

# Innovative Web Frontends for a secure access to High-Performance Computing resources via DASF

Philipp S. Sommer, Björn L. Saß, Markus Benninghoff, Helmholtz-Zentrum Hereon, Institute of Carbon Cycles, Contact: philipp.sommer@hereon.de



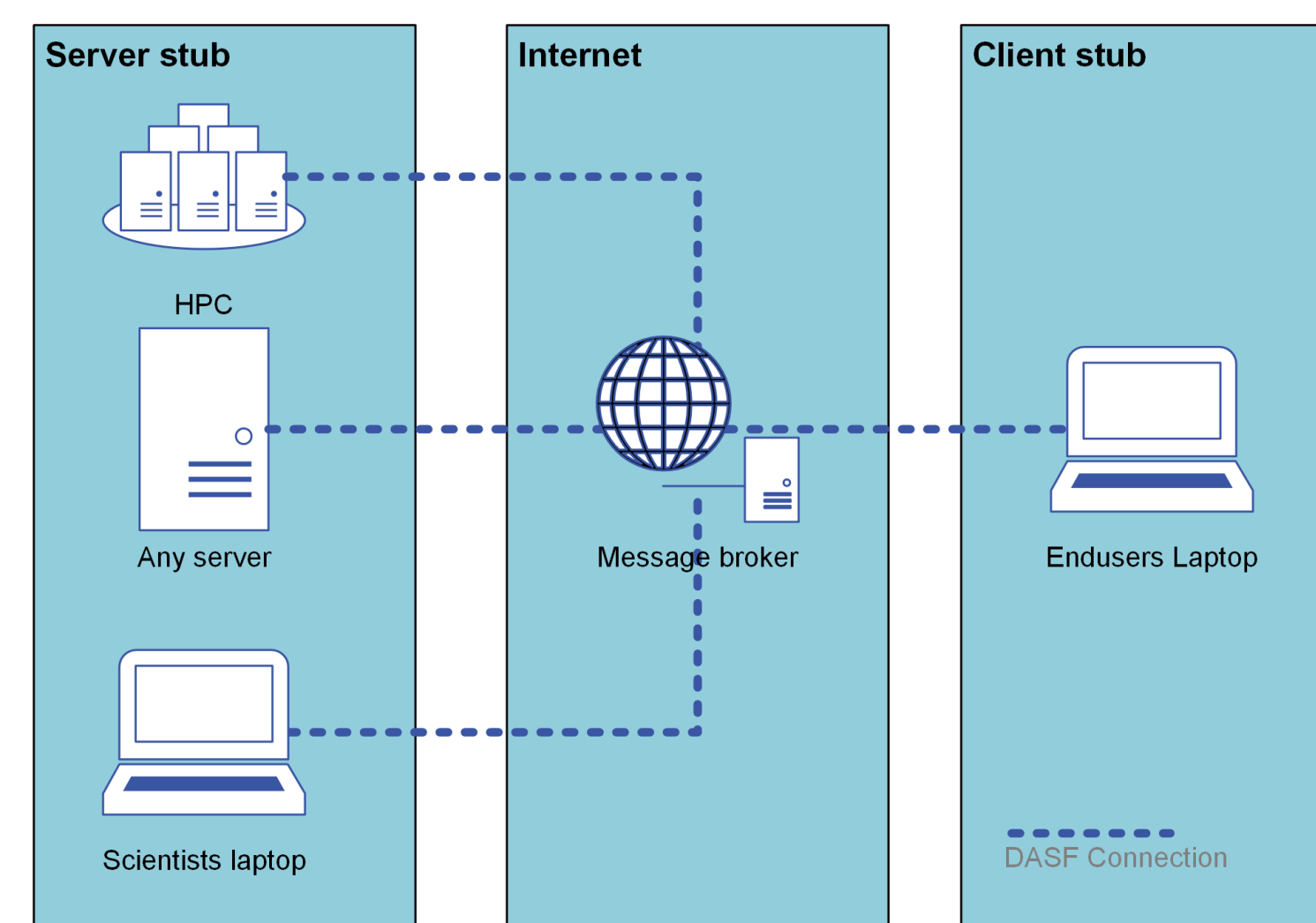
## 20 sec summary

The open-source Data Analytics Software Framework (DASF) is an advanced remote procedure call (RPC) framework designed to abstract Python code and make it securely callable over the internet. The framework ensures that computing resources remain protected from direct exposure to the internet. The latest innovation within DASF is the development of a flexible framework to automatically generate a web frontend for computing resources.

