AI4EU Acumos Hello World "Sudoku Design Assistant"

Peter Schüller, TU Wien

Version 1, 27.11.2020

Click on a cell to change.

?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?

Click on a grey digit to set it, click on a green digit to reset it, click outside the field to cancel.

?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	1 2 3 4 5 6 7 8 9	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?

Click on a cell to change.
Sudoku has multiple solutions

?8	?	?4	?1	?9	?	?2	?6	?5
?1	?	?6	?5	?8	?2	?9	?	?4
?2	?5	?9	?4	6 6	?	?8	?	?1
?5	?6	?7	?8	?2	?1	?3	?4	?9
?4	?8	?3	?9	?7	?5	?1	?2	?6
?9	?2	?1	?6	?3	?4	?5	?8	?7
?6	?1	?5	?2	?4	?8	?7	?9	?3
?3	?9	?2	?7	?5	?6	?4	?1	?8
?7	?4	?8	?3	?1	?9	?6	?5	?2

Click on a cell to change.
Sudoku has multiple solutions

?2	?1	?6	?3	?4	?8	?5	?7	?9
?3	?8	?9	?1	?7	?5	?6	?4	?2
?4	?7	?5	?9	6	?2	?8	?3	?1
?9	?2	?8	?7	?3	?4	?1	?6	?5
?6	?5	?3	?8	?9	?1	?4	?2	?7
?7	?4	?1	?5	2	?6	?3	?9	?8
?8	?9	?7	?4	?1	?3	?2	?5	?6
?5	?6	?4	?2	?8	?	?	?1	?3
?1	?3	?2	?6	?5	?	?	?8	?4

Click on a cell to change.
Sudoku has no solution

?	7	?	?	?	?	?	?	?
?	6	5	?	?	?	?	?	?
?	9	?	?	?	?	?	?	?
4	?	8	?	?	?	?	?	?
?	?	1	?	?	?	?	?	?
9	?	2 2	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?

Principles

- Lots of Readme
- Easy to build (with docker or with conda)
- Easy to startup, runs in Linux or Windows Subsystem for Linux
- Hopefully a starting point for other protobuf pipelines
  - Runnable completely without Acumos
  - Migration to Acumos will require small changes
