AI4EU Experiments
Sudoku Hello World - Tutorial

Peter Schüller

peter.schueller@tuwien.ac.at
contact@peterschueller.com

Technische Universität Wien, Austria

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

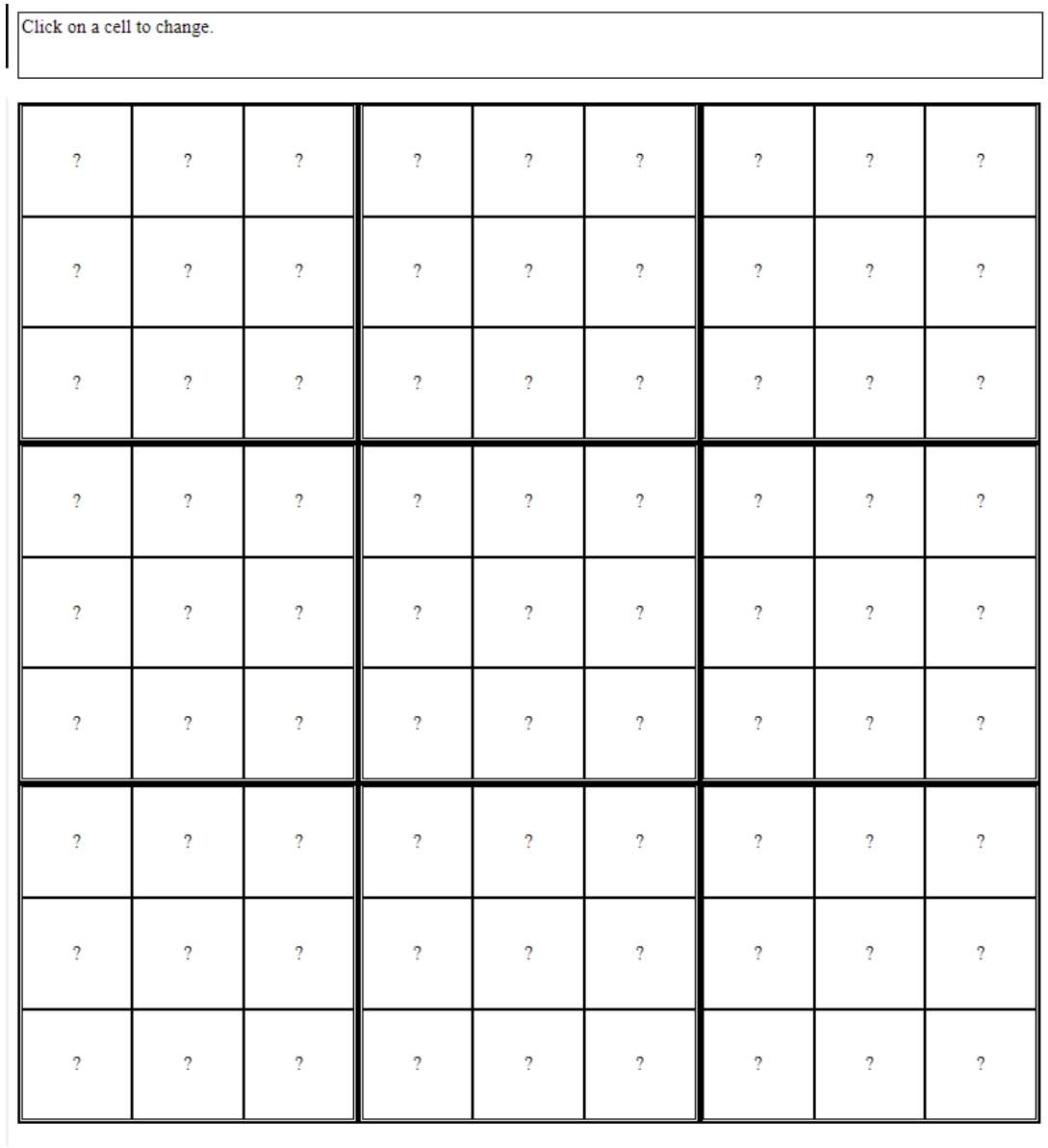
Sudoku Hello World

- A simple ”Hello World” example of an AI4EU Experiment.
- The purpose is to show how to use the Platform.
- The example is …
  - … Open Source: https://github.com/peschue/ai4eu-sudoku/
  - … Public Domain: MIT License
- You can use it as a template for your own Experiments.
- You can use parts of it.
- A Cheatsheet (all steps/commands) is available in the video description and in the above repository

