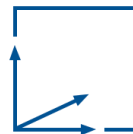


Development and Implementation of a Reusable Web-API for Online 3D-Reconstruction

Moritz Krüger

16.09.2021



Final: Bachelor Informatics

Supervisor: Prof. Gudrun Klinker, Ph.D.

Advisor: Linda Rudolph, M.Sc.

Introduction and Motivation

- Fields of application for 3D models have been increasing in recent years
 - Rise in demand for 3D content
 - 3D reconstruction software gaining importance
- Several photogrammetry tools available (Agisoft Metashape, AliceVision Meshroom, COLMAP, ...)

Introduction and Motivation

- Lack of a common interface to use and compare 3D-reconstruction software
- Local limitations (RAM, CPU, GPU)
- Tedious work involved with local setups
- Rise of cloud-based services



Necessity for web-based solution

Existing Solutions / Related Work

Related Publications:

- “3D reconstruction from photographs by CMP SfM web service” (J. Heller, M. Havlena, M. Jancosek, A. Torii, and T. Pajdla)
- “3DNOW: Image-based 3D reconstruction and modeling via web” (Y. T. Tefera, F. Poiesi, D. Morabito, F. Remondino, E. Nocerino, and P. Chippendale.)

Existing Solutions / Related Work

Related Publications:

- “3D reconstruction from photographs by CMP SfM web service” (J. Heller, M. Havlena, M. Jancosek, A. Torii, and T. Pajdla)
- “3DNOW: Image-based 3D reconstruction and modeling via web” (Y. T. Tefera, F. Poiesi, D. Morabito, F. Remondino, E. Nocerino, and P. Chippendale.)



Code not available and little implementation details

Existing Solutions / Related Work

Existing cloud-based services:

- Autodesk ReCap Photo
- Agisoft Cloud
- PIX4Dcloud
- WebODM Lightning

Existing Solutions / Related Work

Existing cloud-based services:

- Autodesk ReCap Photo
- Agisoft Cloud
- PIX4Dcloud
- WebODM Lightning



Mostly paid and/or proprietary software

Goals of this Thesis

Develop and implement a web-based service with the following properties:

- Allows management of image-sets and related image-metadata
- Allows execution of reconstruction tasks using different photogrammetry software through a common interface
- Is maintainable and extensible

Goals of this Thesis

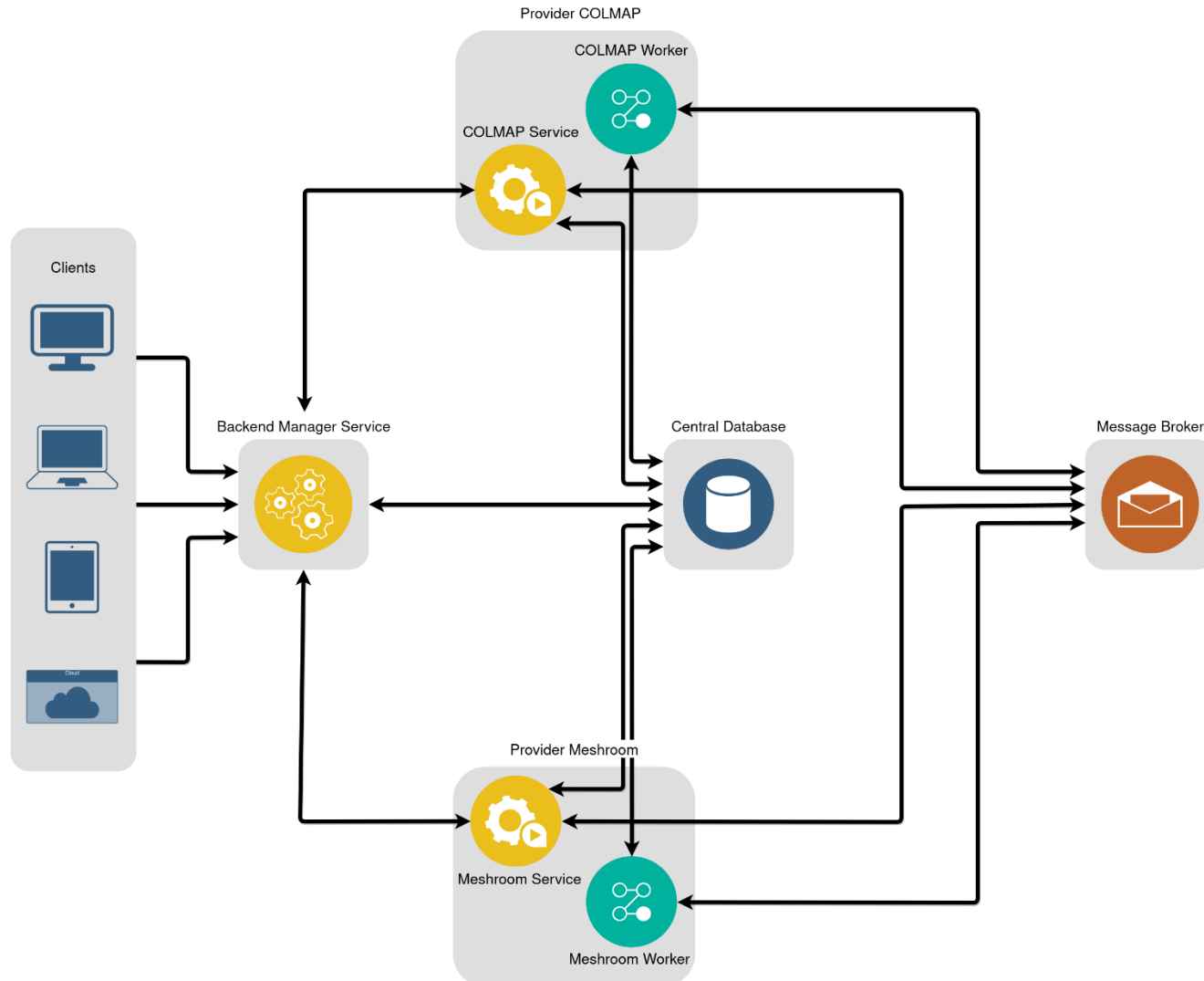
Develop and implement a web-based service with the following properties:

- (higher-level) Help expert and non-expert users to use a given photogrammetry software
- (higher-level) Help develop new software

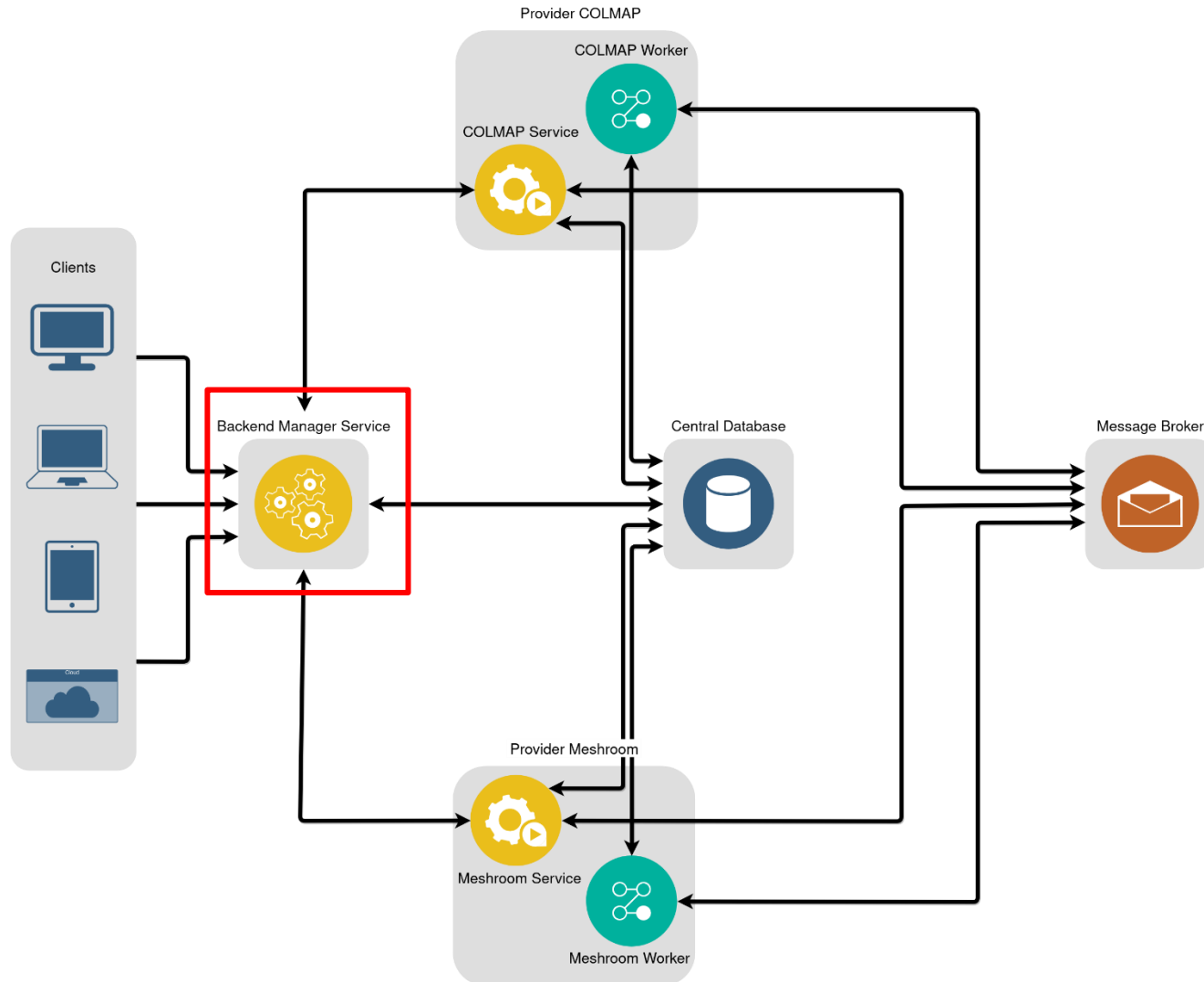
Critical Research Issues

- Reusability aspect
 - Suitable for different client applications
 - Generalized functionalities
 - Well documented API
- Future-oriented development
- Functional implementation

