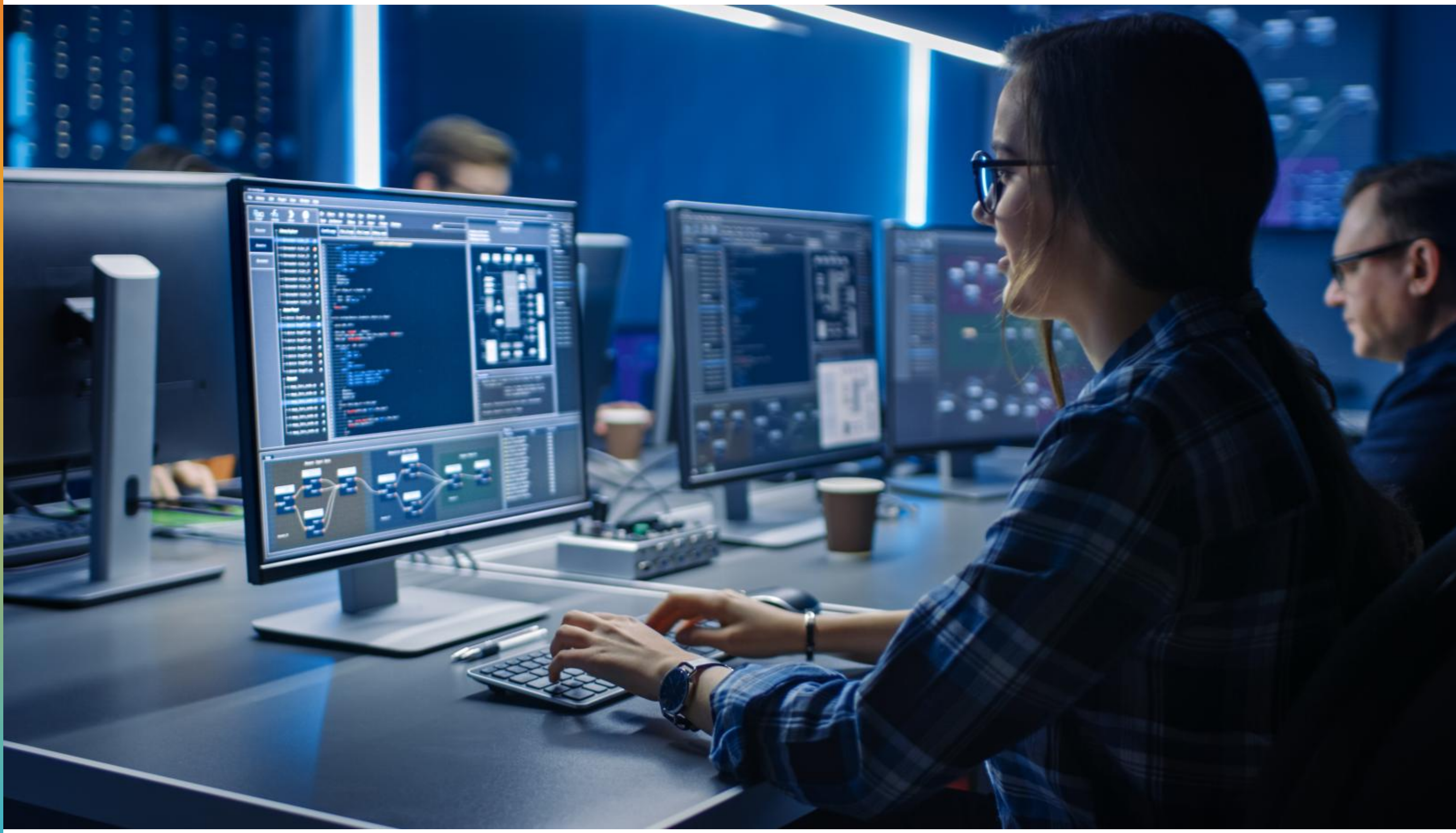
Development & Integration:

A Developer Guide

A comprehensive guide for developers and third party software providers to integrate their own City Climate Services and components into the CityCLIM ecosystem.



September 2024

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No *101036814*

Foreword

Dear Reader,

The CityCLIM project introduces an innovative approach to developing technologies that mitigate urban climate change. This developer guide serves as both an introduction and an invitation to developers and third-party software providers interested in extending and enriching our Generic City Climate Platform by integrating additional components and services to build together our ecosystem of next generation City Climate Services.

Our initiative is following the principles of collective expertise and co-development. Therefore, we have outlined the tools and support mechanisms CityCLIM offers to ensure seamless integration for software contributors like you. This developer's guide includes development guidelines with practical examples of integrations and is designed to simplify your first contributions to this dynamic platform.

Sincerely,

The CityCLIM Consortium

Table of Contents

1 Introduction 5

2 What CityCLIM offers to third-party software providers 6

3 Developer Manual 7

3.1 CityCLIM Overview: Architecture and GCCP 7

3.2 Development and integration of third-party components and services 9

3.2.1 Integration and Registration 10

3.2.2 Storage and Data Management 11

3.2.3 Interaction with the CityCLIM ecosystem 13

3.2.4 Monitoring and Alerting 15

4 Conclusion, Next Steps and Contact 19

5 References 20

Abbreviations

AM	Access Management
API	Application Programming Interface
CCS	City Climate Services
D	Deliverable
e.g.	Exempli gratia = for example
etc.	et cetera.
FP	Full Prototype
GCCP	Generic City Climate Platform
HTTP	Hypertext Transfer Protocol
i.e.	id est = that is to say
IM	Identity Management
JSON	JavaScript Object Notation
QoS	Quality of Service
REST	Representational State Transfer
RFC	Request for Comments
SDK	Software Development Kit
UHD	Ultra High Definition
URL	Uniform Resource Locator

1 Introduction

This document is a guideline and invitation for developers and third-party software providers who are interested in participating in the CityCLIM ecosystem with their own software components and services. It shows the benefits of participation and addresses directly software contributors by providing a comprehensive guide for the development and integration of their own services and software components into the CityCLIM ecosystem.

CityCLIM realised a digital ecosystem which is not limited to components and services developed within the origin project but is open for further developers and third-party software providers who are interested in participating and becoming part of the CityCLIM community. The CityCLIM ecosystem and community offer a platform suitable to exchange knowledge, generate and establish new business models, and cooperate and collaborate for the realisation of further next-generation City Climate Services.

As an enabler, CityCLIM realised the Generic City Climate Platform (GCCP) which is a software platform for City Climate Services (CCS) and backend components that offers standardised, secured and trusted communication within the CityCLIM ecosystem. The genericity of the GCCP enables software providers to participate in the CityCLIM ecosystem with own backend components and own services based on standardised interfaces.

How third-party software providers can benefit from CityCLIM, how they can participate, how they develop and integrate own services and components, and who are contact points if further support is needed, is in detail explained in this document following the structure below:

- **Chapter 2** describes the benefits of a participation in CityCLIM as a software provider. It also introduces which features are offered and gives an impression of what can and need to be done for their own software contributions.
- **Chapter 3** describes technical details of how third-party services and components can be developed, integrated, and registered in the CityCLIM ecosystem and also provides a range of implementation examples as additional material to simplify and support the development process.
- **Chapter 4** shows how to reach our support and contact points for any further information and support.

2 What CityCLIM offers to third-party software providers

The participation as a software provider in CityCLIM comes with a range of benefits and opportunities. First, you become part of a large community with already strong and well know partners (including the CityCLIM core team; see https://www.cityclim.eu/de/about) driving the generation and establishment of next-generation City Climate Services. Together we exchange ideas, and cooperate and collaborate on the development, linkage and roll-out of new software components and City Climate Services.

We come with several already implemented real-world City Climate Services successfully deployed in our European wide reference cities serving as initial exploitation areas for our services:

- Luxembourg City in Luxembourg
- Thessaloniki in Greece
- Valencia in Spain
- Karlsruhe in Germany

Becoming a participant in CityCLIM will accelerate your time to this new evolving market with wide exploitation fields. This we achieve by offering the usage of our GCCP to third-party software providers which comes already with a range of usable features and characteristics including:

- A comprehensive RESTful communication API which enables overall communication within components and services in the CityCLIM ecosystem.
- A secured and trusted communication environment realised through encrypted communication, unique software identifiers and API keys.
- An Identity Manager as OpenID Connect Provider offering a Single-Sign-On based authentication and authorisation mechanism usable for customers of your services.
- A data warehouse reachable over a RESTful-API to create, delete, use and monitor scalable storage spaces in our cloud infrastructure enabling to manage your data just for storing or to share with other components or services.
- A monitoring system which provides comprehensive Quality of Service (QoS) information, live data and alerts for your services and components.
- Several software components and services already in use, which data could be used as a basis/input for your developments.