- Public Repository https://github.com/peschue/ai4eu-sudoku
- Ugly! – feel free to make Pull Requests

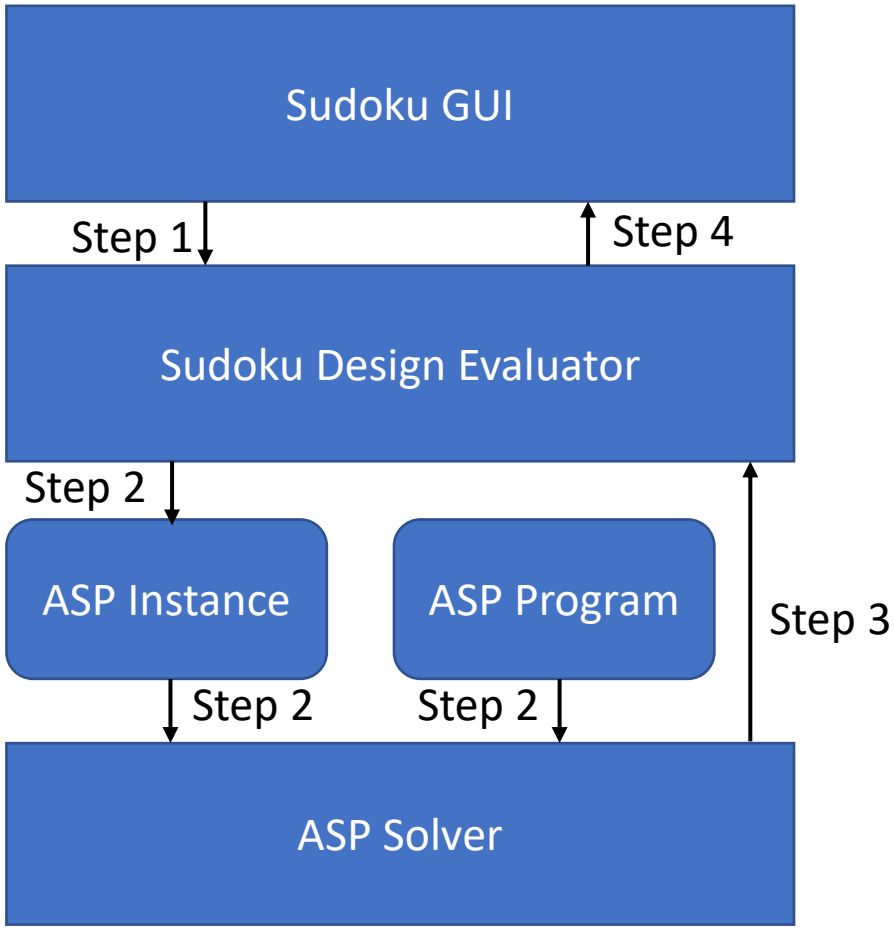
ps@DESKTOP-FO797RD: ~/_ai4eu-sudoku

```
./(base) ps@DESKTOP-FO797RD:~/_ai4eu-sudoku$ ./docker-run-all-detached.sh
+ for component in gui evaluator aspsolver
+ pushd gui
~/_ai4eu-sudoku/gui ~/_ai4eu-sudoku
+ ./docker-run-detached.sh
fab6d45922a4cf43d792c4e27145770f9d60fb8ac43ad4a0c47e2fbb7f390a91
+ popd
~/_ai4eu-sudoku
+ for component in gui evaluator aspsolver
+ pushd evaluator
~/_ai4eu-sudoku/evaluator ~/_ai4eu-sudoku
+ ./docker-run-detached.sh
42b56d91a822863dc18ab2714e50425bf459a03ce2d89a884842eb00485a6df7
+ popd
~/_ai4eu-sudoku
+ for component in gui evaluator aspsolver
+ pushd aspsolver
~/_ai4eu-sudoku/aspsolver ~/_ai4eu-sudoku
+ ./docker-run-detached.sh
f0905f9765165ee8b58b6c5ce83aa5f094ac93760d78eaea95601eee2182314a
+ popd
~/_ai4eu-sudoku
(base) ps@DESKTOP-FO797RD:~/_ai4eu-sudoku$ less README.md
(base) ps@DESKTOP-FO797RD:~/_ai4eu-sudoku$ cd orchestrator/
(base) ps@DESKTOP-FO797RD:~/_ai4eu-sudoku/orchestrator$ conda activate ai4eusudoku
(ai4eusudoku) ps@DESKTOP-FO797RD:~/_ai4eu-sudoku/orchestrator$ python3 orchestrator.py
INFO:root:importing generated protobuf modules
INFO:root:calling SudokuDesignEvaluationRequestDataBroker.requestSudokuEvaluation() with empty dummy
INFO:root:calling SudokuDesignEvaluationProblemEncoder.evaluateSudokuDesign() with guijob
INFO:root:calling OneshotSolver.solve() with parameters number_of_answers: 2
```