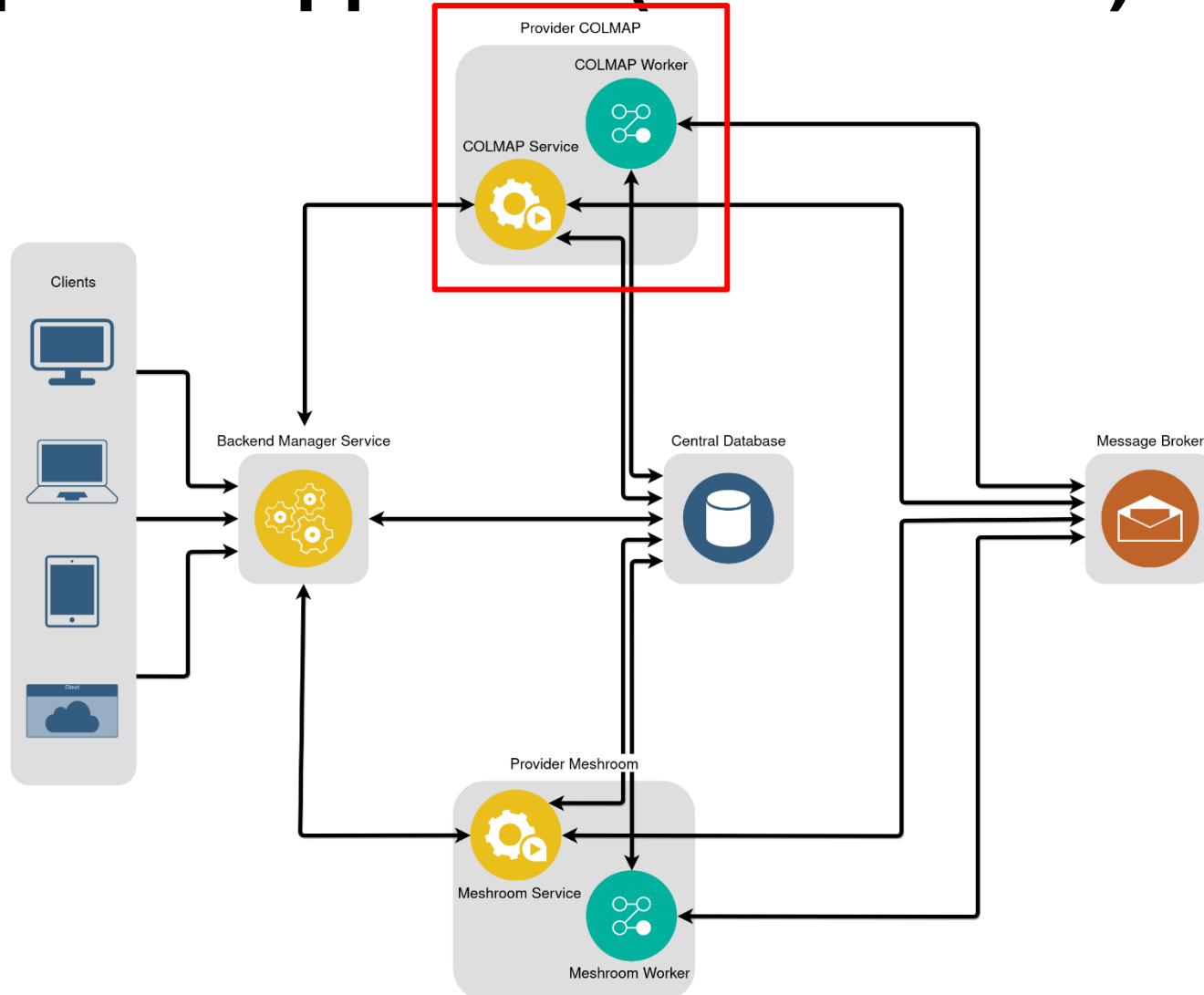
Proposed Approach (Architecture)



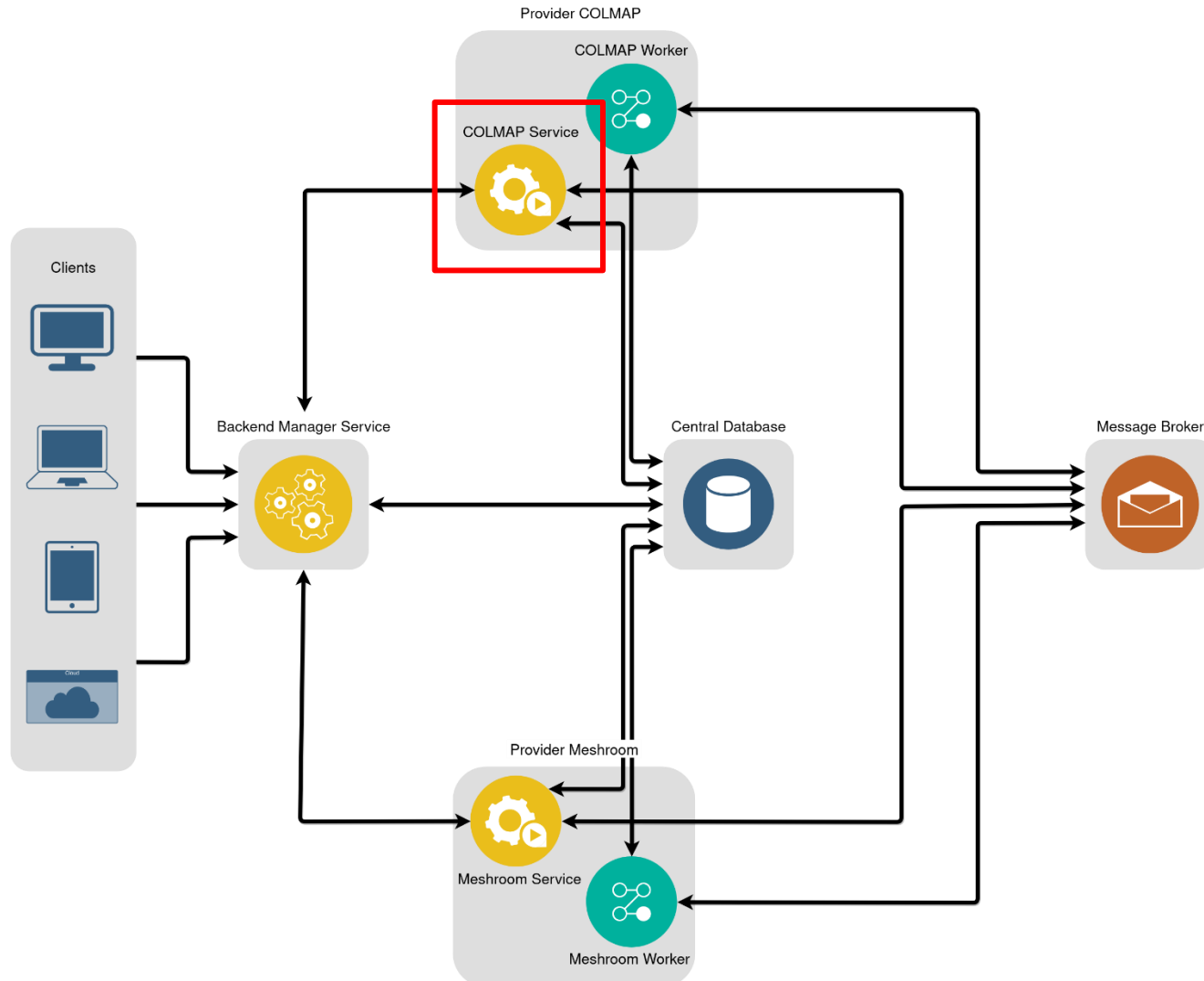
Proposed Approach (Architecture)



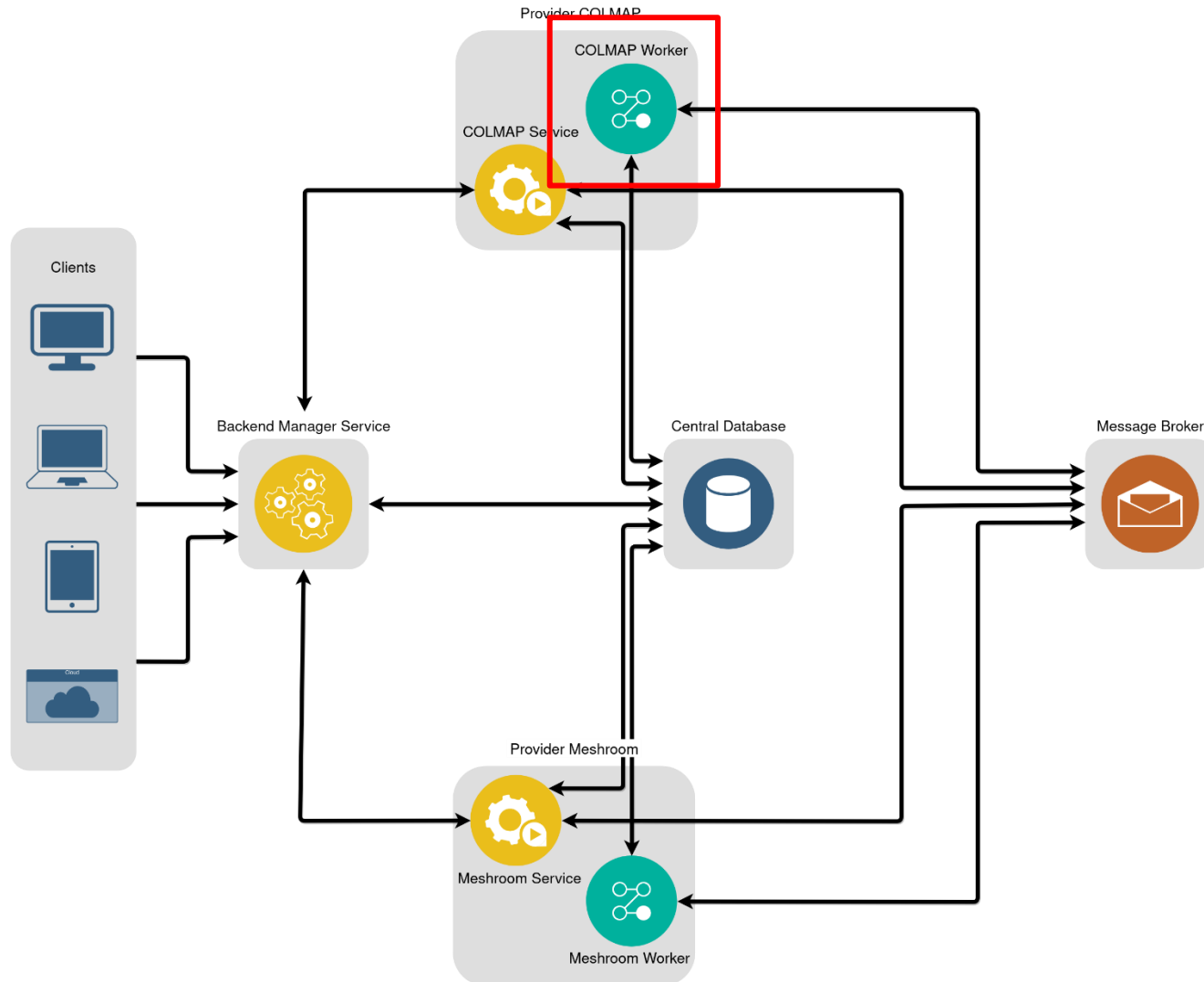
Proposed Approach (Architecture)



