

D2BI

Exercice Scientifique et technique

La théorie des types

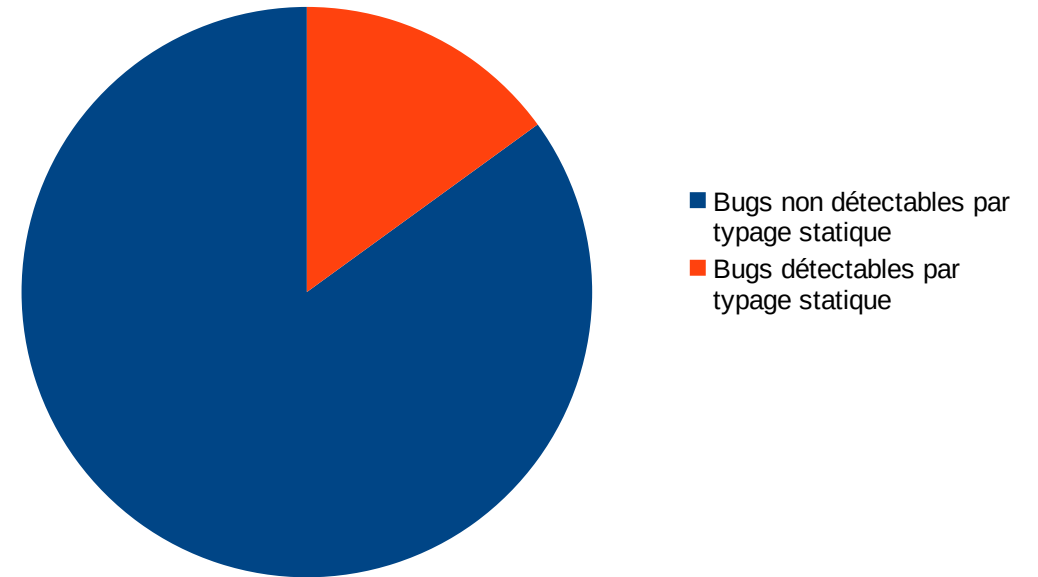
Mingotaud Gabriel

Pourquoi typer son code ?

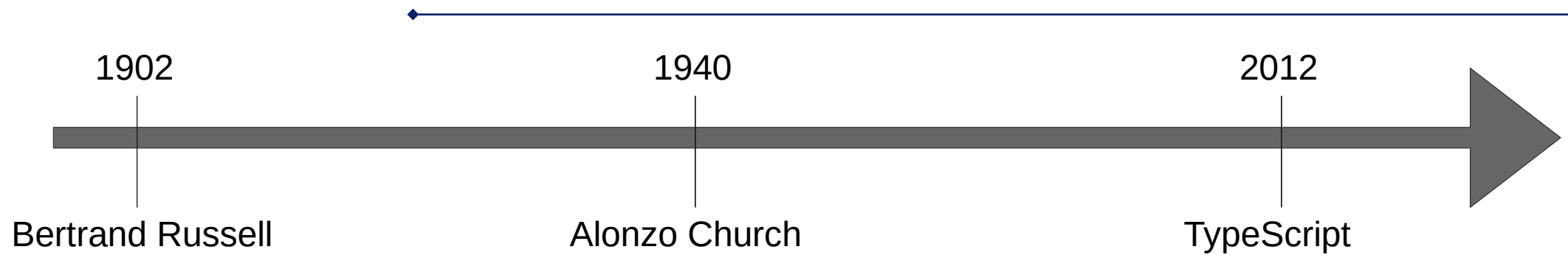
"To Type or Not to Type: Quantifying Detectable Bugs in JavaScript" - Zheng Gao, Christian Bird, Earl T. Barr - IEEE ICSE 2017

```
1 function addNumbers(x, y) {  
2   return x + y;  
3 }  
4 console.log(addNumbers(3, "0"));
```

Proportion de bugs détectables par le typage statique



Qu'est-ce que la théorie des types ?



Bertrand Russell :

- Découvre un paradoxe : "l'ensemble de tous les ensembles qui ne se contiennent pas eux-mêmes"
- Pour le résoudre : classer les objets mathématiques en niveaux hiérarchiques = les types
- Objectif : rendre les mathématiques cohérentes, ne pas programmer

Alonzo Church :

- Invente le lambda-calcul typé
- Fondation théorique de tous les langages fonctionnels modernes (Haskell, OCaml...)
- Première formalisation du concept de type en informatique

TypeScript :

- Microsoft applique ces concepts au développement web
- La théorie mathématique devient un outil quotidien du développeur

Les grands systèmes de types

Typage statique

```
function double(x: number): number {  
  return x * 2;  
}  
double("3");
```

TypeScript

Erreur détectée lors du développement

Typage dynamique

```
def double(x):  
  return x * 2  
  
double("3")
```

Python

Erreur détectée à l'exécution

La relation de Curry-Howard

MATHÉMATIQUES

PROGRAMMATION

Proposition	=	Type
Preuve	=	Programme
A implique B	=	Fonction $A \rightarrow B$
A et B	=	Tuple (A, B)
A ou B	=	Union $A \mid B$

