



# TRANSFORMATION DES PROCESSUS ACTUARIELS AVEC PYTHON

Tendances du marché et retour d'expérience

Mélina Restoueix  
Santiago Fiallos



# 1. Introduction

# Quelques définitions pour commencer



Outil	code ou logiciel qui donne à l'utilisateur la possibilité d'effectuer une tâche précise
Langage ouvert	langage dont la licence permet d'accéder librement au code source, de créer des travaux dérivés et de le redistribuer
Solution propriétaire	logiciel mis à disposition par une partie tierce qui détient la propriété intellectuelle

# Rejoignez-nous sur beekast!

24<sup>e</sup> CONGRÈS DES ACTUAIRES



183504

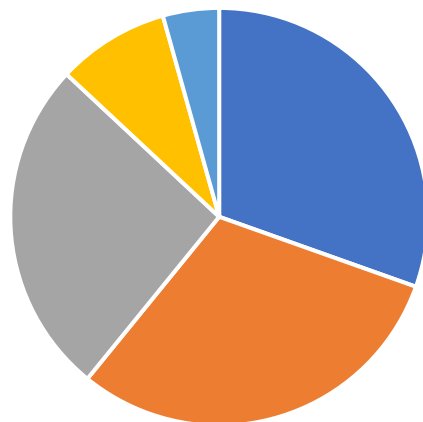
## 2. Utilisation des langages ouverts sur le marché – résultats de l'enquête aVB

# Trois quarts des répondants utilisent des langages ouverts pour développer des outils internes



29

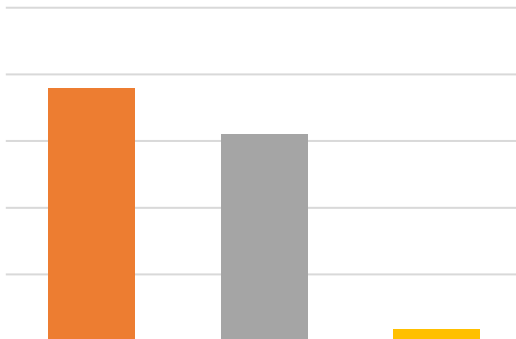
répondants



- Mutuelle
- Assurance
- Bancassurance
- Institution de prévoyance

76%

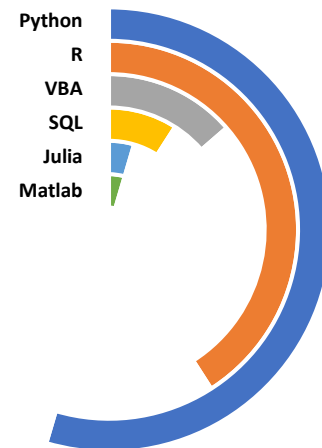
utilisent des outils développés en interne à partir de langages ouverts