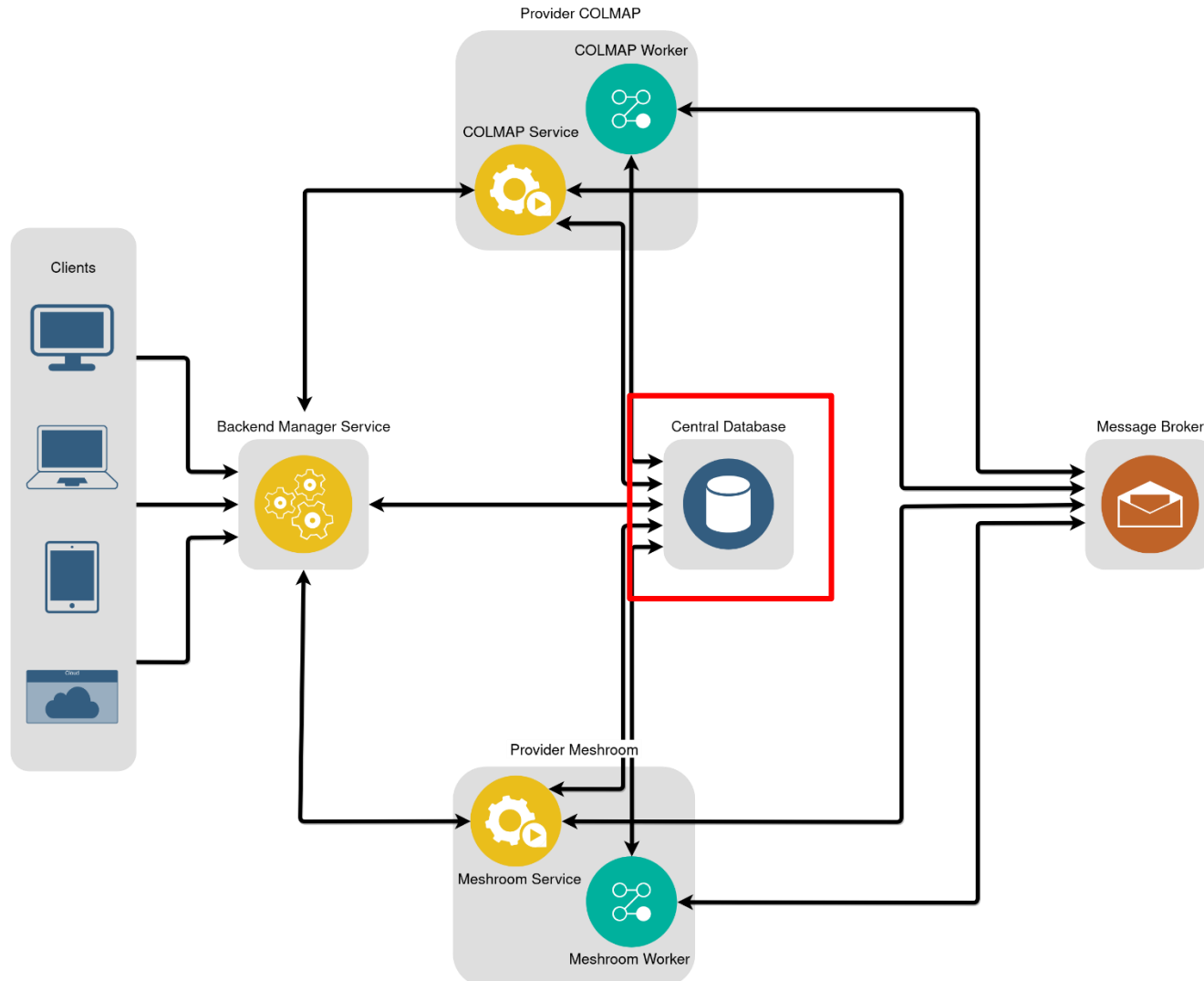
Proposed Approach (Architecture)



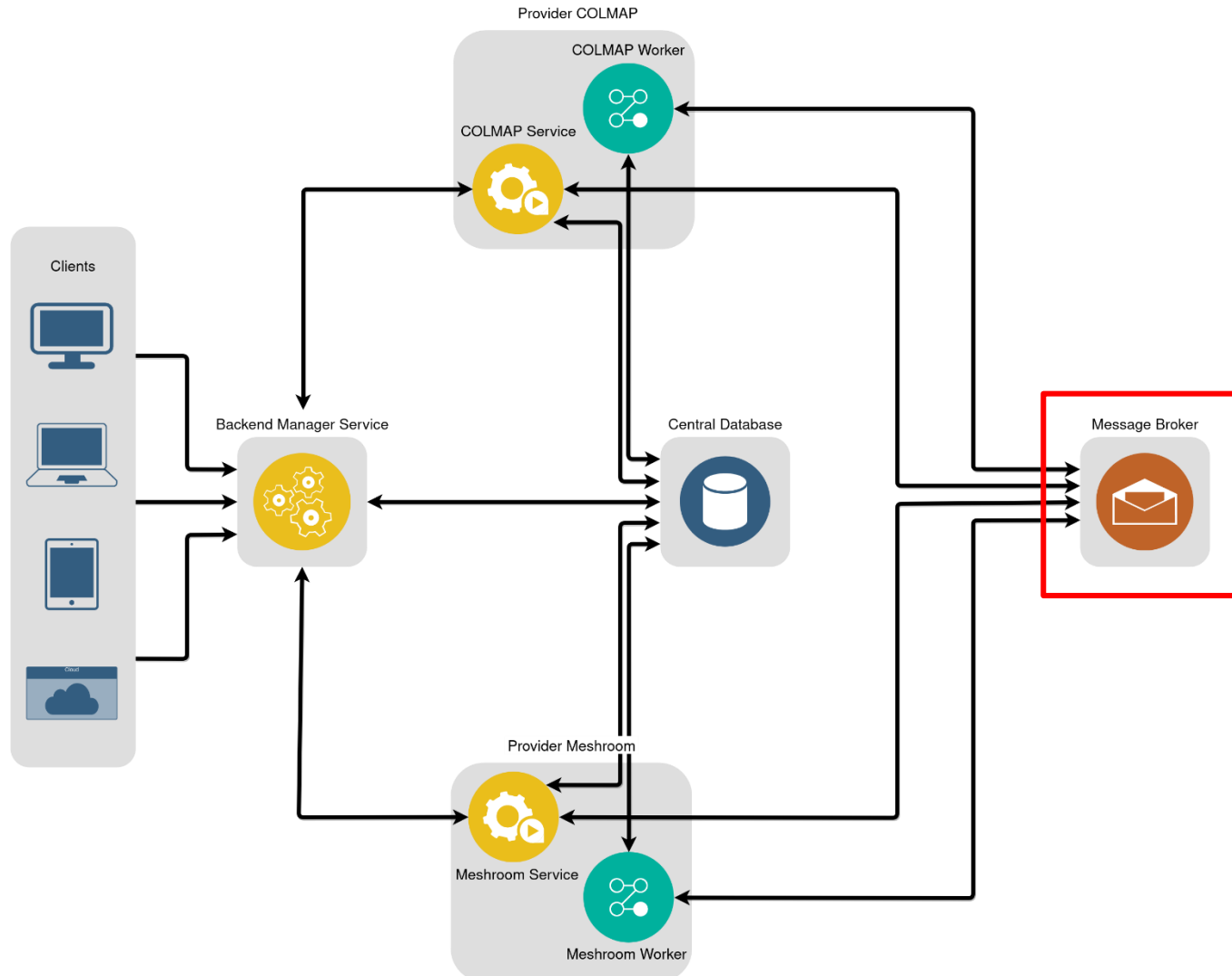
Proposed Approach (Architecture)



Proposed Approach (Architecture)



Proposed Approach (Architecture)



Proposed Approach (Architecture)

