

Understanding AI Experiment Assets - A « Hello world » Project

- IMT Teralab - Assistant Project

AloD - TCB - 22 September 2023

ASSISTANT, being executed by 12 partners from 7 nations, has received funding from European H2020 programme



The ASSISTANT Project

“develop breakthrough solutions for the **manufacturing industry**, using **artificial intelligence** to optimize **production systems**”



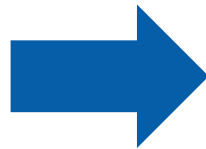
- One of the keystones of ASSISTANT is the creation of intelligent digital twins by combining machine learning, optimization, simulation, and domain models
- ASSISTANT develops tools and solutions providing all required information to help production managers design production lines, plan production, and improve machine settings for effective and sustainable decisions that guarantee product quality and safety.



Context

- On May 17, 2022 an Integration Workshop took place around AI4Europe
 - Presentation of assets from ICT projects
 - Presentation of AI4Europe CMS
 - Presentation of AI4EU Experiments

One feedback received after the integration Workshop is that the development of assets to AI Experiments « seems » complicated.



We produced a guide to describe how to publish on AloD shared through the consortium

We started a work on a « hello word » project:

- Simple usecase, light code
- Focus on GRPC and Pipelines
- User friendly and easy to understand

What is the «hello world » project?

- A working implementation of two containers **exchanging very simple text message in GRPC**
- Simple code through UX, written in NodeJS with comments explaining the most important lines of code.
- Two different setups highlighting the key concepts to be run locally or through execution spaces

*“All you need **<and nothing more>** to develop your own models and solutions to be executed within AI4Experiments”.*

Open Source : <https://gitlab.telecom-paris.fr/teralab/ai-experiment-hello-world>

Scenario

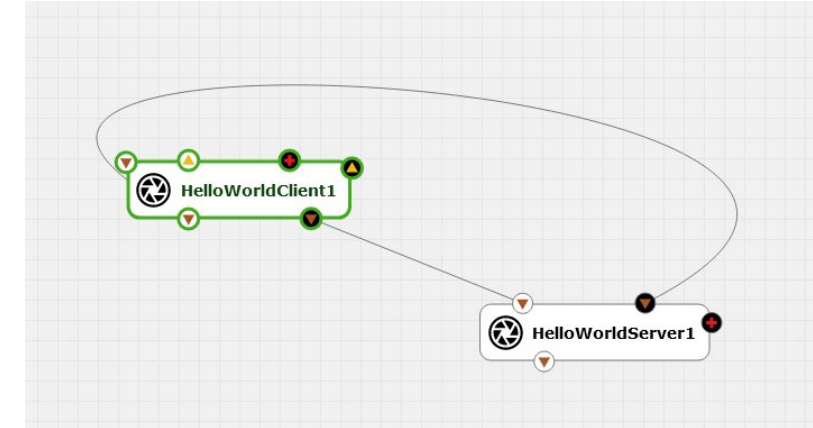
Two containers : a server (SRV) and a client (CLIENT) each implementing their own web interfaces.

In the client's web interface:

1. The user enter his name.
2. After validation, the client invoke the sayHello method.
3. The sayHello method just reply "hello <user>".

The server web interface

- displays a history of all the messages received and processed.
- Run the « sayHello » method



Use case

Hello World example - Client View

By IMT TERALAB

An introduction to AIoD AI Experiments

[Source code](#) - [Contact](#)

Request

Response

Hello again Benoit (for the 2 nd time)

GRPC
(Orchestrator)

Hello World example - Server View

By IMT TERALAB

An introduction to AIoD AI Experiments

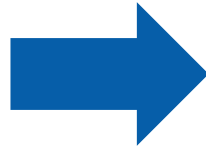
[Source code](#) - [Contact](#)

