Grouplanner: A Group Recommender System for Tourism with Multi-agent MicroServices

Patrícia Alves[1](✉), Domingos Gomes[1], Catarina Rodrigues[1], João Carneiro[1], Paulo Novais[2], and Goreti Marreiros[1]

[1] GECAD – Institute of Engineering of Polytechnic of Porto, Porto, Portugal
{pat,1190534,cadra,jrc,mgt}@isep.ipp.pt
[2] ALGORITMI Centre – University of Minho, Guimarães, Portugal
pjon@di.uminho.pt

Abstract. To provide recommendations to groups of tourists is a very complex task, especially due to conflicting preferences and the group's heterogeneity. The introduction of Multi-Agent Systems (MAS) can be the leverage we are looking for. Their autonomy, isolated state, distribution, and loose coupling make them suitable for the development of distributed systems, being the concept similar to a Microservices architecture. This connection brought a new approach, the Multi-Agent Microservices (MAMS) architecture, which exposes agents as resources through REST endpoints, changing the way MAS are seen and implemented, facilitating the user ↔ agent interaction, with a more efficient interoperability, bringing faster and more intelligent systems. In this demonstration, we propose the use of a MAMS architecture to represent the tourists in a mobile Group Recommender System for Tourism prototype, Grouplanner, exposing their agents and knowledge as resources that can be consumed by HTTP clients by directly communicating with the tourists' agents using REST endpoints, in order to provide faster and better recommendations.

Keywords: Group recommender systems · Multi-agent microservices · Tourism

1 Introduction

Recommender Systems (RS) are an increasingly important tool to help users obtain personalized recommendations in a variety of domains. The travel and tourism domain are a good example, however, when groups of tourists are involved, conflicts related to the users' preferences and the group's heterogeneity arise. Several authors proposed the use of Multi-Agent Systems (MAS) to help with those limitations, and proactively make recommendations based on the tourists' profile and context, enriching the tourists' experience [1, 2].

It can be seen in literature that the use of MAS is becoming more frequent and applied to a variety of different domains [3–5]. One of the greatest advantages of using MAS is that their autonomy, isolated state, distribution, and loose coupling [6] allows the development of distributed systems [7], being the concept similar to MicroServices

F. Dignum et al. (Eds.): PAAMS 2022, LNAI 13616, pp. 454–460, 2022.
https://doi.org/10.1007/978-3-031-18192-4_37

(MS), as argued by W. Collier, O'Neill, Lillis and O'Hare [6], who believe that "an agent can be viewed as a type of Microservice that can be deployed seamlessly within any Microservice ecosystem". According to the authors, there are many similarities between MAS and MS, such as the isolated state, distribution, elasticity, automated management, and loose coupling, as well as how easy it is for MS to meet the social ability, autonomy, and reactivity of agents, with the advantage that agents can be proactive. These similarities led to the Multi-Agent MicroServices (MAMS) architecture, that consists on Agent-Oriented MS exposed through REpresentational State Transfer (REST) endpoints [6], where requests can be responded asynchronously by the agents. This type of architecture can change the way MAS are seen and implemented, facilitating the implementation of agile development methods [6], and the way agents are accessed and used, bringing faster and more intelligent systems.

This paper presents an improvement and demonstration of part of our work, on modeling a mobile Group Recommender System (GRS) for tourism using the tourists' personality to predict their preference for tourist attractions [8] and intelligent agents with MicroServices [3, 9], where we propose the use of a MAMS architecture to expose the agents and their capabilities as resources, facilitating the user ↔ agent interaction, the access to the agents' knowledge and capabilities, and a more efficient interoperability, to provide faster and more personalized recommendations.

2 Main Purpose

This demonstration intends to show how we managed to implement a MAS using MAMS in a mobile GRS for tourism prototype, Grouplanner, allowing direct communication with the tourists' agents using REST endpoints, and how the agents use their social abilities to interact with each other and provide individual and group recommendations.

Each tourist is represented by an agent in the MAMS (Tourist Agent), modelled with the tourist's profile (demographic data, personality, travel-related preferences and concerns, and motivations for traveling), which is accessible through a REST endpoint from the MAMS REST Application Programming Interface (API) (see Fig. 1). The objective is to expose Tourist Agents as resources through a REST API [6], so the information on the tourists can be shared easily and faster. The agents can share their knowledge and interaction data, either internally and externally, and collaborate to provide more accurate and satisfying recommendations to the tourists, reducing the group conflicts, without the need of a broker agent in the environment to communicate with the other agents.

