Centrum

Content

[Introduction 2](#_Toc292958457)

[Guidelines 2](#_Toc292958458)

[server and CLIENT 2](#_Toc292958459)

[Initialization 2](#_Toc292958460)

[Initiating the server 2](#_Toc292958461)

[Initiating the client 3](#_Toc292958462)

[Security Design 3](#_Toc292958463)

[Contact procedures 4](#_Toc292958464)

[“Hello” messages 4](#_Toc292958465)

[Constructing a “Hello” message 4](#_Toc292958466)

[Client score 4](#_Toc292958467)

[“Hello” procedure 4](#_Toc292958468)

[Client – Protocol of message 4](#_Toc292958469)

[Server - Receiving an “Hello” message 5](#_Toc292958470)

[User interface 5](#_Toc292958471)

[References 6](#_Toc292958472)

# Introduction

Centrum, from the Latin translation of “center”, is a role assigned to a machine that is considered as the central point of a given network, let it be a clan or castrum.

Actors inside a given network will resort to a specific centrum whenever in need of enrolling for the first time on a clan or castrum. This act is also valid for cases when the connection between two actors is broken and needs to be reestablished.

# Guidelines

On this chapter we detail how the centrum should work from a conceptual point of view.

## server and Client

We divide the Centrum component in two separate roles: server and client.

The server is intended to receive the “Hello” messages from each client in the network. The client has the responsibility of sending these “Hello” messages at periodic time intervals.

The “Hello” messages are described on their own section at this document.

During the “Hello” procedure, the centrum may use the opportunity to request clients to activate other components such as the triumvir. The “Hello” may also contain instructions pertinent to clients and other components on each system.

## Initialization

When a given remedium instance starts, there might not exist the information regarding where the centrum is located. On this situation, we define that the instance itself should be the centrum server by default.

Therefore, when the centrum component is initializing we will launch the server role regardless if the component is a designated centrum or not.

After the server role is launched, the client role is initiated.

### Initiating the server

The centrum server does not keep any persistent records. This means that our server will gather information while it is running that will not be available upon restart of the service.

Data is not persistent since it is not be consistent when the component re-initializes. The next start might not be immediate and only occur hours/days after the previous run time, rendering the cached information to become unreliable.

There are no particular clean up steps required for the initialization, since there is also no persistency of information.

After the initialization, our server is ready to register new clients.

### Initiating the client

The first step for the client role is to read the persistent settings of the component and retrieve the location for the designated Centrum server. If no such value exists, it will register the remedium instance itself as being the Centrum server.

## Security Design

We define that each Centrum should only allow registration for 1000 clients. This restriction provides a safety mechanism to ensure that attackers do not overrun the resources from the machine running the Centrum service.

An attacker might still fake the registration of Centrum clients up to the maximum number of available slots; however, this action should trigger an alarm message that is dispatched to alert the system administrator.

# Contact procedures

On this section we detail the contact interactions between the server and client roles.

## “Hello” messages

Centrum clients are expected to say “Hello” to the Centrum server at periodic intervals of time. In the “*Hello*” message we include details about the remedium instance.

### Constructing a “Hello” message

Typically, we dispatch a message that contains specific information about the client. The properties of this message are listed below:

* Remedium instance location (IP address and port number)
* Centrum ID (created dynamically)
* Client score (defined on the remedium structure document)
* Version of the “Hello” message

These values allow the server to process the message and register the client on the remedium network.

### Client score

The client score is a procedure calculated independently from the “Hello” procedure that uses its own thread. This value is requested by the Centrum client when needed for inclusion on the “Hello” message.

## “Hello” procedure

Contact is performed at specific time intervals and we detail on the following subsections the expected scenarios of interaction.

If contact with the remote server is successful, then it will signal the operation as completed and wait for the next scheduled “Hello”.

If the value exists but we fail to retrieve an answer from the remote server, then it will continue retrying to contact the server at periodic time intervals.

## Client – Protocol of message

The client will use asynchronous messaging to perform the “Hello” procedure. This design decides adds extra complexity on both ends of the message handlers that is supported in order to:

* Allow both ends to work in a decoupled manner
* Withstand attempts of DDoS
* Perform complex operations that might require minutes before being completed

## Server - Receiving an “Hello” message

Whenever the Centrum server receives a message of type “Hello”, it will first look for the version property. We intend to provide backward compatibility so that older clients are capable of speaking with more modern servers.

The method that handles “Hello” requests is entitled: “”

## User interface

On the remedium instance assigned as Centrum it shall be possible for users to:

* List registered clients and when they last said “Hello”
* Allow to write specific component settings
* View the log of activities for the component (make this as a default view?)

# References