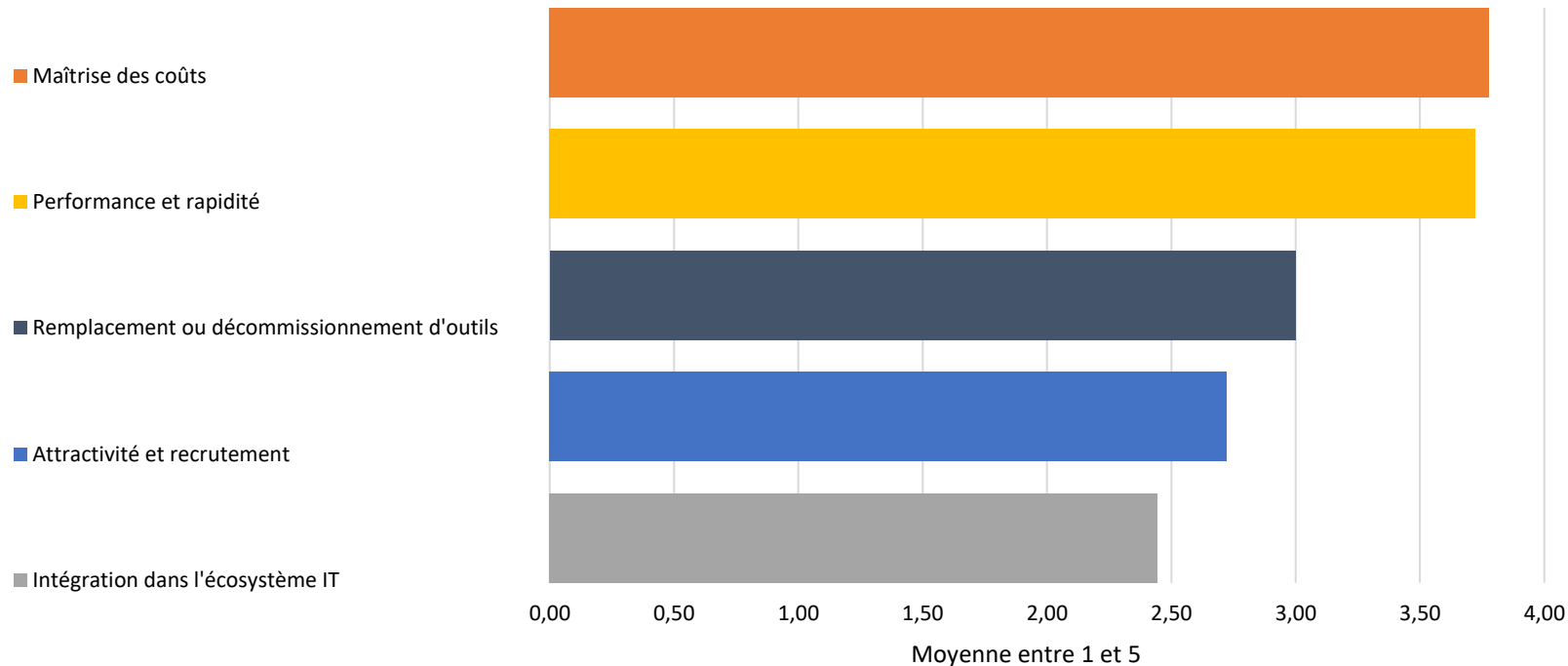
Langages ouverts Solutions propriétaires Non

55%

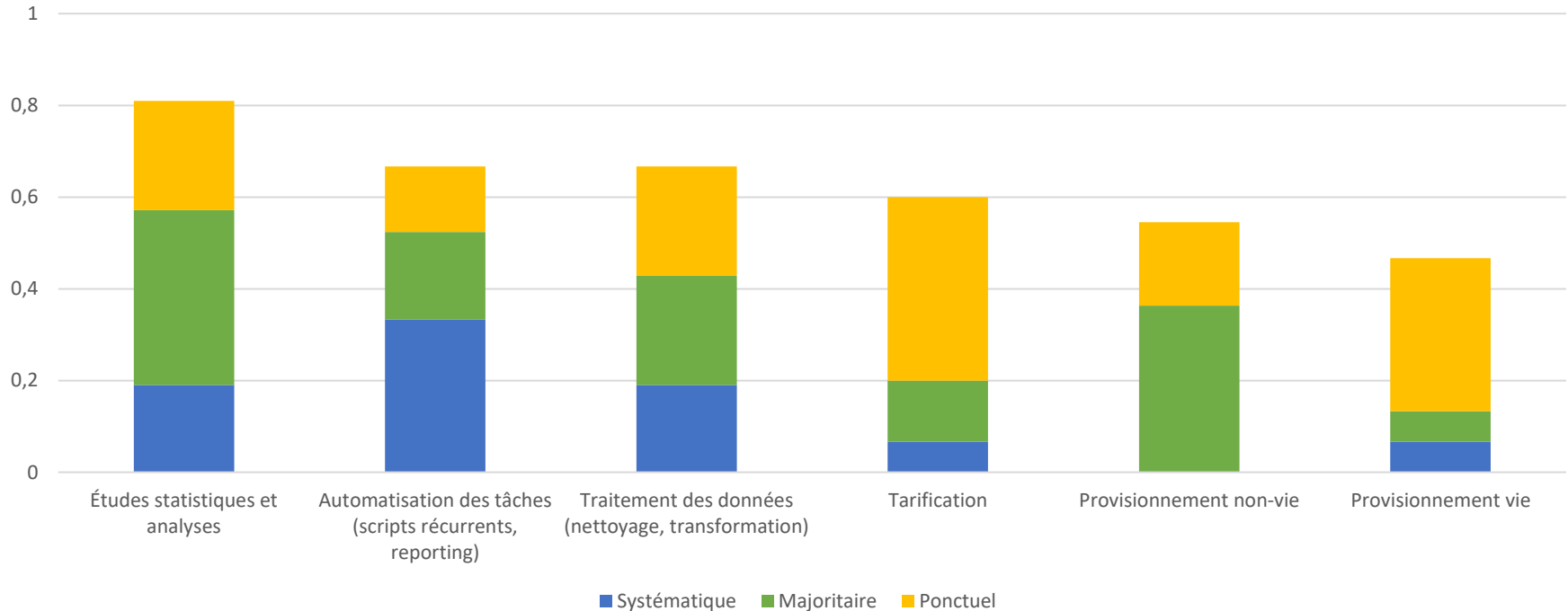
ont développé des outils à partir de python



