General Overview

1. **Can you summarize the main goal of your thesis in a few sentences?**
   - The main goal of my thesis is to simplify the development of serverless applications by abstracting and orchestrating heterogeneous BaaS services.
2. **What inspired you to choose this particular topic for your thesis?**
   - Cloud Computing was always an interesting topic for me and where I work they we also use a lot of semantic stuff with ontologies and knowledge graphs so I thought it could be really helpful to gain insights into that topics.

Problem Statement and Motivation

3. **What specific problems are you aiming to solve with the ServerlessIntent framework?**
   - The main problems I am trying to solve are challenges in portability and vendor lock-in and I also want to enhance the usability of these BaaS services by making them easier to use.
4. **Why is the issue of vendor lock-in significant in cloud computing?**
   - Vendor lock-in is a big issue in cloud computing because it limits flexibility and can increase costs. When you're stuck with one cloud provider, switching to another one is hard and expensive. This means you might miss out on better, cheaper services from other providers.
   - Relying on a single vendor also increases risks if they have outages or security problems. It can be tough to integrate new technologies and meet different regulatory requirements. Being locked in can stifle innovation since you can't easily try new services.
   - In short, avoiding vendor lock-in keeps your options open, helps manage costs, reduces risks, and encourages innovation.

Technical Aspects

5. **How does the ServerlessIntent framework manage to abstract the complexities of multiple BaaS services?**

The ServerlessIntent framework simplifies multiple BaaS services by:

1. **Using Intents**: Users specify what they want in simple terms.
2. **Orchestrator Module**: Automatically creates workflows from these intents.
3. **Ontology Management System**: Finds the best services and sequences.
4. **Service Registry**: Keeps an updated list of services.
5. **Workflow Formatter**: Converts workflows into ready-to-use files.

6. **Can you explain how the Ontology Management System works within your framework?**

The Ontology Management System (OMS) in the ServerlessIntent framework:

1. **Stores Knowledge**: Keeps data about BaaS services and their capabilities.
2. **Determines Best Services**: Uses this data to find the optimal services and sequences for a given task.
3. **Guides Workflow Creation**: Helps the orchestrator module construct effective workflows.

In short, the OMS ensures the framework uses the right services in the right order.

7. **What are the main challenges in automating the orchestration of BaaS services, and how does your framework address them?**

8. **How do you handle the limitations of individual BaaS providers, such as character limits in translation services?**

   - As mentioned in the presentation I would for example need to implement some serverless functions myself for example split, merge and things like that, but the limitations with their mitigations will be output of research question 2

Research Questions and Methodology

9. **Can you elaborate on your research methodology, specifically the Design Science Research approach?**

The Design Science Research (DSR) approach in my methodology:

1. **Problem Identification**: Identify the challenge of orchestrating BaaS services.
2. **Requirements Establishment**: Define what the ServerlessIntent framework needs to do.
3. **Design and Implementation**: Create and build the framework.
4. **Demonstration**: Show how the framework works with examples.
5. **Evaluation**: Assess the framework's effectiveness and usability.
6. **Communication**: Document and share the findings.

In short, DSR helps systematically create, test, and refine the ServerlessIntent framework.

10. **How do you plan to evaluate the effectiveness and efficiency of your framework?**
    - With a small experiment and survey, as well as the time logging
11. **What are the key parameters for BaaS services that you identified in your research?**
    - Output of research question 1

Implementation and Evaluation

12. **What technologies and tools are you using to develop the ServerlessIntent framework?**
    - Java, Protégé, GraphDb maybe, Apache Jena, RDF4J
13. **How will you measure the usability and time savings of your framework?**
    - Not yet to 100% decided how the survey will look like. Probably qualitative analysis as there are not that many participants. Time savings will be logged
14. **Can you describe the planned experiment involving computer science students and how it will contribute to your evaluation?**

Potential Impact and Future Work

14. **What do you see as the main contributions of your thesis to the field of cloud computing?**
    - The automated workflow generation using BaaS-enabled serverless functions from different providers
16. **How do you envision the future development of the ServerlessIntent framework?**
17. **Are there any potential commercial applications for your framework?**

Challenges and Risks

18. **What are the main risks associated with your project, and how do you plan to mitigate them?**
19. **Have you encountered any significant technical challenges so far, and how have you addressed them?**

Practical Applications and Use Cases

20. **Can you provide a real-world example or use case where the ServerlessIntent framework would be particularly beneficial?**

21. **How do you ensure that your framework remains adaptable to new cloud services and technologies?**

Intro

Which complexity of inputs will be handled? Text & Speech only?