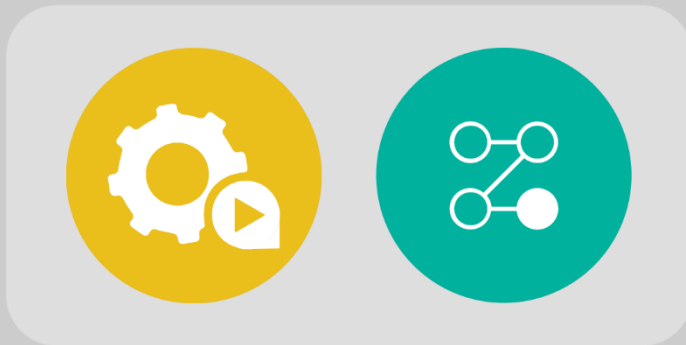
Backend Manager Service



- Main entrypoint of the application
- Manages images and image-metadata
- Integrates different “Providers”

Proposed Approach (Architecture)

Provider



- Encapsulates 3D-reconstruction software
- Responsible for implementation of task execution
- Consists of “Service“ and “Worker“ component

Proposed Approach (Architecture)

Provider Service



- Offers functionalities via REST calls
- Manages task execution and scheduling
- Controls the “Provider Worker”

Proposed Approach (Architecture)

Provider Worker



- Responsible for actual reconstruction process
- Use the software CLI

Proposed Approach (Architecture)

Database



- Saves the related metadata for image and project related data

Proposed Approach (Architecture)

Message Broker



- Responsible for communication between “Provider Service” and “Worker”
- Implements a message-queue for task-scheduling

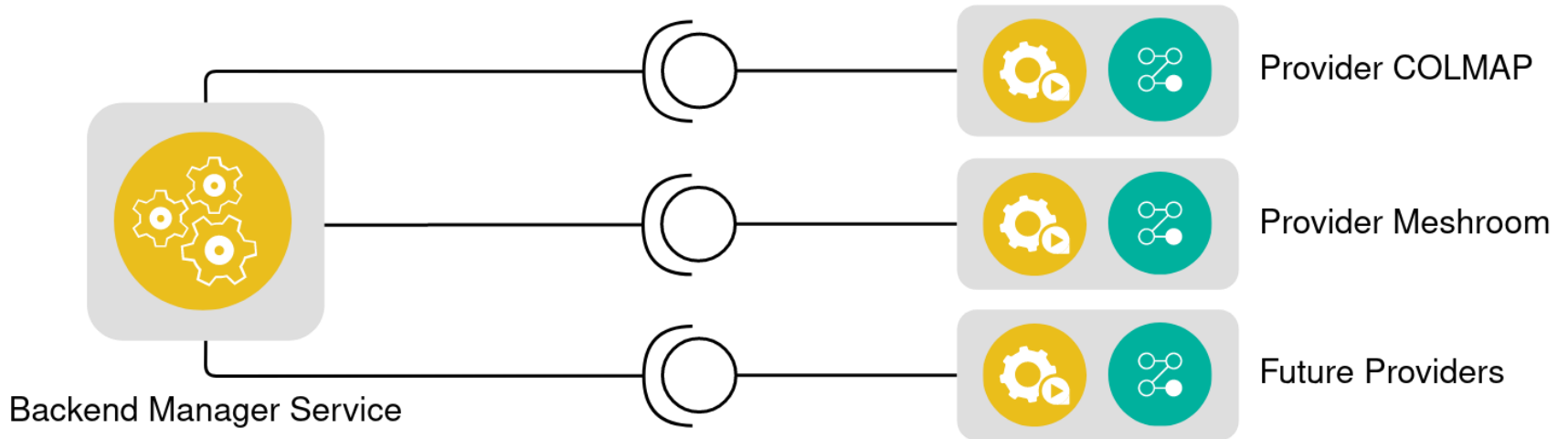
Proposed Approach (Design Decisions)

- Microservice Architecture
- Container-based deployment (Docker)
- “Provider Interface” – requirements for a “Provider”
- Task queue scheduling for reconstruction tasks
- Provider Subdivision (Worker and Service component)

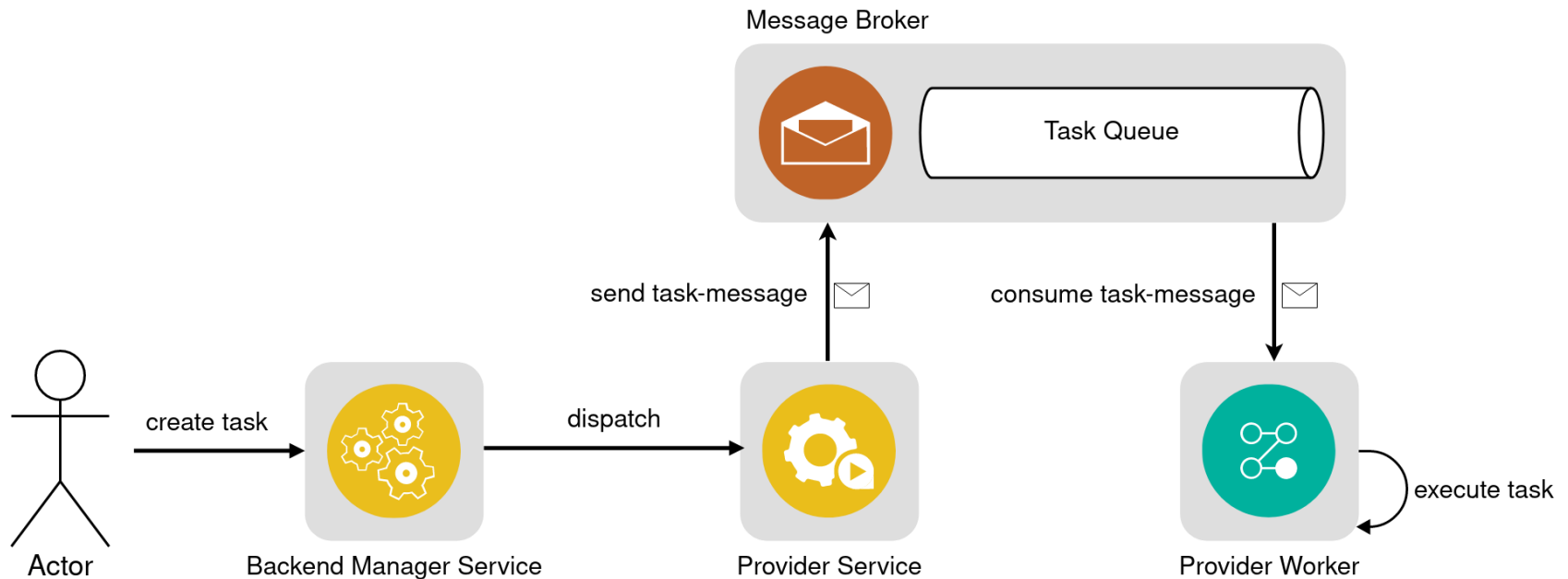
Proposed Approach (Provider Interface)

Required Endpoints

- /api/tasks (POST)
- /api/tasks/{task_id}/status (GET)
- /api/tasks/{task_id}/actions/cancellation (PUT)
- /api/task_types (GET)



Proposed Approach (Task Queue)



Proposed Approach (Task Queue)

Task Queue:

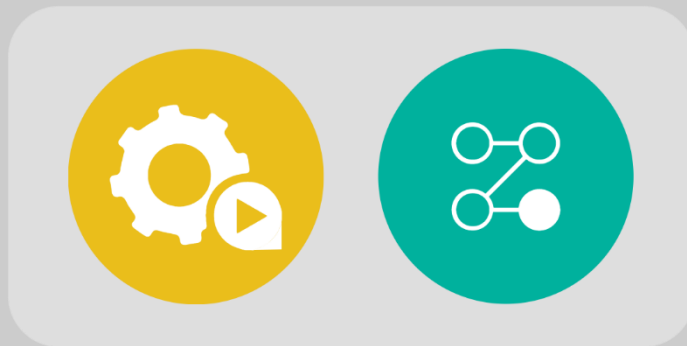
- Advisable for:
 - Resource-intensive background computations
 - Long and blocking tasks
- Allows for:
 - Improved scalability
 - Sequential or parallel execution by workers

Proposed Approach (Provider Interface)

- “Provider Interface” requires routes for task execution, status and cancellation as well as a list of offered offered task types
- Provider integration in the “Backend Manager Service” is managed in a single config file

Proposed Approach (Provider Subdivision)