Using all features of the GCCP enables third-party software providers to focus on their core business activities and reduces the work to few tasks, e.g.:

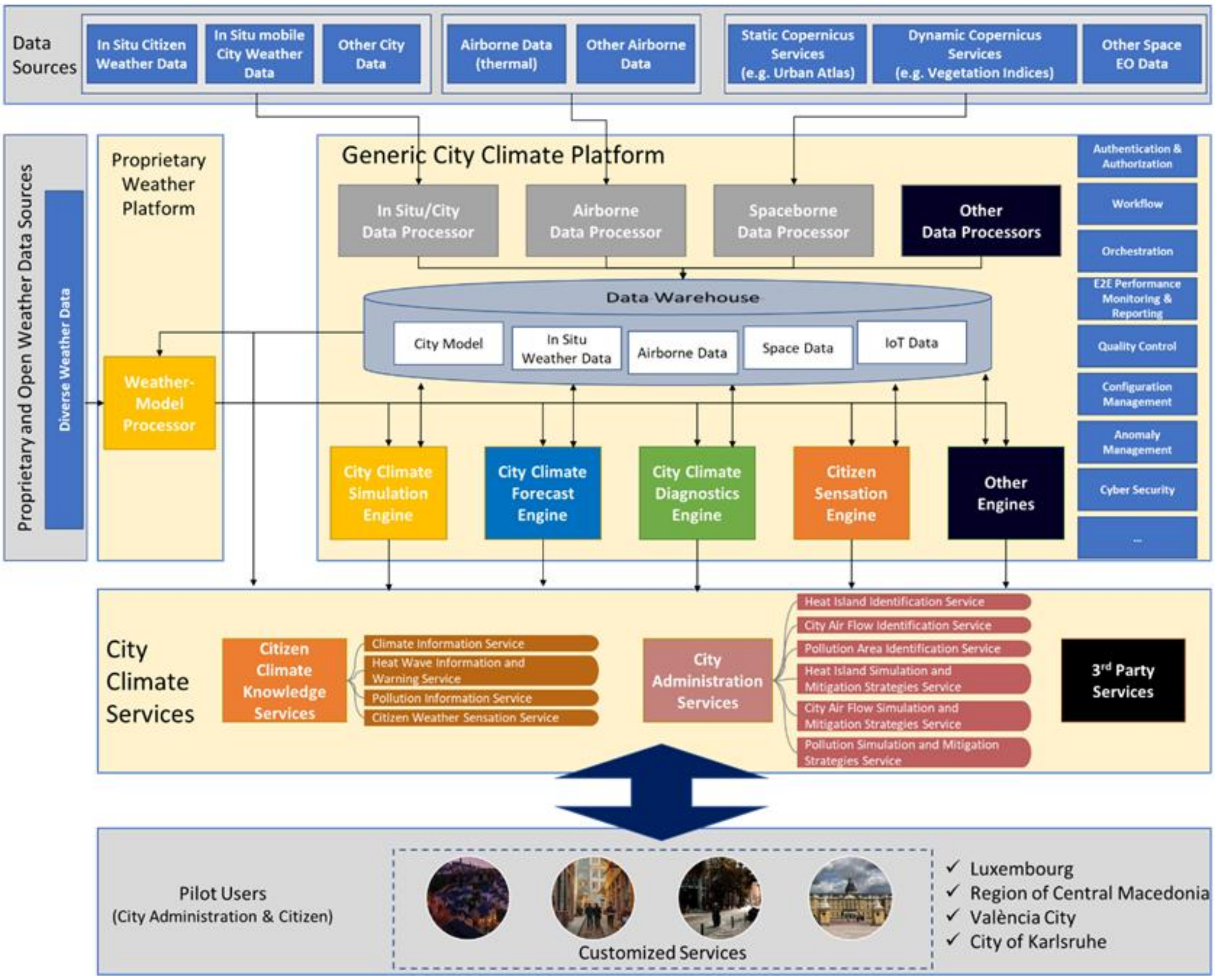
- Development and optimisation of software components and services.
- Easy integration and registration of the software components and services into the CityCLIM ecosystem.
- Attracting your target group to use your components and services.

3 Developer Manual

This chapter provides a technical overview of the CityCLIM software architecture, provided interfaces, processes and methods and describes how third-party software providers can develop, integrate and register further software components or services, how they can use the features of the GCCP and how they can use/link with components and services already operating in the CityCLIM ecosystem.

3.1 CityCLIM Overview: Architecture and GCCP

To introduce the CityCLIM ecosystem, this section describes the overall CityCLIM architecture and the software architecture of the GCCP.



Data Sources
In Situ Citizen Weather Data
In Situ mobile City Weather Data
Other City Data
Airborne Data (thermal)
Other Airborne Data
Static Copernicus Services (e.g. Urban Atlas)
Dynamic Copernicus Services (e.g. Vegetation Indices)
Other Space EO Data
Proprietary and Open Weather Data Sources
Diverse Weather Data
Proprietary Weather Platform
Weather-Model Processor
Generic City Climate Platform
In Situ/City Data Processor
Airborne Data Processor
Spaceborne Data Processor
Other Data Processors
Data Warehouse
City Model
In Situ Weather Data
Airborne Data
Space Data
IoT Data
City Climate Simulation Engine
City Climate Forecast Engine
City Climate Diagnostics Engine
Citizen Sensation Engine
Other Engines
Authentication & Authorization
Workflow
Orchestration
E2E Performance Monitoring & Reporting
Quality Control
Configuration Management
Anomaly Management
Cyber Security
...
City Climate Services
Citizen Climate Knowledge Services
Climate Information Service
Heat Wave Information and Warning Service
Pollution Information Service
Citizen Weather Sensation Service
City Administration Services
Heat Island Identification Service
City Air Flow Identification Service
Pollution Area Identification Service
Heat Island Simulation and Mitigation Strategies Service
City Air Flow Simulation and Mitigation Strategies Service
Pollution Simulation and Mitigation Strategies Service
3rd Party Services
Pilot Users
(City Administration & Citizen)
Customized Services
✓ Luxembourg
✓ Region of Central Macedonia
✓ València City
✓ City of Karlsruhe

Figure 3-1: Overall CityCLIM architecture.

Figure 3-1 shows the conceptual CityCLIM architecture which is a composition of four major modules:

- **Generic City Climate Platform** – The GCCP is the platform that connects operational software components such as Data Processors, which prepare raw data from connected data sources for further use within the CityCLIM ecosystem, and Engines, which create value out of prepared data that can be used by connected CCS, the end user services. The GCCP also acts as a data warehouse and provides a range of features to ensure a secure and trusted communication environment.
- **Data Sources** – Multiple data sources are connected to the CityCLIM ecosystem ranging from in-situ data to earth observation data.

- **Proprietary Weather Platform** – Showcases the connection of self-hosted third-party component and provides weather forecast information to engines hosted in the GCCP.
- **City Climate Services –** Are representing the next-generation City Climate Services and show a range of already implemented CCS.
- **Pilot users** – Represent the end users of CCS in our 4 reference cities. CCS may be customised for specific end-users in terms of language, usage behaviour, designs or further specific requirements.

A more detailed description of the overall CityCLIM architecture is given in “D1.3 – Public CityCLIM Concept” (CityCLIM Consortium, 2022).

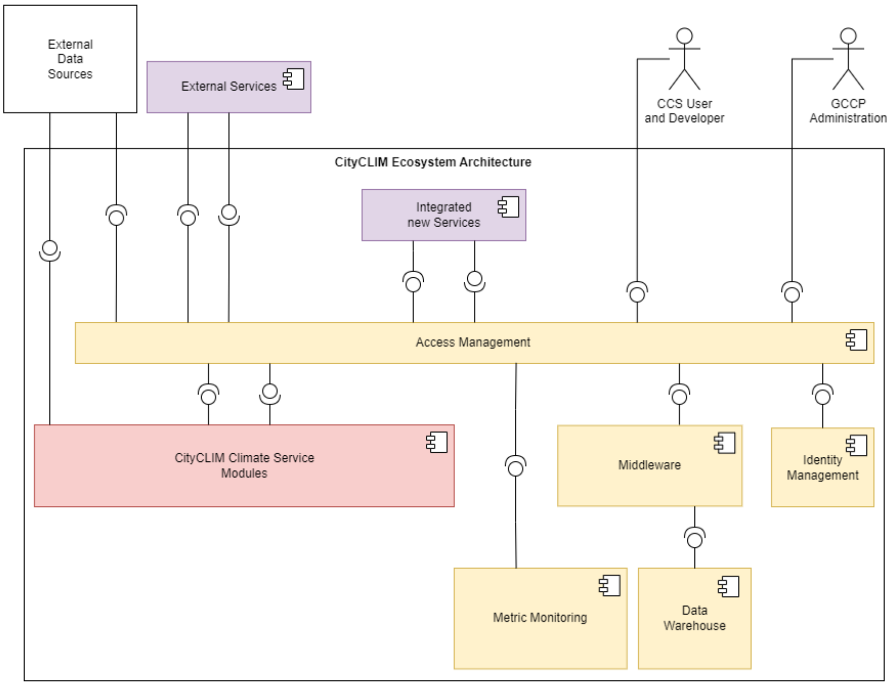


Figure 3-2: GCCP Software Architecture.