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825619

User Interface

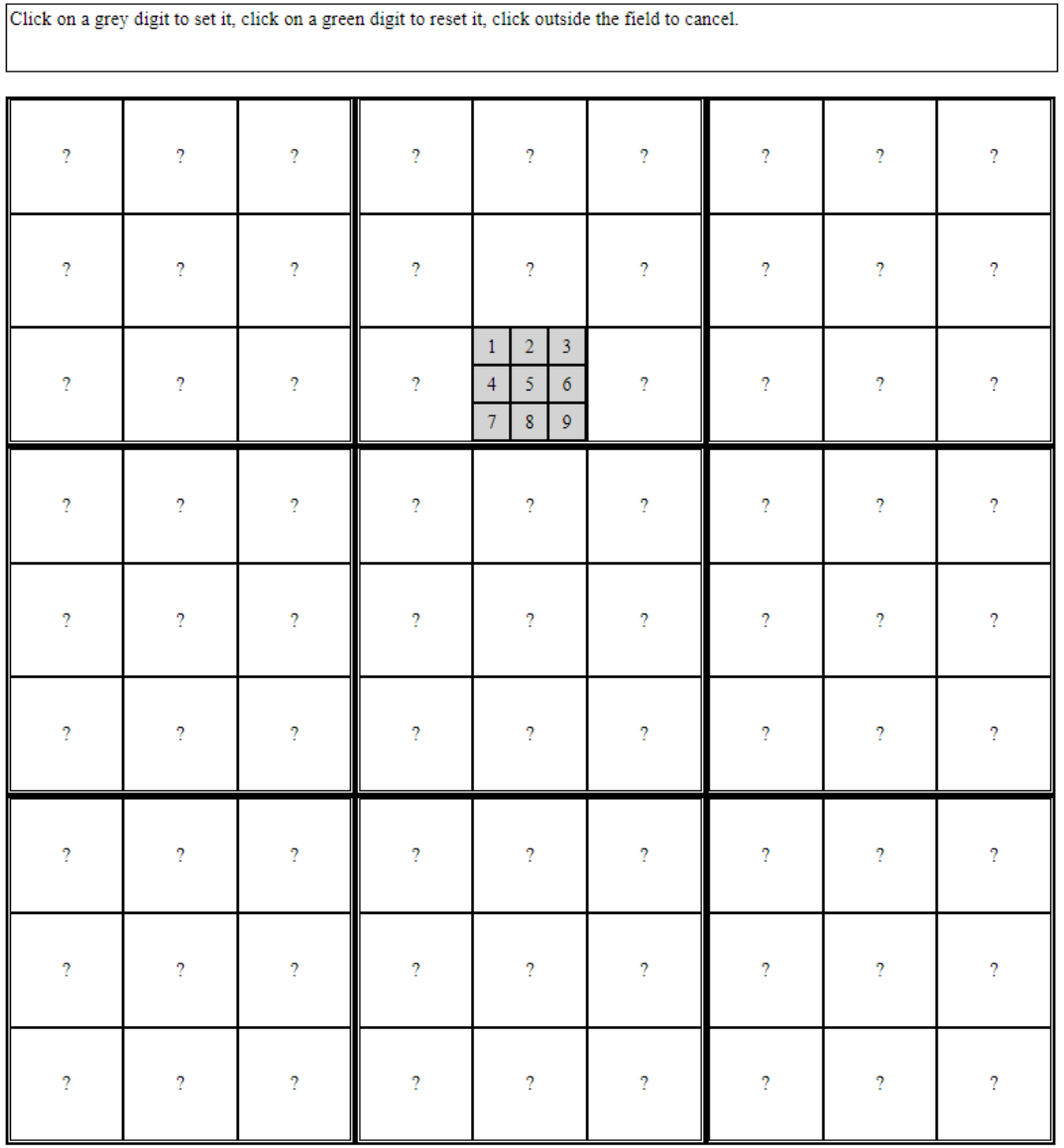
- Status Text on top
- Sudoku Field below



