

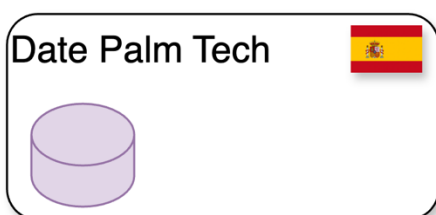
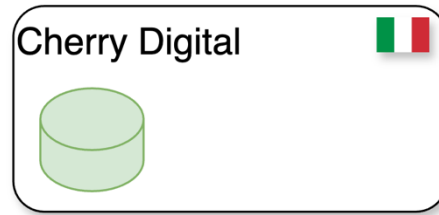
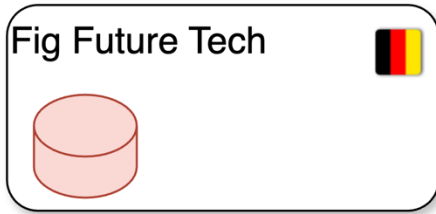
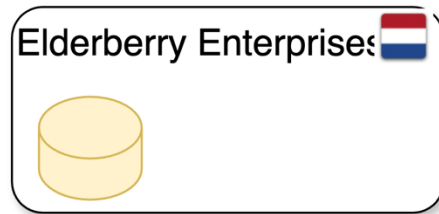
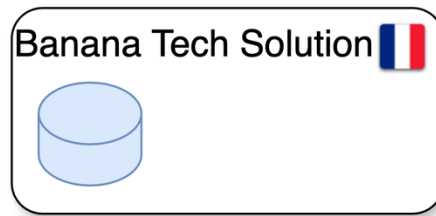
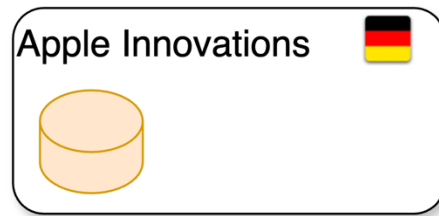
# From theory to practice

Running a PET analysis (without programming) on vantage6



# A theoretical scenario

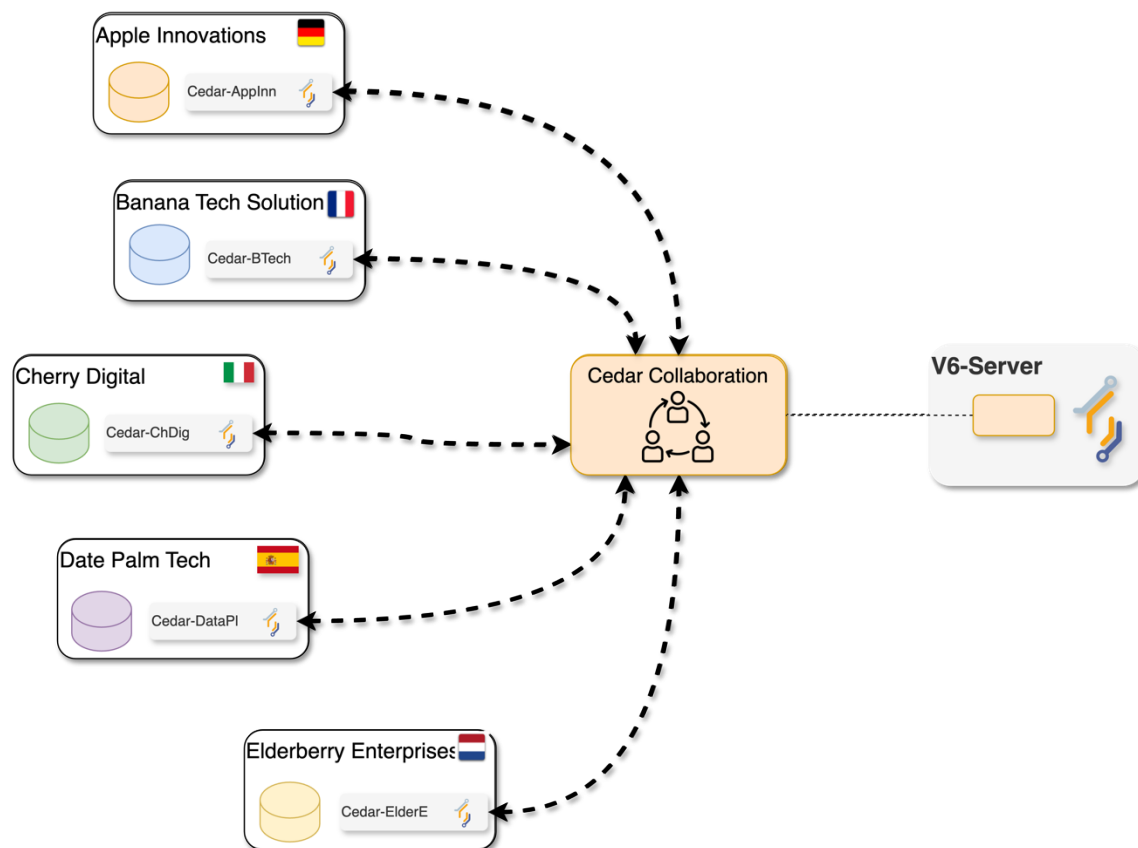
- An international consortium of research organizations working together on aging research.




gender	age	height	weight	isOverweight	ageGroup
M	39	152	108	False	30 - 40
M	8	118	106	False	0 - 10
M	16	161	110	True	10 - 20
M	94	110	115	True	90 - 100
M	47	117	152	True	40 - 50
F	29	127	110	True	20 - 30
M	5	95	65	False	0 - 10
M	39	142	196	False	30 - 40
F	20	189	112	False	20 - 30
F	84	145	116	False	80 - 90



# The scenario – checking v6 concepts!



- Six organizations, one collaboration.
- How many nodes need to be set up?

Node 

- Where are these nodes installed?

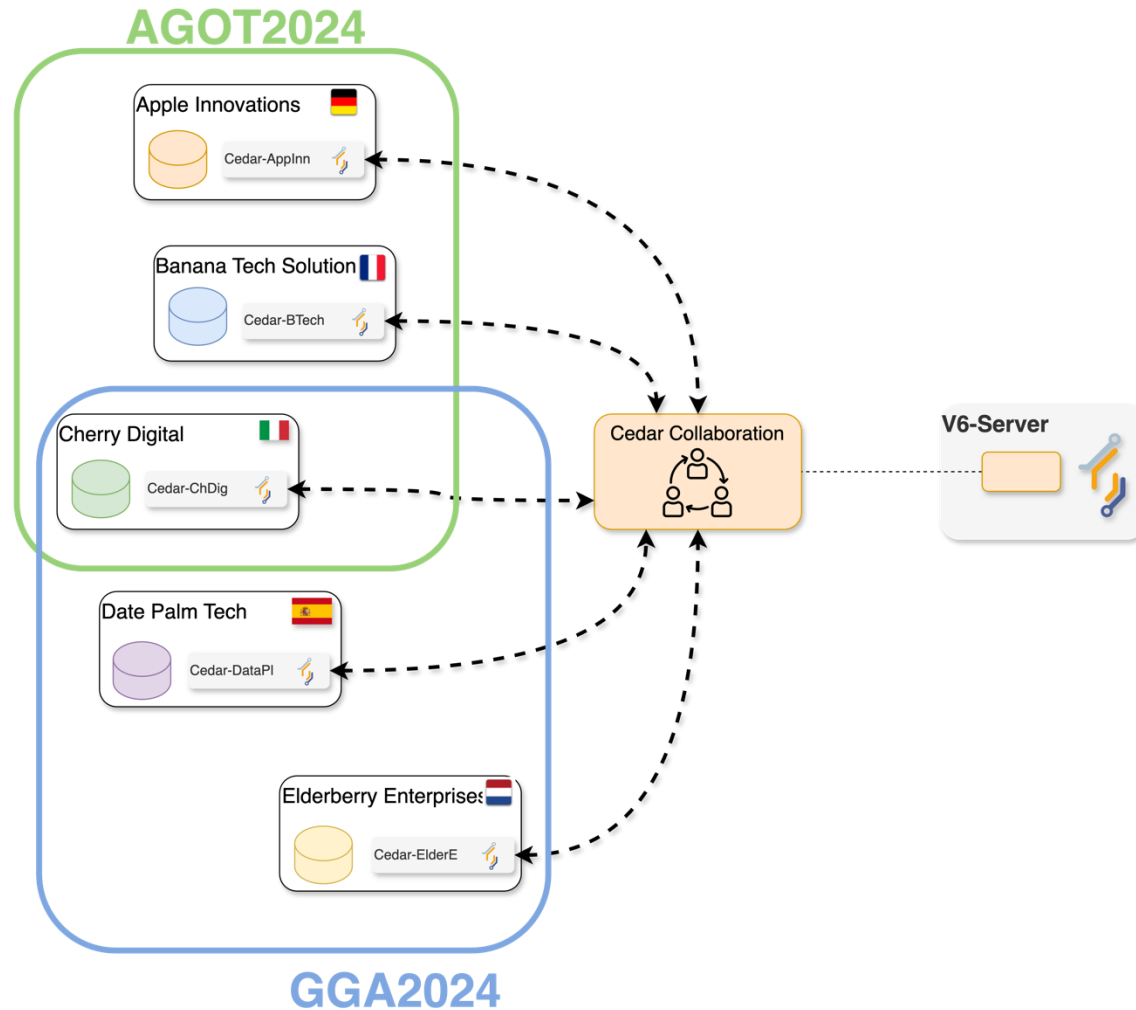
# The scenario – checking v6 concepts!

