universität innsbruck

Reimplementing CoAP for C# with the Task-based Asynchronous Pattern

Is it worth to await?

Philip Wille

Introduction

- Synchronous and asynchronous execution

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
  - Asynchronous:

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
    - **Waiting for completion** of method before continuing with program flow.
  - Asynchronous:

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
    - **Waiting for completion** of method before continuing with program flow.
  - Asynchronous:
    - Can **perform other tasks** while the execution is running.

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
    - **Waiting for completion** of method before continuing with program flow.
    - **Busy waiting** $\rightarrow$ Thread is marked as **blocked**.
  - Asynchronous:
    - Can **perform other tasks** while the execution is running.

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
    - **Waiting for completion** of method before continuing with program flow.
    - **Busy waiting** $\rightarrow$ Thread is marked as **blocked**.
  - Asynchronous:
    - Can **perform other tasks** while the execution is running.
    - **No busy waiting** $\rightarrow$ Thread is **free** for other tasks.

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
    - **Waiting for completion** of method before continuing with program flow.
    - **Busy waiting** $\rightarrow$ Thread is marked as **blocked**.
  - Asynchronous:
    - Can **perform other tasks** while the execution is running.
    - **No busy waiting** $\rightarrow$ Thread is **free** for other tasks.
- **T**ask-based **A**synchronous **P**attern (TAP)

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
    - **Waiting for completion** of method before continuing with program flow.
    - **Busy waiting** $\rightarrow$ Thread is marked as **blocked**.
  - Asynchronous:
    - Can **perform other tasks** while the execution is running.
    - **No busy waiting** $\rightarrow$ Thread is **free** for other tasks.
- **T**ask-based **A**synchronous **P**attern (TAP)
  - Developed by Microsoft.

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
    - **Waiting for completion** of method before continuing with program flow.
    - **Busy waiting** $\rightarrow$ Thread is marked as **blocked**.
  - Asynchronous:
    - Can **perform other tasks** while the execution is running.
    - **No busy waiting** $\rightarrow$ Thread is **free** for other tasks.
- **T**ask-based **A**synchronous **P**attern (TAP)
  - Developed by Microsoft.
  - Simple usage.

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
    - **Waiting for completion** of method before continuing with program flow.
    - **Busy waiting** $\rightarrow$ Thread is marked as **blocked**.
  - Asynchronous:
    - Can **perform other tasks** while the execution is running.
    - **No busy waiting** $\rightarrow$ Thread is **free** for other tasks.
- **T**ask-based **A**synchronous **P**attern (TAP)
  - Developed by Microsoft.
  - Simple usage.
  - Built-in in C#.

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
    - **Waiting for completion** of method before continuing with program flow.
    - **Busy waiting** → Thread is marked as **blocked**.
  - Asynchronous:
    - Can **perform other tasks** while the execution is running.
    - **No busy waiting** → Thread is **free** for other tasks.
- **T**ask-based **A**synchronous **P**attern (TAP)
  - Developed by Microsoft.
  - Simple usage.
  - Built-in in C#.
- **Co**nstrained **A**pplication **P**rotocol (CoAP)

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
    - **Waiting for completion** of method before continuing with program flow.
    - **Busy waiting** → Thread is marked as **blocked**.
  - Asynchronous:
    - Can **perform other tasks** while the execution is running.
    - **No busy waiting** → Thread is **free** for other tasks.
- **T**ask-based **A**synchronous **P**attern (TAP)
  - Developed by Microsoft.
  - Simple usage.
  - Built-in in C#.
- **Co**nstrained **A**pplication **P**rotocol (CoAP)
  - Subset of **H**yper**t**ext **T**ransport **P**rotocol (HTTP).

Introduction

- Synchronous and asynchronous execution
  - Synchronous:
    - **Waiting for completion** of method before continuing with program flow.
    - **Busy waiting** → Thread is marked as **blocked**.
  - Asynchronous:
    - Can **perform other tasks** while the execution is running.
    - **No busy waiting** → Thread is **free** for other tasks.
- **T**ask-based **A**synchronous **P**attern (TAP)
  - Developed by Microsoft.
  - Simple usage.
  - Built-in in C#.
- **Co**nstrained **A**pplication **P**rotocol (CoAP)
  - Subset of **H**yper**t**ext **T**ransport **P**rotocol (HTTP).
  - Specialized for **I**nternet **o**f **T**hings (IoT) and **M**achine-**t**o-**M**achine (M2M) devices.