After giving an overview on the overall CityCLIM architecture, Figure 3-2 focuses on the software architecture of the GCCP where third-party components and services need to be integrated and registered. The yellow boxes are highlighting the components which are building the GCCP. The red boxes are highlighting the components and services connected to the platform and operating within the CityCLIM ecosystem. The purple boxes show how your self-hosted services and components would be integrated into the CityCLIM ecosystem. Furthermore, outside of the software architecture are located different users, as well as connected external data sources.

Table 3-1 gives a brief introduction into the yellow boxes/components that are building the GCCP.

Table 3-1: Introduction of components building the GCCP.

GCCP Component	Brief description
Metric Monitoring (MM)	Provides GCCP metric collection capacities, and support for visualization of these metrics (e.g., performance, service usage, data access, …).
Access Management (AM)	The GCCP component for managing external requests, which includes management aspects as routing, rate limiting, load balancing for APIs and DDoS protection for website endpoints.
Identity Management (IM)	The identity management system to integrate authentication and authorization mechanisms (e.g., Single-Sign On, account and token management)

GCCP Component	Brief description
Middleware	The management of external services by generic standardised APIs, access to data warehouses, and tracking of relevant information for metric monitoring. Usage of cloud-provider specific Software Developments Kits (SDK).
Data Warehouse	Data spaces for structured and unstructured data types.

As a third-party software provider who wants to operate their own self-hosted components and services within the CityCLIM ecosystem, you will in get in contact with these components in the development and integration process.

3.2 Development and integration of third-party components and services

Following an overview of the CityCLIM architecture and GCCP, this section outlines the steps that third-party service and component providers need to take to develop, integrate, and operate their components and services within the CityCLIM ecosystem.

Figure 3-3 shows the GCCP use cases relevant for third-party service and component provider which are described in Table 3-2.

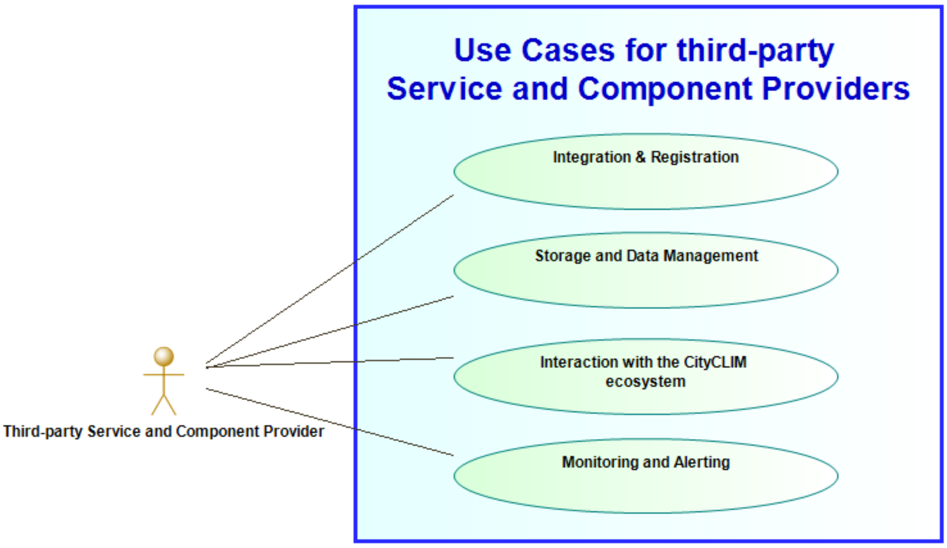

Use Cases for third-party
Service and Component Providers
Integration & Registration
Storage and Data Management
Interaction with the CityCLIM ecosystem
Monitoring and Alerting
Third-party Service and Component Provider

Figure 3-3: Use Cases for third-party Service and Component Providers.

Table 3-2: Descriptions of Use Cases for third-party Service and Component Providers.

Use cases	Description
Integration & Registration	Integration and registration of self-hosted third-party components or services into the GCCP ecosystem.
Storage and Data Management	Storage and Data management using the GCCP, which includes the creation/reservation of new storage spaces, data upload and data download, data space monitoring and further storage organisational aspects.
Interaction with the CityCLIM ecosystem	Interaction/communication with other components and services operating in the CityCLIM ecosystem.
Monitoring and Alerting	Monitoring and visualisation of metrics (e.g., performance, service usage, data access, etc.) and alerting for third-party components and services.

The following sections explain in detail the implementation of these use cases.

3.2.1 Integration and Registration

The GCCP allows participation to the CityCLIM ecosystem by the interaction with a personal user account or when implementing new services by a machine-to-machine communication. Both ways rely on state-of-the-art security policies and protocols.

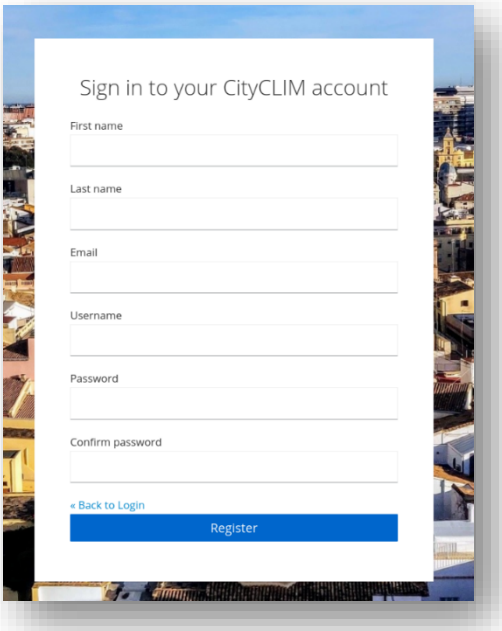
All necessary links, access data and relevant data for configuration will be provided after you got in contact with us (see section 4).

3.2.1.1 Interaction by a personal user account

A dedicated webpage allows users to register to the CityCLIM ecosystem.

On the bottom of the sign-in page (see Figure 3-4) click on “Register” to get to the registration page (see Figure 3-6).

Complete the registration form and click on “Register”. After a few moments, an email with a confirmation link will be sent. Clicking on this link completes the registration process.



Sign in to your CityCLIM account

First name

Last name

Email

Username

Password

Confirm password

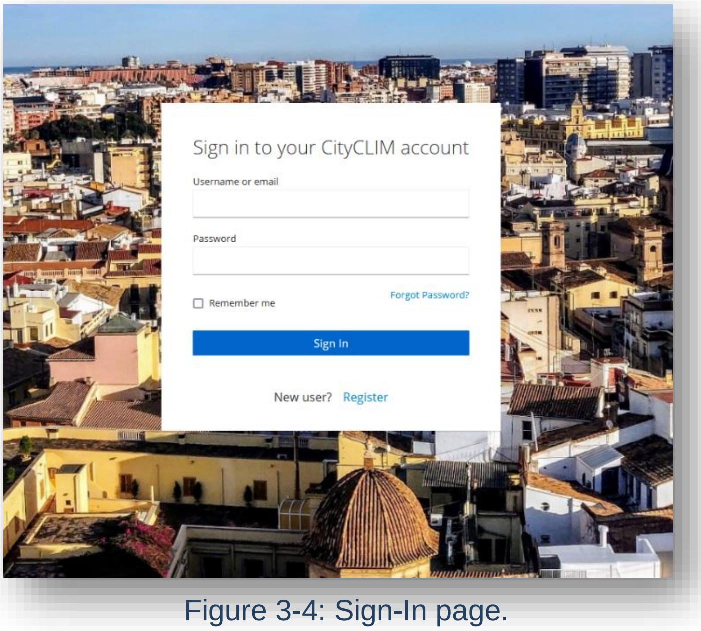
« Back to Login

Register

Figure 3-6: Registration page.