Provider



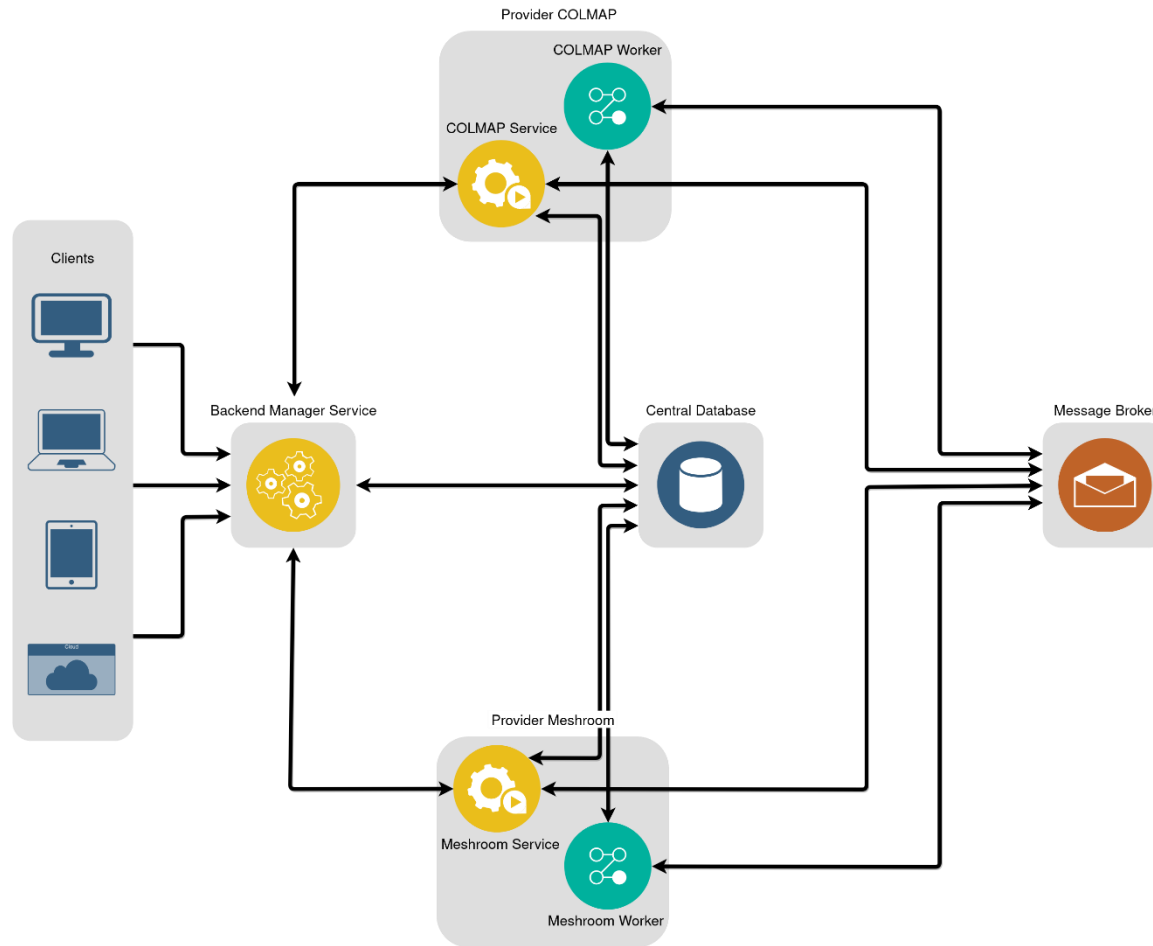
Service

Worker

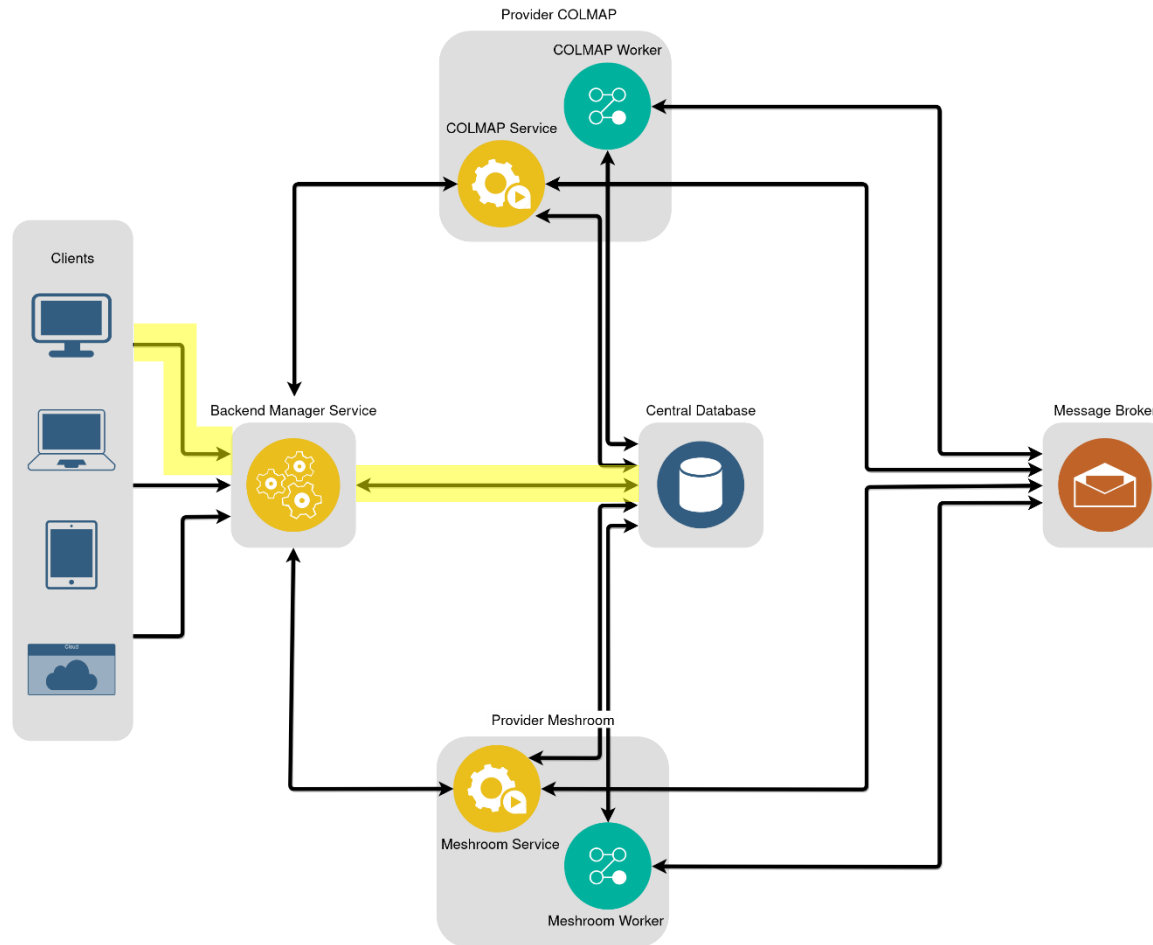
Provider is split up into a “Worker” and “Service” component:

- Allows independent scaling of “Service” (=managing) and “Worker” (=executing) components

Proposed Approach (Sample Workflow)

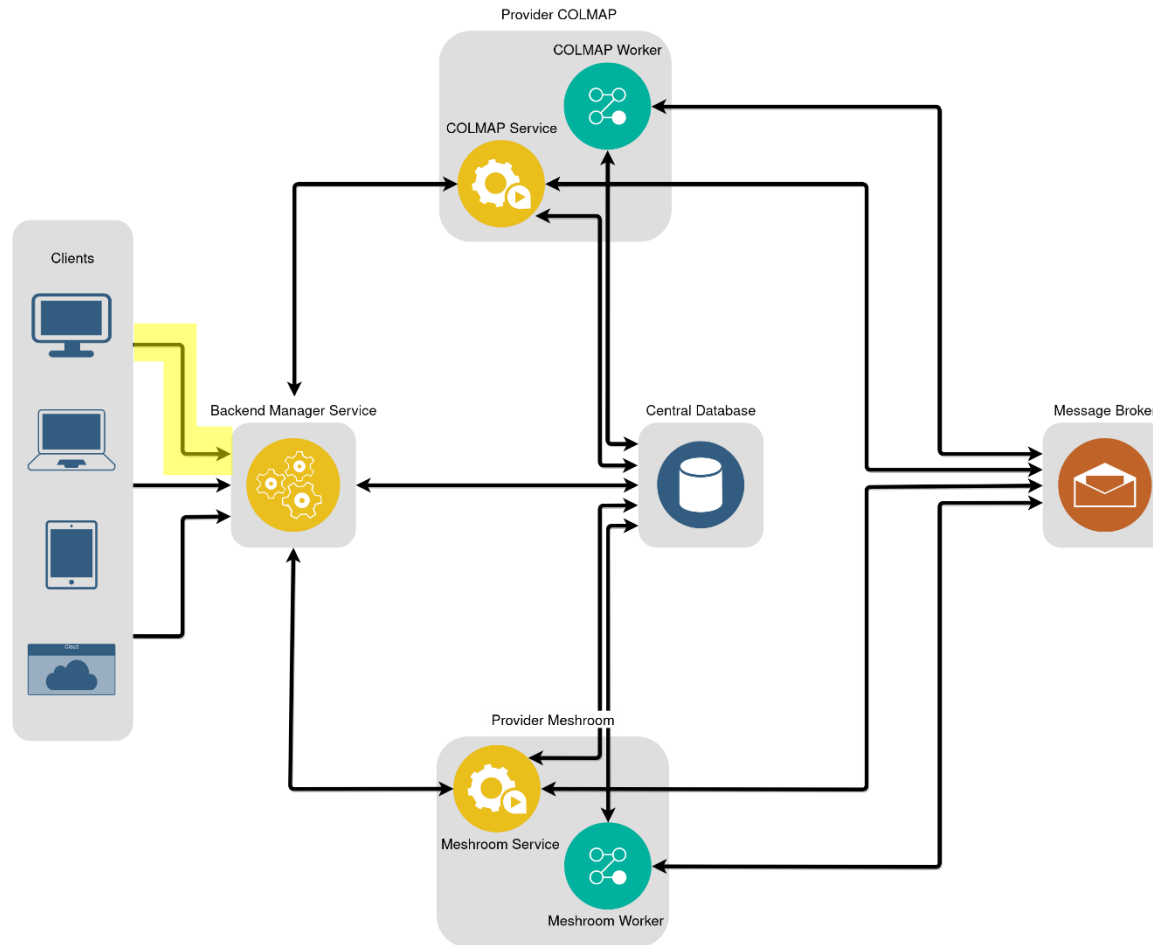


Proposed Approach (Sample Workflow)