The consortium (from now on also referred as the **collaboration**) will initially focus on two different projects, each one involving different organizations:

- *Age-Related Variations in Overweight Prevalence: A Comparative Analysis Across Gender and Age Groups (**AGOT2024**) : four organization with relevant data.*
- *The Effect of Gender on Height Development Across Various Age Groups (**GGA2024**): three organization with relevant data.*

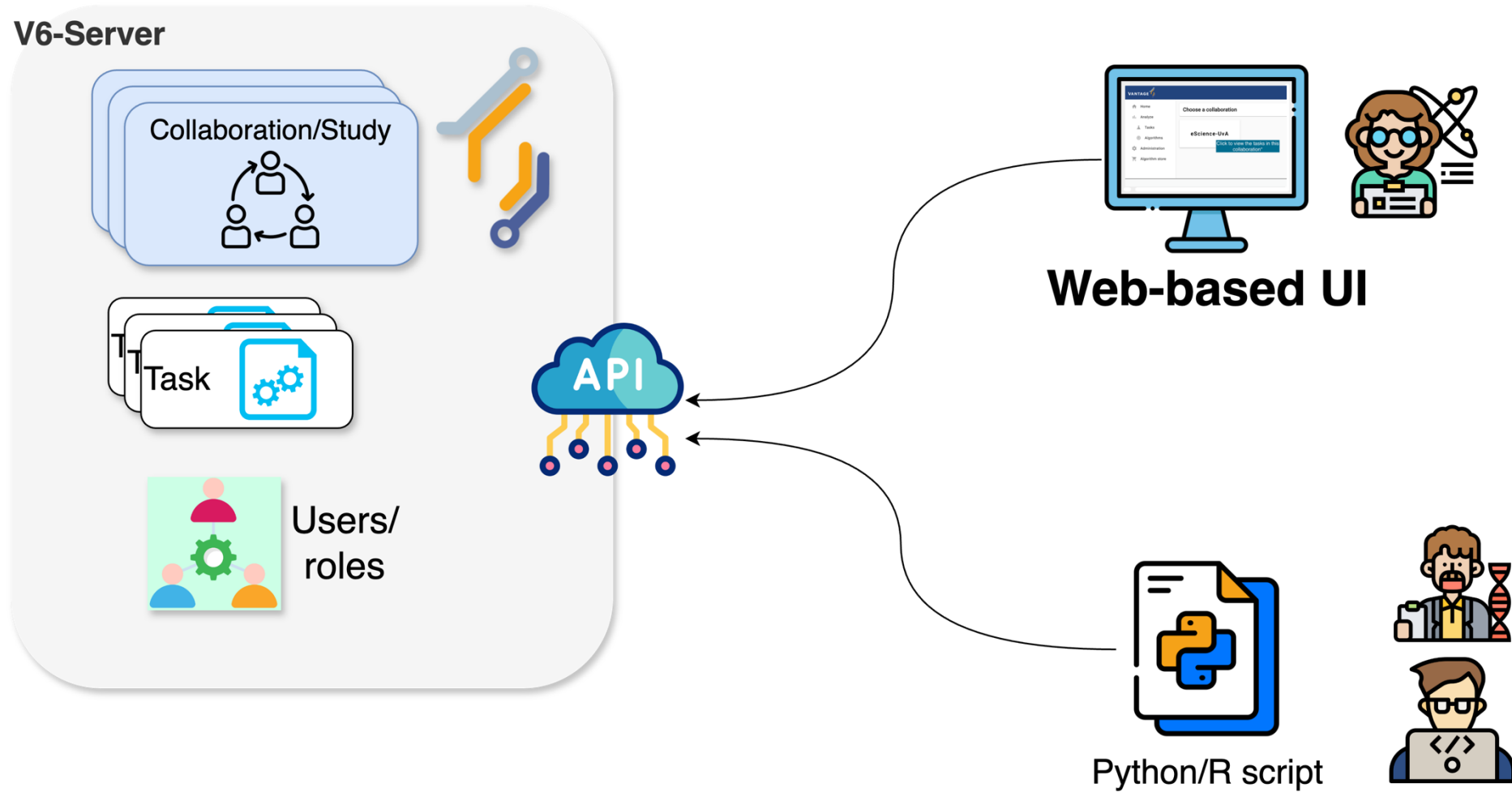
Which V6 concept is applicable here?

# The scenario – checking v6 concepts!

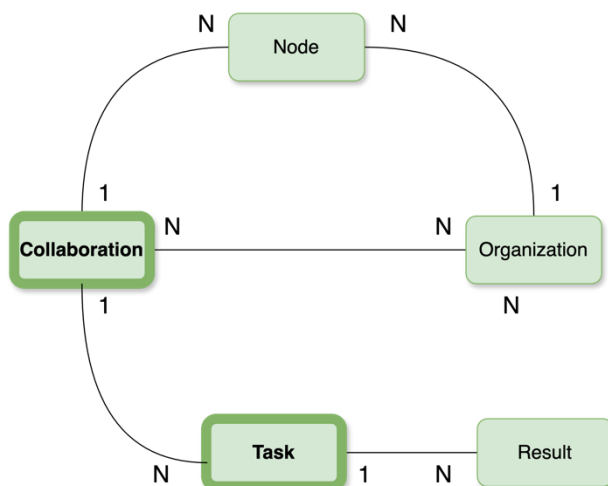


- Two studies: GGA2024, AGOT2024.
- Your setup may be different!

# How to conduct an analysis?



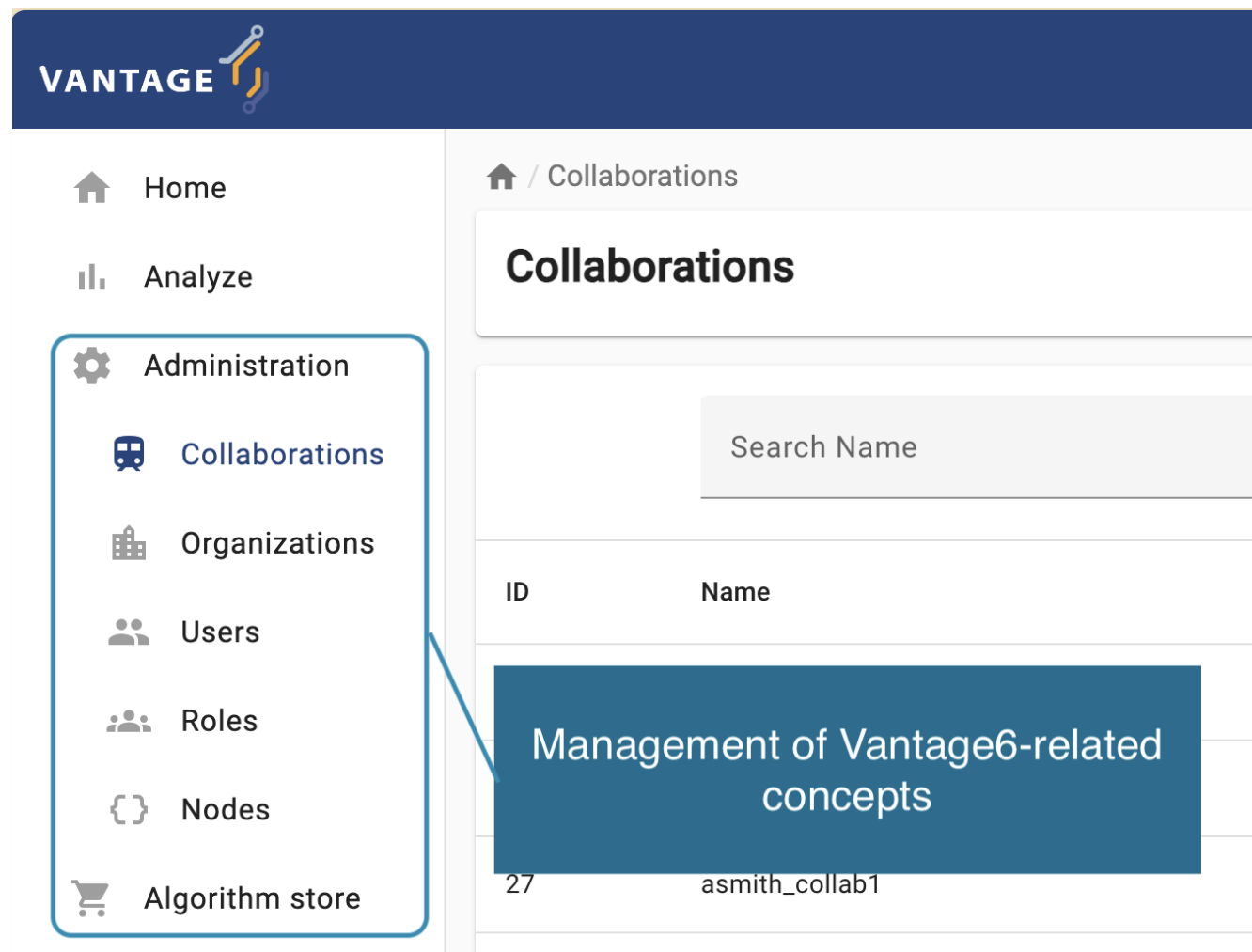
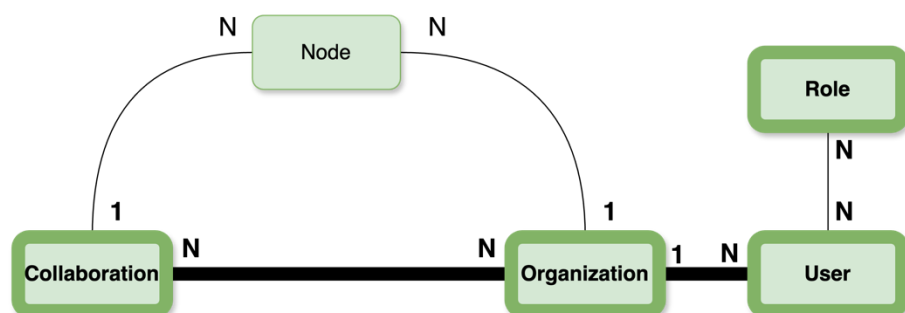
# In this episode: web-based UI



