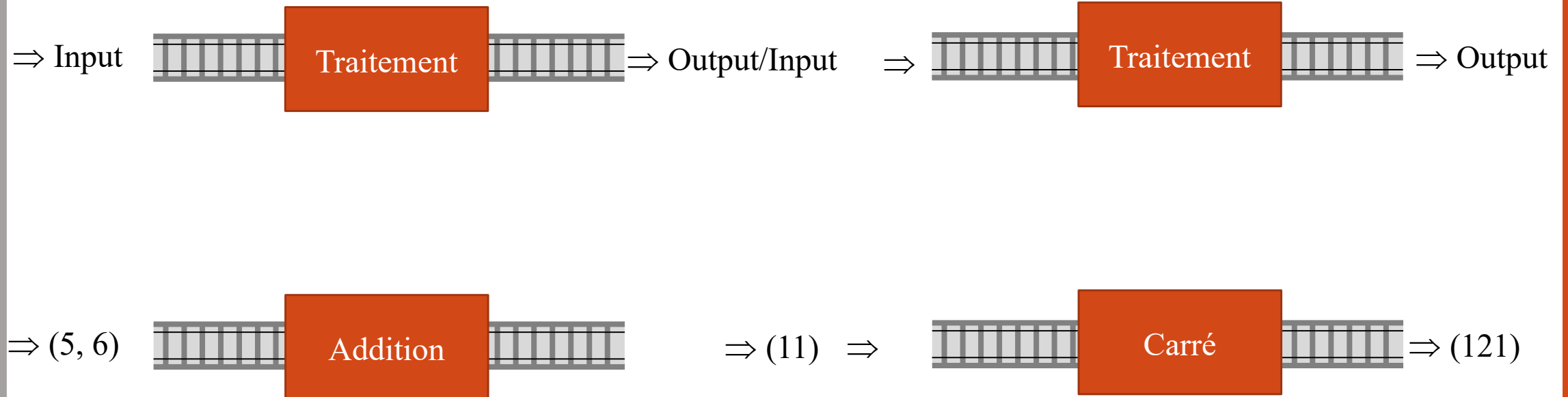


Let's do railways!

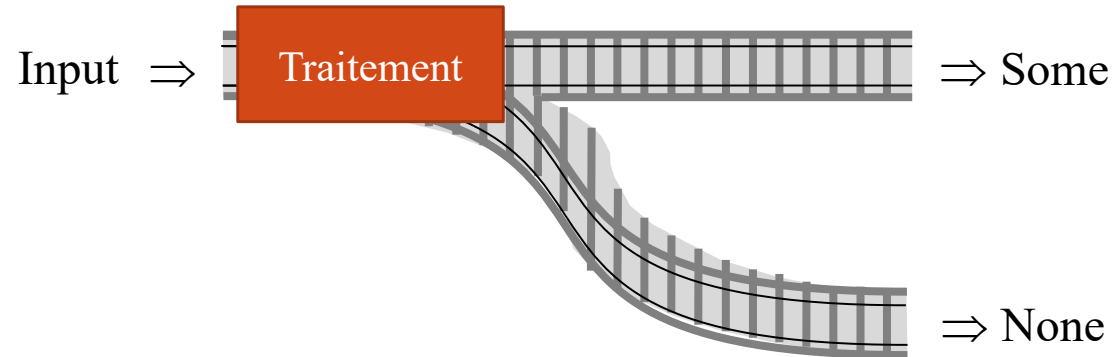
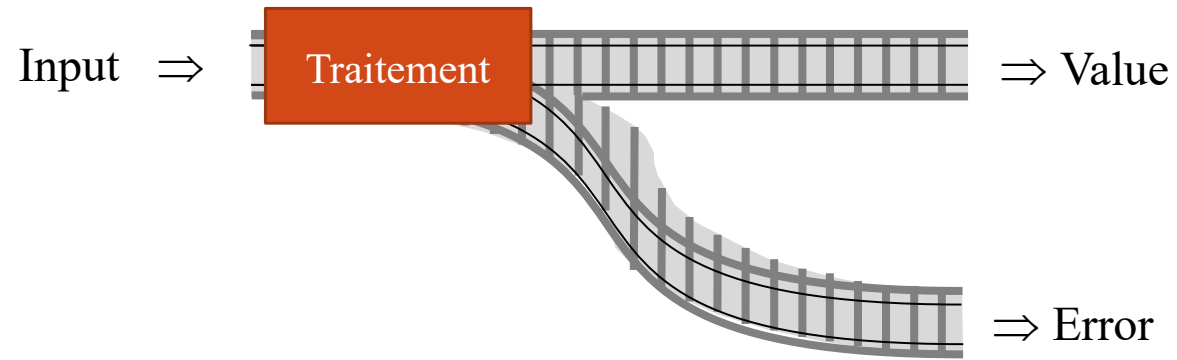
Fonction simple