Processing History

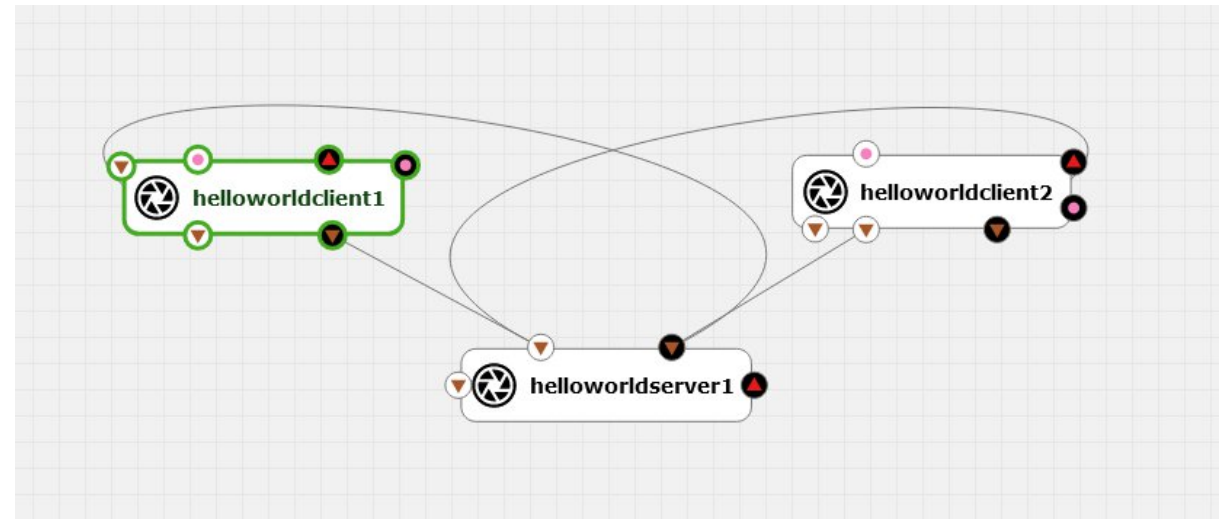
Time	Input	Output
19/09/2023 17:31:47	Benoit	Hello Benoit
19/09/2023 17:31:47	Benoit	Hello again Benoit (for the 2 nd time)

How to use it

1. Follow our instruction in the readme
2. Test and adapt the code to fit your usecase
3. Onboard your new model
4. Create new pipelines



Example of a Hello World with two clients and one server



You can execute the solution in any AI Experiment playground

Find the code on our gitlab instance:
<https://gitlab.telecom-paris.fr/teralab/ai-experiment-hello-world>

Find the models and the solutions on AI Experiments <https://aiexp.ai4europe.eu>

Deployment: HelloWorld

Status: Ready | Run | Reset | Delete | Logs

Status Check	Nodename	Status Details	Logs	WebUI/Folder
✓	helloworldclient1	show	view	
✓	helloworldserver1	show	view	
✓	orchestrator	show	view	

Benoit

Where?

Find the code on our gitlab instance:

<https://gitlab.telecom-paris.fr/teralab/ai-experiment-hello-world>

Find the models and the solutions on AI Experiments & AI4Europe (in a few days)

<https://aiexp.ai4europe.eu>

Thanks

ASSISTANT, being executed by 12 partners from 7 nations, has received funding from European H2020 programme



main ▾ ai-experiment-hello-world / srv / + ▾ History Find file Edit ▾ ⬇ ▾ Clone ▾

Name	Last commit	Last update
..		
node_modules	added some more text about deployments + a little more details about ai experiments	7 months ago
utils	init .	7 months ago
views		
.env		
Dockerfile		
greeter-srv.proto		
index.js		
package-lock.json		
package.json		

```
function sayHello(call, callback) {
  var name = call.request.name, message = '';
  console.log('gRPC server | Name :', name);
  var n = history.filter(e=>e.name==name).length;
  if (n == 0){
    message = `Hello ${name}`;
  }
  else {
    var th = (n+1) % 10 == 1 ? 'st' : (n+1) % 10 == 2 ? 'nd' : (n+1) % 10 == 3 ? 'rd' : 'th';
    message = `Hello again ${name} (for the ${n+1} ${th} time)`
  }

  var h = { date: Date.now(), name, message };
  history.push(h);
  //
  // Alert UI of new request processed
  //
  mainEvent.emit('entry', h)
  callback(null, { message });
}
```