.NET is one of the most used object-oriented programming frameworks for implementing web services and interoperable cross-platform applications, and it was therefore used to implement the Grouplanner prototype. As for the development of the MAMS, many multi-agent platforms can be found in literature [10]. We chose ActressMAS [11] for its simplicity and ease of use, and because it was one of the few agent frameworks that uses C# programming language, being more suitable for integrating with the.NET framework.

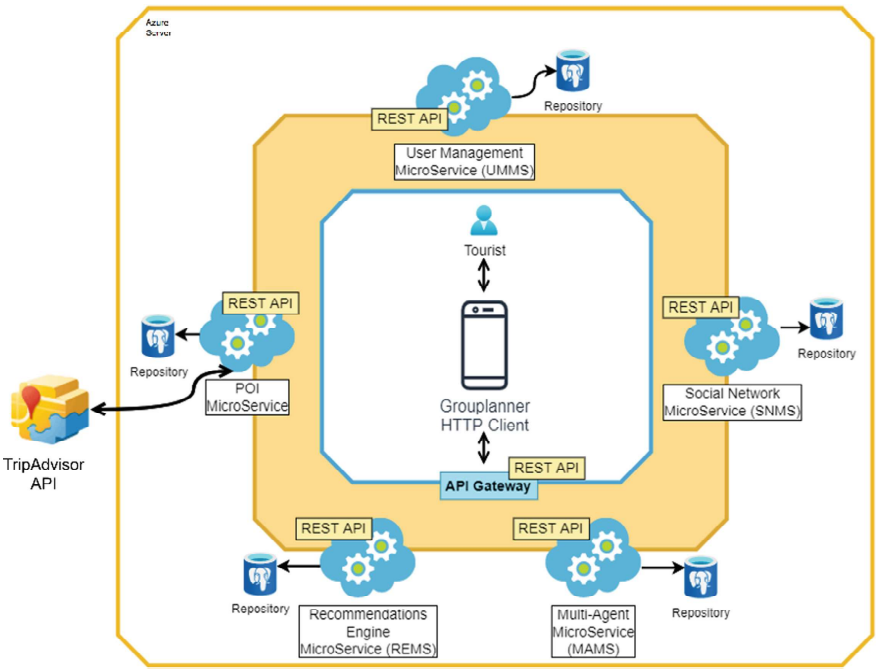

Fig. 1. Grouplanner MicroServices architecture.

3 Demonstration

When a tourist user registers in the Client App, he is first registered and authenticated/authorized in the system, using the ASP.NET Core Identity and JSON Web Tokens (JWT), through the API Gateway, which was developed using the ASP.NET Core framework, that acts as a single-entry point into the system, having all the HTTP Clients requests to pass through it.

After the registration, a JSON with the tourist's profile is built and sent at the same time to the User Management MS (UMMS) and the MAMS, via an HTTPS POST request. The MAMS parses the JSON file and brings life to a Tourist Agent (TourAg) in the environment, modelled with the tourist's profile, registering it in the environment's concurrent dictionary [11]. The UMMS stores the tourist's data in the corresponding repository, necessary for consults and updates in the app.

When a tourist requests individual recommendations of Points-Of-Interest (POI), a JSON message is sent, via HTTPS, directly to her TourAg in the MAMS, using the URL: "https://<domain>/mams/agents/tourists/<TourAg-name>/requestRecommendation". Having access to the tourist's knowledge base, the respective TourAg compiles the tourist's profile into a JSON and asks the Recommendations Engine MS (REMS) for the 10 best POI that match her personality[1], as predicted in the "Personality vs Tourist Attractions Preference" model proposed in our previous work [8], and according to a POI Ontology. The REMS responds to the TourAg with a JSON containing the suggested POI, after which the TourAg uses her social ability to ask the other

[1] To facilitate the demonstration, we are not considering the tourists' demographic data, motivations, nor travel-related preferences and concerns.

TourAg if there are better suggestions based on their current knowledge, updating them accordingly. The final recommendations are then sent to the requesting user.

When a tourist creates an excursion group (e.g.: "One Adventure"), becoming the group owner, a JSON containing the group's information is sent to the MAMS (URL: "https://<domain>/mams/groups/registerGroup"), which immediately creates a Travel Agent (TrvAg) with the group's data, responsible for that group (see next dialogue).

> (12/05/2022 10:20:20) Travel Agent 1 has been added
> (12/05/2022 10:20:20) [TrvAg-1]: Greetings, I am the Travel Agent for the "One Adventure" excursion group.