Composition



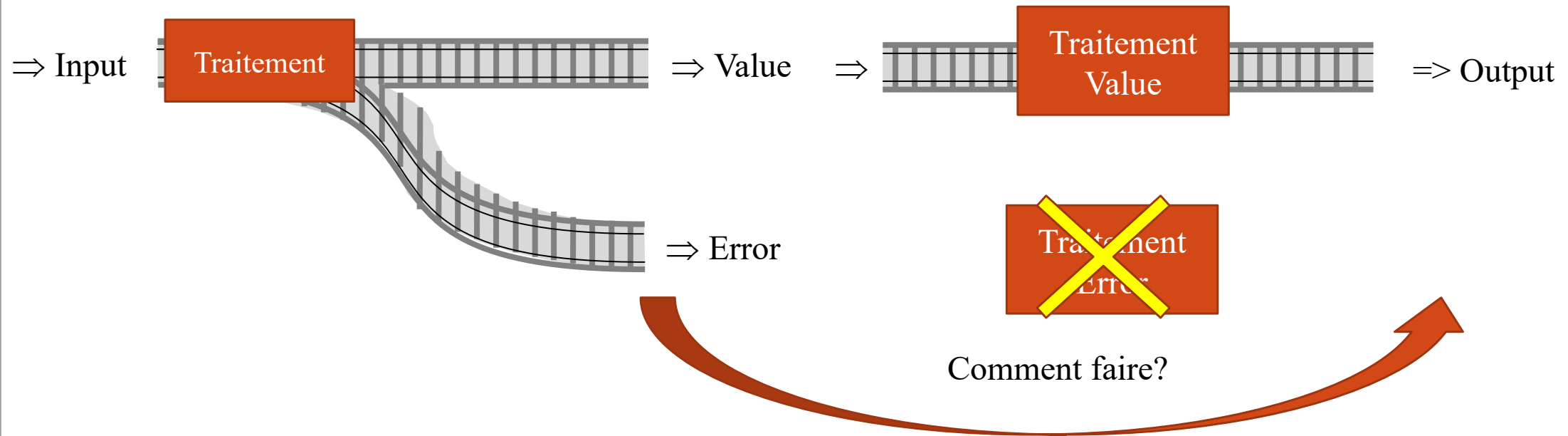
Changeons les Output : Result ou Option



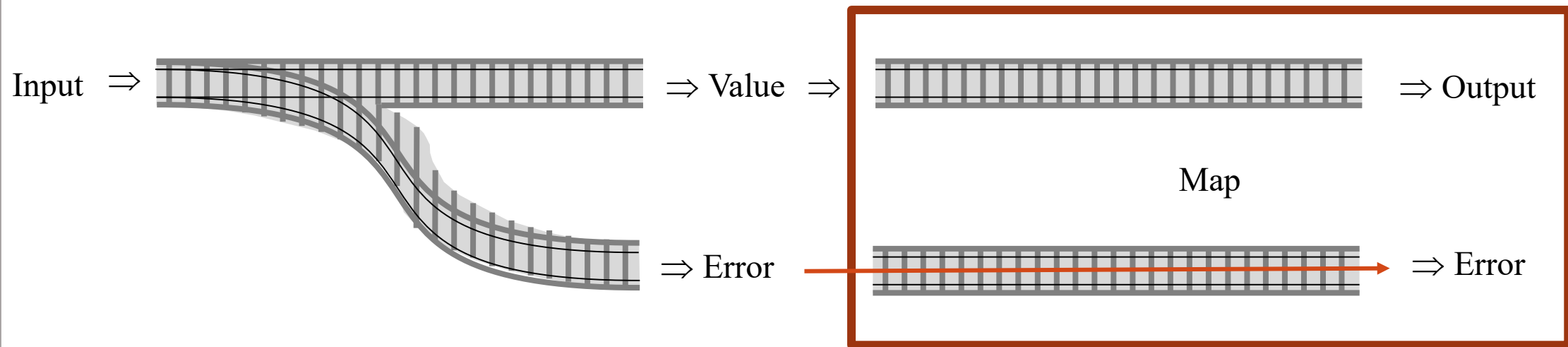
Don't do this!