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

User Interface

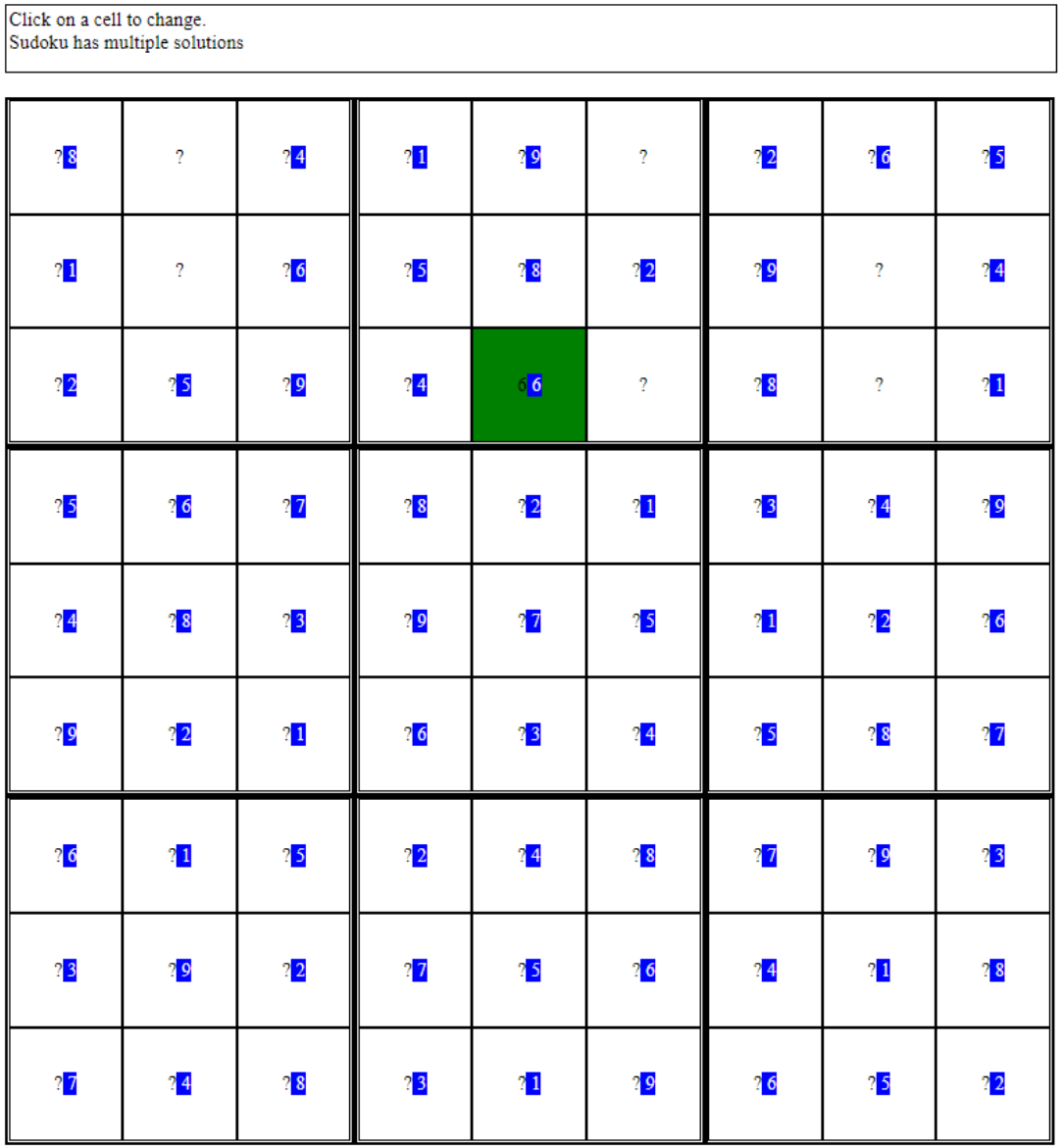
- Status Text on top
- Sudoku Field below
- Click on Field permits to set cells to certain numbers



