

# C4 Model in AsciiDoctor

## Purpose

The purpose of this project is to show how AsciiDoc can be used to generate C4 diagrams from PlantUML, both as html and pdf.

## C4 Example

The following example is based on the [System Context Diagram](#) example at the C4 Model home page. Consequently, this C4 diagram:

# System Context diagram for Internet Banking System

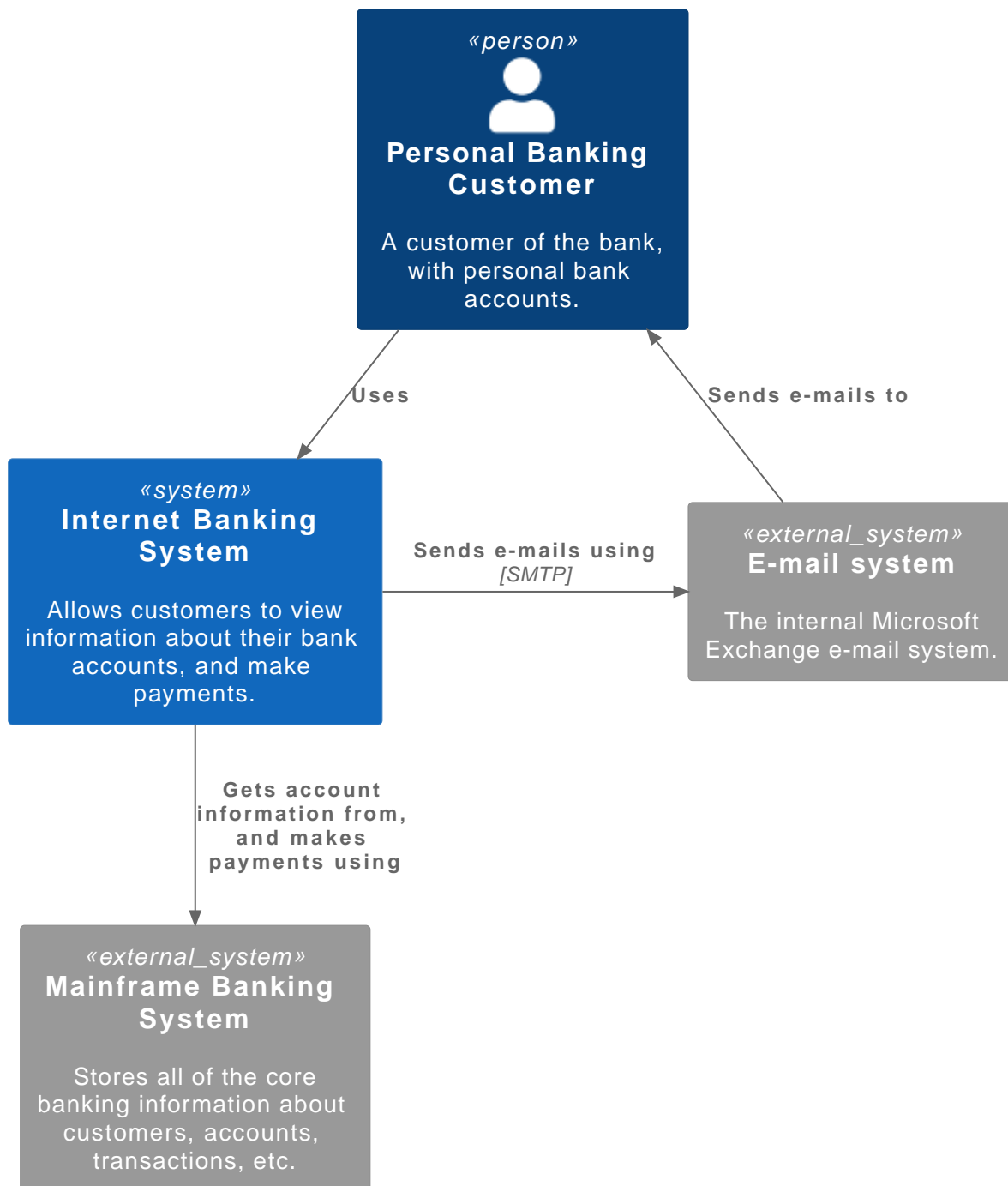


Figure 1. C4 Example

is generated from this PlantUML file:

```
1 @startuml
2 !include <C4/C4_Container>
3 title System Context diagram for Internet Banking System
4
5 Person(customer, "Personal Banking Customer", "A customer of the bank, with
  personal bank accounts.")
6 System(banking_system, "Internet Banking System", "Allows customers to view
  information about their bank accounts, and make payments.")
7
8 System_Ext(mail_system, "E-mail system", "The internal Microsoft Exchange e-mail
  system.")
9 System_Ext(mainframe, "Mainframe Banking System", "Stores all of the core banking
  information about customers, accounts, transactions, etc.")
10
11 Rel(customer, banking_system, "Uses")
12 Rel_Back(customer, mail_system, "Sends e-mails to")
13
14 Rel_Neighbor(banking_system, mail_system, "Sends e-mails using", "SMTP")
15 Rel(banking_system, mainframe, "Gets account information from, and makes payments
  using")
16 @enduml
```