How to transform your Application to Acumos

1. Identify Computational Blocks and Data Flow
2. Identify Generic Blocks vs Application-specific Blocks
3. Define Datatypes in protobuf
4. Define Services in protobuf
5. Implement all blocks and test them independently
6. Implement the orchestrator (only passes around protobuf)
   ➔Now your application runs!
7. Put everything into docker (usually quite easy)

1. Identify Computational Blocks and Data Flow

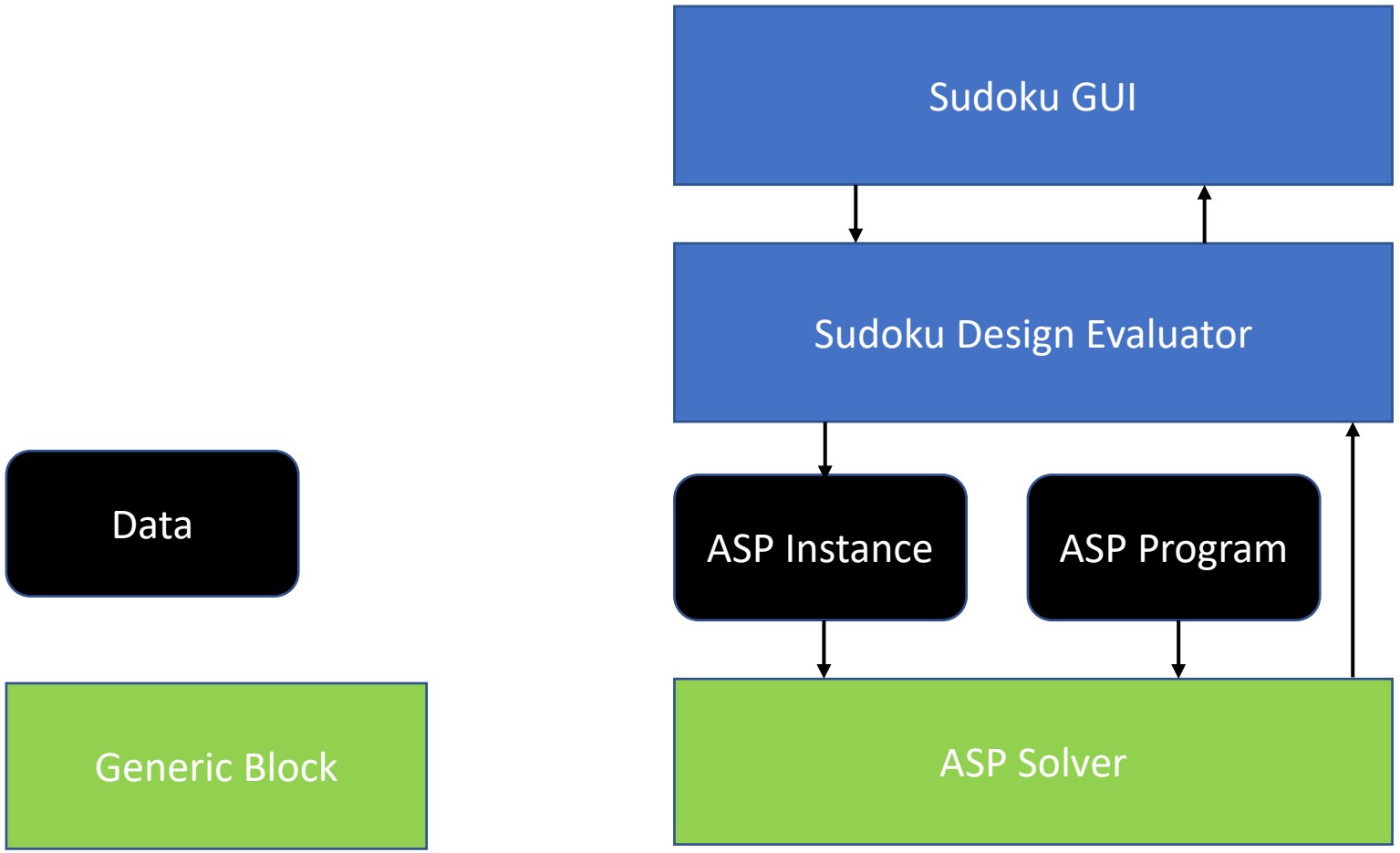


Pipeline =
Step 1 -> Step 2 -> Step 3 -> Step 4

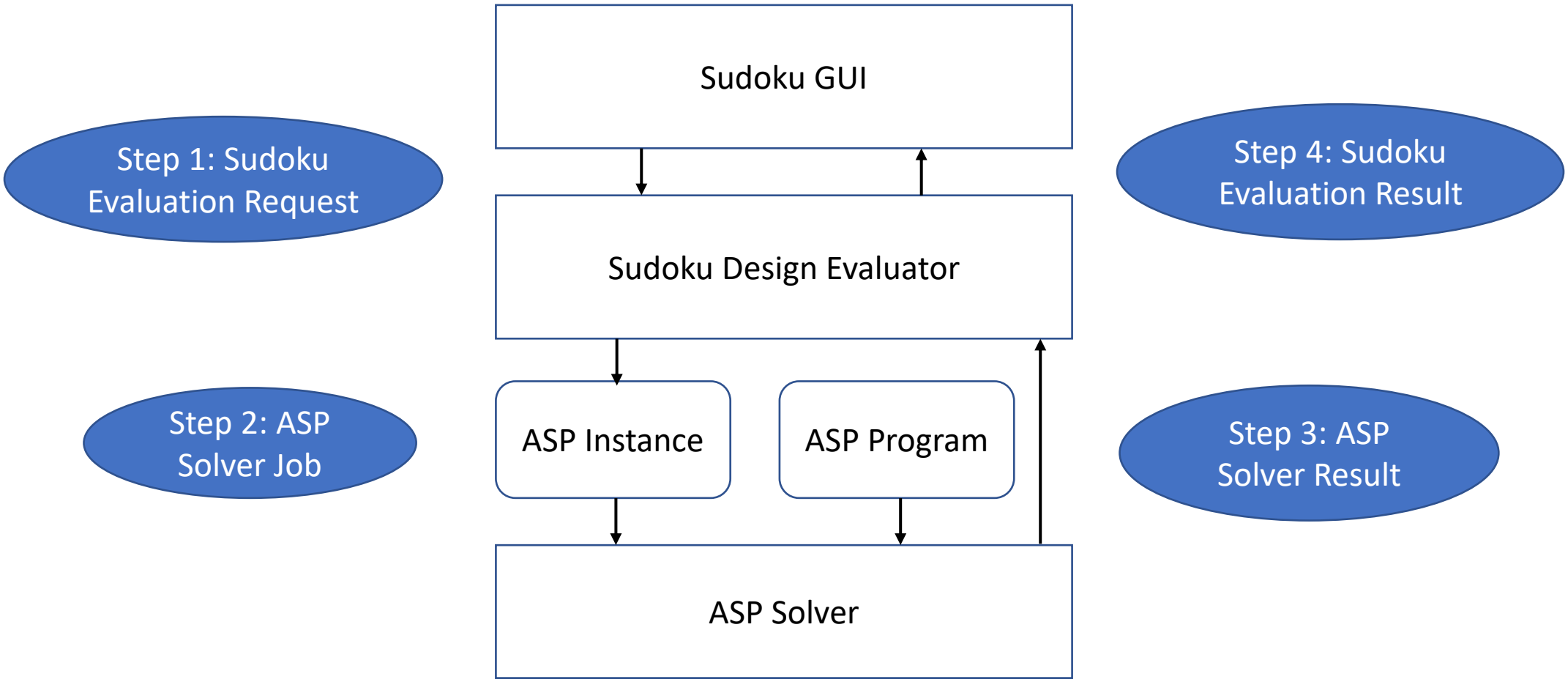
can run multiple times