To change account information, such as name or email address, or to delete the account, navigate to the sign-in page, sign in, and on the account page, click “Personal info”. In the upcoming page (see Figure 3-5), perform the needed actions.

After successful registration, contact the technical team of CityCLIM again so that an authorized use of the requested interfaces specified in OpenAPI is ensured.



Sign in to your CityCLIM account

Username or email

Password

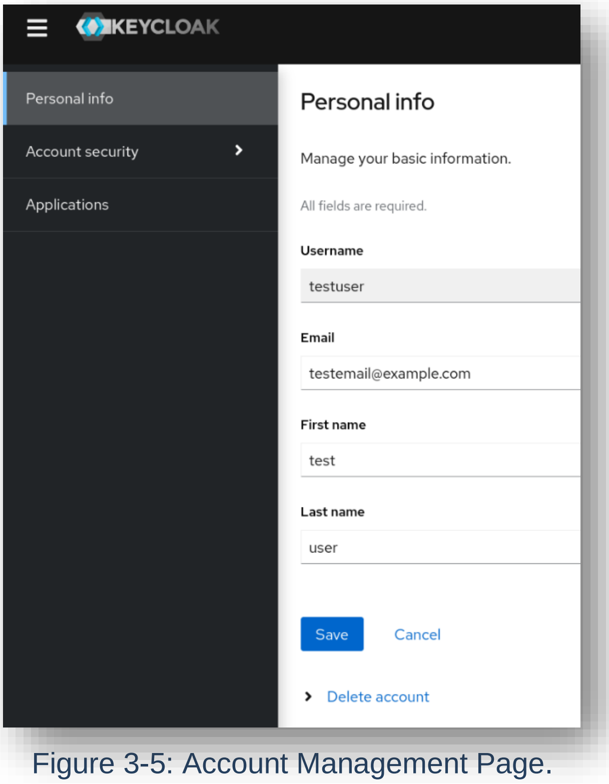
□ Remember me

Forgot Password?

Sign In

New user? Register

Figure 3-4: Sign-In page.



KEYCLOAK

Personal info

Account security

Applications

Personal info

Manage your basic information.

All fields are required.

Username: testuser

Email: testemail@example.com

First name: test

Last name: user

Save

Cancel

Delete account

Figure 3-5: Account Management Page.

To interact with RESTful CityCLIM interfaces, include your user credentials in the headers of each request by assigning your username and password to the header keys “`USERNAME`” and “`PASSWORD`”. Additionally, a middleware and security software developer kit (in Python) (introduced in section 3.2.2) can be provided to facilitate the communication with the platform.

The personal user account login is designed to be used especially for manual, low-frequent interaction with the CityCLIM ecosystem, which could be interesting, e.g., for data acquisition or testing.

3.2.1.2 Interaction via machine-to-machine communication

When extending the CityCLIM ecosystem by new services or components, authentication processes may occur at high frequency and require a time-efficient implementation. In that case, the GCCP supports a machine-to-machine communication approach, which involves the assignment of a unique API key. This way enables communication between the new service or component and the existing portfolio of CityCLIM and is in fact applied to all processing components of CityCLIM.

This registration process requires contact with the administration team of CityCLIM to ensure effective integration to the platform. The following steps are required:

- **Step 1:** Contact the technical support team of CityCLIM using the contact provided in section 4 for registration of the new service or component.
- **Step 2:** The technical support provides a token we refer as the “API key” which can be used for authentication and authorisation.
- **Step 3:** Attach the API key to the header key “`api_key`” of each request.

If, moreover, the service shall be integrated to the metric monitoring system of the GCCP, then the following information is required:

- URL of the service or component
- specification of interfaces
- service name.

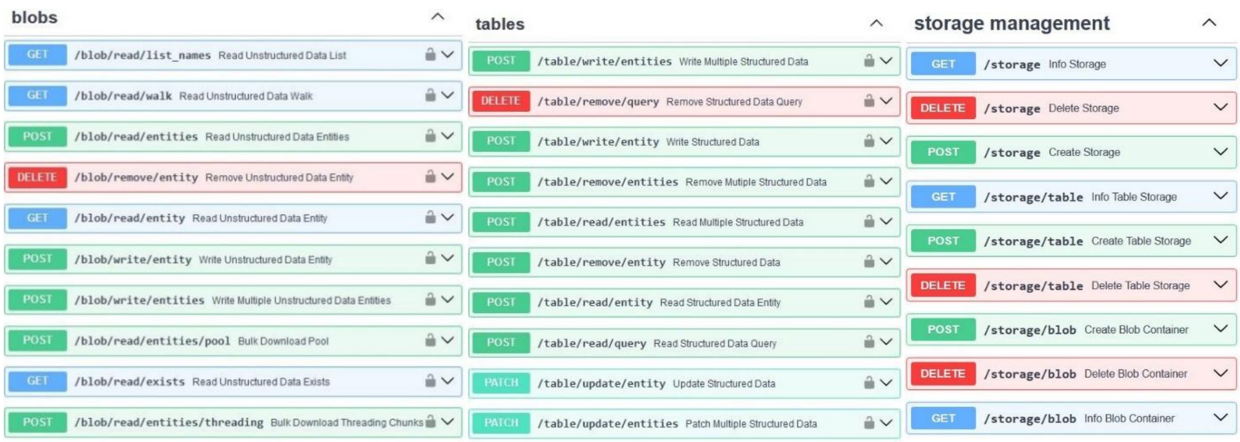
Additionally, registration of a CityCLIM user account is needed (see Section 3.2.1.1), which is used for entering the monitoring dashboards.

3.2.2 Storage and Data Management

The GCCP provides an API framework that handles data management and operations of integrated components and 3rd party services with respect to the Data Warehouse by APIs. These APIs are established by using cloud provider specific SDKs, and so are generic in the sense, that they offer data management and data operations that have no specific use cases, but rather cover the following general functionalities:

- Management of storage space (e.g., creation, deletion, internal storage-type specific organisation)
- Upload, patching, download, and deletion of (multiple) data (with bulk operations support)
- Querying data with filtering options to receive requested entities or corresponding metadata.

The functionality of the middleware is not one-to-one with the integrated cloud provider specific SDKs. In fact, it extends the portfolio by adding additional functionalities (e.g., bulk operations), which are not offered by the selected cloud providers. Each storage type has dedicated APIs for data operation actions and management.



blobs

GET	/blob/read/list_names	Read Unstructured Data List
GET	/blob/read/walk	Read Unstructured Data Walk
POST	/blob/read/entities	Read Unstructured Data Entities
DELETE	/blob/remove/entity	Remove Unstructured Data Entity
GET	/blob/read/entity	Read Unstructured Data Entity
POST	/blob/write/entity	Write Unstructured Data Entity
POST	/blob/write/entities	Write Multiple Unstructured Data Entities
POST	/blob/read/entities/pool	Bulk Download Pool
GET	/blob/read/exists	Read Unstructured Data Exists
POST	/blob/read/entities/threading	Bulk Download Threading Chunks

tables

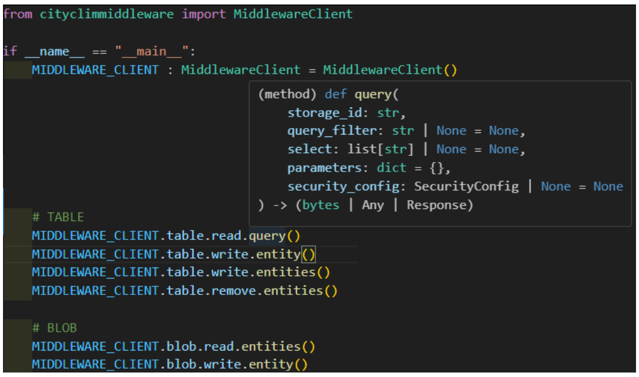
POST	/table/write/entities	Write Multiple Structured Data
DELETE	/table/remove/query	Remove Structured Data Query
POST	/table/write/entity	Write Structured Data
POST	/table/remove/entities	Remove Mutiple Structured Data
POST	/table/read/entities	Read Multiple Structured Data
POST	/table/remove/entity	Remove Structured Data
POST	/table/read/entity	Read Structured Data Entity
POST	/table/read/query	Read Structured Data Query
PATCH	/table/update/entity	Update Structured Data
PATCH	/table/update/entities	Patch Multiple Structured Data

storage management

GET	/storage	Info Storage
DELETE	/storage	Delete Storage
POST	/storage	Create Storage
GET	/storage/table	Info Table Storage
POST	/storage/table	Create Table Storage
DELETE	/storage/table	Delete Table Storage
POST	/storage/blob	Create Blob Container
DELETE	/storage/blob	Delete Blob Container
GET	/storage/blob	Info Blob Container

Figure 3-7: Overview of generic APIs available in the middleware FP.