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

User Interface

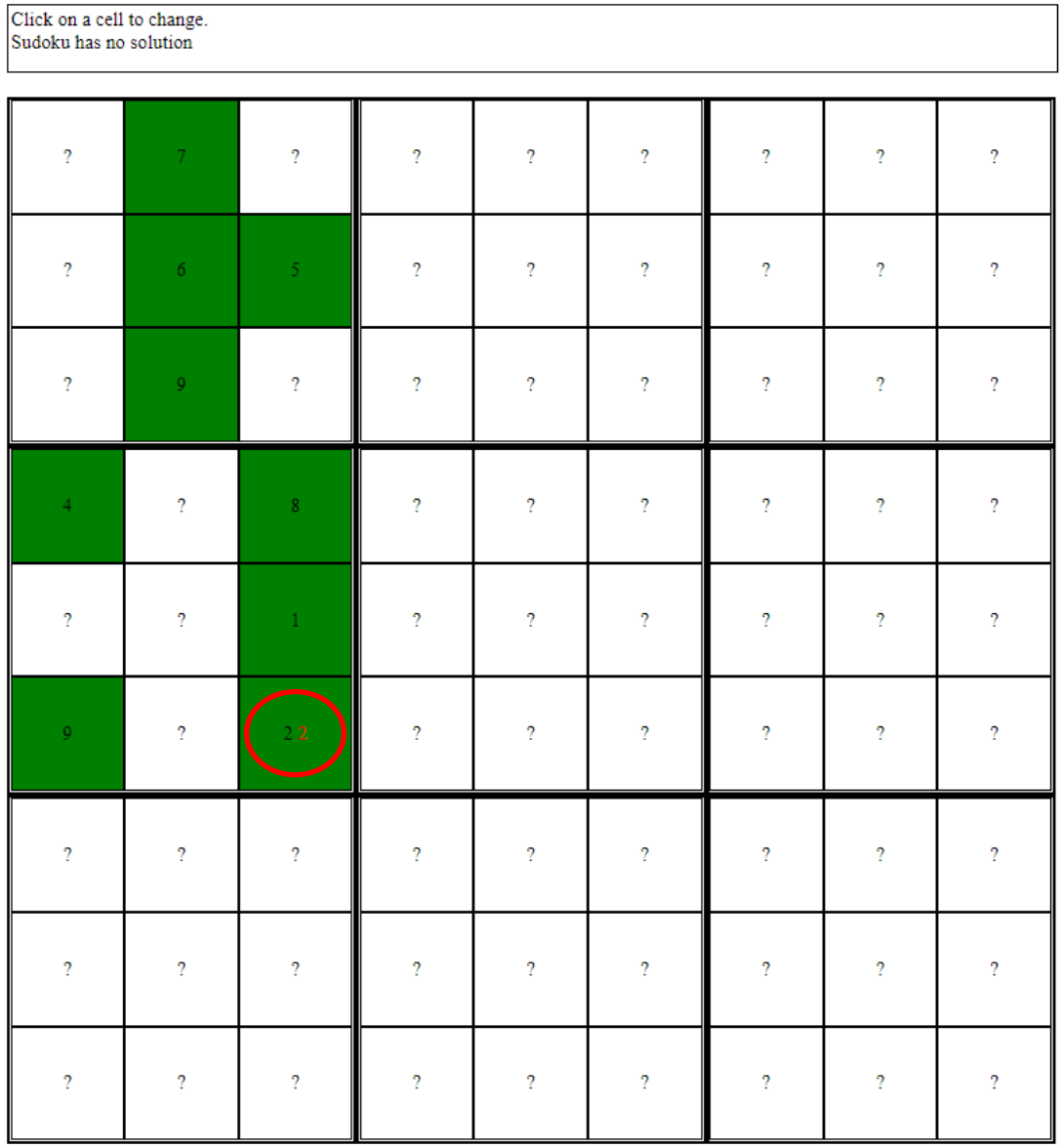
- Status Text on top
- Sudoku Field below
- Click on Field permits to set cells to certain numbers
- Changing the fields shows a
  - partial solution, if one exists
  - full solution, if a unique one exists