The screenshot displays the VANTAGE web-based user interface. The top navigation bar includes the VANTAGE logo and a sidebar menu with options: Home, Analyze, Tasks, Algorithms, Administration, and Algorithm store. The main content area is titled 'Choose a collaboration' and features a button for 'eScience-UvA'. A blue arrow points from this button to the 'Tasks' section below. The 'Tasks' section includes a search bar labeled 'Search Name' and a table of tasks.

ID	Name	Status
23	r2	Active
21	sda	Active
20	sss	Completed
19	asdas	Failed

# Web-based UI



The screenshot displays the Vantage6 web-based user interface. The top navigation bar includes the VANTAGE logo and a home icon. The left sidebar contains a menu with the following items: Home, Analyze, Administration (highlighted with a blue box), Collaborations, Organizations, Users, Roles, Nodes, and Algorithm store. The main content area shows the 'Collaborations' page, which includes a search bar labeled 'Search Name' and a table with columns 'ID' and 'Name'. A blue callout box points to the 'Collaborations' menu item and the table, containing the text 'Management of Vantage6-related concepts'.

ID	Name
27	asmith_collab1



# Getting familiar with the UI

## CHALLENGE 1

Log into the UI using the information provided and navigate to the administration page. Then, update your email, first name, and last name.

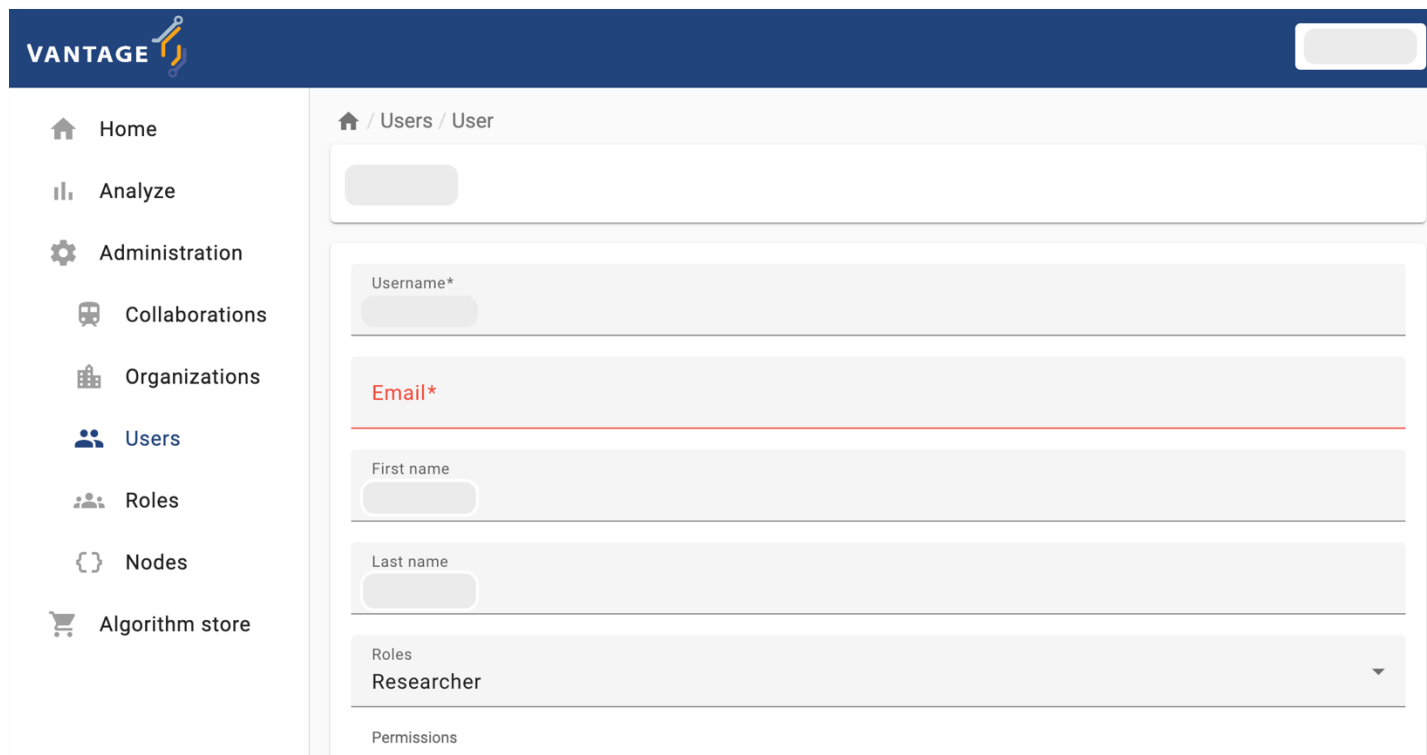


### HINT

Use the **Users** option on the Administration panel on the left.