- Un programme qui compile = une preuve qui est correcte

- Un bug de type = une erreur logique dans une preuve

Cas professionnel concret

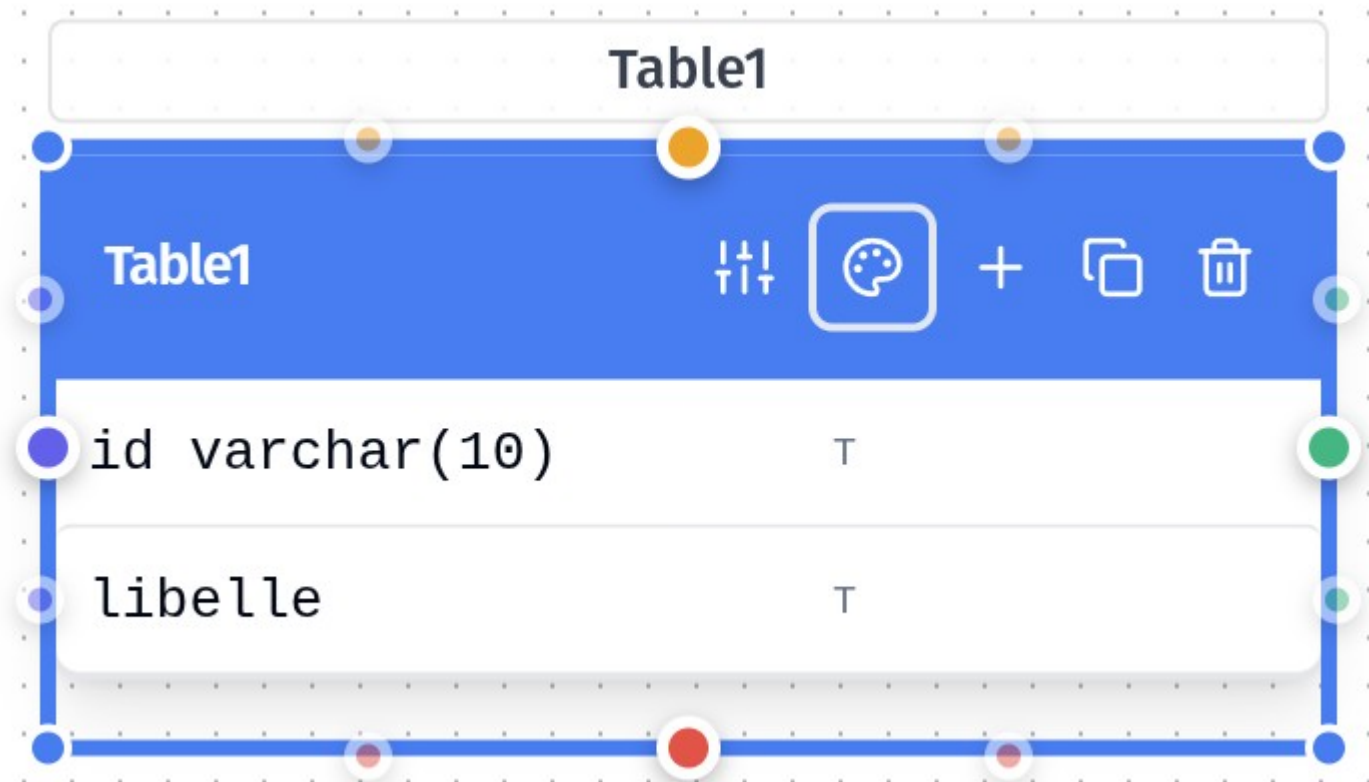
varchar et int sur SQL Server

```
SELECT libelle  
FROM Table1  
WHERE id = 0000000000
```

→ CAST de libelle en int

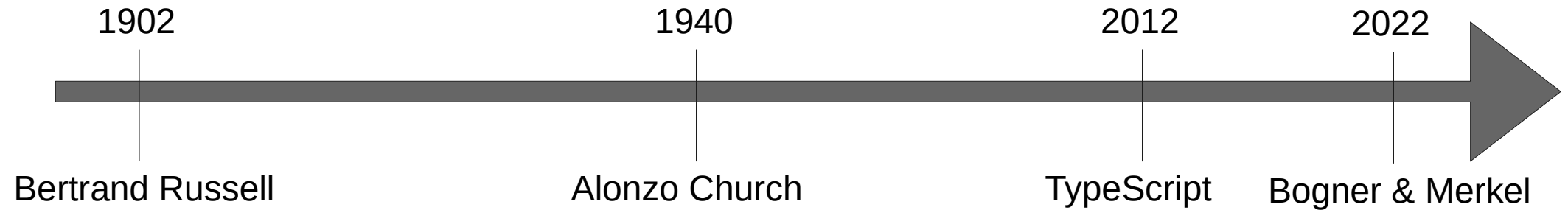
```
SELECT libelle  
FROM Table1  
WHERE id = 9999999999
```

→ CAST de libelle en int impossible



Limites & débats

"Characterizing the Usage of TypeScript in Open Source Projects" - Justus Bogner & Manuel Merkel - MSR 2022



- 60 % de commit de correction de bugs en plus en TypeScript

- Les bugs TypeScript prennent un jour de plus à corriger (31,86 vs 33,04 jours)

Des mathématiques de Russell aux langages modernes

Le typage statique détecte 15% des bugs en production

Typer son code c'est raisonner mathématiquement

Ce n'est pas une solution miracle

Sources

"To Type or Not to Type: Quantifying Detectable Bugs in JavaScript"

- Zheng Gao, Christian Bird, Earl T. Barr
- IEEE ICSE 2017 :

<https://ieeexplore.ieee.org/document/7985711>

"Characterizing the Usage of TypeScript in Open Source Projects"

- Justus Bogner & Manuel Merkel
- MSR 2022 :

https://www.researchgate.net/publication/359389871_To_Type_or_Not_to_Type_A_Systematic_Comparison_of_the_Software_Quality_of_JavaScript_and_TypeScript_Applications_on_GitHub