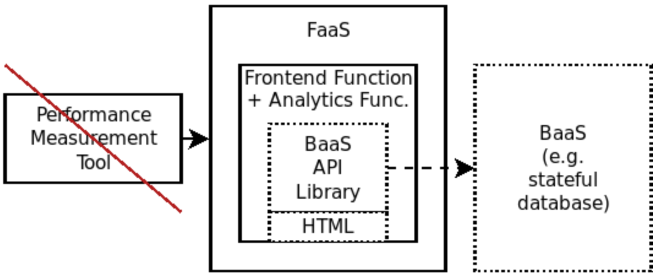
SCAD P02 – Cloud Function-based Application Development

This is a one-week lab, i.e. SW 3, submission before SW 4 lecture.

1. Growing your Application

You fast-grow your project from single function(s) to a useful stateful application, a feedback survey. In addition to a FaaS runtime and multiple cloud functions, you will now need a frontend, a first BaaS (database) and a library or framework to talk to it from the function (unless you prefer to talk to it in plain HTTP). Make sure to understand the following figure. The frontend should be implemented as one or more of the functions. Moreover, you will extend your application to be either cross-cloud or polyglot, as detailed below. (If you are brave, you can do both.)


~~Performance Measurement Tool~~
FaaS
Frontend Function + Analytics Func.
BaaS API Library
HTML
BaaS (e.g. stateful database)

Make sure to divide your team to manage this task. You will need an «HTML expert», a «BaaS expert», an «analytics expert», and a «cross-cloud/polyglot» expert to let your application shine.

2. Feedback Form Application Development

1.1 Goal and FaaS/BaaS selection

Based on the initial experiment function development in P01, the goal is to develop a user-facing web-based feedback form with stateful backend that will store the feedback messages. The backend, a database, is considered the BaaS. Users first load a form which is rendered by the function. The form data is then submitted to the same function (which detects if form data was submitted) and conditionally added to the backend with timestamp, and the function returns a confirmation. Moreover through a password-protected link on the initial form, the entire state information, i.e. all feedback messages, is conveyed to the user through the web interface as well. A second password-protected link should offer analytics - how many feedbacks are there, what is the average word size per feedback, what is the feedback frequency per time unit and similar metrics.

The task shall be implemented as FaaS+BaaS combination, best using a FaaS like GCF, ICF, CW or Fission, all of which have a useful web action mode for individual functions, or AWS Lambda that can emulate the same using API Gateway. It is recommended to look for a simple document-oriented or relational database and other appropriate backend services - you might have to scout a bit or refer to the HOWTO. You can also connect to BaaS on other providers that does not support FaaS to build your first cross-cloud application while all functionality is embedded in a single cloud function.

To get around this “cloud function monolith”, it is advised to explore an alternative design with more fine-grained structures where each functionality is implemented as separate cloud function. How many more lines of code will you need? How can the functions be decomposed and decoupled? Try building your first polyglot application by implementing the functions in multiple languages.

If you dare and your team is strong/large enough, you can of course implement a dual cross-cloud and polyglot application.

1.2 Development process

Joint parallel work on the application is recommended according to the team specialisation. First, implement a coarse-grained structure with an entry function and specialised language-level functions for displaying forms, processing/persisting form data and retrieving the persisted data. Then, individually fill the functions with useful code.

Note: This is meant to be a team sprint task. You will need to finish with a basic solution after 90 minutes by submitting all code into your team’s Git repository. A final solution can be submitted within a week after the lab.

1.3 Submission

As usual, submit your extended application in a folder named P02 into your group’s Git repository. Give a brief documentation, including on why you chose either the polyglot or the cross-cloud path. Also give a link from the documentation to a live demo (linking to the frontend).