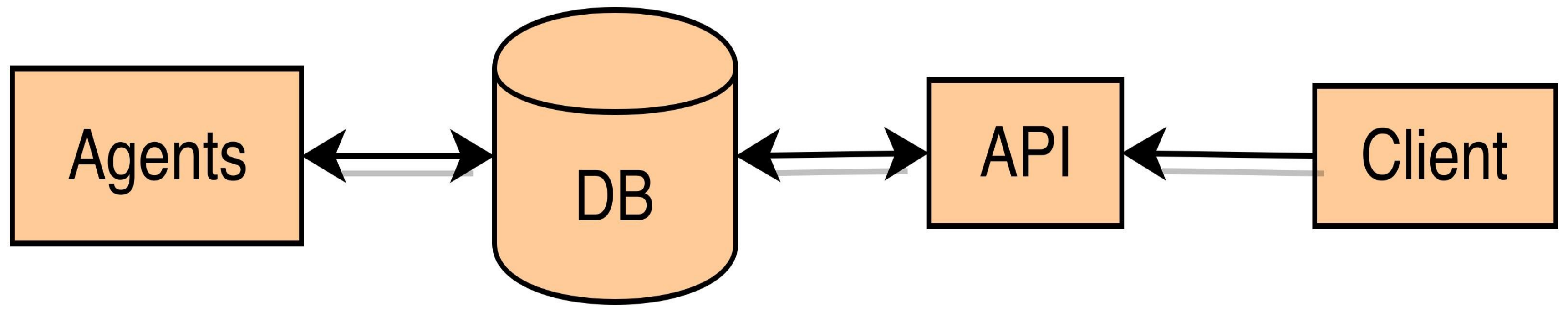
A Multi-Agent System Architecture for Carpooling Solutions

Brendan Devenney, 40053530, BEng Software Engineering, School of Computing, Napier University

Supervisor: Dr Neil Urquhart

Aims and Objectives

1. Understand the factors for and against past and existing carpooling systems.
2. Research the identified advantages and disadvantages of applying agent technology to systems heavy on negotiation.
3. Architect an abstract system which empowers developers to deploy their own bespoke solutions.
4. Apply this architecture to a use-case in the form of a taxi-sharing system between Napier's campuses.



Technologies

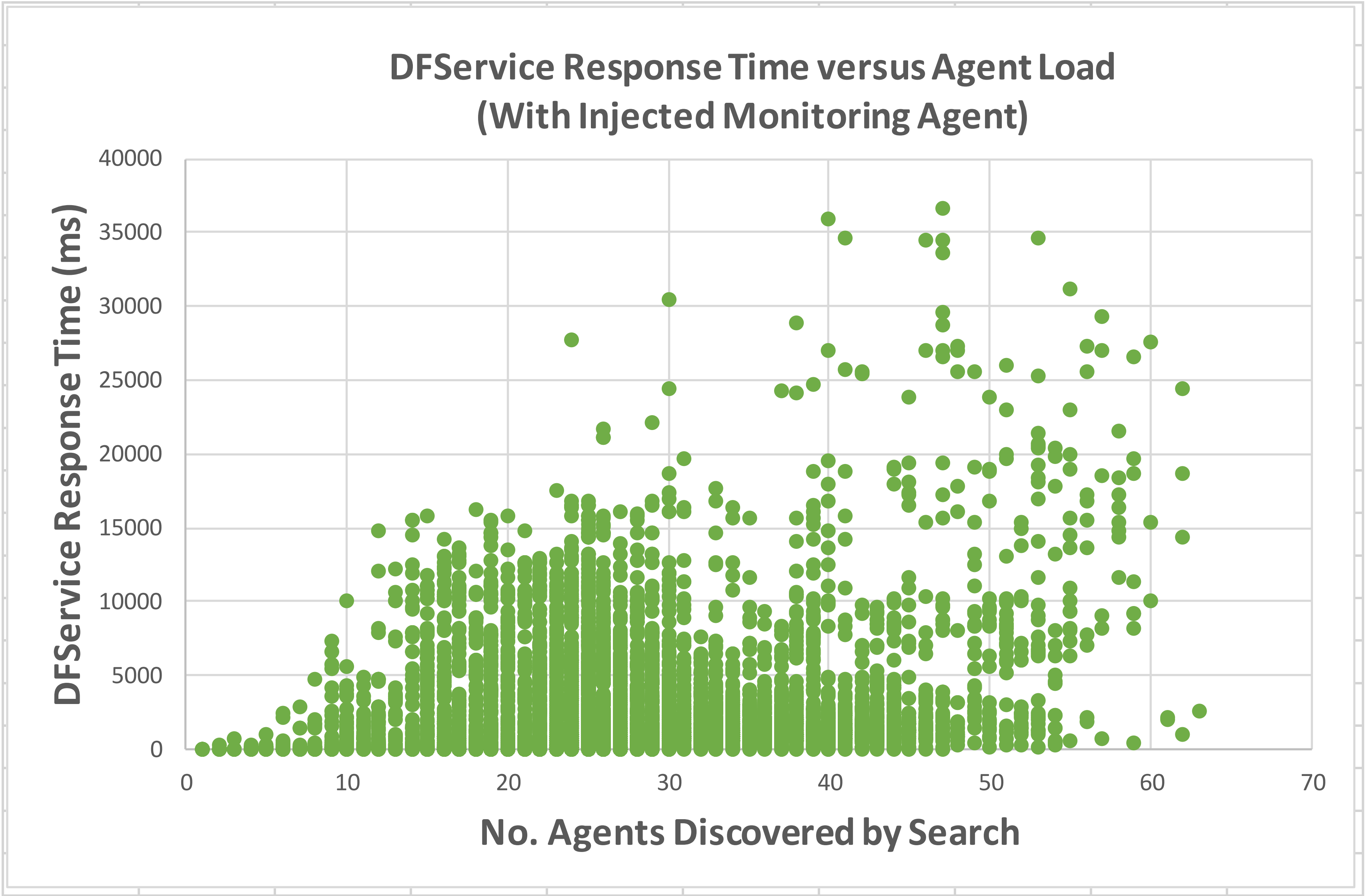
- **Agents:** Java Agent Development Framework (JADE)
- **Database:** PostgreSQL
- **API:** Python (Flask / Boto3 / SQLAlchemy)
- **Client:** IONIC (HTML5 / AngularJS) / Apache Cordova
- **Deployment:** Amazon Web Services