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

User Interface

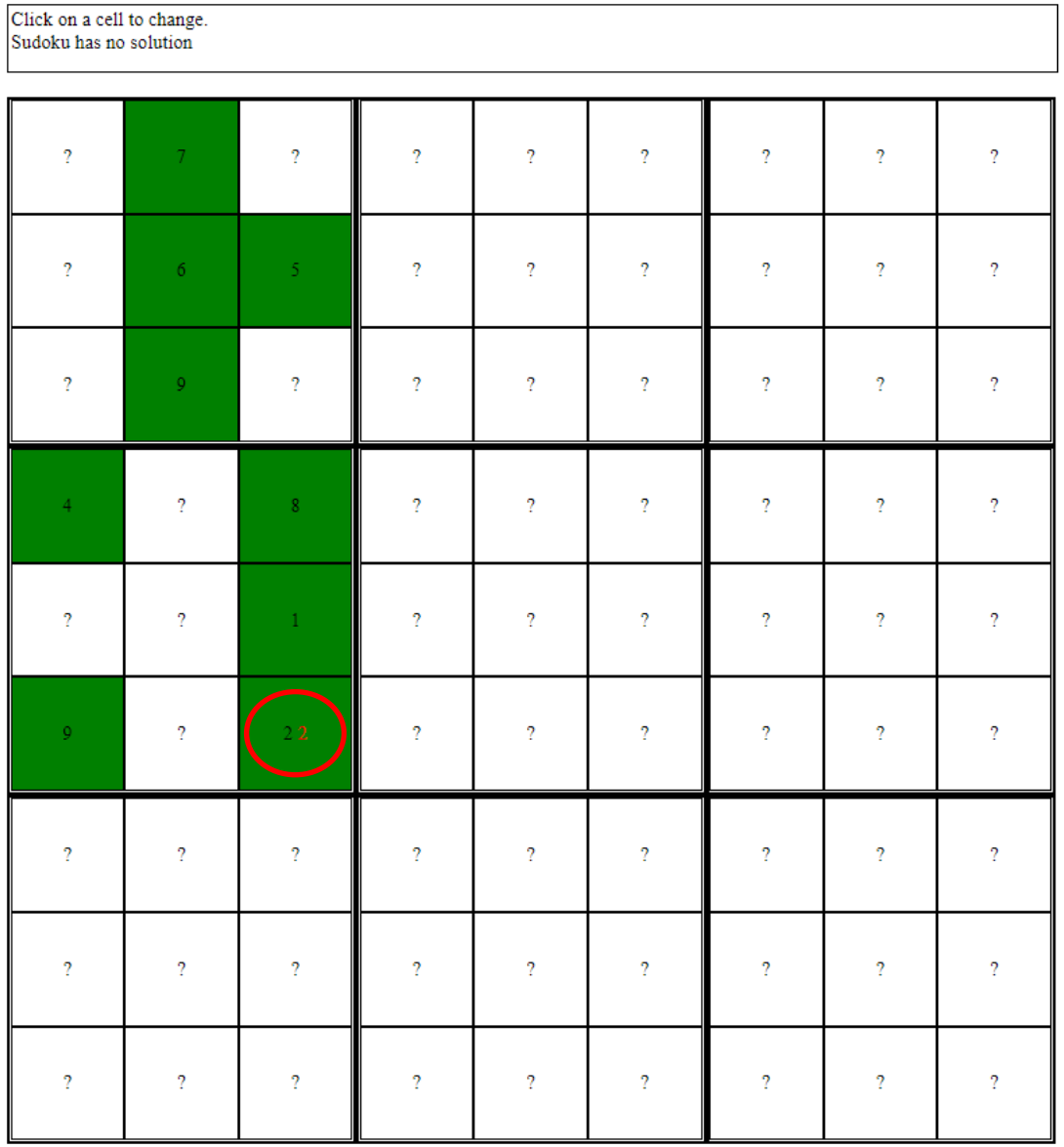
- Status Text on top
- Sudoku Field below
- Click on Field permits to set cells to certain numbers
- Changing the fields shows a
  - partial solution, if one exists
  - full solution, if a unique one exists
  - Conflict (red), if no solution exists



