New approaches for distributed data analysis with the DASF Messaging Framework

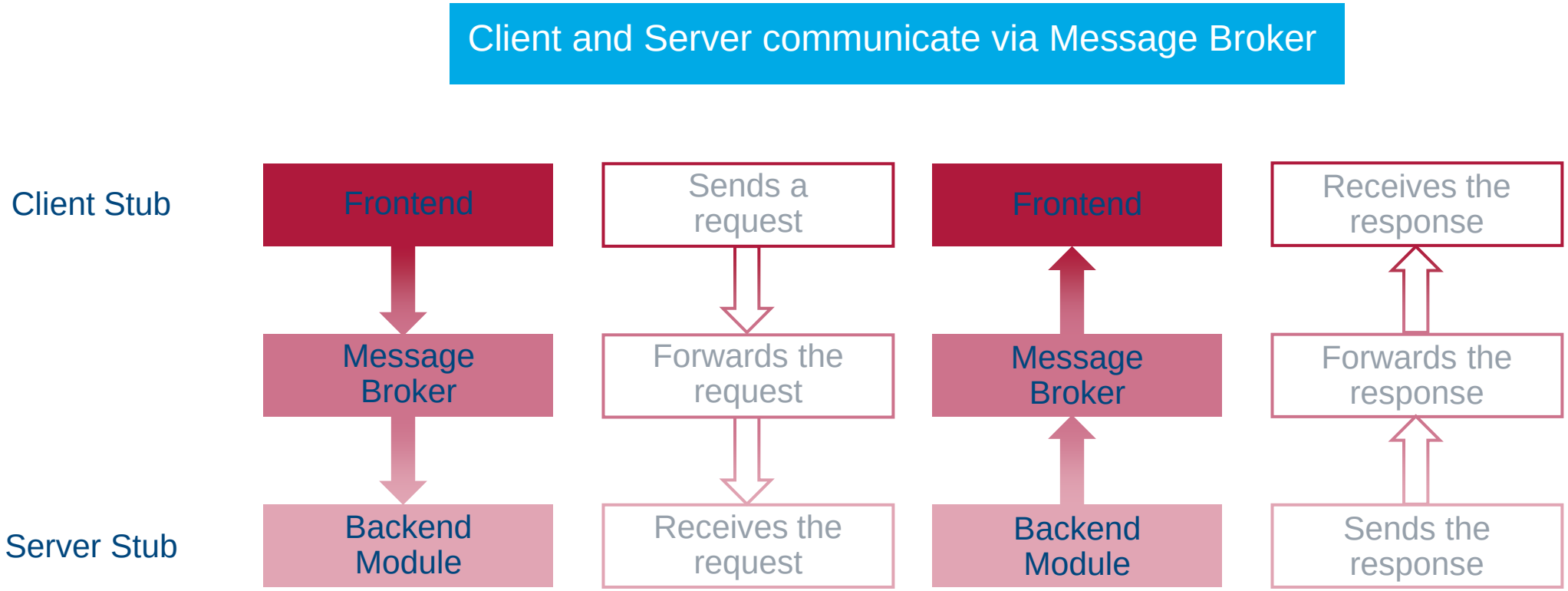
Philipp S. Sommer

Helmholtz Coastal Data Center
Helmholtz-Zentrum Hereon

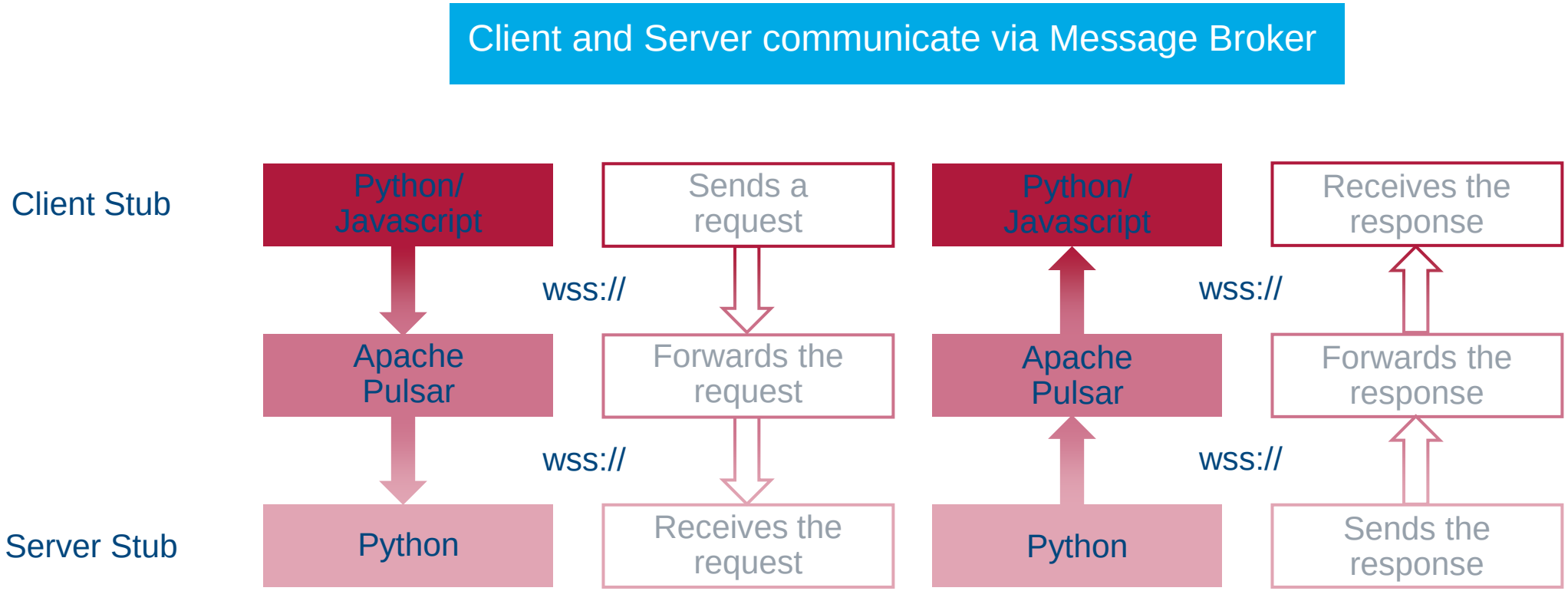
Data Science Symposium No. 7,
June 28th, 2022

Helmholtz-Zentrum
hereon

Basic Workflow



Basic Workflow



Use the scientists methods

- abstract standard python functions and classes into web requests
- everything's basic python, (almost) no need for special stuff
- Client stub is automatically generated
- Requests are abstracted and standardized (JSONschema)

```python
from demessaging import main

def compute_sum(
    da: demessaging.types.xarray.DataArray,
) -> demessaging.types.xarray.DataArray:
    """
    Compute the sum over a data array.

    Parameters
    ----------
    da : DataArray
        The input data array

    Returns
    -------
    DataArray
        The sum of the data array
    """
    request = {
        "member": {
            "func_name": "compute_sum",
            "da": da,
        }
    }

    model = BackendModule.parse_obj(request)
    model.compute()

    return model.member.func_returns  # type: ignore
```

Where are we now?

- **DASF-Progress API implemented**
  - Server stub can send information on the progress and it will be displayed in the frontend

- **Messages are automatically splitted into multiple chunks when they are too big (currently only available for JavaScript clients)**