## Build Local

### Prerequisites

- You need to have Docker installed and running.
- Download the [asciidoctor/docker-asciidoctor](#) image:

```
docker pull asciidoctor/docker-asciidoctor
```

### Render html

This project can be rendered as html using this bash command:

*generate html*

```
1 docker run --rm \
2   --volume "$(pwd)":/documents/ \
3   asciidoctor/docker-asciidoctor \
4   asciidoctor \
5   --require asciidoctor-diagram \
6   --destination-dir target \
7   docs/index.adoc
```

## Explanation

Line by line explanation of the command above:

1. `docker run` starts a Docker container. The `--rm` option removes the container after the command has completed.
2. `--volume` mount a volume to the container, i.e. the current directory (as returned by `$(pwd)` command) is mounted to the `/documents/` directory in the running Docker container. The `/documents/` directory is the default working director of the `asciidoctor/docker-asciidoctor` Docker image.
3. `asciidoctor/docker-asciidoctor` the name of the Docker image that is used to create the Docker container.
4. `asciidoctor` the AsciiDoctor command to render html.
5. `--require asciidoctor-diagram` AsciiDoctor argument that requires the `asciidoctor-diagram` to be used (necessary for PlantUML rendering).
6. `--destination-dir target` asciidoctor argument that defines the destination directory of the output files in the Docker container. In this case the `target` directory.
7. `docs/index.adoc` the name of the AsciiDoc source file to generate the file from. Only the root document needs to be included since the other files are linked from it.

## Render pdf

Alternatively, this project can be rendered as a pdf:

*generate pdf*

```
1 docker run --rm \
2   --volume "$(pwd)":/documents/ \
3   asciidoctor/docker-asciidoctor \
4   asciidoctor-pdf \
5   --require asciidoctor-diagram \
6   --destination-dir target \
7   docs/index.adoc
```

## Explanation

As can be seen, rendering a pdf is very similar to rendering html. The only difference is the asciidoctor command on line 4 that has been changed to `asciidoctor-pdf`.

# Build GitHub Actions

Build using GitHub Actions and publish to GitHub Pages:

```
1 name: Build and deploy
2
3 on:
4   push:
5     branches:
6       - main
7
8 jobs:
9   build:
10    runs-on: ubuntu-20.04
11
12    container:
13      image: asciidoctor/docker-asciidoctor:1.18
14      volumes:
15        # /documents/ is the default working directory of the docker-asciidoctor
        container
16        - ${GITHUB_WORKSPACE}:/documents/
17
18    env:
19      BUILD_PATH: target
20      DEPLOY_PATH: _site
21
22    steps:
23      - name: Checkout
24        uses: actions/checkout@v3
25
26      - name: Generate html
27        run: >
28          asciidoctor
29          --require asciidoctor-diagram
30          --destination-dir ${BUILD_PATH}
31          docs/index.adoc
32
33      - name: Generate pdf
34        run: >
35          asciidoctor-pdf
36          --require asciidoctor-diagram
37          --destination-dir ${BUILD_PATH}
38          docs/index.adoc
39
40      - name: Copy artifacts
41        run: >
42          apk add rsync &&
43          rsync -av --exclude="*" ${BUILD_PATH}/ ${DEPLOY_PATH}
44
45      - name: Deploy
46        uses: JamesIves/github-pages-deploy-action@v4.4.0
47        with:
48          folder: ${DEPLOY_PATH}
```

# Explanation

Line s	Purpose
3-6	Trigger the action every time a <a href="#">commit is pushed</a> to the <a href="#">main</a> branch at GitHub.
10	Use Ubuntu as the GitHub runner environment.
12-16	Use the <a href="#">asciidoctor/docker-asciidoctor</a> image so that the AsciiDoctor tools are available during the build.
18-20	Define paths used during the build as environment variables.
23-24	The <a href="#">checkout action</a> used by the GitHub workflow to checkout the source code.
26-31	Use the <a href="#">asciidoctor</a> command to generate the docs as html in the <a href="#">BUILD_PATH</a> directory.
33-38	Use the <a href="#">asciidoctor-pdf</a> command to generate the docs as pdf in the <a href="#">BUILD_PATH</a> directory.
40-43	Install <a href="#">rsync</a> and copy the generated files to the <a href="#">DEPLOY_PATH</a> directory.
45-49	Use <a href="#">GitHub Pages Deploy Action</a> to deploy the files in the <a href="#">DEPLOY_PATH</a> to GitHub Pages.

# References

Link	Comment
<a href="https://github.com/matsev/c4-model-adoc">https://github.com/matsev/c4-model-adoc</a>	This project at GitHub
<a href="https://github.com/asciidoctor/docker-asciidoctor#readme">https://github.com/asciidoctor/docker-asciidoctor#readme</a>	AsciiDoctor Docker at GitHub
<a href="https://plantuml.com">https://plantuml.com</a>	PlantUML
<a href="https://github.com/plantuml-stdlib/C4-PlantUML#readme">https://github.com/plantuml-stdlib/C4-PlantUML#readme</a>	C4 PlantUml
<a href="https://docs.github.com/en/actions">https://docs.github.com/en/actions</a>	GitHub Actions
<a href="https://pages.github.com">https://pages.github.com</a>	GitHub Pages