# Getting familiar with the UI

 SOLUTION

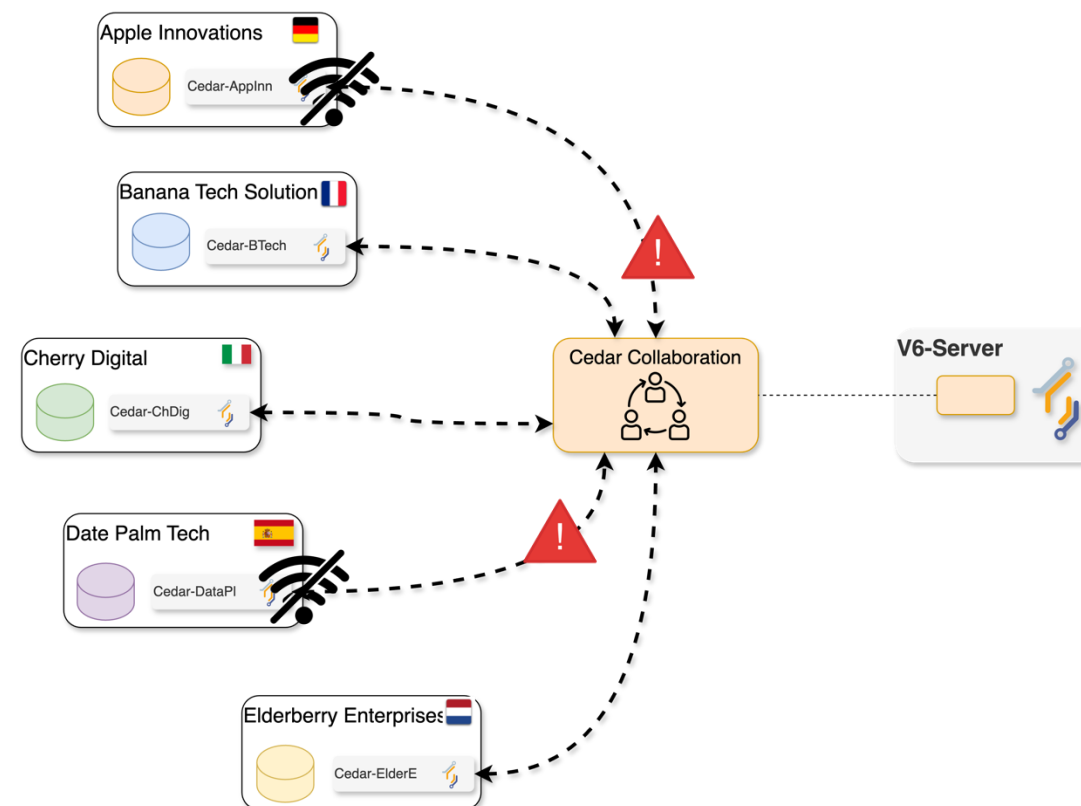


The screenshot displays the VANTAGE user management interface. On the left is a sidebar with navigation links: Home, Analyze, Administration, Collaborations, Organizations, Users (highlighted), Roles, Nodes, and Algorithm store. The main content area shows the 'Users / User' page. It includes a breadcrumb trail, a search bar, and several input fields: 'Username\*' (with a light blue placeholder), 'Email\*' (with a red placeholder and a red underline), 'First name' (with a light blue placeholder), and 'Last name' (with a light blue placeholder). Below these is a 'Roles' dropdown menu currently set to 'Researcher'. At the bottom, there is a 'Permissions' section.

# Running a PET: federated analysis

Are the nodes on my collaboration or study ready for a federated analysis?

- Node within a collaboration: autonomously managed by its corresponding organization.
- Some could be offline.
- UI is handy for checking this!



# Running a PET: federated analysis

## CHALLENGE 2

With your researcher credentials, explore the collaboration you have access to. Check which organizations are part of it and if they are online. Then, check which organizations were assigned to each study: **AGOT2024, GGA2924**.

1. Which study is ready for a federated analysis?
2. If you need to perform an analysis for the study that is **not** ready, which organization you would need to contact to fix this situation?

*\* Take note of which Study is not ready, and the name of the 'offline' organization.*



### HINT

Use the **Collaborations** option on the Administration panel on the left.

# Getting familiar with the UI

 SOLUTION

gtest-AGOT2024

Details

ID	Name	Collaboration
24	gtest-AGOT2024	gui-test-collaboration

Organizations

Apple Innovations

Banana Tech Solutions

Cherry Digital

Date Palm Technologies

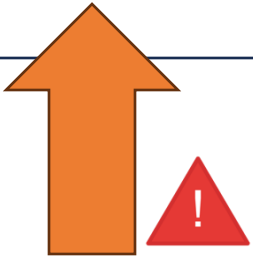
Nodes

gui-test-collaboration - Banana Tech Solutions

gui-test-collaboration - Apple Innovations

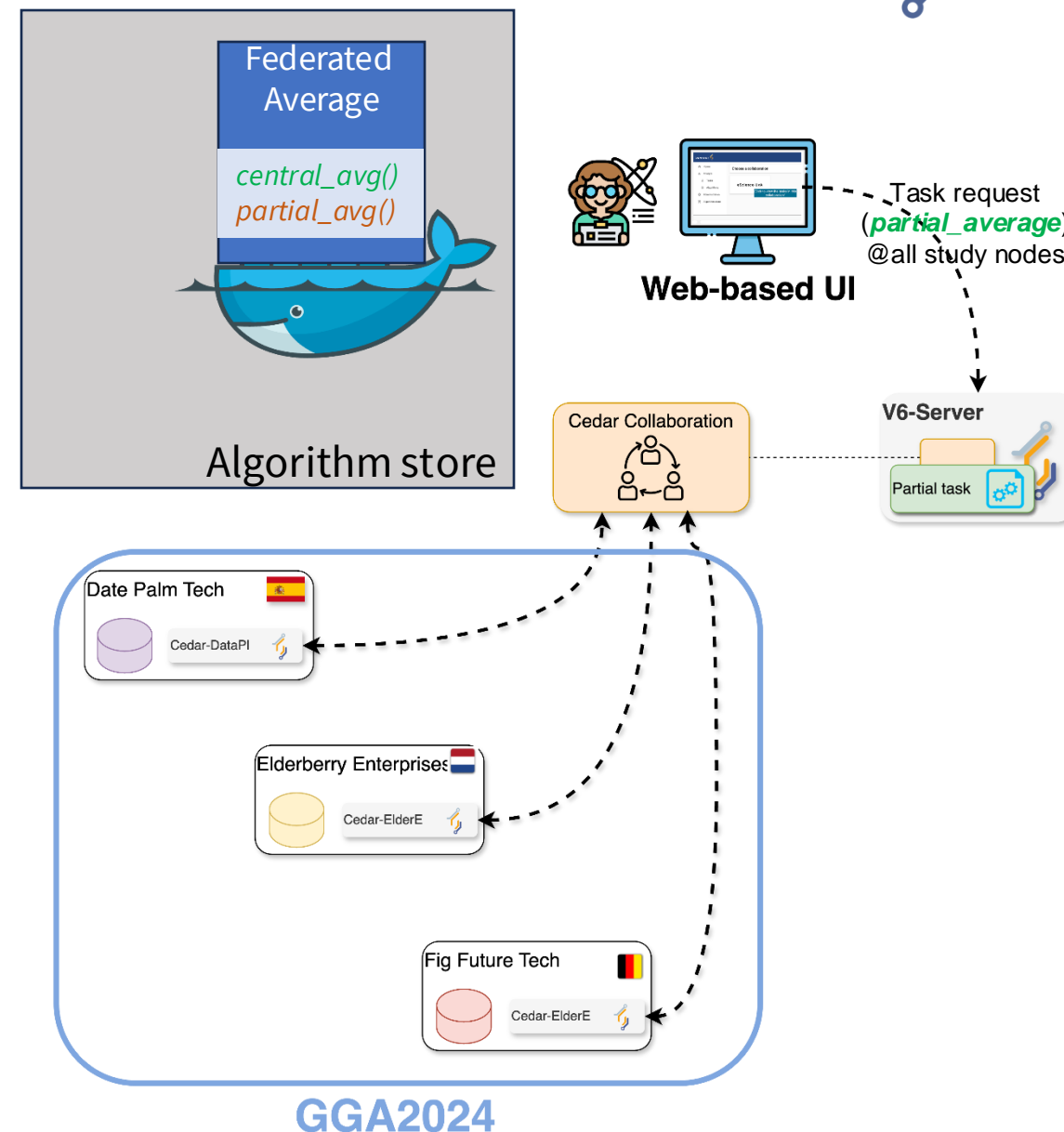
gui-test-collaboration - Date Palm Technologies

gui-test-collaboration - Cherry Digital (Offline since 13/09/2024, 19:30:11)



# Running a PET: federated analysis

- Creating a new task:
  - You will run the **partial\_average** function, of the **Federated Average algorithm**, on **all** the nodes of a study (the one with no offline nodes).



# Running a federated algorithm

## ⚡ CHALLENGE 3

### Create your first task!

Tasks

+ Create task

- Use the *study* with **no** offline nodes.
- **Give your own name to the task so you can find it later.**
- Choose the Average algorithm.
- Select the *partial\_average* function and all the organizations within the study.
- Choose the **default** database.
- Choose a numerical variable as the input.

The task you just requested should be listed with a ‘pending’ status. Once finished, download the JSON results and open them on a text editor.

- What does the content of these files mean?  
Why the results are formatted like that?

gender	age	height	weight	isOverweight	ageGroup
M	39	152	108	False	30 - 40
M	8	118	106	False	0 - 10
M	16	161	110	True	10 - 20
M	94	110	115	True	90 - 100
M	47	117	152	True	40 - 50
F	29	127	110	True	20 - 30
M	5	95	65	False	0 - 10
M	39	142	196	False	30 - 40
F	20	189	112	False	20 - 30
F	84	145	116	False	80 - 90

⚡ HINT

The data on ‘default’ databases on all the nodes looks like this 🙌.

# Running a federated algorithm

## SOLUTION

### Create task

1 Select study

Your collaboration d  
Select the study you

Study\*  
AGOT2024

Next

Home / Tasks / Create task

Selected collaboration **workshop-collab** ⓘ The conten  
sk for.