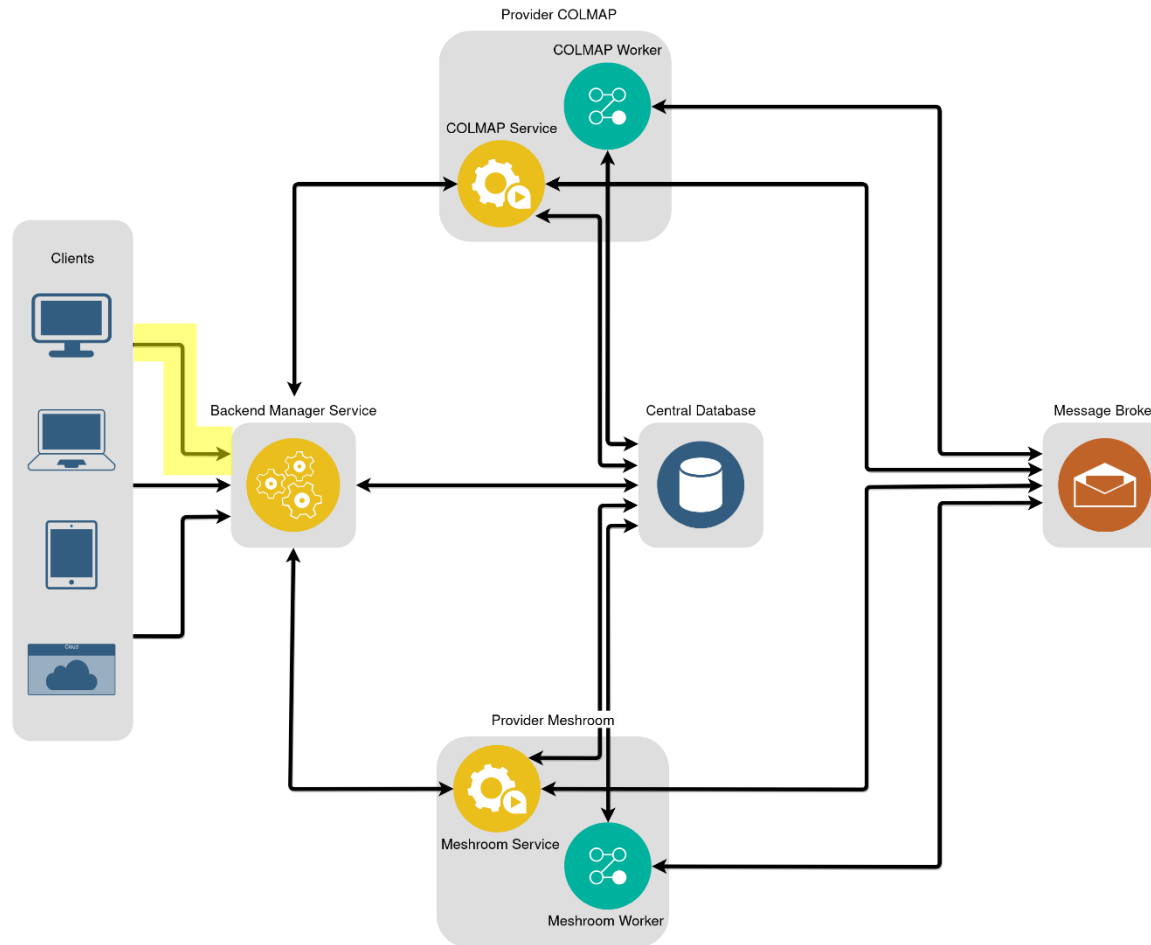
1. Create a project (image-set)

Proposed Approach (Sample Workflow)



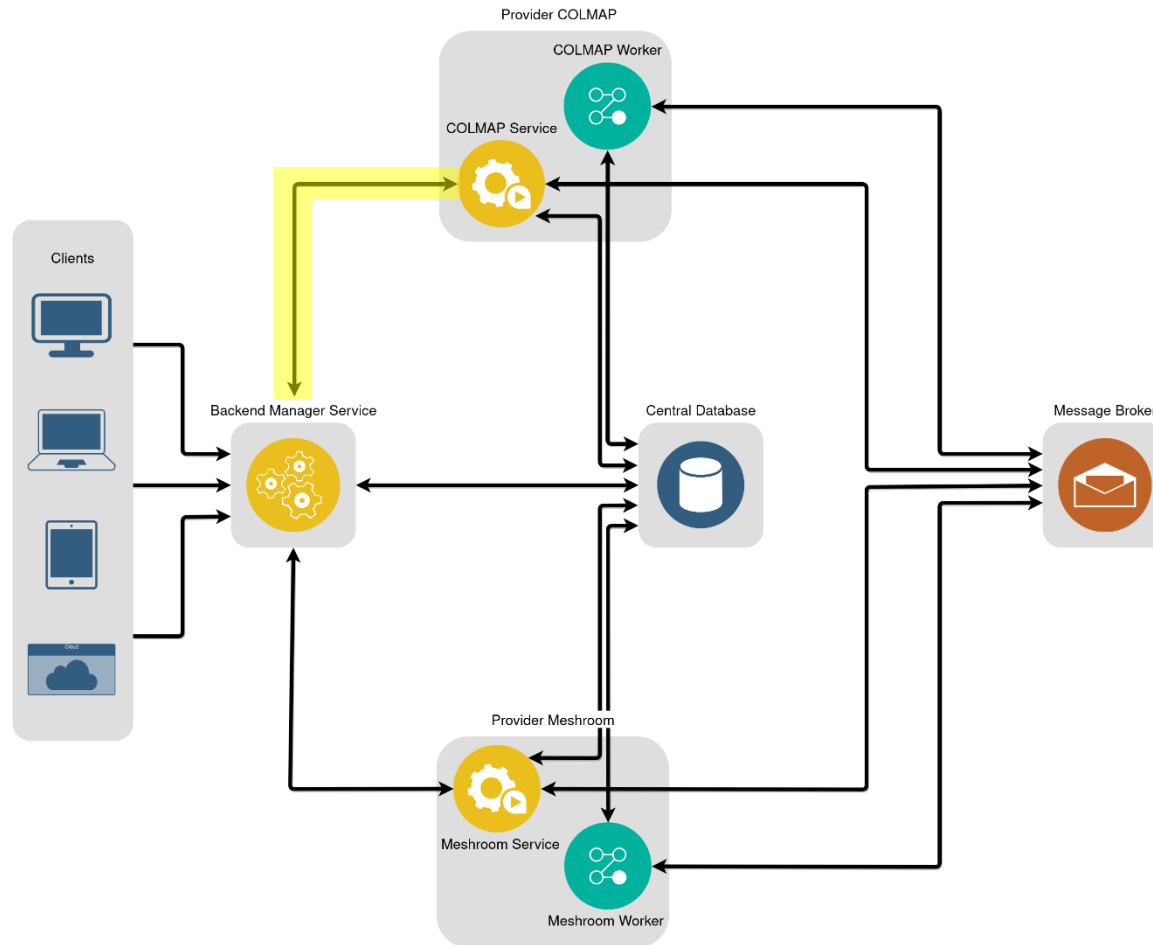
2. Upload images with corresponding metadata

Proposed Approach (Sample Workflow)



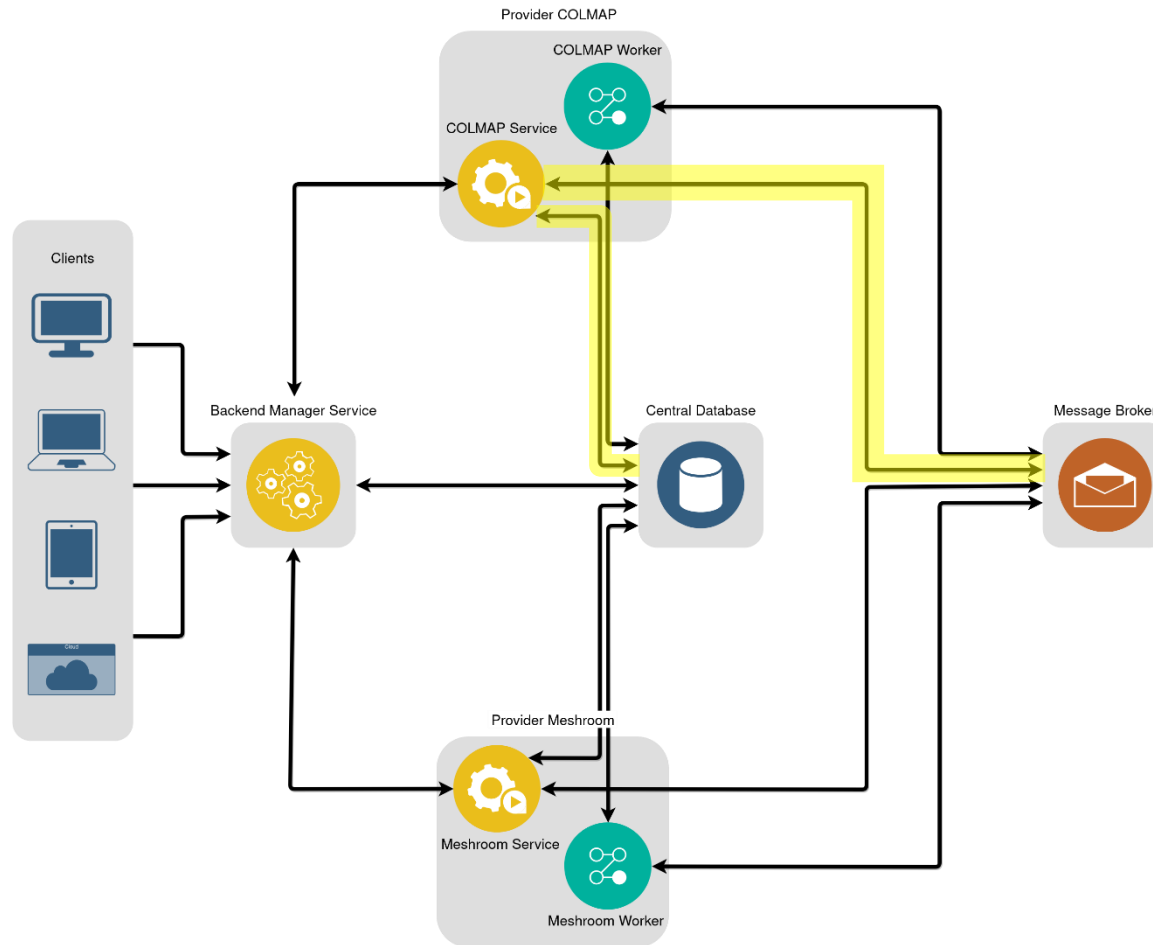
3. Trigger task execution with a selected provider

Proposed Approach (Sample Workflow)