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

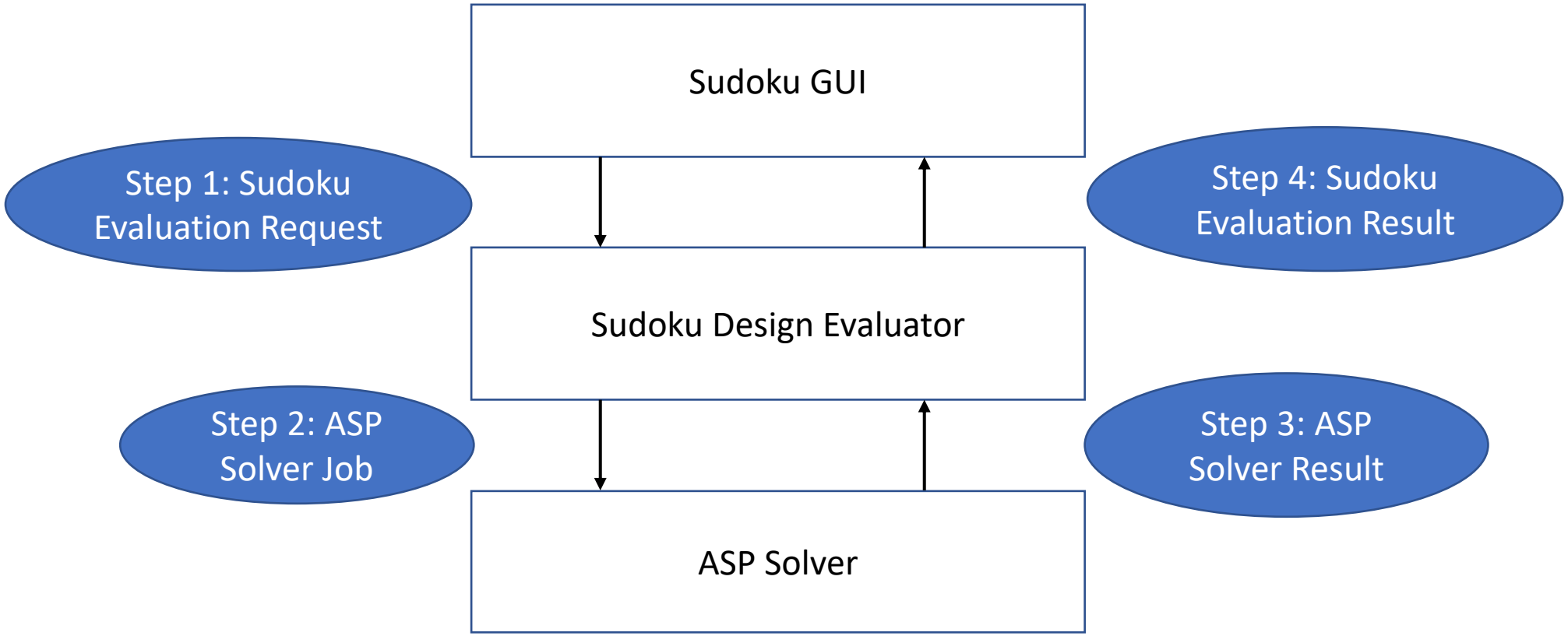
User Interface

- Status Text on top
- Sudoku Field below
- Click on Field permits to set cells to certain numbers
- Changing the fields shows a
  - partial solution, if one exists
  - full solution, if a unique one exists
  - Conflict (red), if no solution exists
- How to set a cell to “?”:
  set the cell to the same number



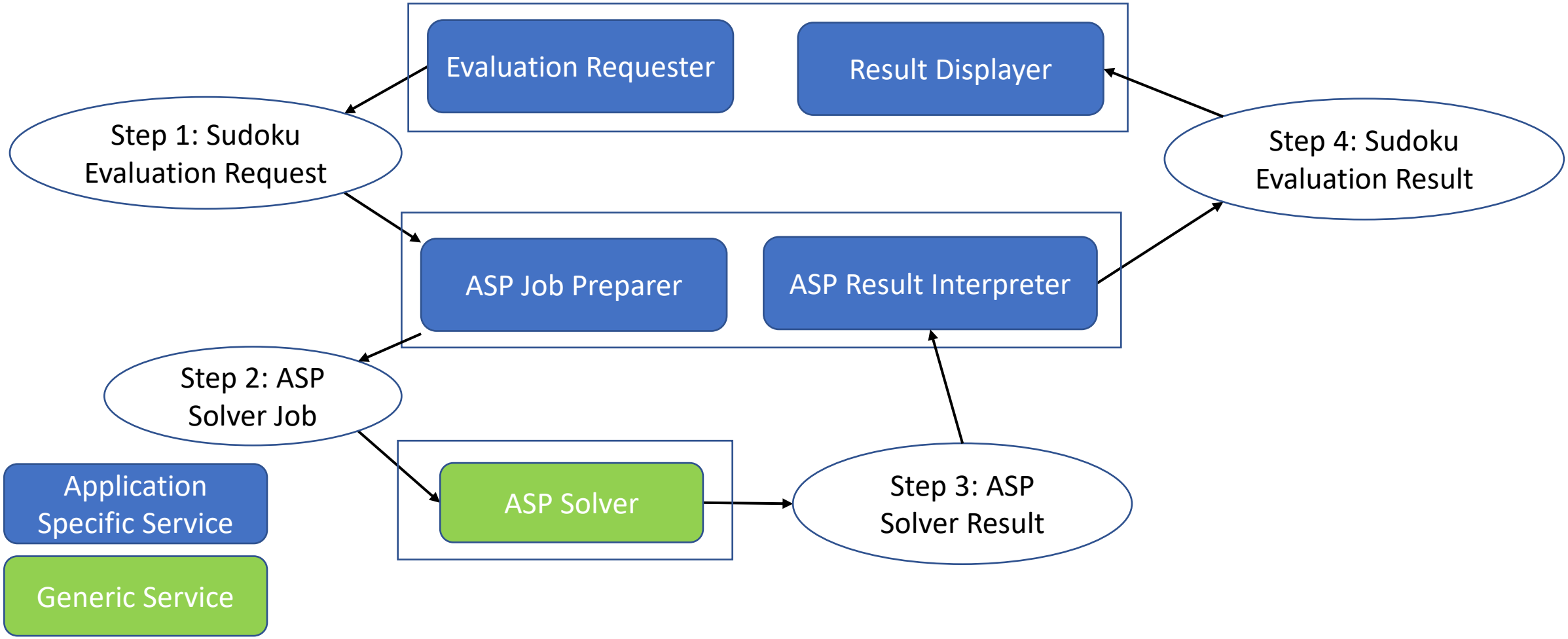
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825619

Components and Messages



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Services – Generic and Application-Specific



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Protobuf Services

```
service SudokuGUI {
  rpc requestSudokuEvaluation(Empty)
       returns(SudokuDesignEvaluationJob);
  rpc processEvaluationResult(SudokuDesignEvaluationResult)
       returns(Empty);
}

service SudokuDesignEvaluator {
  rpc evaluateSudokuDesign(SudokuDesignEvaluationJob)
       returns (SolverJob);
  rpc processSolverResult(SolveResultAnswersets)
       returns(SudokuDesignEvaluationResult);
}

service OneshotSolver {
    rpc solve(SolverJob) returns (SolveResultAnswersets);
}
```