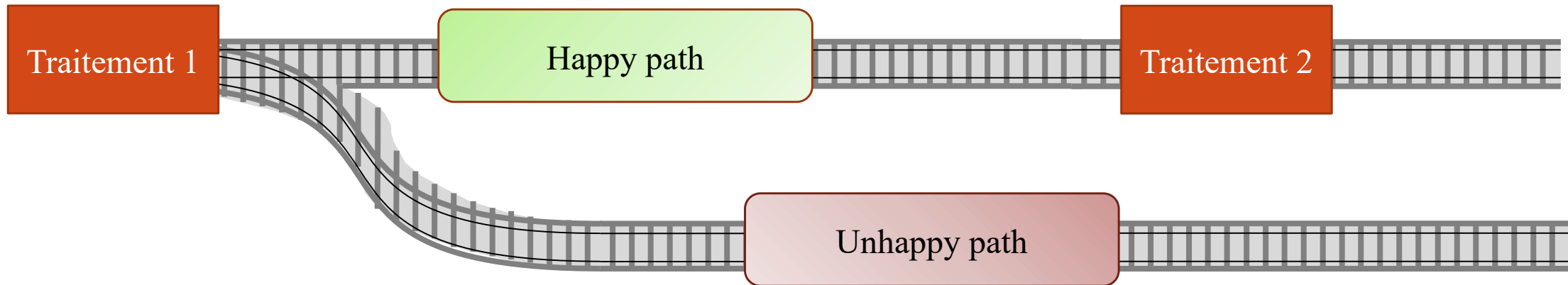
Composition avec un Result



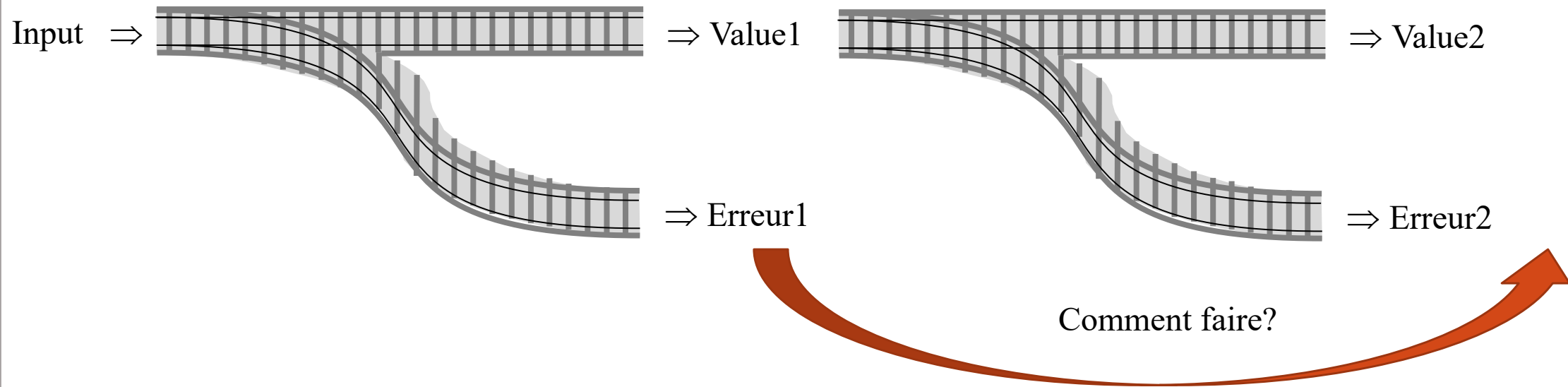
Composition avec un Result



