



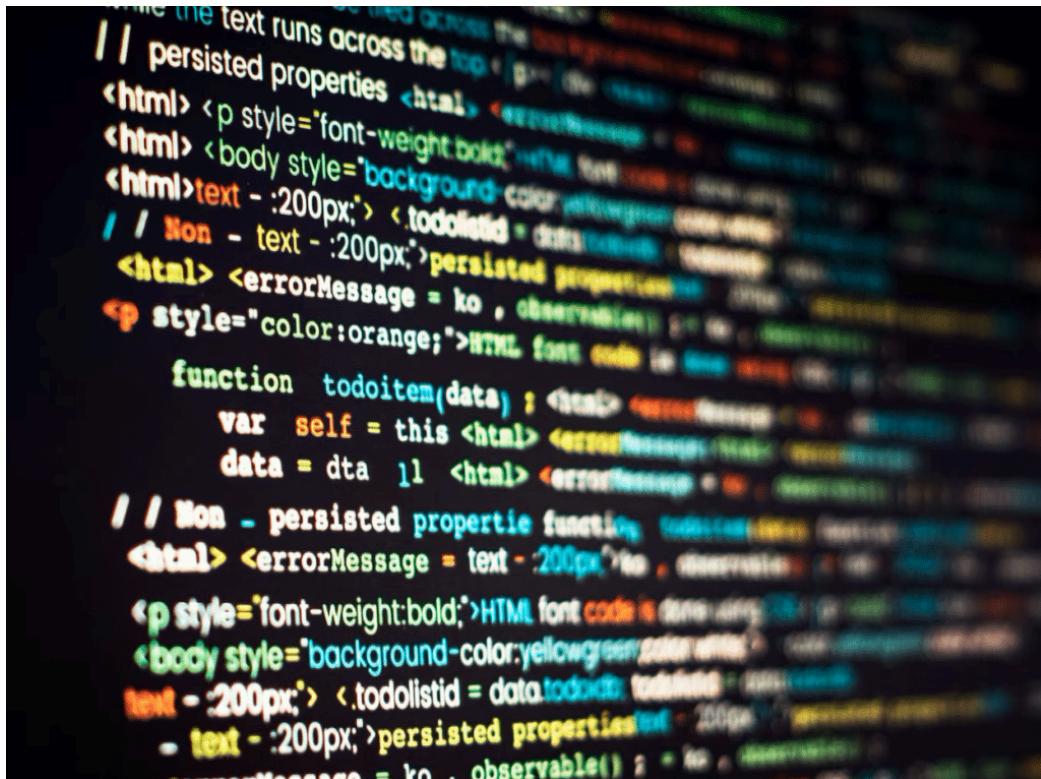
Recherchez un article...

[Retour aux articles \(/blog-data-ia-actualites\)](#)

Le créateur du C++ s'en remet à sa communauté pour sauver son langage

5 Mar 2025 - 2 m de lecture - Actualités (<https://datascientest.com/category/news>)

Gabin Paul(<https://datascientest.com/author/gabpaul>)



```

// persisted properties
<html> <p style="font-weight:bold;">HTML font code is done using</p>
<html> <body style="background-color:yellowgreen; color:white;>
<html>text - :200px;> <.todolistid = data.todolistId>
<html> <errorMessage = ko . observable()>
<p style="color:orange;">HTML font code is done using</p>
function todoitem(data) {
  var self = this <html> <errorMessage = ko . observable()>
  data = dta || <html> <errorMessage = ko . observable()>
  // Non - persisted properties function
  <html> <errorMessage = text - :200px;>ko . observable()
  <p style="font-weight:bold;">HTML font code is done using</p>
  <body style="background-color:yellowgreen; color:white;>
  text - :200px;> <.todolistid = data.todolistId>
  - text - :200px;>persisted properties
  <errorMessage = ko . observable()>

```

Sous pression des agences de cybersécurité et confronté à une remise en question de sa fiabilité, le langage C++ est dos au mur. Son créateur, Bjarne Stroustrup, tire la sonnette d'alarme et appelle la communauté des développeurs à agir pour renforcer sa sécurité. Face à des recommandations de plus en plus insistantes pour abandonner le C++ au profit de langages jugés plus sûrs comme Rust ou Swift, la question se pose : le C++ peut-il encore s'adapter ou est-il voué à être oublié ?

Un manque croissant de cybersécurité

Depuis plusieurs années, les **experts en cybersécurité** (<https://datascientest.com/cybersecurite-tout-savoir>) **alertent sur les failles du C++**, notamment en ce qui concerne la gestion de la mémoire. La NSA et la CISA ont même appelé les entreprises à **migrer vers des alternatives plus sûres d'ici 2026**, arguant que les

vulnérabilités inhérentes au C++ (<https://datascientest.com/langage-c-plus-plus>) sont trop importantes pour être ignorées. Ce constat met en péril l'avenir du langage, encore massivement utilisé dans les infrastructures critiques, les systèmes embarqués et les logiciels à haute performance.



Pour contrer ces critiques, Bjarne Stroustrup a proposé l'**adoption du framework Profiles**, censé offrir une meilleure gestion de la sécurité mémoire tout en préservant la flexibilité du langage.



Des solutions qui divisent

L'initiative Profiles ne fait pas l'unanimité. Certains experts, comme Robin Rowe, à l'origine du projet TrapC, estiment qu'elle est trop contraignante et pourrait **limiter certaines fonctionnalités essentielles** du langage. D'autres alternatives ont émergé, comme FilC, Mini-C et Safe C++, chacune avec une approche différente pour renforcer la sécurité mémoire.

D'autres chercheurs, à l'image de David Chisnall de l'Université de Cambridge, insistent sur un point clé : **la sécurité ne peut pas uniquement être gérée au niveau du langage**. L'interopérabilité entre différents langages devient un enjeu majeur, et des architectures hybrides (comme l'association de Rust avec Lua) posent des problèmes de compatibilité.

Si cet article vous a plu et si vous envisagez une carrière dans la Data Science ou tout simplement une montée en compétences dans votre domaine, n'hésitez pas à découvrir nos offres de formations (<https://datascientest.com/formations-data>) ou nos articles de blog (<https://datascientest.com/blog-data-ia-actualites>) sur DataScientest.

Découvrir nos formations en Data Science
(<https://datascientest.com/formations-data>)

[Facebook](#)

[Twitter](#)

[LinkedIn](#)

Poursuivre la lecture