A technical documentation for all APIs is available in accordance with the OpenAPI specification[1] and will be provided when you contact us. To support the usage of the GCCP APIs a SDK (in Python) has been development, which additionally accelerate the development of new services based on the CityCLIM platform. The so-called middleware SDK offers a programmatic approach to its API layer and includes all security relevant aspects. Figure 3-8 gives an impression of the SDK.



```
from cityclimmiddleware import MiddlewareClient

if __name__ == "__main__":
    MIDDLEWARE_CLIENT : MiddlewareClient = MiddlewareClient()
                            (method) def query(
                                storage_id: str,
                                query_filter: str | None = None,
                                select: list[str] | None = None,
                                parameters: dict = {},
                                security_config: SecurityConfig | None = None
                            ) -> (bytes | Any | Response)
    # TABLE
    MIDDLEWARE_CLIENT.table.read.query()
    MIDDLEWARE_CLIENT.table.write.entity()
    MIDDLEWARE_CLIENT.table.write.entities()
    MIDDLEWARE_CLIENT.table.remove.entities()

    # BLOB
    MIDDLEWARE_CLIENT.blob.read.entities()
    MIDDLEWARE_CLIENT.blob.write.entity()
```

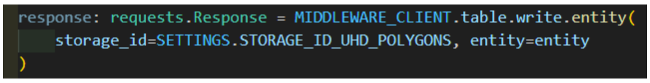
Figure 3-8: Initialisation of the Middleware Client using the Middleware SDK, including of an indication of different methods simplifying the access to the GCCP Data Warehouse.

To provide an implementation example for the storage and data management, the workflow of the Simulation Engine of the CityCLIM ecosystem is used. The Simulation Engine of the CityCLIM ecosystem is responsible for managing the workflows required to enable the so-called Simulation and Mitigation Services (see CityCLIM Consortium, 2024a). A prominent workflow of the simulation engine is the manipulation of earth observation data based on user-provided inputs. These inputs are given by polygons in JSON format, which define manipulated changes to local urban structures.

In this example, to accelerate and simplify transactions with the CityCLIM Data Warehouse, the middleware SDK is used in the implementation of the simulation engine, which now highlights several steps of the mentioned workflow.

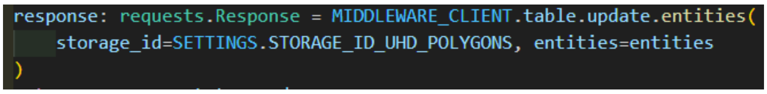
By a dedicated web interface called the Simulation Editor (see CityCLIM Consortium, 2024a), the user can make the manipulated changes to local urban structure. The Middleware SDK supports the maintenance of user-provided changes, providing methods for writing (see Figure 3-9), updating (see Figure 3-10), and removing (see Figure 3-11) structured entities.

[1] See https://github.com/OAI/OpenAPI-Specification/



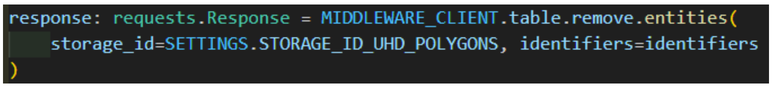
```
response: requests.Response = MIDDLEWARE_CLIENT.table.write.entity(
    storage_id=SETTINGS.STORAGE_ID_UHD_POLYGONS, entity=entity
)
```

Figure 3-9: Middleware SDK example for writing structured entities.



```
response: requests.Response = MIDDLEWARE_CLIENT.table.update.entities(
    storage_id=SETTINGS.STORAGE_ID_UHD_POLYGONS, entities=entities
)
```

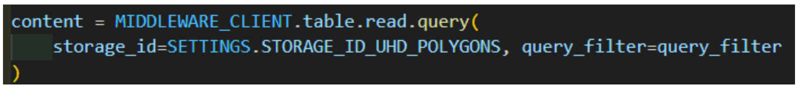
Figure 3-10: Middleware SDK example for updating structured entities.



```
response: requests.Response = MIDDLEWARE_CLIENT.table.remove.entities(
    storage_id=SETTINGS.STORAGE_ID_UHD_POLYGONS, identifiers=identifiers
)
```

Figure 3-11: Middleware SDK example for removing structured entities.

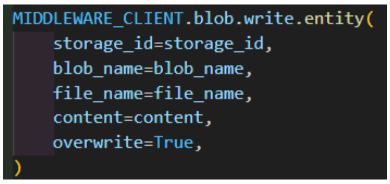
As a preparation of the earth observation data manipulation, user-provided polygons from the Data Warehouse are requested using a method that allows to query structured data based on a query (see Figure 3-12).



```
content = MIDDLEWARE_CLIENT.table.read.query(
    storage_id=SETTINGS.STORAGE_ID_UHD_POLYGONS, query_filter=query_filter
)
```

Figure 3-12: Middleware SDK example for requesting user-provided polygons from the Data Warehouse.

When the manipulation of the earth observation data is completed, the result is uploaded by the following method (see Figure 3-13).



```
MIDDLEWARE_CLIENT.blob.write.entity(
    storage_id=storage_id,
    blob_name=blob_name,
    file_name=file_name,
    content=content,
    overwrite=True,
)
```

Figure 3-13: Middleware SDK example for uploading results.

3.2.3 Interaction with the CityCLIM ecosystem

The CityCLIM ecosystem utilises various data types and offers service applications in the scope of the climate change induced challenges of urban areas by the integration of different processing workflows.

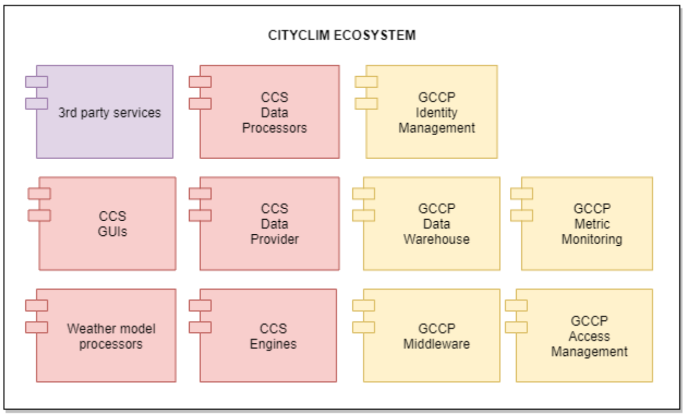


Figure 3-14: Components of the CityCLIM ecosystem with GCCP modules (yellow), and CCS specific modules (red).

Besides graphical web interfaces to the developed services, the CityCLIM ecosystems provides technical access to various datasets and processing workflows, orchestrating all services, components and the GCCP itself (see Figure 3-14).

A selection of such dataset access and processing workflows is listed below:

- On-demand model time series data via API for specific locations for supported regions.
- API for receiving warnings with respect to heat exposure.
- A dedicated interface for searching and downloading earth observation data.
- Beyond-state-of-the-art model-based weather forecast data and post-processed results in image and video formats.
- On-demand statistical analysis over a larger period (e.g., summer months) using aggregated weather model data to identify urban areas heavily affected by the urban heat effect.
- Data injection to proprietary infrastructure (e.g., smart city platforms) after completion of workflows (e.g., daily weather forecast completion)

The technical documentation is following the OpenAPI specification, and up-to-date documentation for relevant interfaces can be provided when requested.

To demonstrate an interaction with a component operating within the CityCLIM ecosystem, the following endpoint in Table 3-3 is specified, which provides time series data of the UHD model run for different locations.

Table 3-3: Specification of an example endpoint for time series data provision.