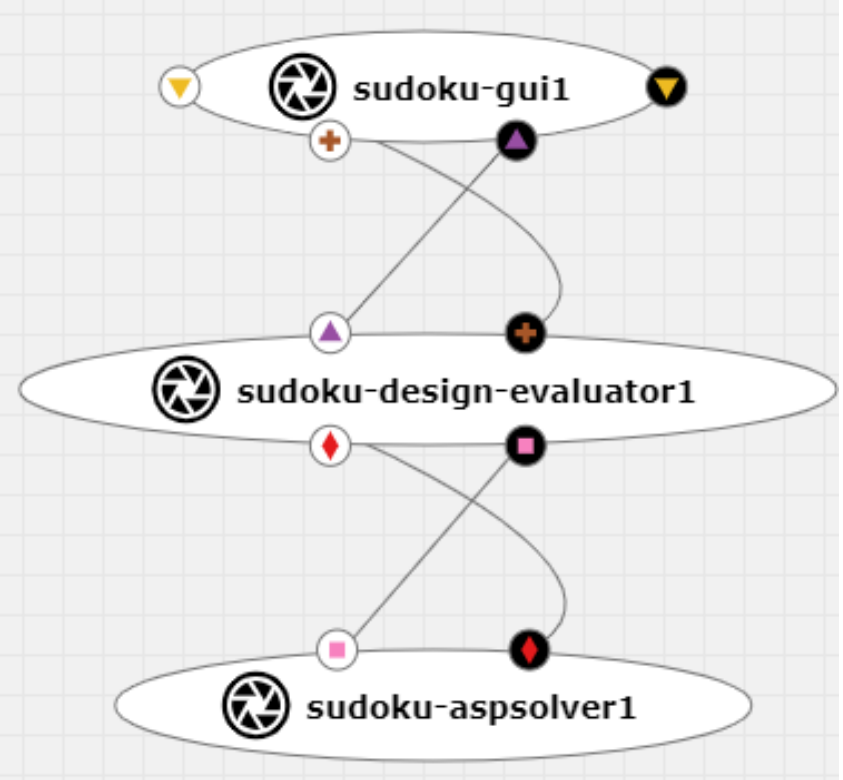

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Protobuf Services

```
service SudokuGUI {
  rpc requestSudokuEvaluation(Empty)
       returns(SudokuDesignEvaluationJob);
  rpc processEvaluationResult(SudokuDesignEvaluationResult)
       returns(Empty);
}

service SudokuDesignEvaluator {
  rpc evaluateSudokuDesign(SudokuDesignEvaluationJob)
       returns (SolverJob);
  rpc processSolverResult(SolveResultAnswersets)
       returns(SudokuDesignEvaluationResult);
}

service OneshotSolver {
    rpc solve(SolverJob) returns (SolveResultAnswersets);
}
```



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Prerequisites

- Building and Onboarding
  - Git: clone https://github.com/peschue/ai4eu-sudoku/
  - Python
  - Docker: https://docs.docker.com/
- Deployment and Running
  - Python
  - Kubernetes environment, for example Minikube https://minikube.sigs.k8s.io/docs/start/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Building

- Configure your docker repository in `helper.py`
  This demo uses the AI4EU internal docker image repository.
  Please contact Martin Welß if you think that you should get an account.
- Log in to your docker repository
  ```
  $ docker login <your-repository>
  ```
- Build 3 docker images for the 3 components:
  ```
  $ ./helper.py build
  ```
- Tag and push the images to the repository:
  ```
  $ ./helper.py tag-and-push
  ```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Onboarding

- Sign in to the AI4EU Experiments Platform.
- Choose “On-boarding Model” on the left.
- For each component:
  - Choose a model name.
  - Enter Host and Port of the docker registry.
  - Enter Path (image) and Tag of the docker image.
  - Browse for the Protobuf File <u>and click “Upload”</u>.
- <u>Wait until the Status is green before uploading the next Component!</u>

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825619

Orchestrator

- The directory “orchestrator” in the repository
  - is useful for development
  - does not need to be onboarded
    (cannot be onboarded)
- AI4EU Experiments Platform
  - will automatically add an orchestrator to the solution
    (if it contains more than one component)
  - That generic orchestrator is publicly available:
    https://github.com/ai4eu/generic-serial-orchestrator

This project has received funding from the European Union’s Horizon 2020
research and innovation programme under grant agreement No 825619