When the group owner requests recommendations for the group, an HTTPS request is sent to the group's TrvAg (URL: "https://<domain>/mams/agents/groups/OneAdventure/requestRecommendation"), which divides the group into subgroups with similar personalities, meaning a group can be divided into two or more subgroups, depending on the tourists' personality (if all the tourists have similar personalities the main group will not be divided). In the example (Fig. 2, left image), the group was divided into three subgroups, where one has more interest in Adrenaline activities, another in Gastronomy events, and the other in Museums, Boat trips & Viewpoints[2].

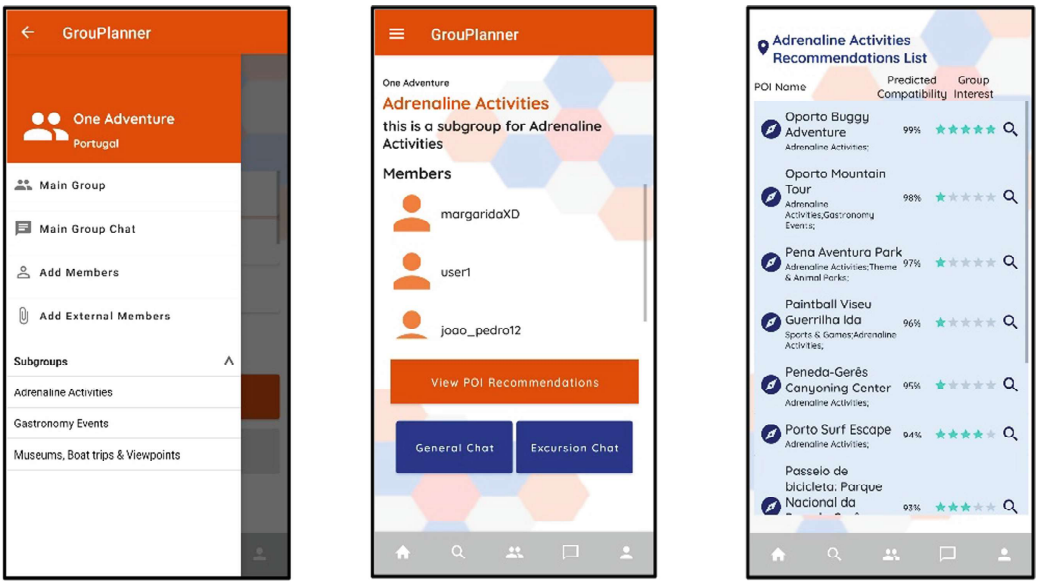

Fig. 2. Left: Grouplanner user interface showing the 3 subgroups formed; Center: Detail of the members in the "Adrenaline Activities" subgroup formed; Right: List of POI recommended to the "Adrenaline Activities" subgroup.

After creating the subgroups, they are sent via HTTPS in a JSON message to the REMS, which proceeds as in the individual recommendations previously mentioned.

[2] This division is also based on the "Personality vs Tourist Attractions Preference" model proposed in our previous work [8].

After assigning the 10 best POI to each subgroup, the service responds with a JSON message to the requesting TrvAg with the recommended POI list. The TrvAg then communicates with the Tourist Agents to learn what should, or should not, be recommended using the same strategy mentioned for the individual recommendations (as can be seen in the next dialogues).

The final POI list is then sent by the TrvAg to the Grouplanner HTTP Client, which creates the proposed subgroups and assigns their respective tourists, and at the same time to the group's Social Network MS (SNMS), responsible for persisting all the (sub)groups data, necessary for future consults within the app. Each subgroup can then consult its respective recommended POI list (see Fig. 2, right side image).

4 Conclusions

In this paper, we briefly demonstrated how we implemented a MAMS in a mobile GRS prototype using a MicroServices architecture, and how the agents were exposed as resources and exchanged information between them.

Just by using HTTP requests, the Tourist Agents knowledge and capabilities can be accessed by any HTTP client, meaning the proposed MAMS can be used by recommender systems exposing a REST API, to obtain knowledge on tourists' preferences according to their personality, demographic data, etc. The ability to make individual recommendations is a plus, as the respective Tourist Agent can learn with the feedback/evaluation given by the tourist to the suggested/visited POI, and therefore gain knowledge to help provide better group recommendations.

Due to space limitations, the remaining implementation details, the Recommendations Engine algorithms and the POI Ontology will be presented in future works.

