

The Weather Archive

Semester Project Specification

Course: Serverless Computing (SERVL-S5)

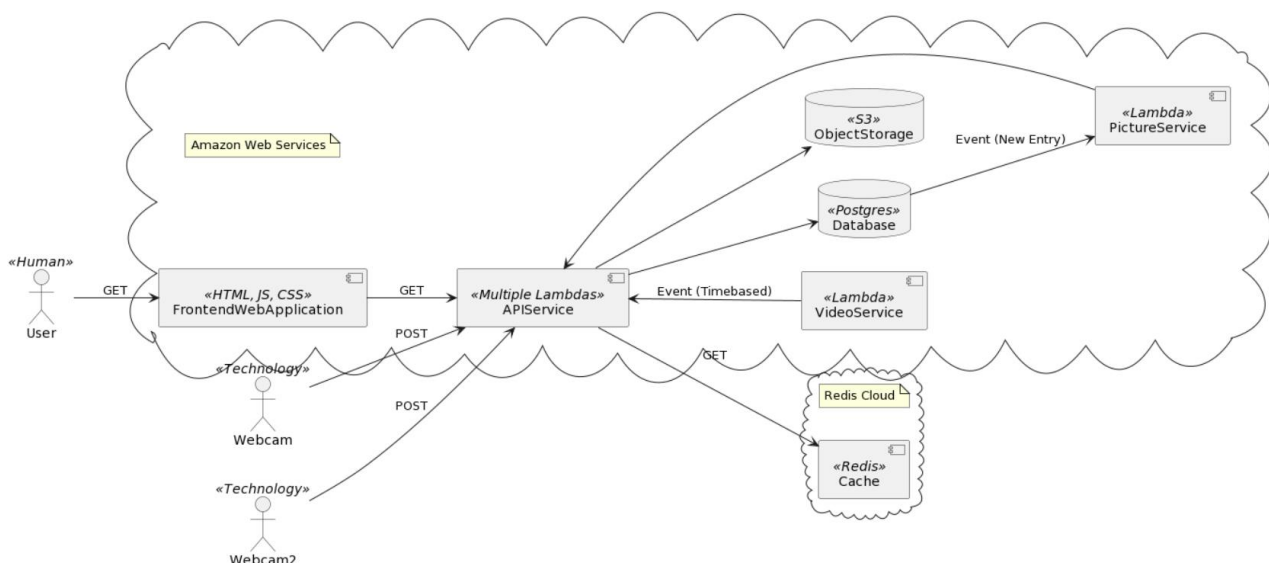
Context and Motivation

Your agency was hired by an Austrian company to create a new web application. This web application (the weather archive) will be used to display historical weather data. The company installed webcams in every major city in Europe. Including Vienna, Berlin, Paris and London. Instead of streaming the data live, these webcams will take pictures once every 5 minutes. This weather data will be used as a weather data archive with visuals. The goal is to have an API that can retrieve pictures (from the webcams, e.g. a picture of a location within a city), a topic (the cities, e.g. Vienna, Berlin, Paris, London, ...) and metadata about those topics (the weather data, e.g. temperature, humidity, air pressure, timestamp ...). The pictures from the webcam are in high resolution and should be compressed to a smaller size. The web app should be able to display a video (that combines all the pictures) and a plot with average values of the weather data (it should generate average values per hour).

Technical Specifications

The project should use the following technologies:

- Amazon Web Services
 - S3 Bucket – for the images
 - RDS Postgres – for data storage
 - Lambda's – where possible
- Redis Cloud
 - Redis – for cache



There are no specifications for the programming language. The Web Application should be accessible via a browser.

User Groups

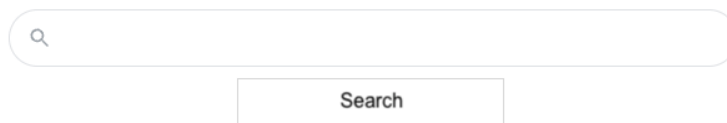
There are two different user groups:

- **Webcams:** can send data to the API (via an API Key).
- **Users:** can retrieve the weather data (video + plots for a specific city)

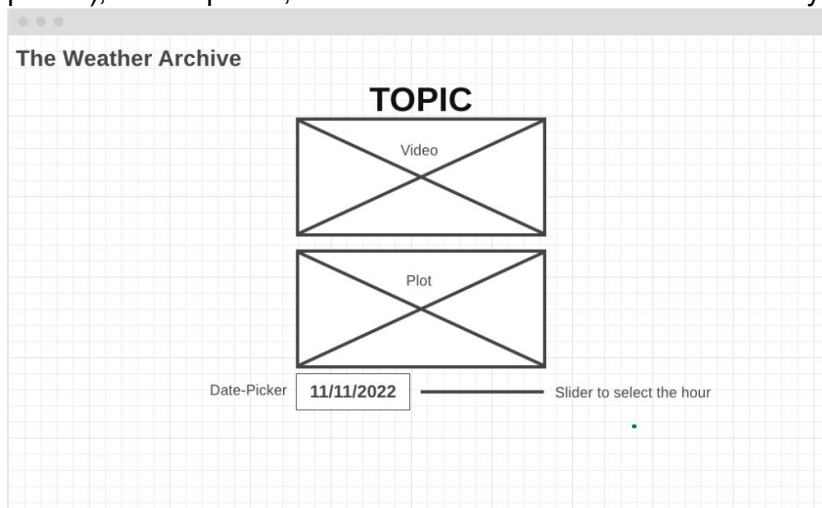
User Stories

- ✓ 1. As a user, I want to navigate to “The Weather Archive” application in the browser and see a logo and a search bar with a button. (similar to the Google Homepage).