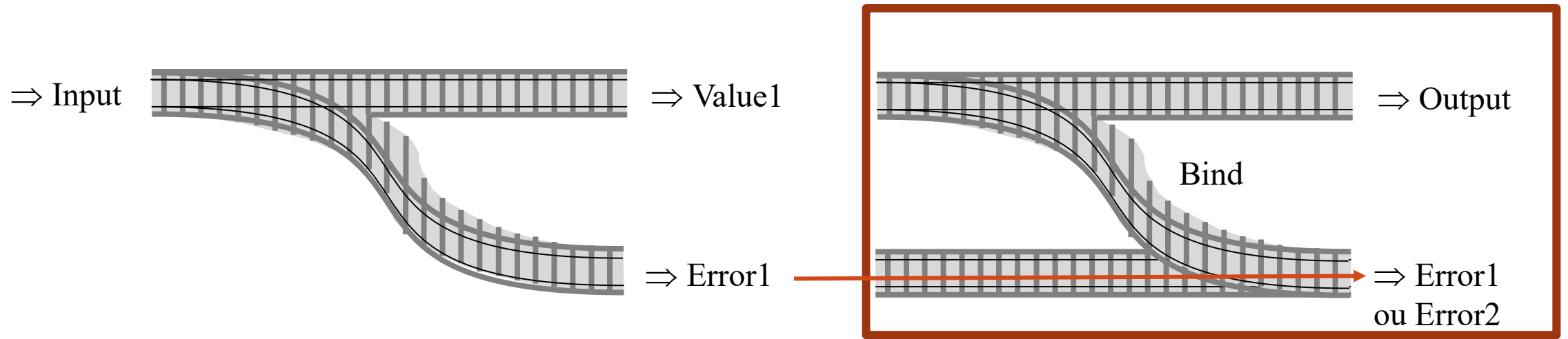
Composition avec un Result : Map



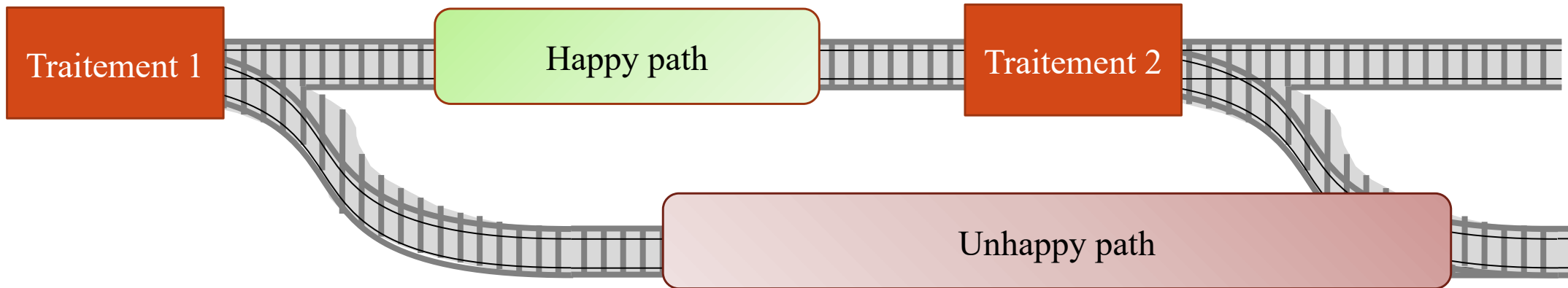
Composition entre deux fonctions à Result