Get UltraHD model run time series	
`POST`	/operational/time-series-pointwise
Query Parameters	
`user_id`	ID of the user
`bbox_name`	Name of the supported region
`parameter_name`	Name of the UltraHD model parameter
`model_run_start_timestamp`	Starting date of the UHD model run in the format. YEAR-MONTH-DAYT00:00:00. E.g., 2023-06-08T00:00:00
`period`	Integer number between 5 and 120 (boundary included). This number describes the sampling rate of the time series.
HEADER	
`USERNAME`	Your username
`PASSWORD`	Your password
BODY	

`[`
`  {`
`    "pointId": "Central area",`
`    "coords": {`
`      "lat": 49.017552773601196,`
`      "lng": 8.399215384591711`
`    }`
`  }`
`]`	

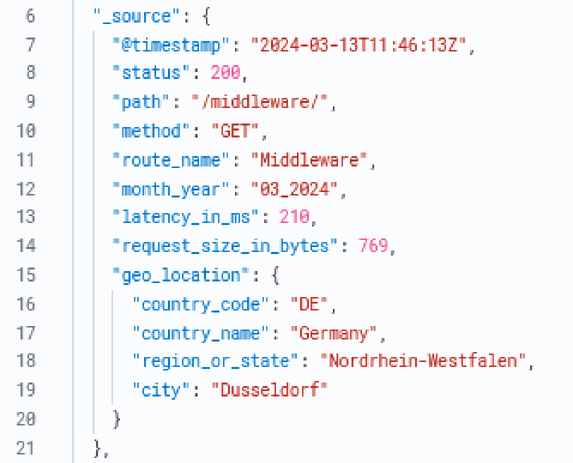
3.2.4 Monitoring and Alerting

The GCCP provides a monitoring and alerting tool for components and services operating in the CityCLIM ecosystem. Each third-party component and service provider can get access to a pre-configured web-based dashboard visualising valuable information about the

- performance (e.g., statistics about component/service calls, latencies, or amount of data exchanged),
- analyse metrics (e.g., debug information and meta-data about messages, anomalies, or error codes, but also interesting metrics for market analyses showing anonymised data on the postal code level from where especially services are accessed, including statistics about top countries and cities) and
- QoS information (like average, min, or max values per hour for several metrics like latencies, data amount transferred, or count of called components and services) of their operating services and components.

A monitoring dashboard can be requested from the GCCP administration team once a component or service has been registered and integrated into Identity Management and is operational within the CityCLIM ecosystem. This allows third-party component and service providers to access the web-based dashboard and view all of the above metrics for their services and components.

In addition, custom monitoring dashboards can be requested to visualise additional metrics or calculations/transformations of them. The following metrics are monitored by default by the GCCP and can be used for this purpose (see Figure 3-15 for an overview and Table 3-4 for the description of the different parameters).



```
"_source": {
  "@timestamp": "2024-03-13T11:46:13Z",
  "status": 200,
  "path": "/middleware/",
  "method": "GET",
  "route_name": "Middleware",
  "month_year": "03_2024",
  "latency_in_ms": 210,
  "request_size_in_bytes": 769,
  "geo_location": {
    "country_code": "DE",
    "country_name": "Germany",
    "region_or_state": "Nordrhein-Westfalen",
    "city": "Dusseldorf"
  }
},
```

Figure 3-15: Example for metrics that are monitored by the GCCP.

Table 3-4: Description of monitored metrics.

Parameter	Description
`@timestamp`	The time the request was received.
`status`	HTTP status code to indicate whether a request was successful or not.
`path`	The requested URL path, which can provide detailed analysis of the specific functionality used by a service.
`method`	HTTP method of the request.
`route_name`	Name of the GCCP component or service addressed.
`latency_in_ms`	Time between the incoming request and the returned response.
`request_size_in_bytes`	The size of an incoming network packet.
`geo_location`	A set of metrics concerning the location from which the request was sent anonymised on postal code level.
`country_code`	A code acting as abbreviation for a country name.
`country_name`	The concrete country name.
`region_or_state`	Region or state in a country.
`city`	City on a specific region.

Finally, the tool provides an alerting feature that can be used to customise alerts based on thresholds or other conditions. This allows third-party components and service providers to configure custom alerts through the web-based interface. The configuration process involves defining several key elements, including the alert rule name, a suitable query to retrieve the necessary data for the alert, the alert condition, the alert evaluation behaviour, and details for your alert rule (such as a summary to facilitate more effective alert management). The alerting tool is based on Grafana, a comprehensive documentation on configuring alerts can be found under https://grafana.com/docs/grafana/latest/alerting.

The following figures show exemplarily the monitoring on a component with the name “Forecast-Engine” with metrics over a 30-day period (generally configurable from live data over specific time periods or over the last few days/weeks/months/years).

Figure 3-16 shows the calls of the component/engine over the selected time period. It shows the total calls (first panel) and the calls as a time series (second panel).

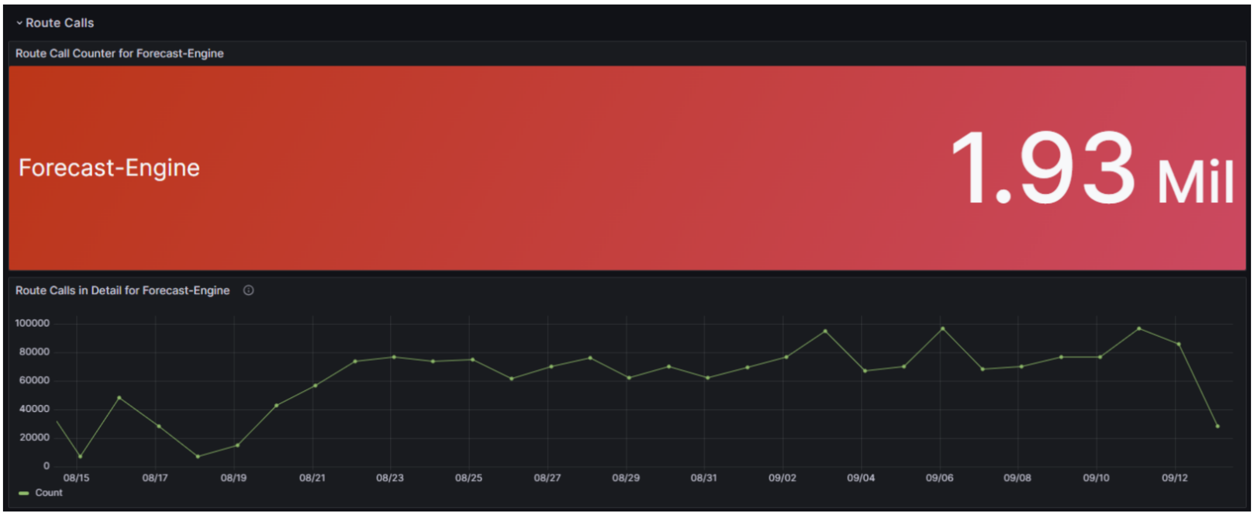
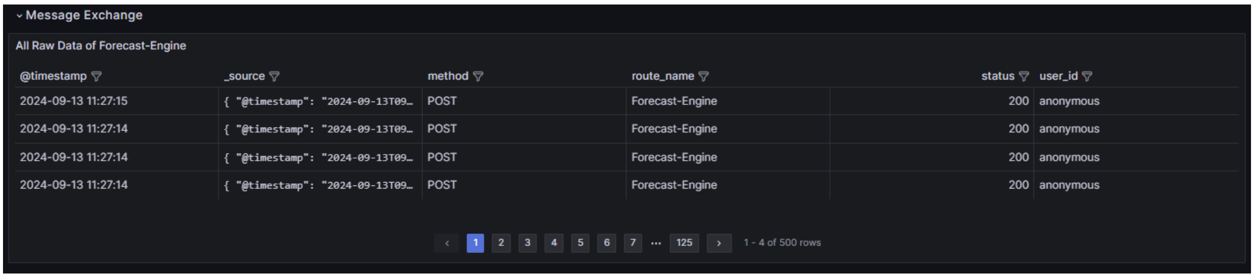


Figure 3-16: Component calls for a selected time period.

Figure 3-17 shows a panel for accessing metadata for every message the component exchanged. It also provides further filtering options to navigate through large amounts of meta-information/messages. A click on a cell in the “_source” column shows the complete meta-data in a new window helpful to analyse single messages in more detail.


Message Exchange

All Raw Data of Forecast-Engine

@timestamp	_source	method	route_name	status	user_id
2024-09-13 11:27:15	{ "@timestamp": "2024-09-13T09…	POST	Forecast-Engine	200	anonymous
2024-09-13 11:27:14	{ "@timestamp": "2024-09-13T09…	POST	Forecast-Engine	200	anonymous
2024-09-13 11:27:14	{ "@timestamp": "2024-09-13T09…	POST	Forecast-Engine	200	anonymous
2024-09-13 11:27:14	{ "@timestamp": "2024-09-13T09…	POST	Forecast-Engine	200	anonymous

Figure 3-17: Meta information of exchanged messages.

Figure 3-18 shows some performance metrics. In this example is show the latencies in milliseconds in the first panel and the sizes of exchanged messages in the second panel as a time series view.

