Java and Spring Boot continue to be in demand in the industry, with Java being one of the most sought-after programming languages and Spring Boot offering features that simplify project setup and configuration[1][2][3][4]. However, Oracle's recent licensing changes have made Java more expensive to use, pushing some towards third-party Java runtimes like Azul, Amazon Coretto, Eclipse Temurin, and IBM Semuru[5][6][7][8].

Open-source alternatives such as Ruby on Rails and Python & Django are indeed gaining traction. Each has its advantages and might be more appealing depending on the context:

- **Learning Curve**:
  - Ruby on Rails (RoR) is often praised for its smoother learning curve due to its convention over configuration philosophy, which could be beneficial for beginners[9].
  - Django, on the other hand, may be easier for those with some prior experience in Python, and its "batteries included" philosophy means developers don't have to plan much ahead for application infrastructure, allowing quicker start to development[10].
- **Development Speed & Flexibility**:
  - RoR tends to be faster in terms of development speed and has a flexible interface[9].
  - Django provides more control in layout and configuration options, which could be seen as a boon or bane depending on the level of customization required[11].
- **Specialization**:
  - RoR is suitable for both front-end and back-end development tasks, making it a more versatile choice[12].
  - Django is more specialized for back-end development but has a huge community support, which could be beneficial for resolving issues and learning[12].
- **Community and Support**:
  - Both frameworks have strong community support which is crucial for solving issues and continuous learning.

The choice between these frameworks and Java & Spring Boot could also be influenced by the project requirements, team expertise, and the long-term goals of the project. For instance, if a project requires rapid development with a smaller learning curve, RoR might be a better choice. However, if the project is more complex and requires a more structured approach, Django or even sticking with Java & Spring Boot might be more suitable.

The comparison between these frameworks is quite nuanced and may require a deeper analysis based on the specific needs and context of your project or learning path.