Algorithm documentation

[Algorithm name]

[Short description]

[Thor A.U., ...]
[Organization]
January 5, 2022

Contents

1 Introduction

[Introduction]

2 Mathematics

2.1 Central

$$f(x) = \frac{1}{\sum_i a_i} \tag{1}$$

2.2 Federated

[Explain the federated derivation]

3 Implementation

3.1 Parameters

Input Parameters			
Parameter	type	example	description
a	string	"a"	a value, with a extra long description
b	int	1	b value
c	float	1.1	...

3.2 Algorithm

Algorithm 1 master

Require: $n \geq 0$
Ensure: $y = x^n$
  $y \leftarrow 1$
  $X \leftarrow x$
  $N \leftarrow n$
  while $N \neq 0$ **do**
    if N is even **then**
      $X \leftarrow X \times X$
      $N \leftarrow \frac{N}{2}$ ▷ This is a comment
    else if N is odd **then**
      $y \leftarrow y \times X$
      $N \leftarrow N - 1$
    end if
  end while

3.3 Output

[table of algorithm output(s)]

4 Risks

1. Issue 1
2. issue 2

5 Validation

```python
import do_stuff

from vantage6.client import Client

# create a client and autenticate
client = Client(...)
client.authenaticate(...)

# create task for algorithm
client.task.create(...)

# poll for results
ready = False
while not ready:
    do_stuff()
```

6 Examples

[Preferable multiple examples of how to run it from R, python and a plain API call]

```r
setup.client <- function() {
  # Define parameters
  username <- 'username@example.com'
  password <- 'password'
  host <- 'https://address-to-vantage6-server.domain'
  api_path <- ''

  # Create the client
  client <- vtg::Client$new(host, api_path=api_path)
  client$authenticate(username, password)

  return(client)
}

# Create a client
client <- setup.client()

# Get a list of available collaborations
print( client$getCollaborations() )
```

```r

# Should output something like this:
#   id      name
# 1  1 ZEPPELIN
# 2  2 PIPELINE

# Select a collaboration
client$setCollaborationId(1)

# vtg.dglm contains the function ‘dglm‘.
model <- vtg.glm::dglm(client, formula = num_awards ~ prog + math,
   family='poisson', tol= 1e-08, maxit=25)
```

```python
import do_stuff

from vantage6.client import Client

# create a client and autenticate
client = Client(...)
client.authenaticate(...)

# create task for algorithm
client.task.create(...)

# poll for results
ready = False
while not ready:
    do_stuff()
```