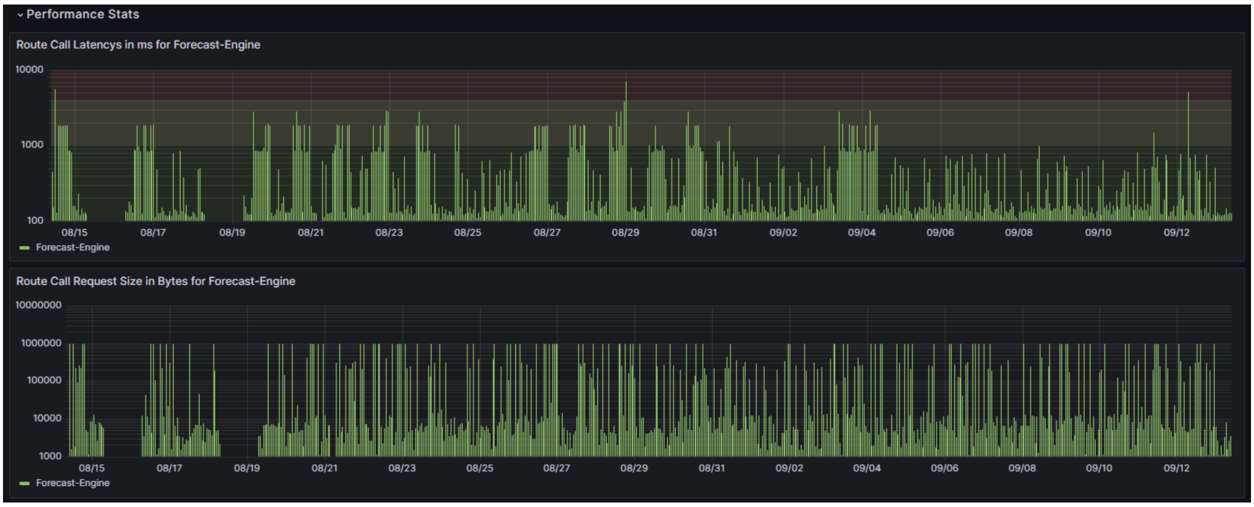
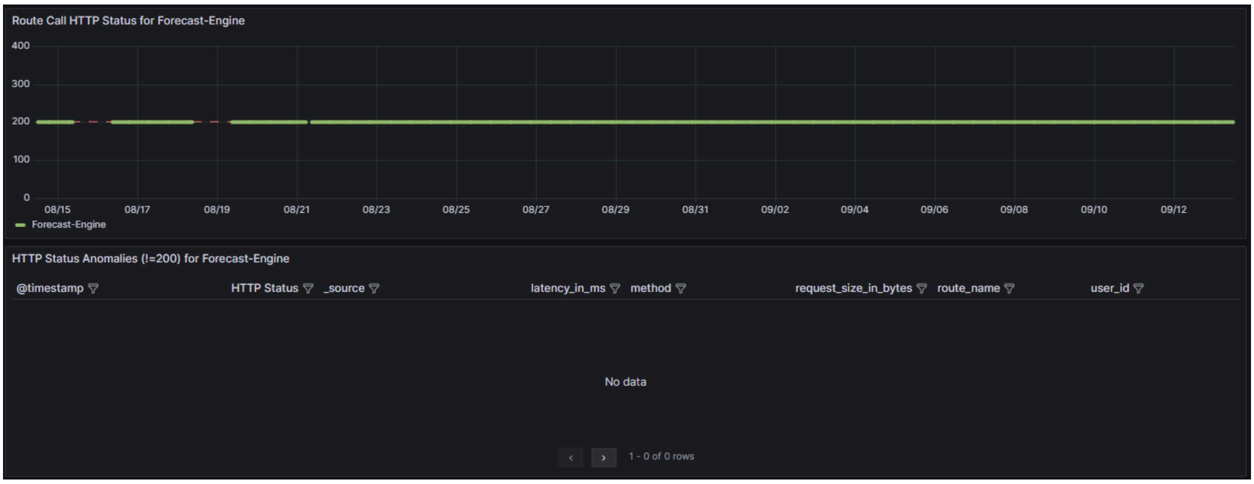


Figure 3-18: Performance metrics about latencies and sizes of exchanged messages.

Figure 3-19 shows panels for the HTTP response status codes[2] that support monitoring of a successful communication. The first panel shows in a time series view the codes and the second panel shows only anomalies which are classified as “not status 200” indicating an unsuccessful communication helpful as additional information to identify errors and anomalies.



@timestamp	HTTP Status	_source	latency_in_ms	method	request_size_in_bytes	route_name	user_id

No data

1 - 0 of 0 rows

Figure 3-19: Metrics about HTTP codes and anomalies.

Figure 3-20 and Figure 3-21 show panels on the usage of a component or service. Figure 3-20 shows the panel for visualising the geolocation of accesses on an anonymised postcode level, and Figure 3-21 shows panels for statistics such as the top countries and cities accessing the component or service. Insights into the usage can help third-party components and service providers to make better business decisions.

[2] Various status codes are existing. The monitoring and alerting tool monitors successful responses (200–299), indicating the request was successfully received, understood, and accepted; redirection messages (300–399), informing the client that further action is needed to complete the request; client error responses (400–499), indicating that there was an error with the request from the client's side; and server error responses (500–599), indicating that the server failed to fulfil a valid request due to an error on the server side. See for more details in RFC 9110: https://httpwg.org/specs/rfc9110.html#overview.of.status.codes

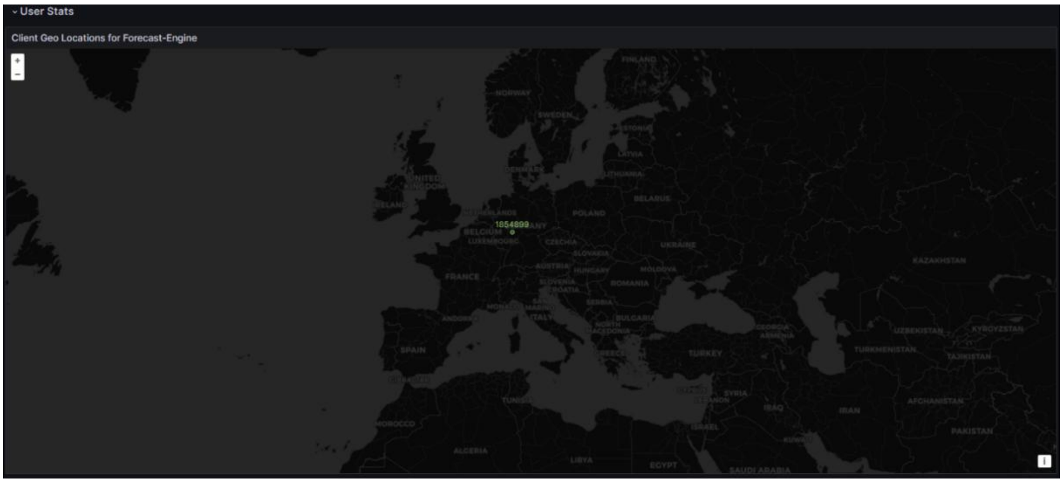
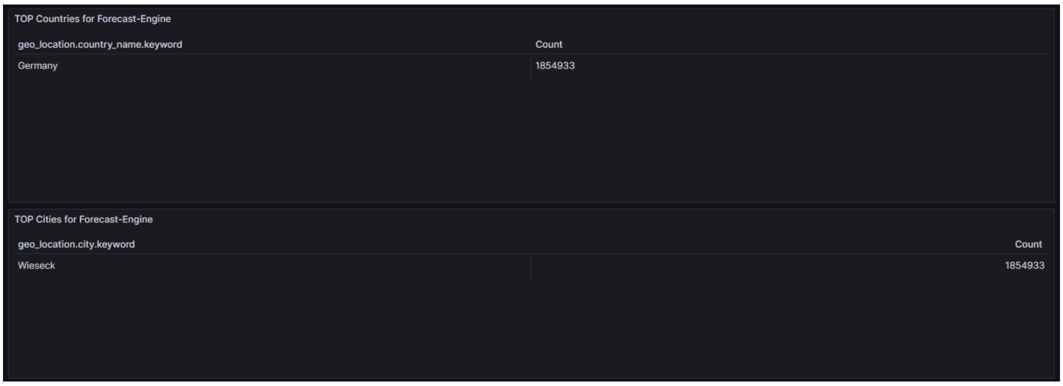


Figure 3-20: Geo-location showing from where services and components are called.



