

## Initial Post for

### Collaborative Discussion 2: Agent Communication Languages

Agent Communication Languages (ACLs) like KQML were developed to help autonomous agents communicate in a more meaningful way. Instead of just calling methods like in Python or Java, ACLs allow agents to share *intent*—they can ask questions, make requests, or offer information using structured message types called *performatives* (Finin, Labrou and Mayfield, 1994).

This kind of communication is powerful in distributed systems, where agents might come from different platforms but still need to collaborate. ACLs offer a level of flexibility and independence that method invocation lacks. Traditional programming languages work well in controlled environments, but they assume shared structures and tight integration, which doesn't always work in open, autonomous agent systems.

However, ACLs like KQML aren't perfect. They're harder to set up, require consistent ontologies, and can suffer from misinterpretation if developers use the same terms differently (Cohen and Levesque, 1995). In contrast, method calls in Java or Python are faster, easier to debug, and better suited for simpler or tightly connected systems.

In short, ACLs are great for agent autonomy and cooperation, especially when systems are complex and open-ended. But for smaller, well-defined applications, traditional programming still does the job more efficiently.

#### References:

Cohen, P.R. and Levesque, H.J. (1995) 'Communicative actions for artificial agents', *Proceedings of the International Conference on Multiagent Systems*, San Francisco, CA, 12–14 June. Cambridge, MA: MIT Press, pp. 65–72.

Finin, T., Labrou, Y. and Mayfield, J. (1994) 'KQML as an agent communication language', *Proceedings of the Third International Conference on Information and Knowledge Management*, New York: ACM, pp. 456–463. Available at: <https://doi.org/10.1145/191246.191322> (Accessed: 10 March 2025).