Using dockerhub as a repository

- Building: omit the server name and port:
  - configure "`<username>/<repository>`" for `REMOTE_REPO` in `helper.py`
  - login without specifying a server name:
    `$ docker login`
- Onboarding:
  - Use `hub.docker.com` as `host`
  - Omit the `port` number (leave the field empty)
- Deployment should then work just fine

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Configure Onboarded Models

- Choose “My Models” on the left.
- For each model:
  - Click “Manage My Model”
    ➔ “Publish to Marketplace”
    ➔ “Model Category”
  - Select “Data Transformer” and “Scikit-learn”.
    Because we must choose something for the next step, and this combination is known to work.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825619

Assemble Solution

- Choose “Design Studio” ➔ “Launch” Acu-Compose
- Drag the components to the Canvas and connect matching ports
  - Input ports are circles with white background.
  - Output ports are circles with black background.
  - Symbols indicate matching ports.
- Click “Save” and enter a solution name and version.
- Click “Validate”.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825619

Congratulations!

- Your solution is ready
  - to be deployed,
  - to be automatically executed/orchestrated/run, and
  - to be published or privately shared.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Deployment – Getting solution.zip

- Load the Solution in Design Studio / Acu-Compose
- Click “Deploy”
  ➔ “Deploy to local”
  ➔ “Export To Local”
  ➔ “Download Solution Package”
- Store solution.zip to a new folder
- Unzip the file.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825619

Deployment - Kubernetes

- Create a new namespace for Sudoku
```
$ kubectl create namespace sudoku
```
- Run deployment script:
```
$ python kubernetes-client-script.py -n sudoku
```
- Observe the end of the output:
```
Node IP-address : 192.169.49.2
Orchestrator Port is : 30002
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Deployment - Orchestrator

- Recall the output of the deployment script:

```
Node IP-address : 192.168.49.2
Orchestrator Port is : 30002
```

- Run the orchestrator:

```
$ cd orchestrator_client
$ python orchestrator_client \
  192.168.49.2:30002
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Deployment – Access the GUI

- Get access to the Web-GUI (minikube on Windows/MacOS):

```
$ minikube -n sudoku service \
  --url sudoku-gui1webui
```

  - Navigate to the address given by the command.
- Get access to the Web-GUI (in general):
  - List Node, PODs, and Services

```
$ kubectl -n sudoku get \
  node,service,pod -o wide
```

  - Navigate to

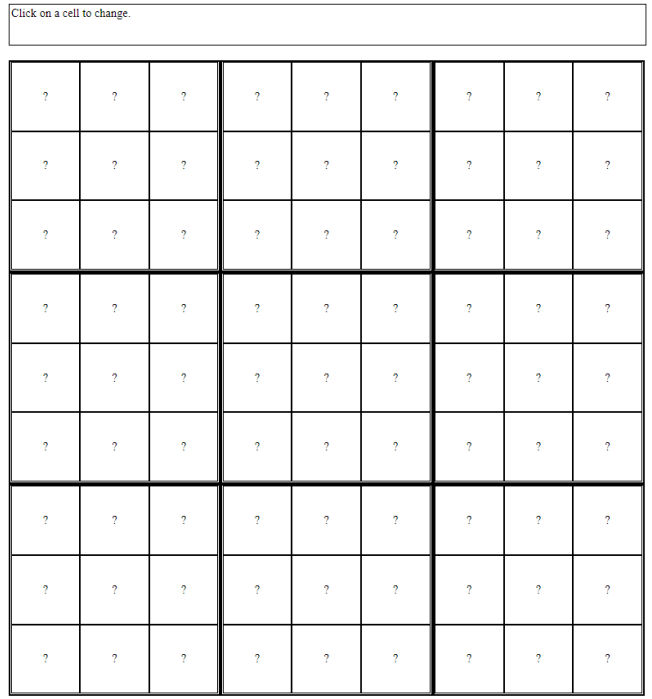
```
http://<node-IP>:<sudoku-gui1webui-port>/
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Test it

- You should see the Sudoku GUI.
- Click on one field and select a number.
- You should see a partial solution of the Sudoku that respects your input.
- For each click, the orchestrator client must be started again.

```
$ while sleep 1; do \
  python orchestrator_client…;\
  done
```



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619

Final Comments

- This is the status of the March 2021 Release.
- The easiest way might be minikube on native Ubuntu LTS, e.g., Ubuntu 20.04 / “Focal Fossa”.
- This demo was created with minikube in WSL2 (Windows Subsystem for Linux Version 2):
  - docker was installed within WSL2 (not Docker Desktop)
  - minikube was installed within WSL2 (not the Windows Installer)
- Contact: peter.schueller@tuwien.ac.at
  contact@peterschueller.com

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825619