### Create task

1 Select algorithm

Select which algorithm you

Algorithm\*  
Average

Provide a suitable name fo

Name\*

Optionally, describe your ta

Description

Back Next

3 Select function

Select which function you want to run

Function\*  
partial\_average (federated)

Select the organizations that will carry out a partial function

Organization(s) to run this function\*

☐ Apple

☐ Banana

☐ Cherry

☐ Elderberry

Submit

4 Select databases

Select the database(s) you want to use. The options in the dropdown

Main database

Database\*

default (csv)

kaplan\_meier (csv)

Submit

5 Define parameters

Provide parameter values for the selected algorithm function

Name of the column to calculate average for

column\_name\*

Back

Submit

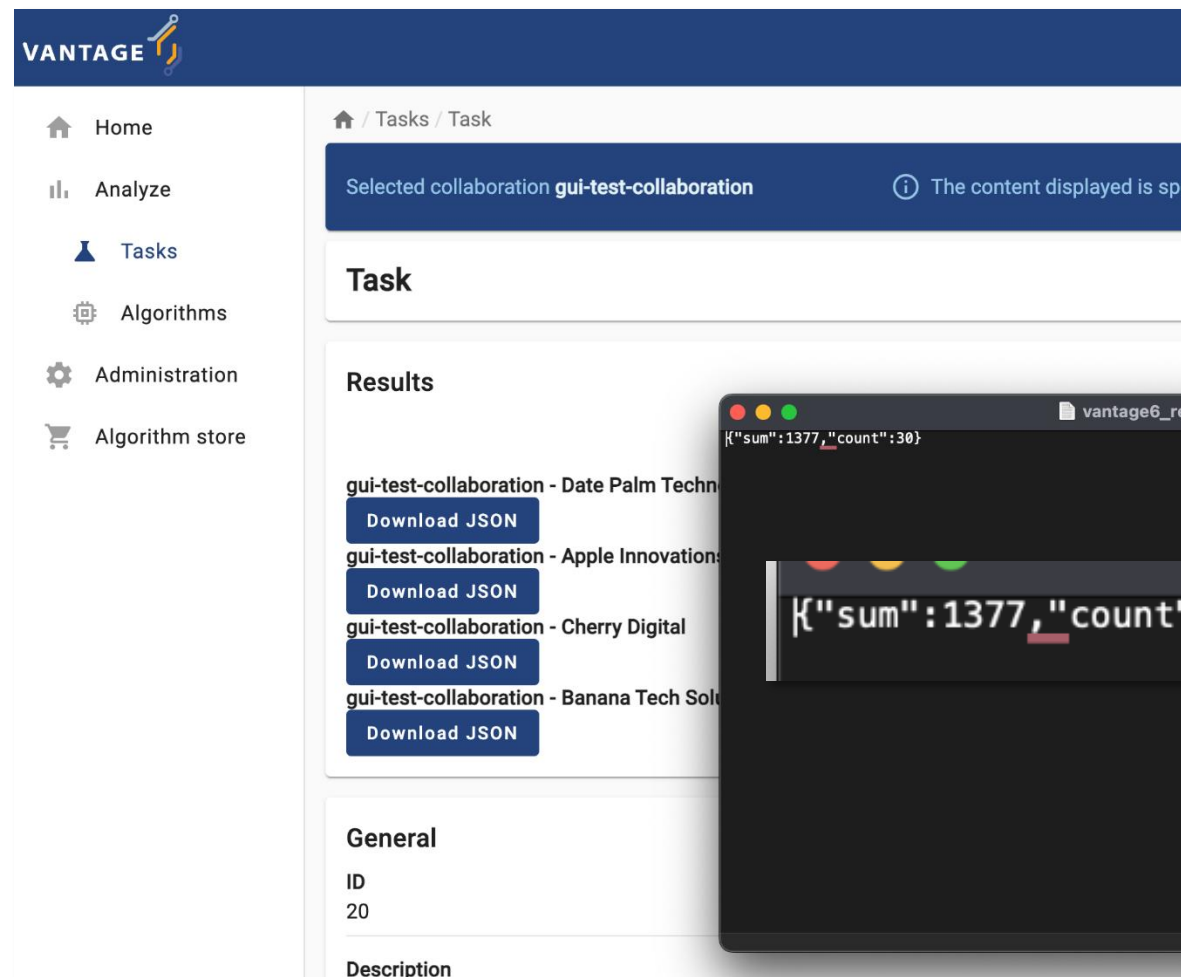


# Getting familiar with the UI

## ⚡ SOLUTION

The ***partial\_average*** returns the two values needed by the central function of the ‘federated average’ algorithm, as described in Chapter 2: the number of records within the database, and their sum.

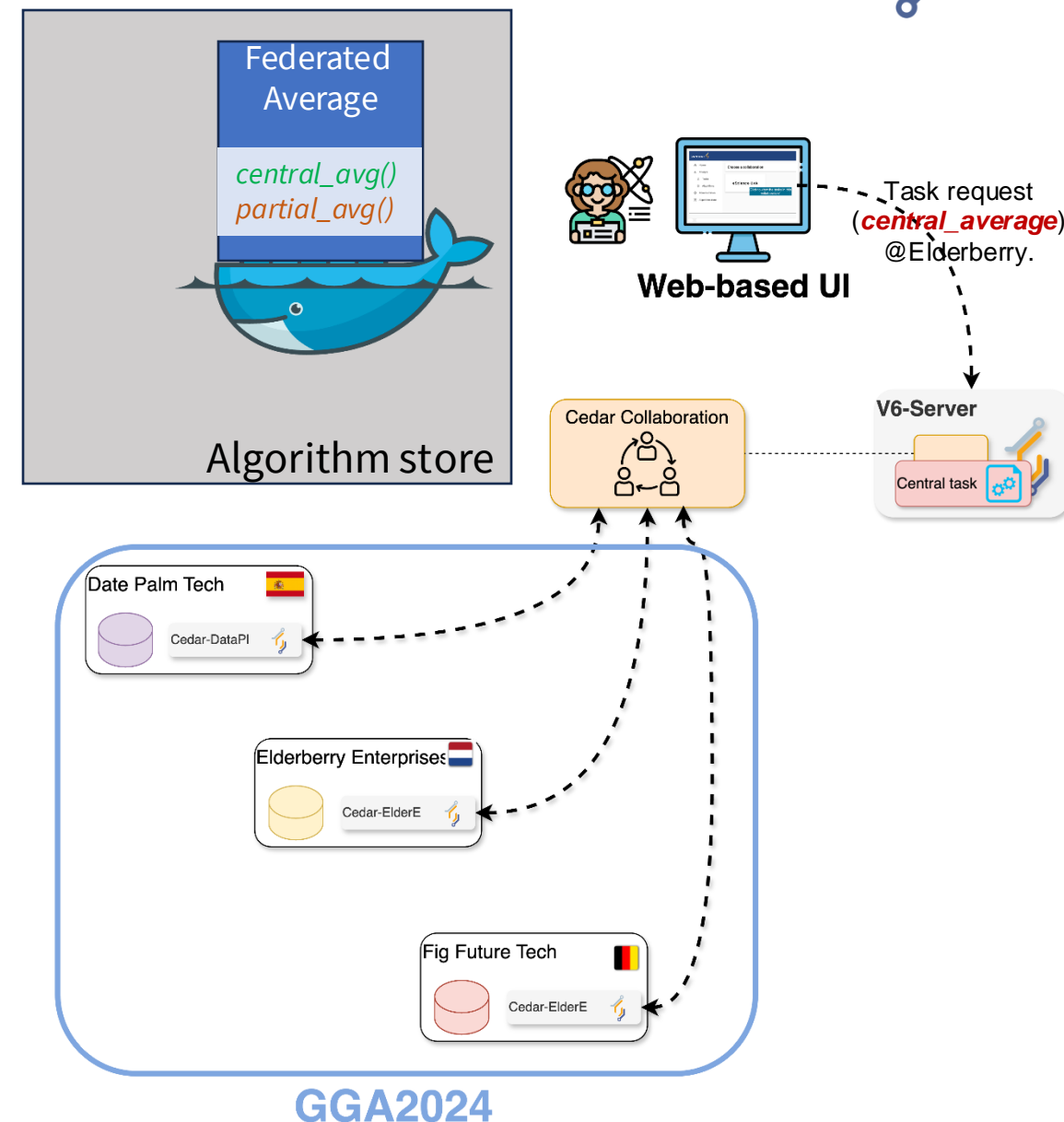
The algorithm is ‘encoding’ these values as a JSON document, so they can be used later (e.g., by another function or program).



The screenshot displays the VANTAGE web application interface. On the left is a navigation sidebar with links: Home, Analyze, Tasks, Algorithms, Administration, and Algorithm store. The main content area shows the 'Task' details for 'gui-test-collaboration'. It includes a 'Results' section with a list of tasks and their corresponding 'Download JSON' buttons. A modal window is open, showing a JSON document: `{"sum":1377,"count":30}`. The 'General' section below the results shows the ID as 20. The 'Description' section is partially visible at the bottom.

# Running a PET: federated analysis

- Let's create yet another task.
  - This time you will run the **central\_average** function, of the **Federated Average algorithm**, on **one** of the study nodes (once again, on the study with **no** offline nodes).