Evaluation

- Bottleneck: JADE platform scales linearly in all aspects other than directory facilitation.
- Solution is necessary before applying JADE to communication-heavy problems.
- Mengistu et al. (2008) solved the problem with local search caching—not good enough.

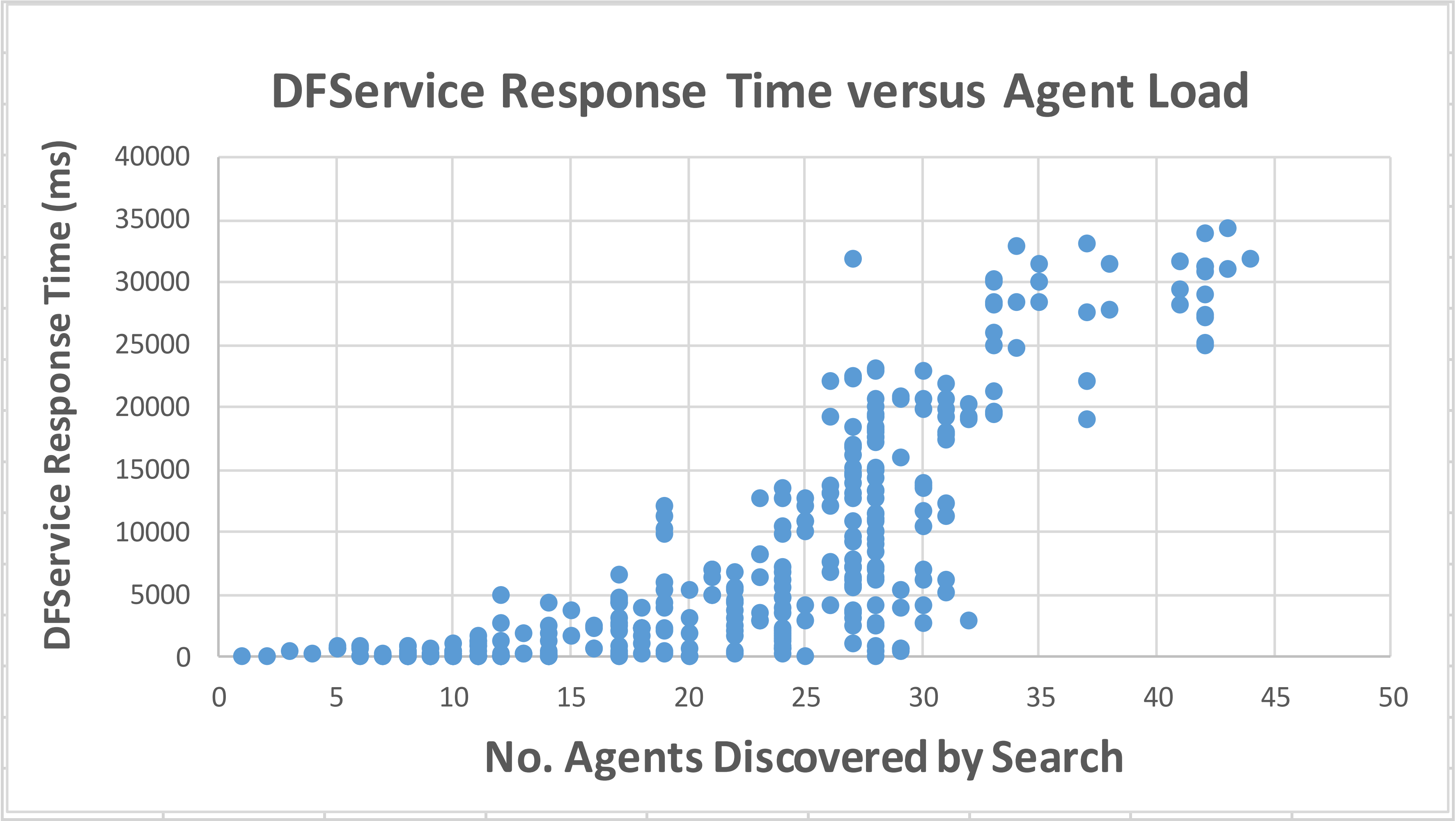
Solution: A monitoring agent which restricts searches when response times suggest DF nearing instability.

No collapse; recovery possible.



Issue: Response times greater than 30000ms throw an exception in JADE. Timeout can be increased—but this only delays the issue.

System collapse; no ability to recover.



Conclusions

1. JADE requires some tweaks for communication scalability.
2. JADE scales linearly and is perfectly suited for problems which can be distributed, such as carpool negotiation.
3. The architecture meets all requirements at a high level.
4. A vast amount of deeper research would be beneficial as scope changed to focus on DF issues.

References

Mengistu, D., Troger, P., Lundberg, L., & Davidsson, P. (2008). *Scalability in Distributed Multi-Agent Based Simulations: The JADE Case.* (pp. 93-99). IEEE.