The Weather Archive



- ✓ 2. As a user, I would like to type into the search bar and to retrieve autocomplete results (topics e.g. cities) in a list.
- ✓ 3. As a user, I am only able to choose topics (e.g. cities) from the autocomplete.
- ✓ 4. As a user, I want to retrieve an error message when I am searching for a topic that is not in the autocomplete list.
- ✓ 5. As a user, I would like to be taken to a new site when I select a valid topic from the autocomplete and click on the search button.
6. As a user, I would like to see a heading (topic, e.g. Vienna), a video (a compilation of the time period), a plot of the metadata (temperature, humidity, air pressure - of that time period), a date-picker, and a slider to select the hour of the day on the new page.



7. As a user, I am only able to select dates that are available.
8. As a user, I can only select hours that are available for the selected date.
9. As a user, I want to see the logo of "The Weather Archive" on the new page so that I can navigate back to the homepage when I click on it.
10. As a user, I would like to receive good (human-readable) error messages when something goes wrong.
11. As a webcam, I would like to be able to communicate via a RESTful Interface to POST data (an image, the topic, and metadata) to the API.
12. As a webcam, I want to use an API key when communicating with the API to ensure only authenticated users can POST data.
13. As a webcam, I would like to receive good error codes when something goes wrong.
14. As API, I would like to be able to store an image in an object storage.
15. As API, I want to be able to store data (image location, topic, timestamps, metadata) in a database
16. As API, I would like to be able to retrieve the data from the database
17. As API, I would like to be able to retrieve data from the S3 Bucket
18. As API, I want to be able to store data in the Redis Cache
19. As API, I would like to be able to retrieve data from the Cache
20. As API, I will not access the DB if the requested topic and time period is already in the Redis cache
21. As API, I will return the cached results if the requested topic and time period is already in the Redis cache
22. As PictureService, I want to get notified when a new entry was made to the database (e.g. image location)
23. As PictureService, I would like to be able to retrieve data from the S3 Bucket
24. As PictureService, I would like to be able to store data in the S3 Bucket
25. As PictureService, I want to be able to resize an image (compress)
26. As VideoService, I would like to be scheduled for a specific time period
27. As VideoService, I would like to be able to read the database
28. As VideoService, I would like to be able to create a video based on the images of that topic (that were sent from the webcam).
29. As VideoService, I will only create a video per time period (so a picture that was already used in a past video will not be used for the new schedule).
30. As VideoService, I want to be able to store a video in a S3 Bucket

Evaluation

The semester project will be evaluated based on the assessment criteria's that are available in the Moodle course.

Please note that the following 2 assignments count as part of the semester project

Hand-In 1: Frontend Draft

- You are required to create a draft of your frontend until the 7th unit
- The draft should be hosted on AWS and accessible from the internet
- You only need to upload screenshots from your draft implementation, but you have to do a live demo (open it in the browser & share the link) during the 7th unit
- The draft can have only static HTML and does not have to implement any user stories at this point
- You are free to modify or dismiss this frontend later during the project

Hand-In 2: Arc42 Draft

- You can upload the full specification at this point to get feedback on your planned work
- As this is still a draft, so you don't have to hand in all parts of the documentation. However, it should at least contain the following:
 - 4. Solution Strategy
 - 5. Building Block View

FAQs

Q: Do we have to create a video for the documentation:

A: Yes. As stated in “Assessment Criteria” in Moolde, please create “A 5-minute video of the functionality of the project.” The video should be a brief walkthrough of your project. It should show

- The working frontend in the browser
- The RESTful services you expose to the internet (e.g. output of PostMan or cURL)
- An overview on AWS services from the AWS console (e.g. API Gateway, Lambdas, Buckets ...)

Q: Do we have to use Arc42 for the documentation:

A: Yes. You don't need to fill out all parts of the template, but for the final hand in you should provide content for the following parts:

1. Introduction and Goals
3. Context and Scope
4. Solution Strategy
5. Building Block View
9. Architecture Decisions

The level of detail for the documentation should be sufficient so that a person who is new to the project would understand what was developed. You can start your Building Block View (5) with the diagram from this document and then drill down to component level (Lambdas, Gateway, Buckets ...). The Architecture Decisions (9) should make sure the reader understands why the solution and building blocks look the way they do.

Q: How is PictureService and VideoService connected to the rest of the infrastructure?

The **VideoService** should be triggered by a schedule.

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/RunLambdaSchedule.html>

The **PictureService** was originally supposed to be triggered by inserting data into the RDS database. However, due to limitations in the RDS event system, this may not be possible. You can choose any event source suitable for task.

Q: Does our VideoService have to render the images into a video?

Yes. Your system will expect one image per webcam every 5 minutes. Think of an adequate timespan for rendering videos. Also make sure you render the videos in a format that can be viewed in the browser.