# Running a federated algorithm

## CHALLENGE 4

Let's see what the central function does under the hood!

- Use the study with **no** offline nodes.
- Choose the **Average algorithm**.
- Select the **central\_average** function and one of the organizations within the study.
- Choose the default database.
- Choose a numerical variable as the input.

**Keep an eye on the Tasks section, and see how the processes are created:**

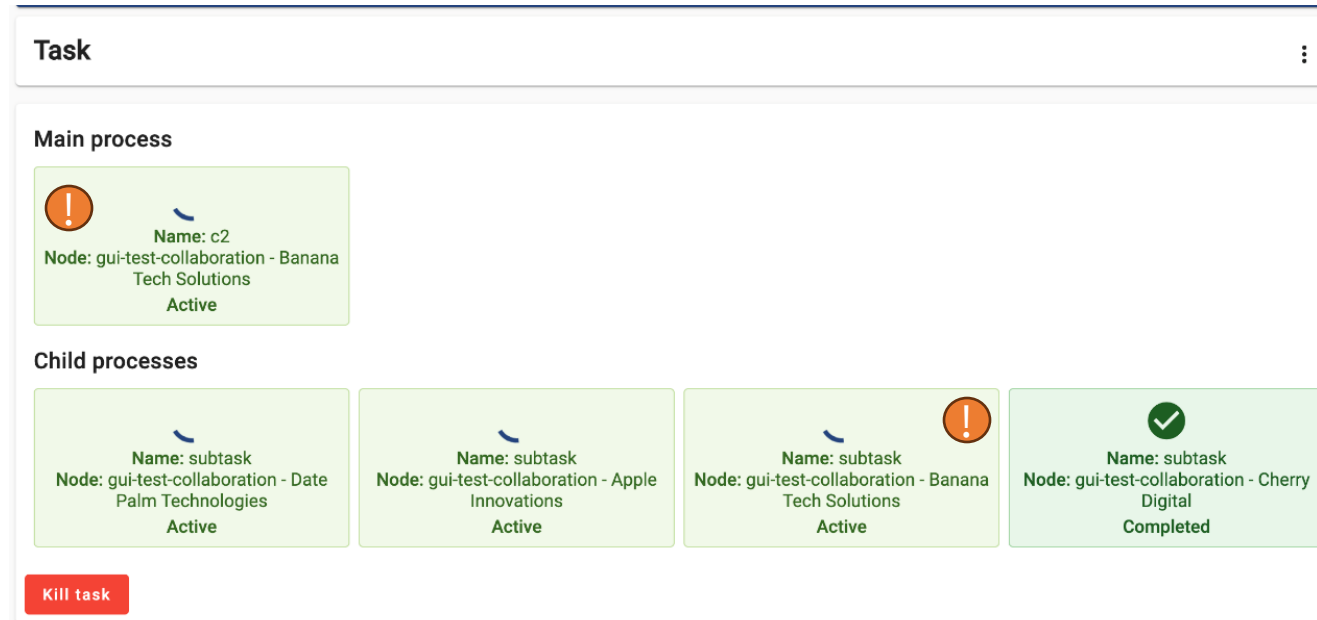
- Why does one of the nodes show up as a Main process and also as a Child one?

## HINT

As soon your task is created, go to the task list and open it. See how 'child processes' are created.

# Running a federated algorithm

## CHALLENGE 4



- Why does one of the nodes show up as a Main process and as a Child one?

# Algorithm pseudo-code

## ⚡ CHALLENGE 4

FUNCTION **partial\_average**(dataset, column\_name)

```
1 INFO("Extracting column", column_name)
2 column_values = dataset[column_name]
3
4 INFO("Computing partials")
5 local_sum = SUM(column_values)
6 local_count = SIZE(column_values)
7
8 RETURN {"sum": local_sum, "count": local_count}
```

END FUNCTION

- Why does one of the nodes show up as a Main process and also as a Child one?

Actual python code?

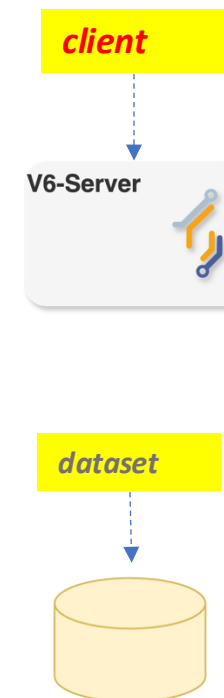
<https://bit.ly/v6-federated-avg>

FUNCTION **central\_average**(client, column\_name)

```
1 study_orgs = client.study.list()
2
3
4 INFO("Requesting partial computation")
5 task = client.task.create(
6     function = "partial_average"
7     column = column_name
8     organizations = study_orgs
9 )
10
11 INFO("Waiting for results")
12 results = client.wait_for_results(task)
13
14 global_sum = 0
15 global_count = 0
16
17 FOR EACH output IN results
18     global_sum = global_sum + output.sum
19     global_count = global_count + output.count
20
21 average = global_sum/global_count
22
23 RETURN {"average": average}
```

END FUNCTION

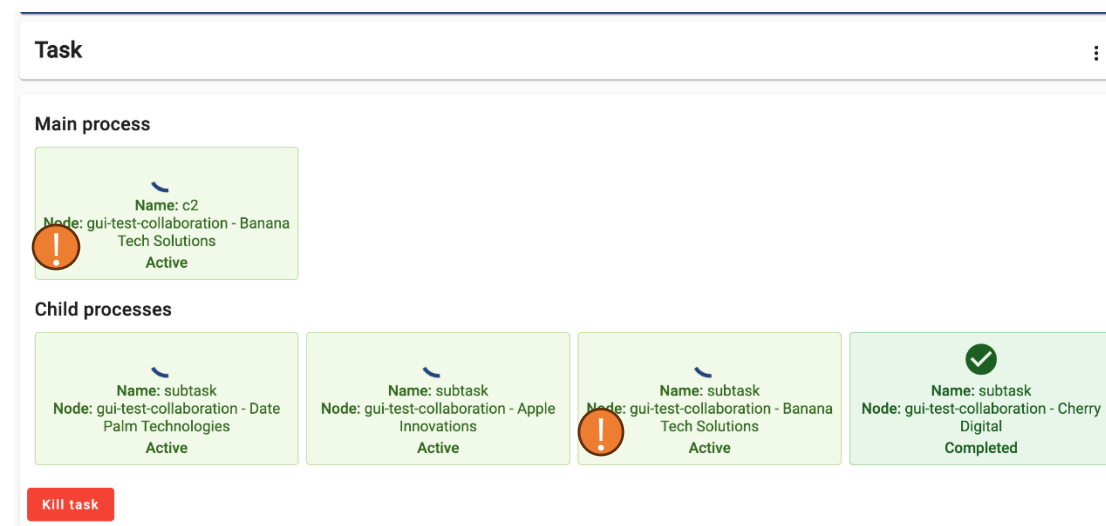
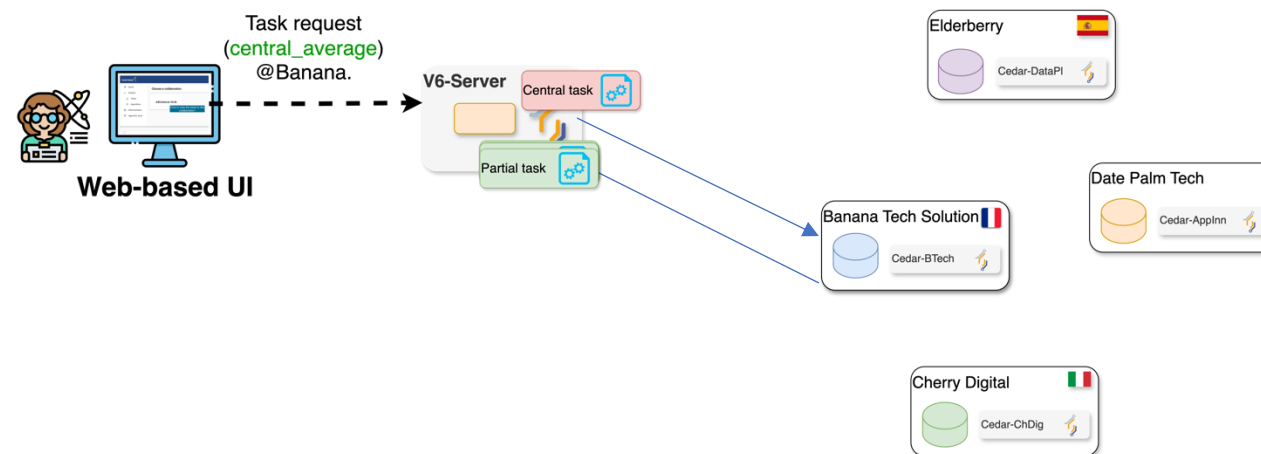
⚡ HINT



# Getting familiar with the UI

## ⚡ SOLUTION

The **Task** with the **central\_average** function, when executed by a node, requests **all** the nodes within the **Study** to run the **partial\_average** one. This particular node is also part of the **Study**, so in the end it gets his own request!