## The Framework

### DASF

DASF connects two resources through a message broker. The server stub (e.g. some HPC system) and the client stub (some other scientist, general public, etc.). The server stub is a custom python script written by the scientist. The generic functions and classes in the script are interpreted by DASF, and DASF connects via websockets to a message broker, a public webserver. Endusers (the client stub) can now connect to the message broker and call the functions that the scientist provided in the script. Validation of the client requests goes via standard static python type annotations, and only the controlled set of functions and classes is available for RPC. So little to no overhead is required to make a function available on the web.

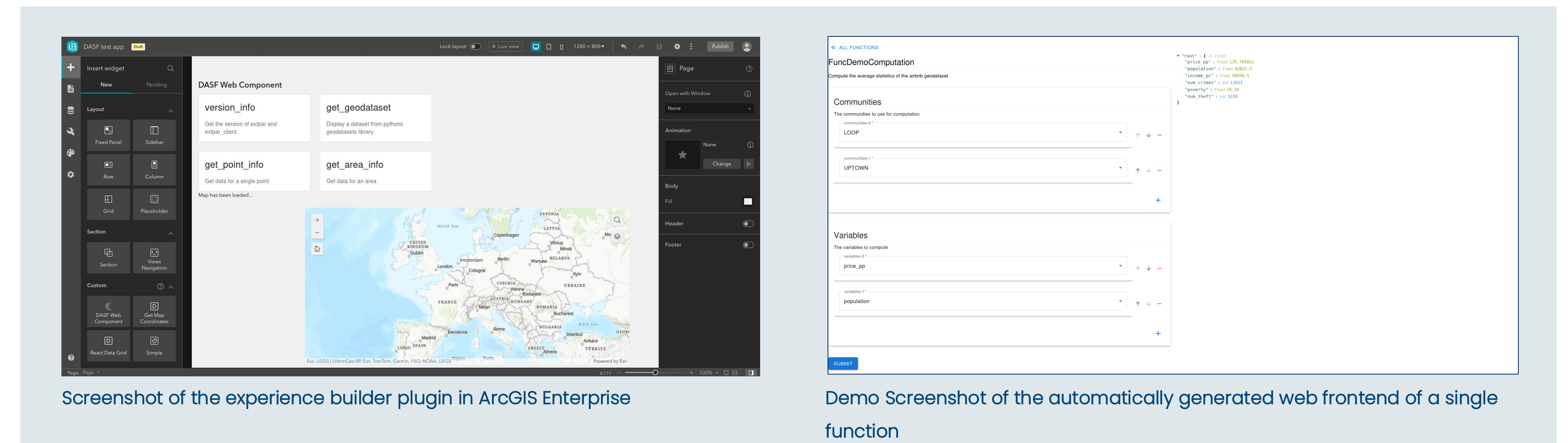
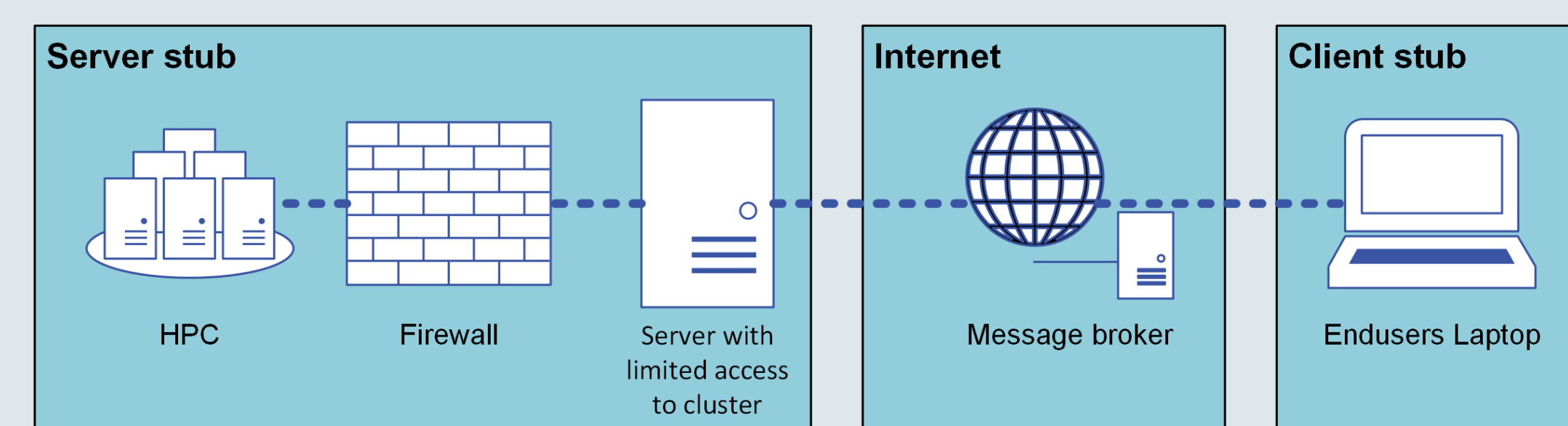