# Les langages ouverts pour une meilleure maîtrise des coûts et une meilleure performance

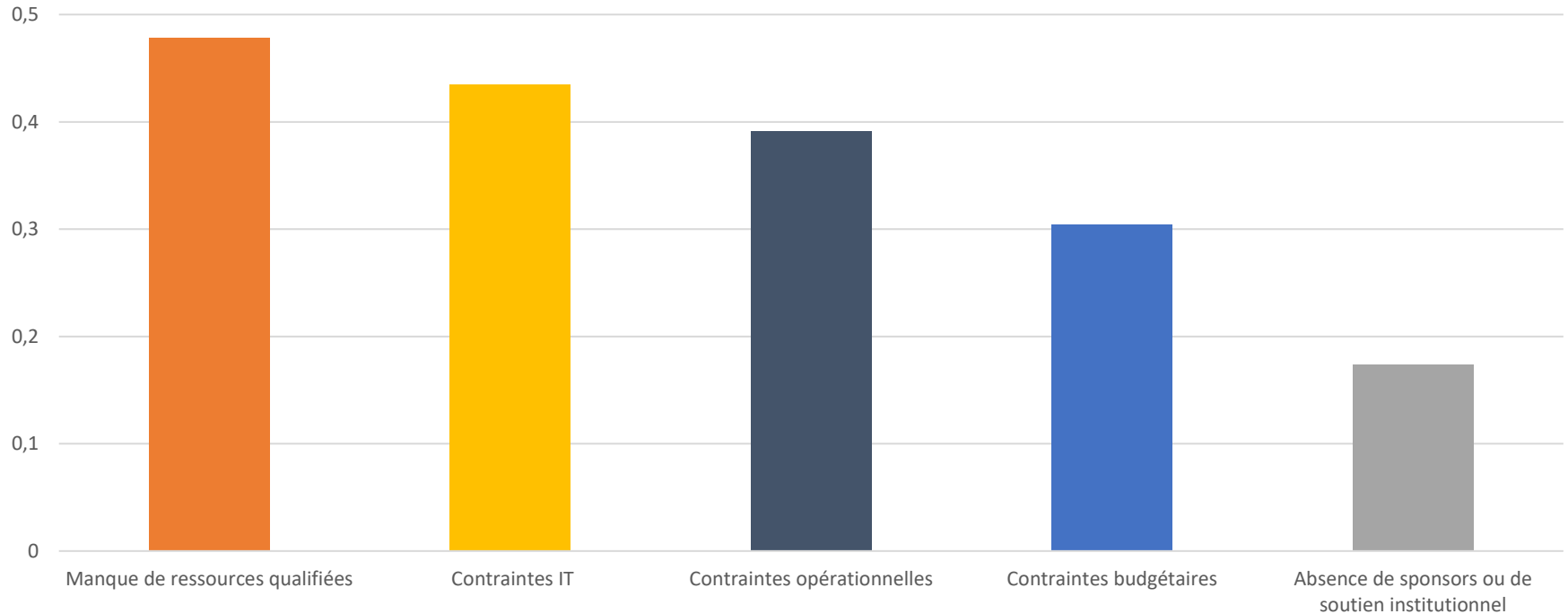


# Aujourd'hui les langages ouverts sont principalement utilisés dans les domaines de la donnée, de l'analyse et de l'automatisation