# Revisiting key concepts...

FUNCTION **partial\_average**(dataset, column\_name)

```
1 INFO("Extracting column", column_name)
2 column_values = dataset[column_name]
3
4 INFO("Computing partials")
5 local_sum = SUM(column_values)
6 local_count = SIZE(column_values)
7
8 RETURN {"sum": local_sum, "count": local_count}
```

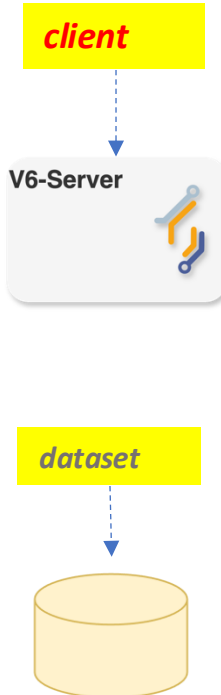
END FUNCTION

- Why does the **central\_average** function, unlike **partial\_average**, not get any dataset as an input?
- Why does the **partial\_average** function, unlike **central\_average**, not get any client as an input?

FUNCTION **central\_average**(client, column\_name)

```
1 study_orgs = client.study.list()
2
3
4 INFO("Requesting partial computation")
5 task = client.task.create(
6     function = "partial_average"
7     column = column_name
8     organizations = study_orgs
9 )
10
11 INFO("Waiting for results")
12 results = client.wait_for_results(task)
13
14 global_sum = 0
15 global_count = 0
16
17 FOR EACH output IN results
18     global_sum = global_sum + output.sum
19     global_count = global_count + output.count
20
21 average = global_sum/global_count
22
23 RETURN {"average": average}
```

END FUNCTION

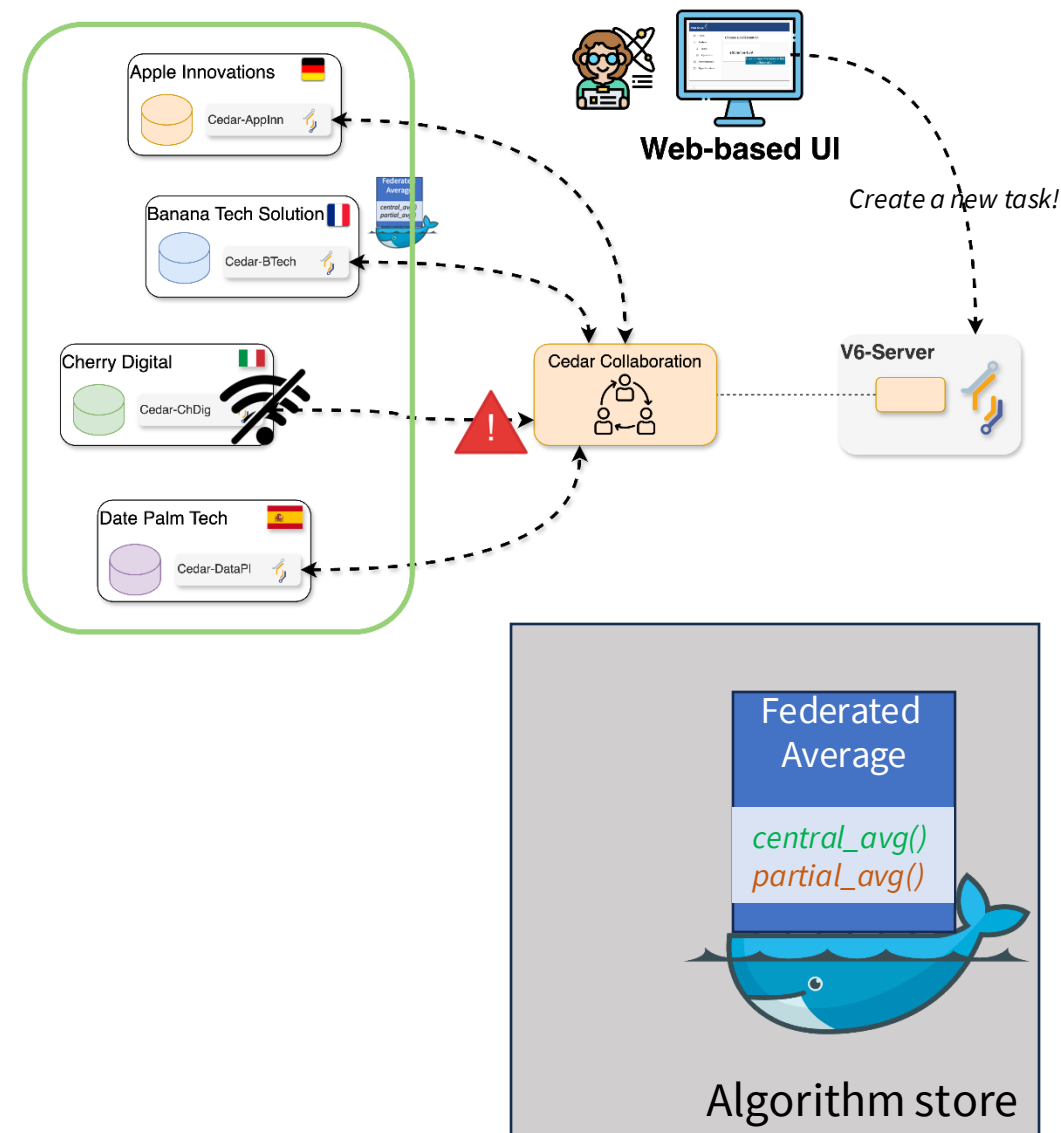


# Running a PET: federated analysis



## Problems on a node?

- What if...
  - You run the **central\_average** function (**Federated Average algorithm**), on the study that has an **offline** node?





# Running a federated algorithm

## ⚡ CHALLENGE 5

handling problems through the UI!

- Use the study that has **an offline node**.
- Choose the **Average algorithm**.
- Select the **central\_average** function and an **online** organizations within the study.
- Choose the *default* database.
- Choose a numerical variable as the input.

What happened with the task? Why?

FUNCTION **central\_average**(**client**, column\_name)

```

1  study_orgs = client.study.list()
2
3
4  INFO("Requesting partial computation")
5  task = client.task.create(
6      function = "partial_average"
7      column = column_name
8      organizations = study_orgs
9  )
10
11  INFO("Waiting for results")
12  results = client.wait_for_results(task)
13
14  global_sum = 0
15  global_count = 0
16
17  FOR EACH output IN results
18      global_sum = global_sum + output.sum
19      global_count = global_count + output.count
20
21  average = global_sum/global_count
22
23  RETURN {"average": average}

```

END FUNCTION

⚡ HINT

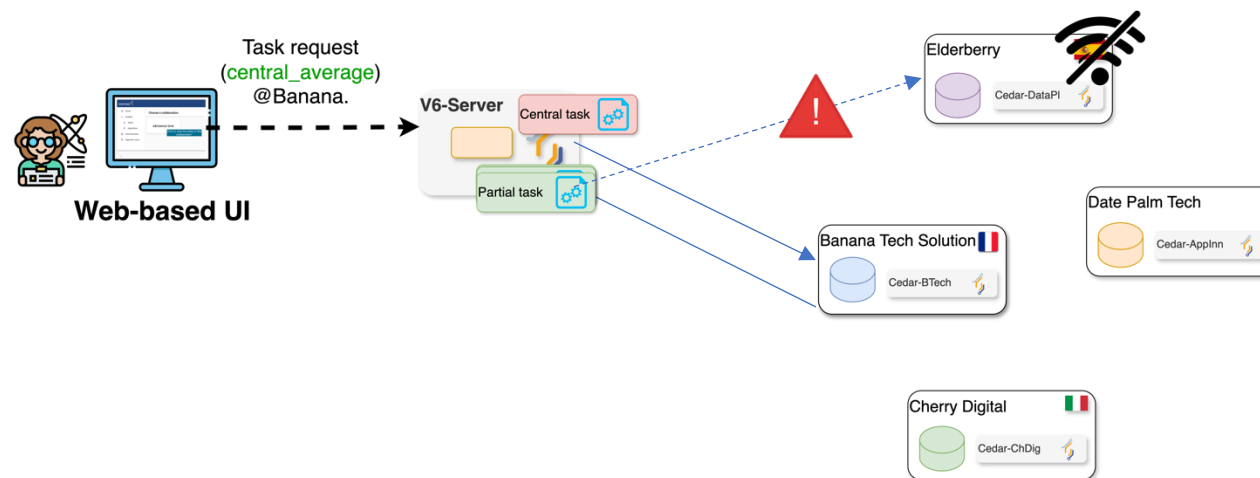
# Getting familiar with the UI

## ⚡ SOLUTION

The algorithm didn't crash.

