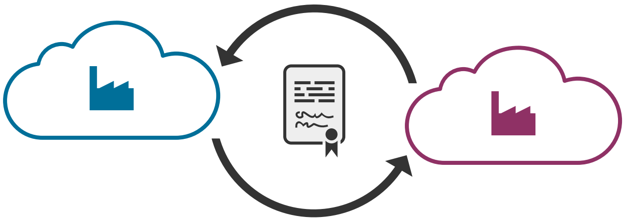
Contractual Inter-Cloud Data Sharing

Assignment D7042E



Abstract

An assignment made for the D7042E course, aimed at setting up two Arrowhead Local Clouds, connecting them using the Gatekeeper and Gateway systems, and then having them coordinate the sharing of data by entering into a contract.

1 Overview

The purpose of this assignment is to produce a concept system that is able to selectively share data between local clouds, theoretically owned by distinct stakeholders, in response to contractual agreements. Concretely, this will require setting up a system-of-systems as the one depicted in Figure 1. It must consist of two Arrowhead local clouds, connected via the Gatekeeper/Gateway systems. Each cloud must have its own Arrowhead Management Tool instance running, not depicted in the just mentioned figure, which enables the clouds to be configured while they are running.

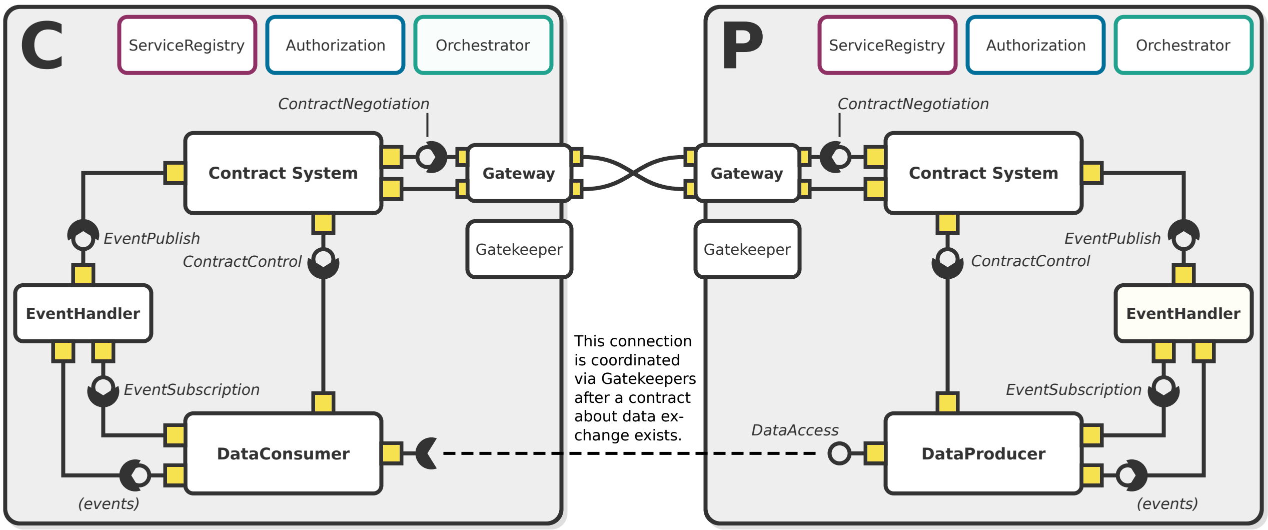


Figure 1: The topology of the two clouds to be setup as part of the assignment. The C cloud contains a *DataConsumer* system that is to first make a contract with the *DataProducer* in the P cloud, and then, using the data in the agreement made, request that data.

2 Custom Systems

Three custom systems part of the desired clouds provide new functionality not found in existing Arrowhead Core systems, namely (1) the *Contract System*, (2) the *Data Consumer* and (3) the *Data Producer*. These must be designed as follows.

1. **The Contract System**. The purpose of this system is to record agreements between pairs of parties. It receives (1) *offer* messages, (2) *reject* messages and (3) *accept* messages. The first kind of message contains a request to get a certain data item, identified by a hash, using a given random identifier, in exchange for a stated number of Euros. When received by the contract system, it saves it and then publishes it using the *EventHandler* system. After receiving an offer, saving it and then publishing it, it waits for some other system in the same local cloud to accept or reject it. When receiving such a reject or accept, it is relayed back to the system that sent the offer. Accept messages must contain whatever details are required to retrieve the data item identified in the accepted offer. The data is to be made available by a *DataProducer* system, which must become contactable via the Gatekeeper/Gateway systems by the *DataConsumer* system in the other cloud.

2. **The Data Consumer**. This system does two things. Firstly, it sends data request offer messages, as described in point (1). Secondly, it actually retrieves the data it offered to pay for if its offer is accepted by the system it sent the offer to.

3. **The Data Producer**. This system holds a key/value map where keys are always the md5 hashes of their corresponding values. The values can be chosen at will. *"Hello, World!"*, *"Secret Data"* or other arbitrary values could be suitable. Apart from holding data, the system significantly (1) listens to offers for buying data via the *EventHandler*, and (2) accepts those offers if payments are above a certain threshold (such as 1000 Euros). Offers below that threshold are rejected. When an offer is accepted, the random identifier in the offer is saved in a table together with the hash of the bought data value. Later, if a request is received directly (not via the EventHandler) that contains the saved random identifier, the random identifier is removed from the table and the data value is sent to the system making the request.

3 Important Repositories

The below table contains the most important sources of code. The so-called *working source repository* contains code you can start from. You can either work against the repository directly in your own git branch, or you can clone it and work on your own copy. When you are done with the assignment, your code should be in the master branch of that repository.

URL	Description
`https://github.com/emanuelpalm/ahf-conet`	**Working source repository.**
`https://github.com/arrowhead-f`	Arrowhead GitHub organization.
`https://github.com/arrowhead-f/core-java-spring`	Released Arrowhead core systems.
`https://github.com/arrowhead-f/mgmt-tool-js`	Arrowhead management tool.
`https://github.com/arrowhead-f/client-library-java-spring`	Client library for Arrowhead systems.

4 Delimitations

Note that it is not a requirement that cryptography or security is used in any way. You may, however, set that up if you want to.

5 Final Words

Please note that this assignment is quite open-ended. You can solve it in a number of ways. You are expected to be mostly independent. You will not be graded, only passed or failed.

If you have questions or concerns you feel that you cannot figure out on your own, please contact me.

Best regards,
Emanuel Palm