TOP Countries for Forecast-Engine

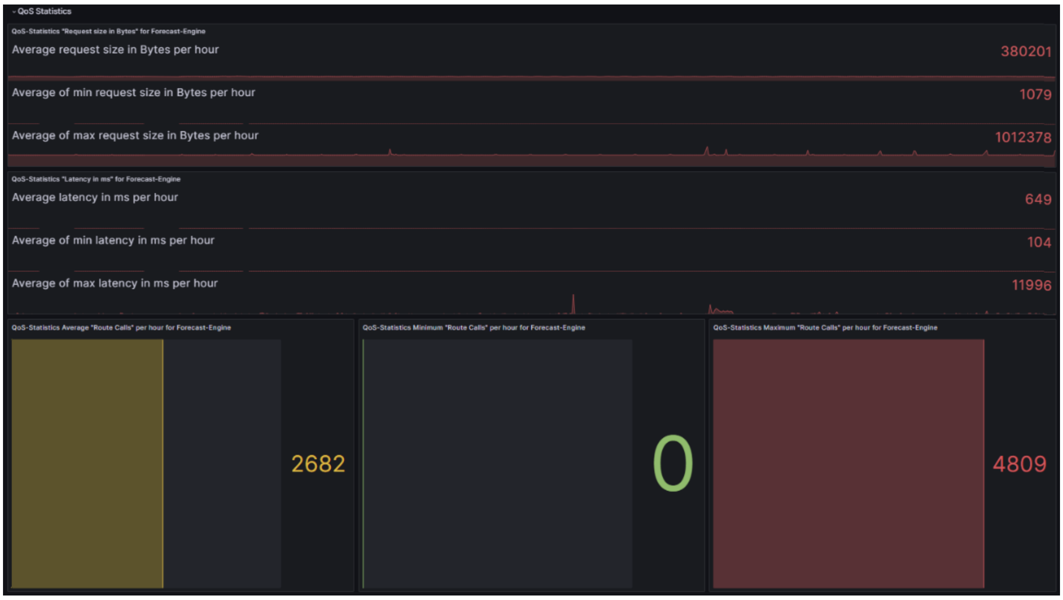
geo_location.country_name.keyword	Count
Germany	1854933

TOP Cities for Forecast-Engine

geo_location.city.keyword	Count
Wieseck	1854933

Figure 3-21: Statistics about top countries and cities.

Finally, Figure 3-22 shows some QoS metrics about average, minimum, and maximum usage per hour in the exemplary 30 days’ time period.



QoS Statistics

QoS-Statistics "Request size in Bytes" for Forecast-Engine

Average request size in Bytes per hour 380201

Average of min request size in Bytes per hour 1079

Average of max request size in Bytes per hour 1012378

QoS-Statistics "Latency in ms" for Forecast-Engine

Average latency in ms per hour 649

Average of min latency in ms per hour 104

Average of max latency in ms per hour 11996

QoS-Statistics Average "Route Calls" per hour for Forecast-Engine 2682

QoS-Statistics Minimum "Route Calls" per hour for Forecast-Engine 0

QoS-Statistics Maximum "Route Calls" per hour for Forecast-Engine 4809

Figure 3-22: QoS statistics for different metrics.

The figures above give an impression of the monitoring and alerting tool. In addition to these default visualisation panels for each component and service, third-party services and component providers can request further customised dashboards and configure alerts to trigger notification via email, Slack, or other communication channel based on individually configured conditions.

4 Conclusion, Next Steps and Contact

Join CityCLIM as a Software Partner
Urban centres globally are facing intricate challenges related to Urban Heat Islands, distinct microclimatic variations, and the pressing imperative of sustainable urban development. CityCLIM offers state-of-the-art, scientifically-advanced next generation city climate services meticulously designed to address the unique climatic complexities of today's urban landscapes. To third-party software providers, CityCLIM offers an exciting opportunity to contribute to this rapidly growing ecosystem and emerging market by collaborating with leading experts and helping shape the future of urban climate resilience.

Designed to enable a seamless, fast, and secure integration of your services and components, the Generic City Climate Platform (GCCP) provides the needed tools to do this with minimal effort using secured and standardised interfaces, robust data management, and strong community support. CityCLIM provides a collaborative environment where you can build and showcase your services while leveraging the strengths of an established network and platform.

How to Get Started
This manual already provided an overview of what CityCLIM offers to you as a third-party software provider and how you are able to develop and integrate third-party components and services. To join CityCLIM as a Software Partner and begin your journey, follow these steps:

- **Understand your benefits and opportunities –** CityCLIM offers a chance to enter a new emerging market for next generation City Climate Services. Explore this opportunity and the benefits offered by CityCLIM, including an already strong and growing community, a feature-rich platform that enables rapid time-to-market, and think about your own promising business models.
- **Get familiar with the GCCP and get in contact with us** – Familiarise yourself with the capabilities and features of the GCCP and how the GCCP supports you in focussing on your goals. If you have any questions, get in contact with us (see below)!
- **Develop & integrate your services in the CityCLIM ecosystem** - Start developing your components and services and start with the integration process using the detailed guidelines in this document.
- **Collaborate with the CityCLIM community** - Exchange ideas with other partners, explore synergies with other existing components and services, and build new solutions that extend and innovate the CityCLIM ecosystem.

Further Information
Refer to further public material as "D2.4 - Optimised Full Prototype of Generic City Climate Platform" (CityCLIM Consortium, 2024b) to get a short fact sheet on the GCCP or the handbooks "Towards a Green Future with CityCLIM: A Handbook for Interested Cities" (CityCLIM Consortium, 2023b) and "Becoming a CityCLIM Citizen Scientist: A Comprehensive Guide" (CityCLIM Consortium, 2023a) to get a broader view of the project.

Get in contact with us
If you have any questions or need advice at any stage, or if you would like to receive updates on CityCLIM, please do not hesitate to contact us! To get updates and further information on CityCLIM, you can visit the CityCLIM website at https://cityclim.eu and subscribe to the newsletter. You can also follow us on social media platforms such as Twitter (X), and LinkedIn, and ResearchGate. If you have any questions about the project or are interested in collaboration, you can contact us by filling out the form available on our website. If you are interested in integrating and registering your own services and components, please send an email to the CityCLIM administration team (see email below).

Contact Us: Email: satelliteservices@ohb-ds.de

Thank you for joining the CityCLIM community.

5 References

CityCLIM Consortium. (2022). *D1.3 -Public CityCLIM Concept*. https://www.cityclim.eu/info-material

CityCLIM Consortium. (2023a). *Becoming a CityCLIM Citizen Scientist: A Comprehensive Guide*. https://www.cityclim.eu/info-material

CityCLIM Consortium. (2023b). *Towards a Green Future with CityCLIM: A Handbook for Interested Cities*. https://www.cityclim.eu/info-material

CityCLIM Consortium. (2024a). *City Administration Services: A User Guide*. https://www.cityclim.eu/info-material

CityCLIM Consortium. (2024b). *D2.4—Optimised Full Prototype of Generic City Climate Platform*. https://www.cityclim.eu/info-material

About CityCLIM

The strategic objective of CityCLIM is to significantly contribute to delivering the next-generation of City Climate Services based on advanced weather forecast models enhanced with data both from existing, but insufficiently used, sources and emerging data sources, such as satellite data (e.g., Copernicus data) or data generated by Citizens Science approaches for Urban Climate Monitoring etc. For City Climate Services, data products of interest related to land surface properties, atmospheric properties (e.g., aerosol optical thickness), geometry etc. For all of those, information of interest concerns e.g., Copernicus data products and services that are already existing (e.g., based on Sentinel-3/OLCI, PROBA-V, SPOT, Sentinel-1, MetopASCAT data), will exist in the near future (based on already flying satellites such as Sentinel-2), or will exist in the mid-term (based on satellites currently under development) and long-term (based on satellites soon starting concept phase) future. The project will establish; (i) an open platform allowing for efficient building of services based on access to diverse data; (ii) enhanced weather models based on data from diverse existing and emerging sources; (iii) a set of City Climate Services customizable to specific needs of users in cities; and (iv) a generic Framework for building next generation of Urban Climate Services. CityCLIM will be driven by 4 Pilots addressing diverse climate regions in Europe (Luxembourg, Thessaloniki, Valencia, Karlsruhe) which will define requirements upon the tools to be developed, support specification and testing of the services and serve as demonstrators of the selected approaches and the developed technologies. The consortium will elaborate business plan to assure sustainability of the platform and services.

Every effort has been made to ensure that all statements and information contained herein are accurate, however the CityCLIM Project Partners accept no liability for any error or omission in the same.

Funded by the Horizon 2020 Framework Programme of the European Union