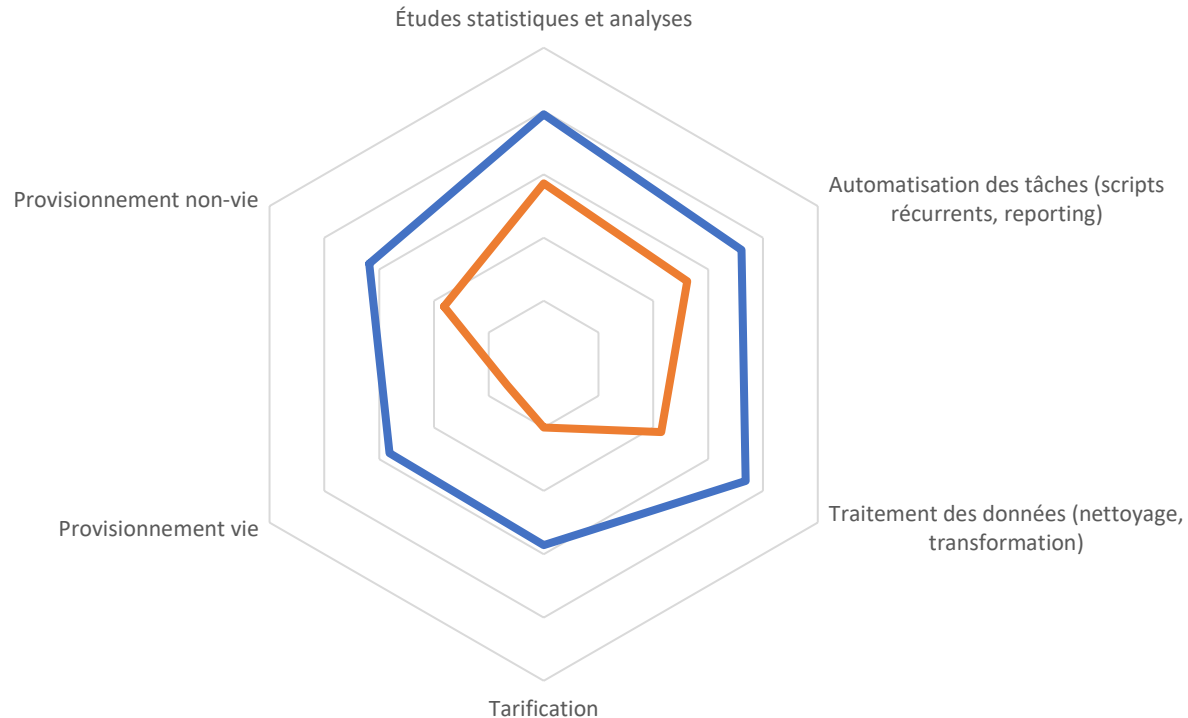




# Le manque de ressources qualifiées et les contraintes IT et opérationnelles sont les principaux freins à une plus grande adoption des langages ouverts



# Les actuaires prévoient de recourir davantage aux langages ouverts dans les 3 ans à venir, et ce sur l'ensemble de leur champ d'action