Composition entre deux fonctions à Result



Composition entre deux Result: Bind

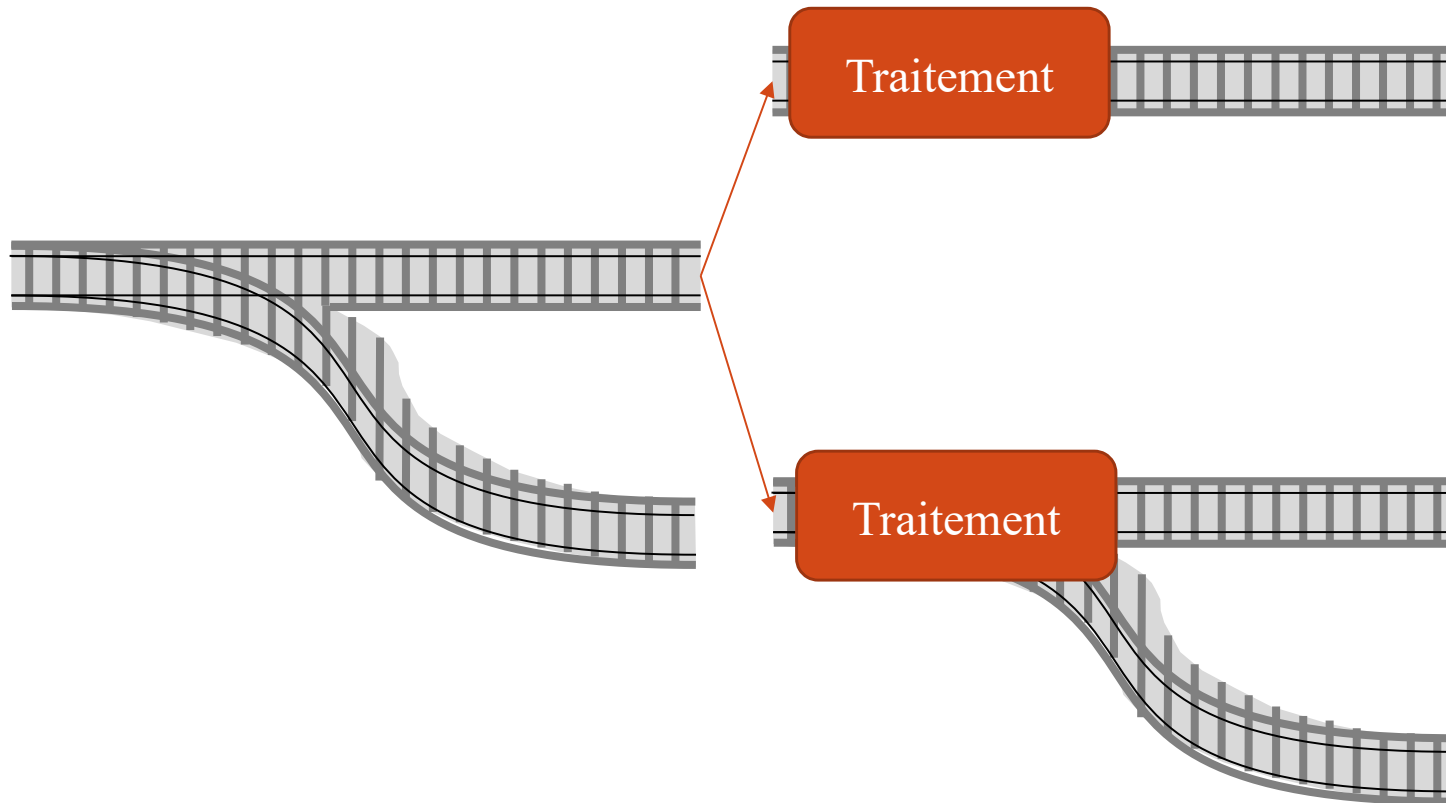


Aide-mémoire

Type de base
(Result ou Option)

Type de transformation
à appliquer

Fonction à
utiliser



Map

Bind

Conclusion

Sources!

- <https://github.com/RomainTrm/Talk-RailwayProgramming>
- <https://twitter.com/ScottWlaschin>
- Domain Modeling Made Functional (Scott Wlaschin)(<https://pragprog.com/titles/swdddf/domain-modeling-made-functional/>)
- <https://fsharpforfunandprofit.com/rop/>
- <https://blog.ploeh.dk/2019/02/11/asynchronous-injection/>
- <https://blog.ploeh.dk/2020/03/02/impureim-sandwich/>
- Théorie des catégories(vous la connaissez déjà):
<https://www.youtube.com/watch?v=DFZ7arg1XFc>