### The new frontend

DASF transforms the functions and classes of the backend script into a JSONschema representation. The new frontend uses this JSONschema to render web forms. These formulars can be used to enter the request parameters, send the request and visualize the results. The new frontend can be used from within another React frontend, or via framework-agnostic custom web elements in any HTML-based application.

### Use-Case: Digital Twin in HPC

Digital twins are virtual representations of physical systems. They rely on data and HPC resources to function effectively. DASF is particularly well-suited for these applications because it allows the code to run on HPC without direct exposure of this sensible infrastructure to the internet. This ensures that expensive resources remain protected while still providing the necessary computational power. The strategy is to have a separate virtual machine or container that has access to limited parts of the cluster only, e.g. some part of the filesystem, some dedicated (scheduler) port, or can only run certain commands via SSH.



## Integrations

### ArcGIS Experience Builder

The Experience Builder is a Content Management System in ArcGIS Enterprise to build (map-centered) web apps. We built a plugin for the experience builder to visualize and combine a DASF module with a web map.

#### Highlights:

- Select location or polygon as input for RPC call from the map
- Visualize results as polygons or raster data from backend module on the map

### Django Broker and CMS integration

We developed a django-based message broker inspired by the Apache Pulsar messaging protocol, tailored towards DASF. The combination with the new frontend enables us to directly provide web interfaces for registered topics.

#### Highlights:

- Automatic web frontend for each backend module without extra coding
- Implementation in djangos Content Management System django-CMS via plugin

## Resources

Source code: <https://codebase.helmholtz.cloud/dasf>

Documentation: <https://dasf.readthedocs.io>

Web frontend: <https://codebase.helmholtz.cloud/dasf/dasf-web-component>

Message Broker: <https://codebase.helmholtz.cloud/dasf/dasf-broker-django>

Experience builder plugin:

<https://codebase.helmholtz.cloud/dasf/dasf-experiencebuilder-plugin>

## References

Eggert et al., (2022). DASF: A data analytics software framework for distributed environments. Journal of Open Source Software, 7(78), 4052, <https://doi.org/10.21105/joss.04052>

Source code repository on  
[codebase.helmholtz.cloud/dasf](https://codebase.helmholtz.cloud/dasf)