The Central task requests all the nodes in the study to run the 'partial' function. As the server is unable to transfer this request to the offline node, this child process is kept on hold, until the node is back online.

Consequently, the Main process is also kept on hold, and the process stays with an 'Active' status indefinitely (or until the node is back online).



**Task**

**Main process**

Name: r2  
Node: gui-test-collaboration - Banana Tech Solutions  
Active

**Child processes**

Name: subtask  
Node: gui-test-collaboration - Elderberry Enterprises  
Pending  
Task can't start, node is offline.

✓  
Name: subtask  
Node: gui-test-collaboration - Cherry Digital  
Completed

✓  
Name: subtask  
Node: gui-test-collaboration - Banana Tech Solutions  
Completed

✓  
Name: subtask  
Node: gui-test-collaboration - Date Palm Technologies  
Completed

**Kill task**

netherlands  
eScience center

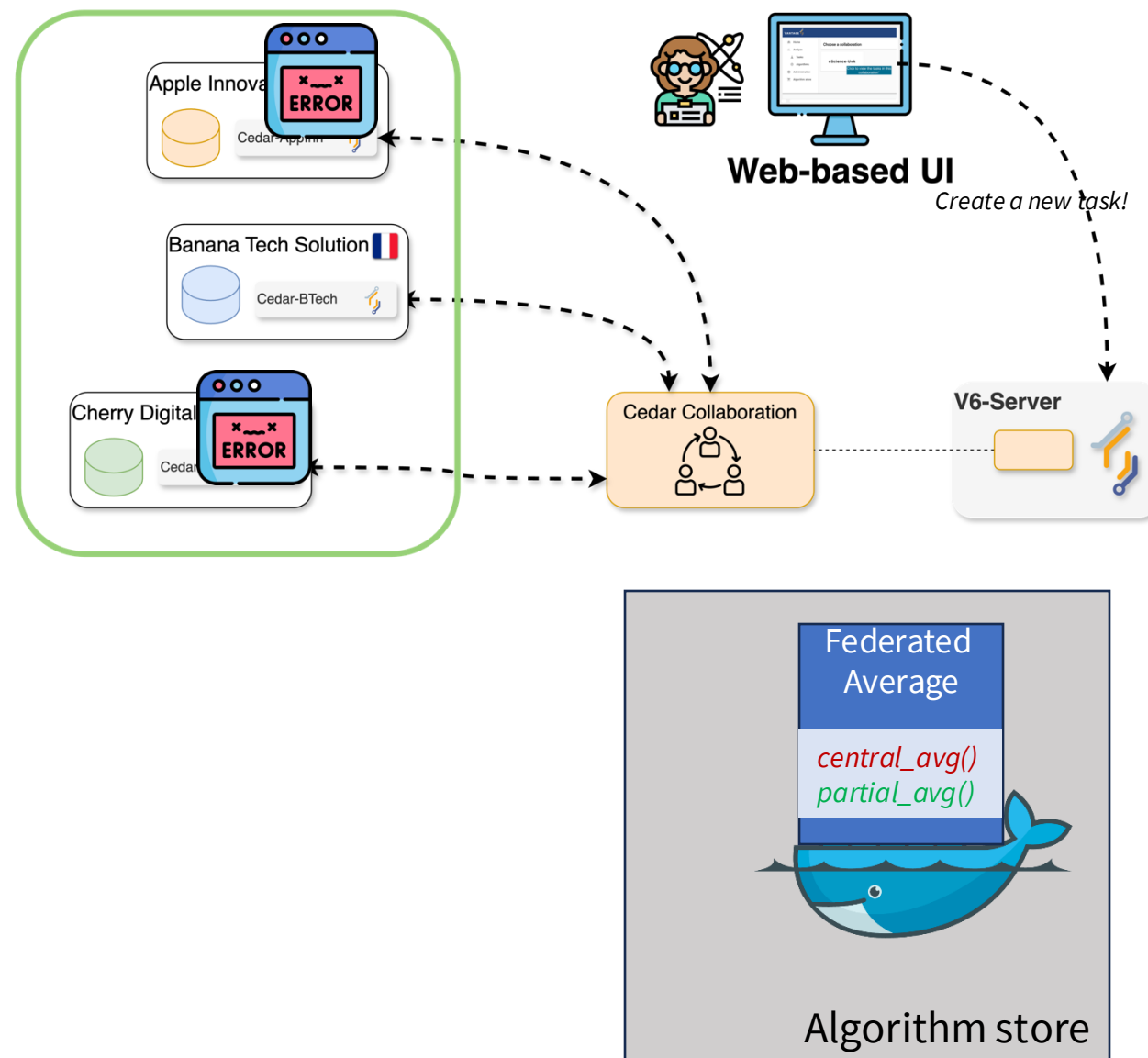
**KNL** integraal  
kankercentrum  
Nederland

# Running a PET: federated analysis



## Problems at the code level?

- What if...
  - You run the **central\_average** function (**Federated Average algorithm**), and later on a node gets a runtime error? (e.g., a code bug, unexpected input, etc.)



# Running a federated algorithm

## ⚡ CHALLENGE 6

### Handling problems through the UI!

- Use the study that has **no** offline nodes.
- Choose the **Average algorithm**.
- Select the **central\_average** function and one of the organizations within the study.
- Choose the default database.
- Choose the **ageGroup** variable as the input.

Look at the error logs.

Show logs

What was the cause of the failure of the **partial\_average**? What about the **central\_average**?

gender	age	height	weight	isOverweight	ageGroup
M	39	152	108	False	30 - 40
M	8	118	106	False	0 - 10
M	16	161	110	True	10 - 20
M	94	110	115	True	90 - 100
M	47	117	152	True	40 - 50
F	29	127	110	True	20 - 30
M	5	95	65	False	0 - 10
M	39	142	196	False	30 - 40
F	20	189	112	False	20 - 30
F	84	145	116	False	80 - 90

### ⚡ HINT

The error logs contain the error messages generated by the Python interpreter while running the algorithm functions. Look for messages that hint the root cause of the problem.

# Running a federated algorithm

## ⚡ CHALLENGE 6

```
FUNCTION partial_average(dataset, column_name)
```

```
1 INFO("Extracting column", column_name)
2 column_values = dataset[column_name]
3
4 INFO("Computing partials")
5 local_sum = SUM(column_values)
6 local_count = SIZE(column_values)
7
8 RETURN {"sum": local_sum, "count": local_count}
```

```
END FUNCTION
```

⚡ HINT

Look at the error logs. What was the cause of the failure of the **partial\_average**? What about the **central\_average**?

```
FUNCTION central_average(client, column_name)
```

```
1 study_orgs = client.study.list()
2
3
4 INFO("Requesting partial computation")
5 task = client.task.create(
6     function = "partial_average"
7     column = column_name
8     organizations = study_orgs
9 )
10
11 INFO("Waiting for results")
12 results = client.wait_for_results(task)
13
14 global_sum = 0
15 global_count = 0
16
17 FOR EACH output IN results
18     global_sum = global_sum + output.sum
19     global_count = global_count + output.count
20
21 average = global_sum/global_count
22
23 RETURN {"average": average}
```

```
END FUNCTION
```

⚡ HINT

# Getting familiar with the UI

## ⚡ SOLUTION

1. This time, the **partial\_average** function on all the data nodes crashes, which lead to the failure of the Main process (**central\_average**).
2. The **partial\_average** crashed while trying to parse the input as a number.  
The **central\_average** crashed due to a division-by-zero.

**Task** ⋮

---

**Main process**

❗

Name: asds  
Node: gui-test-collaboration - Cherry Digital  
Error during execution

Show logs

**Child processes**

❗

Name: subtask  
Node: gui-test-collaboration - Cherry Digital  
Error during execution

Show logs

❗

Name: subtask  
Node: gui-test-collaboration - Date Palm Technologies  
Error during execution

Show logs

❗

Name: subtask  
Node: gui-test-collaboration - Banana Tech Solutions  
Error during execution

Show logs

❗

Name: subtask  
Node: gui-test-collaboration - Apple Innovations  
Error during execution

Show logs

# Play around with other algorithms

## ⚡ CHALLENGE 7

Try to run other (more advanced) algorithms!

### 2 Select algorithm

Select which algorithm you want to run

Algorithm\* ▼

- Contingency table
- Kaplan-Meier
- Average
- Description

age	months	censor	days
56	31	1	926
58	56	0	1690
62	79	0	2374
44	52	0	1569
47	55	0	1646
62	34	1	1010
63	88	0	2646
68	70	0	2105
35	103	0	3075
64	94	0	2832

⚡ **HINT**  
For Kaplan-Meier, choose the  
*kaplan\_meier (csv)* database!

