

Polyglot Development

You are developing an application using a [Microservices Architecture](#). Your application will consist of several different Business Microservices and Backend For Frontends, along with potentially several Adapter Microservices of different types.

What language(s) should you choose for microservices?

The time is long past when single solutions can be put forward for all problems. For most of the 1990's Development pundits were concerned about the question of "What programming language would win out?". However, today the [TIOBE index](#) is a 10+ way fight.

At some point in the 2000's the question for analysts and pundits became "Which Application Server would win"? Today, applications servers are just one solution (increasingly less utilized) among many. Two factors have accelerated this pace: (1) The advent of the Cloud and (2) The advent of Microservices-based approaches. At this point in time, there is no one "right" solution for all programming problems, if there ever was.

Therefore,

Allow for multiple language choices. The Backend For Frontend pattern, for example, encourages teams to re-use the front-end language for the BFF to speed up teams.

Polyglot programming is the notion that the cloud opens up development to several new aspects. Just as Cloud leads to new approaches for scaling and the problems inherent in processing big data leads to new programming models, Open-source automation makes it possible to explore new possibilities for deployment.

Essentially this all boils down to one notion – you figure out what programming task you are doing and choose the best tool for the task

You have to suit the solution to the problem. If you are modeling a business domain with procedural rules and complex interactions, then DDD leading to Object Oriented Programming with Java may be a great fit. If on the other hand, you need to make classification decisions, or determine best choices from incomplete data, then Machine Learning with Python may help you. If you need to manipulate complex mathematical rules, or construct models of physical activities, then functional programming with Scala or even Javascript may be the right approach. There's no reason why you can't use more than one, or all of these approaches (plus approaches like scripting or event-based programming) in the same system!