That was maybe something that was not that clear because I used the same example in the Exposé and in the presentation. We only used this example because the CORE framework which we will use to develop our BaaS enabled serverless functions already supports the BaaS services OCR, Translate, T2S and S2T. But in the end the system is designed in a highly flexible and extensible way to allow all kinds of other inputs for example images to create a workflow that can detect faces and emotions from an image and translate the emotions or do other things with it. Or we could have other type of structured data as input for example User Feedback on a product where we could use AWS Comprehend to do some sentiment analysis on or things like that. Of course these supported input types depend on the BaaS services we support as we cannot support input types for which we don't have a BaaS service to handle them.

In which way is this a "significant advancement in the field of cloud computing"?

This framework is a significant advancement because it automates the orchestration of multiple BaaS services across different cloud providers, addressing key issues like vendor lock-in and portability. And the input as intent really makes it very easy to create complex workflows with minimal manual configuration, enhancing the usability and flexibility of cloud services.

Where does the motivation come from?

Firstly, the motivation comes from the CORE framework which was also developed in a master's thesis from a student of Sashko which simplifys the development of interoperable BaaS enabled serverless functions. Because we

thought the next step would be to wire these BaaS services together and with that simplify the creation of Serverless workflows which consist of a sequence of these BaaS services.
And secondly of course more and more of these BaaS services are developed and launched so developers can outsource more and more of their code to such BaaS services which then of course increases the need of these workflows that use a composition of BaaS services.

How confident are you to have covered all relevant literature?

The most important of the related literatures were given to me by Sashko and from there I did an extensive literature review to find all relevant literatures. However because it was no Systematic Literature Review I cannot be 100% sure that I haven't missed a relevant paper.

Evaluation

Survey: Ease of Use, Intuitiveness -> Have you considered TAM?

Yes for the Survey design I have considered different models also including the Technology Acceptance Model which could be a good fit in my case however as mentioned in the Presentation the Survey Design is not fixed yet and I will most probably sit together with Sashko when the time comes to decide how the Survey should exactly look like.

Proposed Solution

You want to introduce a cloud broker layer, will the choice of provider be manual or are you going to employ some optimization like SkyPilot does?
For the functions where I can use the CORE library there is already some optimization by the function itself. So basically, there I have three options. I

can manually select the provider by passing it to the CORE function, I can pass a set of features which the BaaS service should support then the function chooses the providers BaaS service that supports the set of features that are passed for example for Speech to text AWS supports to create subtitles in a specific format while Google does not. Google however supports spoken emoji which can transform spoken emojis into an emoji in the text. So, if in the function the feature subtitles is passed the CORE library automatically chooses the AWS BaaS service. And if both providers support the features other heuristics like file location, or input and output file size are considered for the provider and region decision.
For additional BaaS services that I will add I am not sure how far the optimization will go and if it is even inside of the scope of this work but technically also there it will be possible to do some optimizations.

You say you base your work on the CORE Framework but there is no reference to it in your Exposé…?

In easy words the CORE programming model abstracts away provider details for the implemented BaaS services so that the developer has one Abstract interface where he can call the BaaS service and then based on the passed parameters the correct provider is selected.

Threats to Validity

The application case is extremely niche (text2speech or speech2speech), whereas the underlying multi-cloud approach is something much broader. Can this Ontology management be transferred to other domains?

The ontology management system is designed to be flexible and extendable. While the initial application cases focus on text and speech, the underlying architecture can be applied to other domains, such as image processing, data analysis, and more, by integrating relevant BaaS services.

SkyPilot offers a broad solution for multi-cloud operation, how does your framework compare in terms of application domain versus deployment domain?

In my opinion these two framworks cannot really be compared as they address different needs. The ServerlessIntent framworks goal is it to make the creation of serverless workflows which use a sequence of BaaS services as easy as possible using the intent as input. That is more about the application domain and less about the most optimal and fastest deployment and on the other hand SkyPilot is more general focusing on running applications across different clouds in the most optimal way.