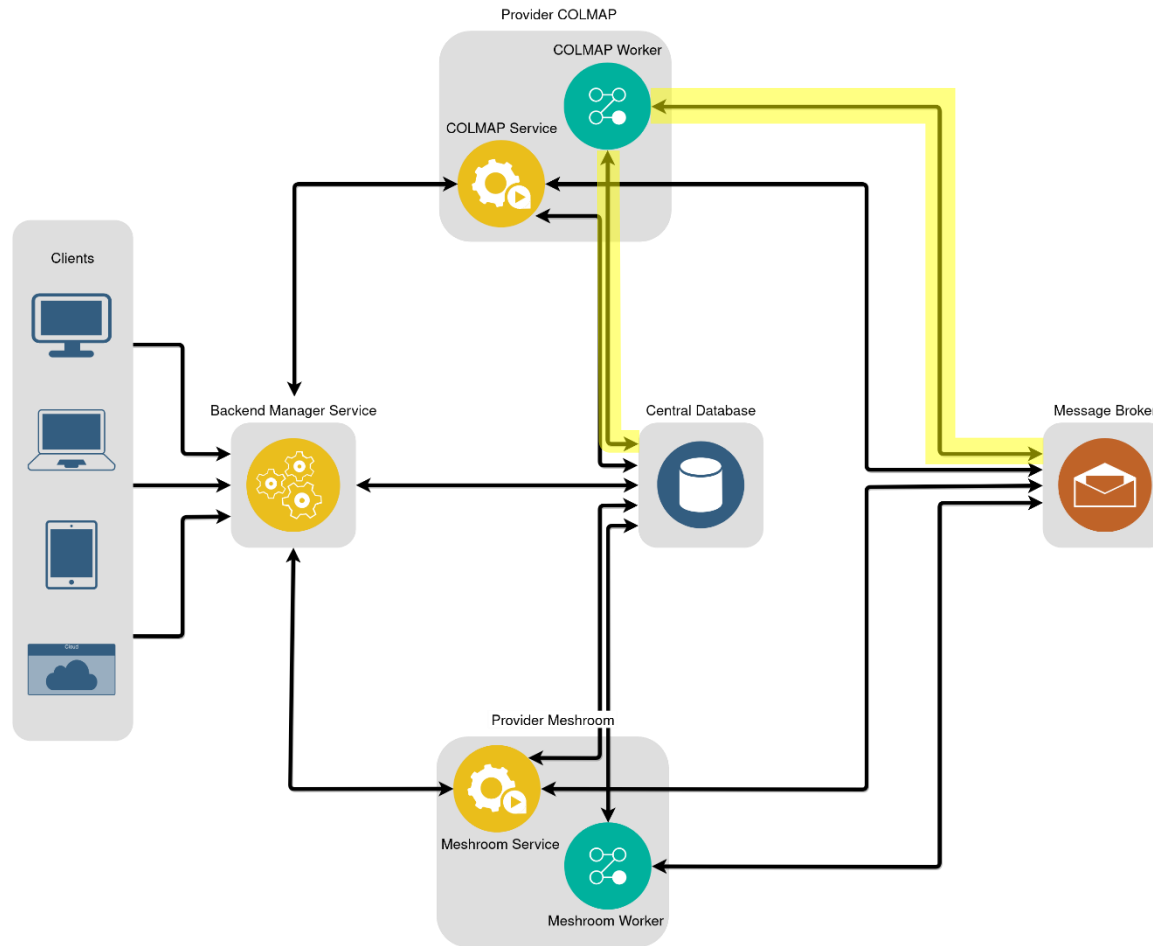
4. BMS redirects task request to provider (service)

Proposed Approach (Sample Workflow)



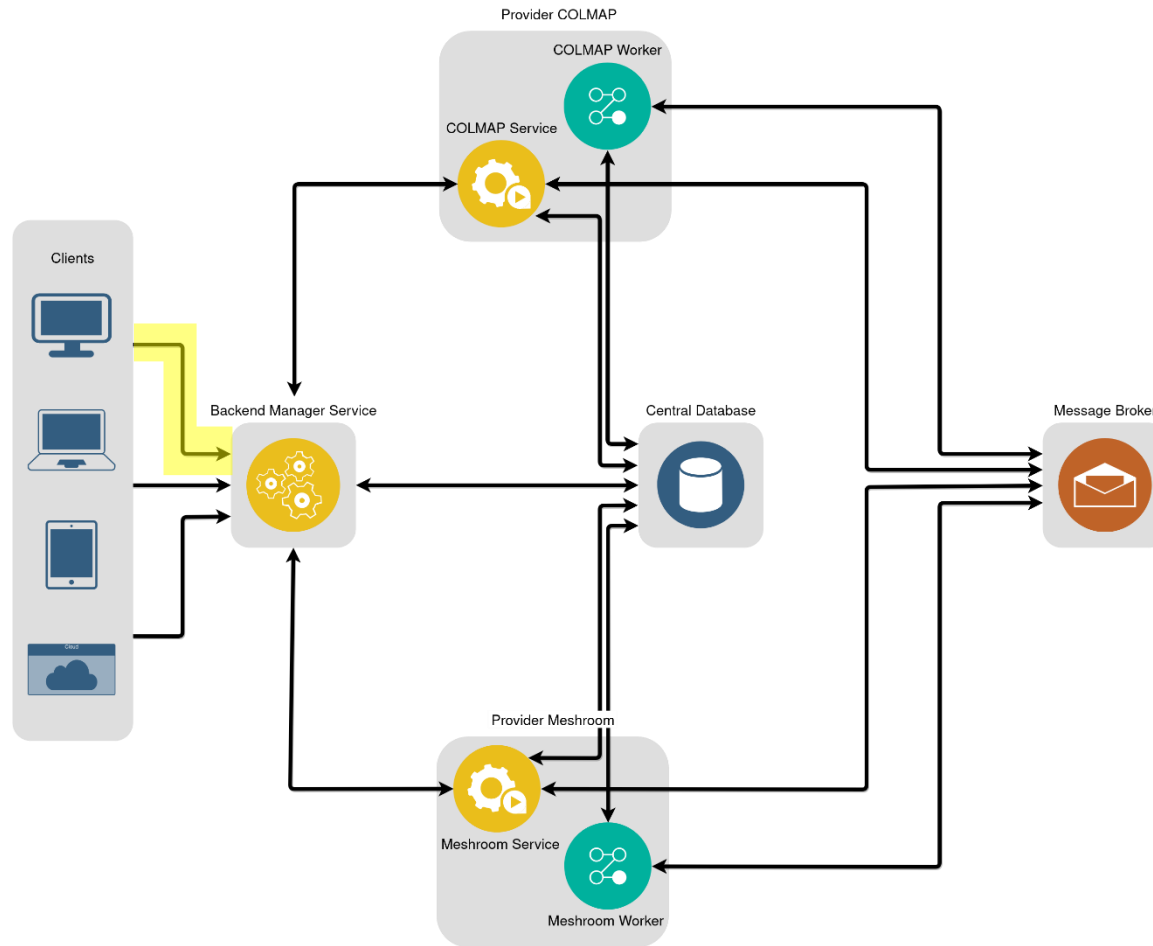
5. Provider Service creates and schedules task

Proposed Approach (Sample Workflow)



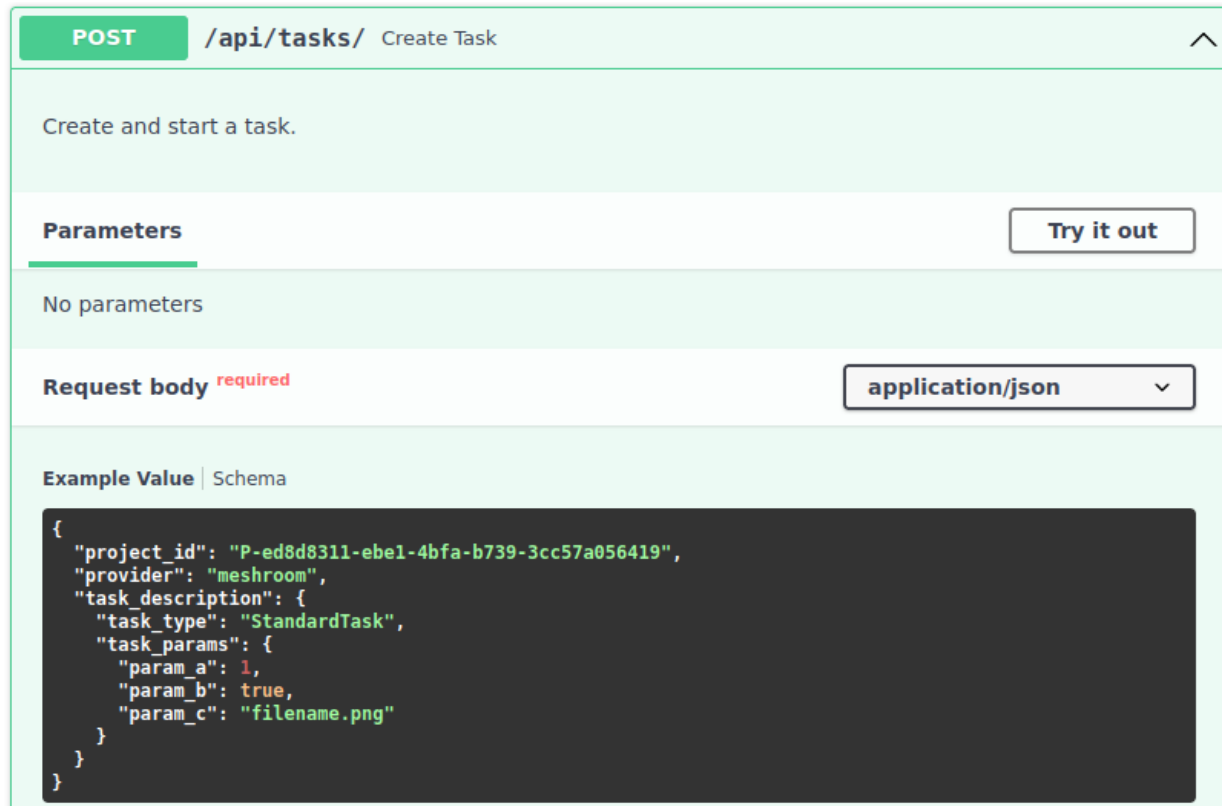
6. Provider Worker fetches and executes task

Proposed Approach (Sample Workflow)



7. (After task execution) Task outputs can be retrieved

Proposed Approach (Task Execution)



POST /api/tasks/ Create Task

Create and start a task.