(12/05/2022 10:20:30) [TrvAg-1]: margaridaXD, please tell me your score for each personality dimension.
(12/05/2022 10:20:30) [TrvAg-1]: user1, please tell me your score for each personality dimension.
…
(12/05/2022 10:20:31) [TourAg-margaridaXD]: Hi TrvAg-1, here are my personality scores: …
(12/05/2022 10:20:31) [TourAg-user1]: Hi TrvAg-1, here are my personality scores: ...
…
(12/05/2022 10:20:32) [TrvAg-1]: margaridaXD, you have a personality very similar to user1, joao_pedro12 and antonio, so I will put you together.
(12/05/2022 10:20:32) [TrvAg-1]: user1, you have a personality very similar to margaridaXD, joao_pedro12 and antonio, so I will put you together.
…
(12/05/2022 10:20:33) [TrvAg-1]: I finished analyzing the group members and decided to create 3 subgroups, one more motivated by "Adrenaline activities", other by "Gastronomy events", and another by "Museums", "Boat trips & Viewpoints".
(12/05/2022 10:20:33) [TrvAg-1]: margaridaXD, user1, joao_pedro12 and antonio, you were assigned to subgroup "Adrenaline activities".

… //After receiving the POI list for each subgroup from the REMS:
(12/05/2022 10:40:10) [TrvAg-1]: I have the following POI recommendation for these subgroups, does anyone has something against the recommendations?
(12/05/2022 10:40:12) [TourAg-margaridaXD]: I am afraid of water activities, can you please recommend other POI?
(12/05/2022 10:40:14) [TourAg- Joana]: I went to Nómadas Adventure Tours, but it's a 1 star to me.
…
(12/05/2022 10:40:24) [TrvAg-1]: According to the feedback received, I have changed the POI list recommendation and will send it to the group.
(12/05/2022 10:40:24) [TrvAg-1]: POI recommendations sent to "One Adventure" group.

References

1. Borràs, J., Moreno, A., Valls, A.: Intelligent tourism recommender systems: a survey. Expert Syst. Appl. **41**, 7370–7389 (2014)
2. Ravi, L., Devarajan, M., Sangaiah, A.K., Wang, L., Subramaniyaswamy, V.: An intelligent location recommender system utilising multi-agent induced cognitive behavioural model. Enterprise Inf. Syst. **15**, 1376–1394 (2021)
3. Carneiro, J., Andrade, R., Alves, P., Conceição, L., Novais, P., Marreiros, G.: A consensus-based group decision support system using a multi-agent MicroServices approach. In: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, pp. 2098–2100 (2020)
4. Zouad, S., Boufaida, M.: Using multi-agent microservices for a better dynamic composition of semantic web services. In: 2020 The 4th International Conference on Advances in Artificial Intelligence, pp. 47–52 (2020)
5. Neto, J., Morais, A.J., Gonçalves, R., Coelho, A.L.: Multi-agent-based recommender systems: a literature review. In: Yang, X.-S., Sherratt, S., Dey, N., Joshi, A. (eds.) Proceedings of Sixth International Congress on Information and Communication Technology. LNNS, vol. 235, pp. 543–555. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-2377-6_51

6. Collier, R.W., O'Neill, E., Lillis, D., O'Hare, G.: MAMS: Multi-Agent MicroServices. In: Companion Proceedings of the 2019 World Wide Web Conference, pp. 655–662 (2019)
7. Batet, M., Moreno, A., Sánchez, D., Isern, D., Valls, A.: Turist@: agent-based personalised recommendation of tourist activities. Expert Syst. Appl. **39**, 7319–7329 (2012)
8. Alves, P., et al.: Modeling tourists' personality in recommender systems: how does personality influence preferences for tourist attractions? In: Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, pp. 4–13 (2020)
9. Alves, P., Carneiro, J., Marreiros, G., Novais, P.: Modeling a mobile group recommender system for tourism with intelligent agents and gamification. In: Pérez García, H., Sánchez González, L., Castejón Limas, M., Quintián Pardo, H., Corchado Rodríguez, E. (eds.) HAIS 2019. LNCS (LNAI), vol. 11734, pp. 577–588. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29859-3_49
10. Pal, C.-V., Leon, F., Paprzycki, M., Ganzha, M.: A review of platforms for the development of agent systems. arXiv preprint arXiv:2007.08961 (2020)
11. Leon, F.: ActressMAS, a. NET multi-agent framework inspired by the actor model. Mathematics **10**, 382 (2022)