### 3. Retour d'expérience AG2R La Mondiale sur un projet de migration de modèle actuariel

# Une migration principalement motivée par des objectifs opérationnels



## Objectifs primaires

Faciliter la prise en main et l'entretien du modèle

Réduire les risques opérationnels

Améliorer l'intégration avec l'écosystème informatique

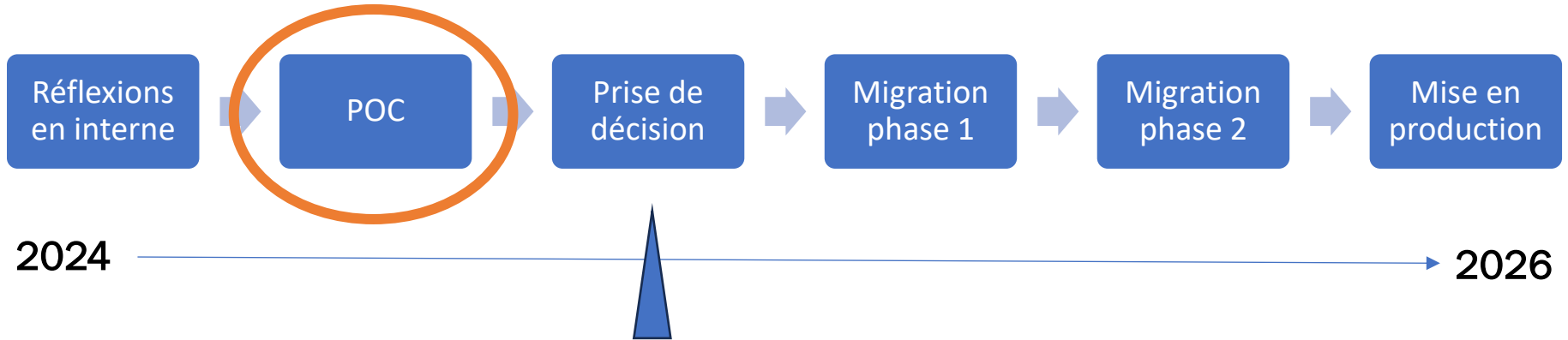
## Objectifs secondaires

S'affranchir d'un modèle construit sur du code propriétaire

Améliorer la qualité du modèle

Améliorer la performance générale du modèle

# Le POC, première étape pour valider la faisabilité technique et opérationnelle de la migration



# Objectifs du POC : lever les doutes sur une migration du modèle Prévoyance Santé sous Python



**Volet technique** : Développer un modèle sous Python

Matérialité

Non régression

Performance

**Volet opérationnel** : Intégrer le modèle Python dans l'écosystème IT du groupe







Auditabilité

Scalabilité

Flexibilité

# Focus sur l'environnement de modélisation construit en amont du POC



	Outils	Usage
	Azure virtual desktop (AVD)	Machine virtuelle Azure (standard groupe)
	Serveur de stockage	Stockage commun sur l'AVD
	Python 3.12	Noyau de calcul
	GIT	Système de contrôle de versions
	GITLAB	Hébergeur de projets versionnés avec GIT permettant la collaboration entre les développeurs
	Environnement de développement intégré (EDI)	Dans le cadre du POC, chaque utilisateur peut choisir l'EDI qui lui convient le mieux (Spyder, VS Code, Pycharm...)

# Construction d'un modèle souple et lisible reposant sur des couches d'abstraction et technique robustes



Souplesse &  
Lisibilité

Modèle  
actuariel

Code orienté objet  
Runlist  
Commentaires

Modularité &  
Factorisation

Couche d'abstraction  
Fonctions propres

Matrices, Vecteurs, Points  
Agrégateur  
Actualisateur  
Exportateur d'objets

Solidité &  
efficacité

Couche technique  
Bibliothèques open source





# Utilisation de l'IA générative comme accélérateur



## Contexte

**Almia** (AG2R LA MONDIALE Intelligence Artificielle) est une plateforme d'intelligence artificielle générative (IA Gen) **interne, sécurisée, responsable et éthique**. Elle a pour objectif d'optimiser les processus internes pour permettre aux collaborateurs de se concentrer sur des missions à forte valeur ajoutée.

## Use Cases

### Développement

génération d'un script python à partir d'un prompt qui décrit un besoin

### Assistance

- débogage
- optimisation du code

### Documentation

Commentaire du code au format sphinx



## 4. Q & A