Parameters Try it out

No parameters

Request body required application/json

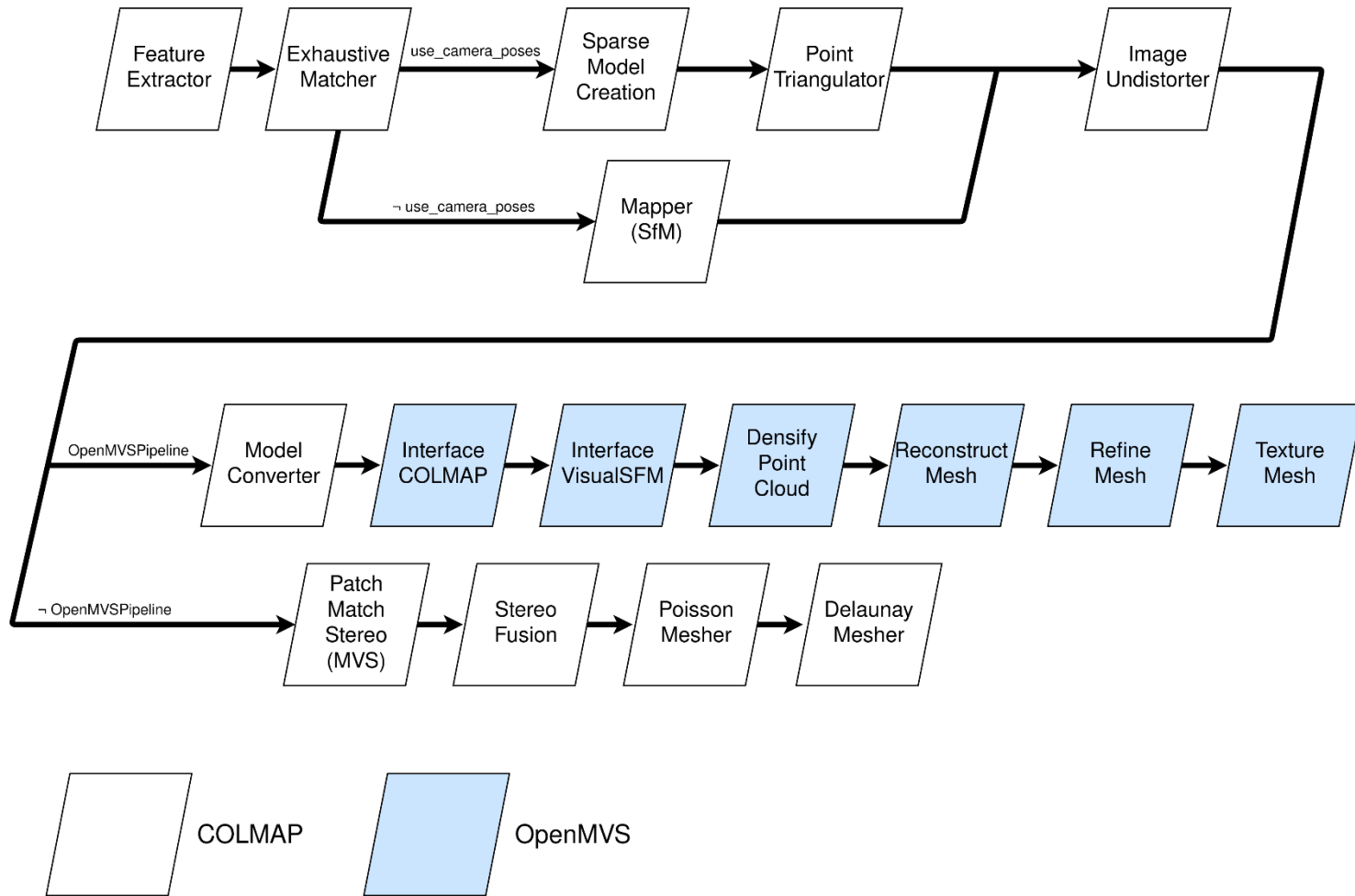
Example Value | Schema

```
{
  "project_id": "P-ed8d8311-ebe1-4bfa-b739-3cc57a056419",
  "provider": "meshroom",
  "task_description": {
    "task_type": "StandardTask",
    "task_params": {
      "param_a": 1,
      "param_b": true,
      "param_c": "filename.png"
    }
  }
}
```

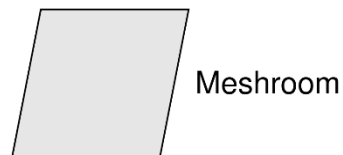
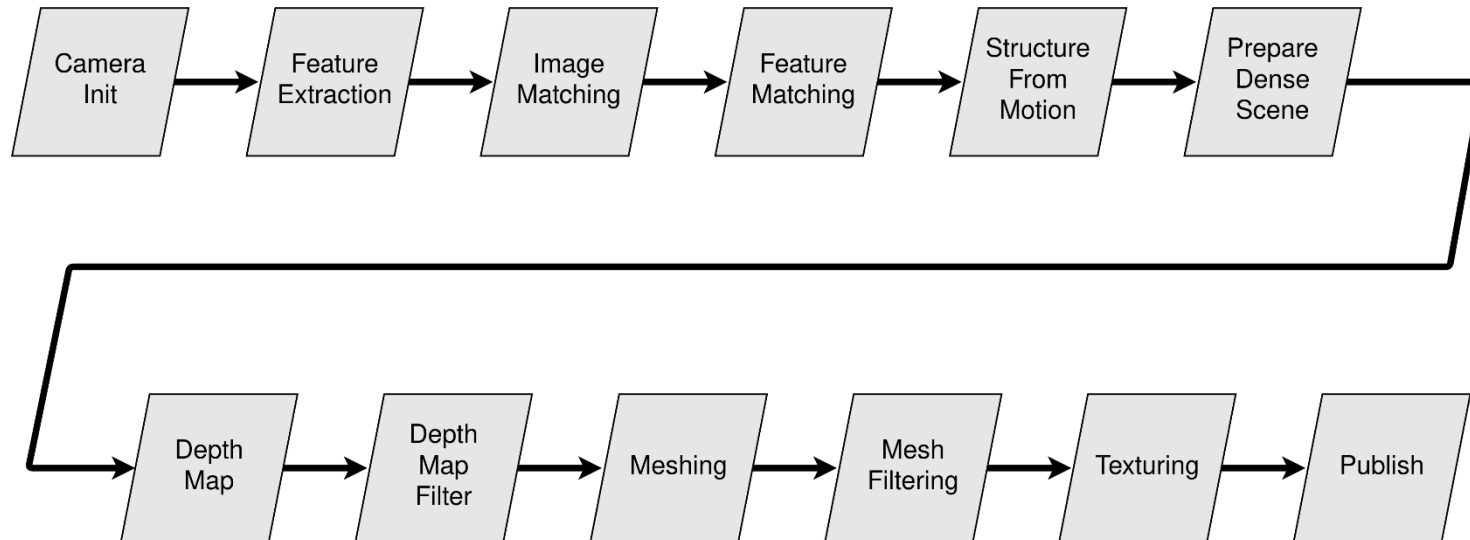
Proposed Approach (Provided Functionalities)

- Upload and manage image-sets and image-metadata
- Perform reconstructions using two providers
- Specify parameters for task execution
- Retrieve results
- Use of camera pose information (COLMAP)

Proposed Approach (Available Pipelines)



Proposed Approach (Available Pipelines)



Implementation

- Python
- FastAPI (Web microservices // Microservice architecture)
 - Pydantic
 - Swagger
- Docker (Containerization)
- Celery + RabbitMQ (Task Queue)
- MySQL (Database)

Evaluation

Structural and functional evaluation:

- Reusability of the API:
 - Evaluate how the API can be used by client projects
 - Evaluate Documentation
- Functional evaluation:
 - Testing API routes
 - Testing reconstruction pipeline

Evaluation (Results)

Structural and functional evaluation:

- API provides a wide range of options to manage image-sets and perform reconstruction tasks
- Well documented
- Errorless pipeline execution for all available pipelines*

*besides issues caused by local limitations (RAM)

Discussion / Reflection

- Error handling
- Logging
- Implementing AliceVision instead of Meshroom

Suggested Future Work

- Integration of more photogrammetry software
- Use of additional information (depth map information)
- Support for more input formats (e.g. video input)
- Modularization of pipeline steps
- Model evaluation
- Parameter tuning
- Scaling and cloud deployment

Conclusion

- Promising architecture
- Future oriented development
- Well-functioning web service and API
- Lots of potential for future features

List of References

1. Agisoft Metashape: <https://www.agisoft.com/>
2. AliceVision Meshroom:
<https://alicevision.org/#meshroom>
3. COLMAP: <https://demuc.de/colmap/>
4. J. Heller, M. Havlena, M. Jancosek, A. Torii, and T. Pajdla. “3D reconstruction from photographs by CMP SfM web service”
5. Y. T. Tefera, F. Poiesi, D. Morabito, F. Remondino, E. Nocerino, and P. Chippendale. “3DNOW: Image-based 3D reconstruction and modeling via web”

List of References

6. Autodesk ReCap Photo: <https://www.autodesk.com/products/recap/overview>
7. Agisoft Cloud: <https://cloud.agisoft.com/>
8. PIX4Dcloud. <https://cloud.pix4d.com/demo>
9. WebODMLighting: <https://webodm.net/>
10. FastAPI: <https://fastapi.tiangolo.com/>
11. Pydantic: <https://pydantic-docs.helpmanual.io/>
12. Swagger: <https://swagger.io/>
13. Celery: <https://docs.celeryproject.org/en/stable/>

List of References

14. RabbitMQ: <https://www.rabbitmq.com/>
15. MySQL: <https://www.mysql.com/>
16. Docker: <https://www.docker.com/>



Questions