triggered by Sudoku GUI

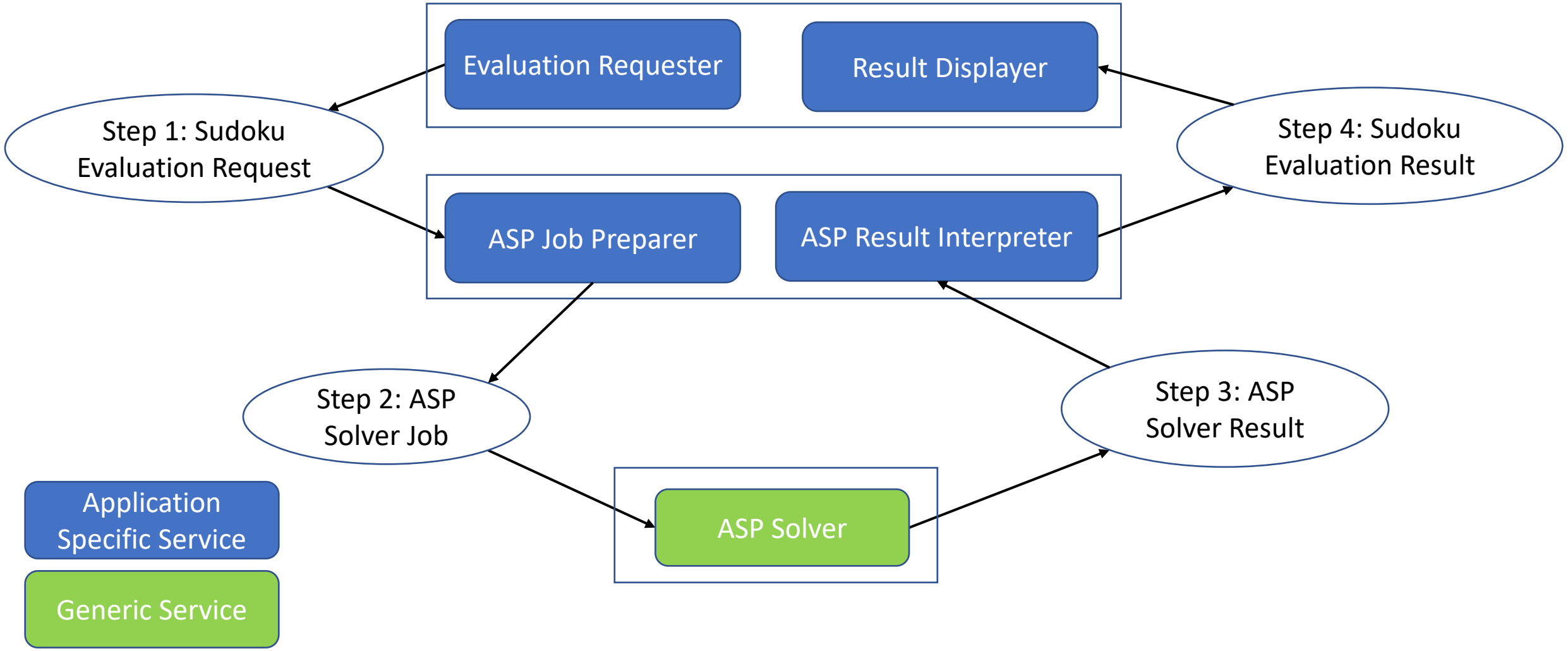
2. Identify Generic vs Application-specific Blocks



3. Define Datatypes in protobuf



4. Define Services in protobuf



Further Steps

5. Implement all blocks and test them independently

➔That also helps to improve the quality of the software

6. Implement the orchestrator

➔This one decides which service runs how often

➔This one should never create protobuf messages, only pass them on

7. Put everything into docker (usually quite easy)

➔ That also helps others to install/run your software (modules)

Scenario with variable # of calls