- **Threading and queueing implemented in backend module**

What is still missing?

Federation

- **How to secure communication between multiple centers?**
  - Example: I want Klaus (Geomar) to access my backend module, but not Andreas (Geomar)

Backend Module Authentication

- Currently, server stubs can subscribe to a topic anonymously -> Requests could be compromised

Installation

- How to install and deploy backend modules in a sustainable manner?
- How to use the framework in HTTPS (i.e. via wss protocol)

Storage

- How to store results of the requests?

Helmholtz-Zentrum hereon

Planned features

Federation

- **How to secure communication between multiple centers?**
  - End-to-End encryption (E2EE) with Django-based Oauth-Provider
  - Detailed Workflow

Backend Module Authentication

- Backend modules register via token to the message broker
- New light-weight Django-Message broker
  - https://gitlab.hzdr.de/hcdc/django/django-dasf-broker

Installation

- Standardize backend module generation via cookiecutter (python package, Docker file)
- Standardize deployment of frontend modules via Django and Django-based Message broker
- Implement setup for Kubernetes

Storage

- Develop Framework for storing requests and results in database (Django)
  - Support E2EE and store results encrypted in the database
  - https://gitlab.hzdr.de/hcdc/django/django-e2ee-framework

Helmholtz-Zentrum hereon

Vielen Dank

www.hereon.de

Helmholtz-Zentrum
hereon