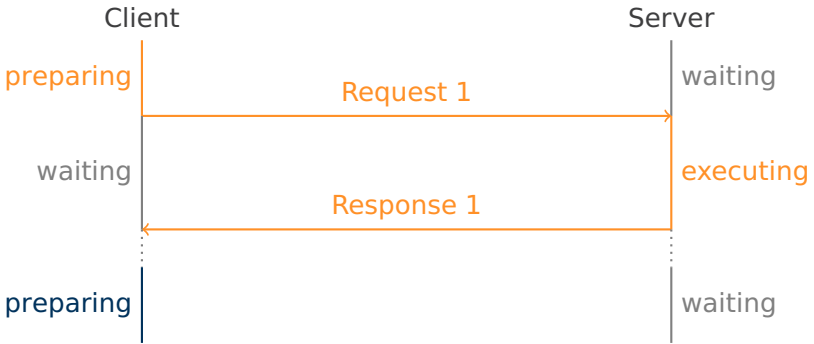
Synchronous execution

Synchronous execution

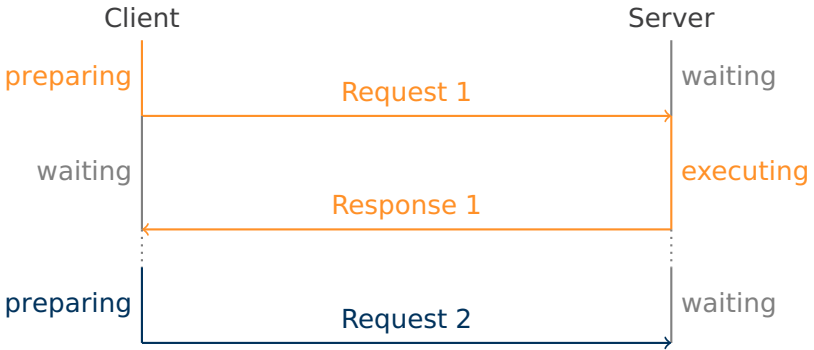
Synchronous execution

Synchronous execution

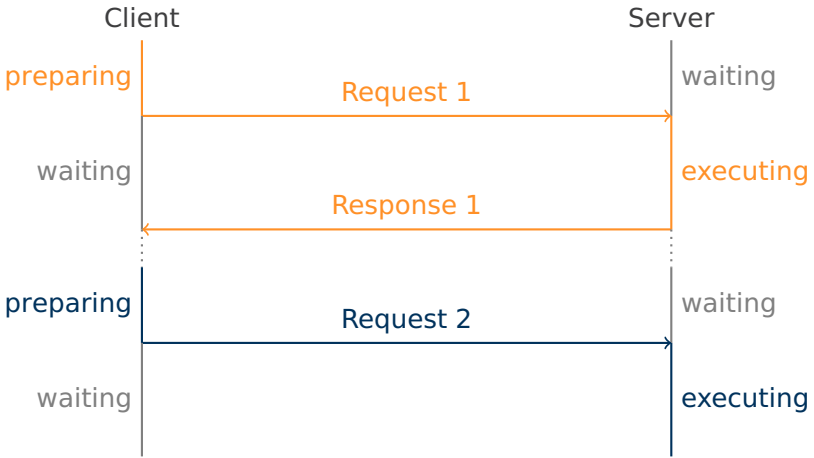
Synchronous execution



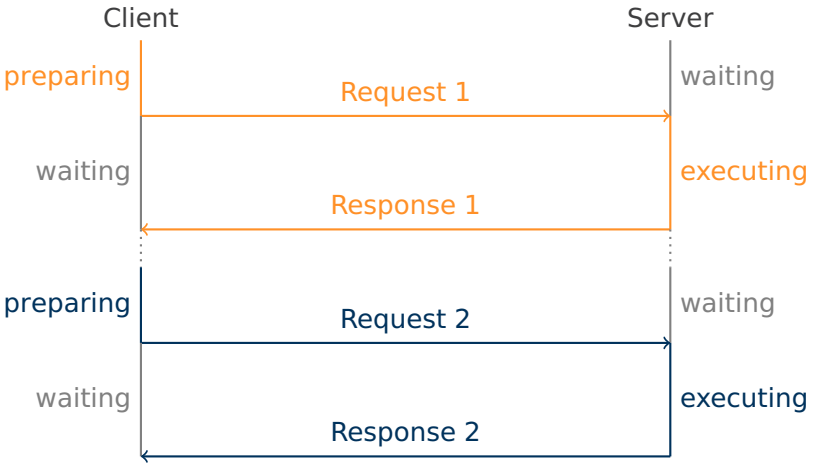
Synchronous execution



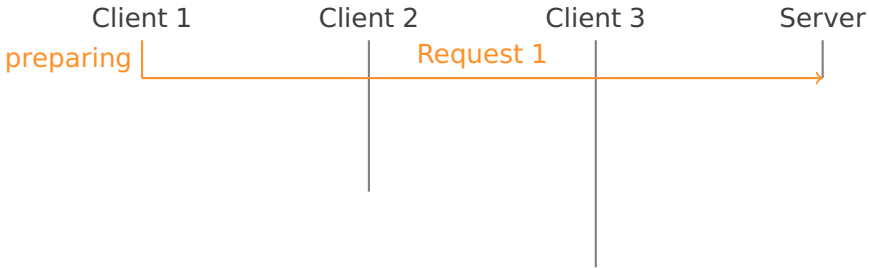
Synchronous execution



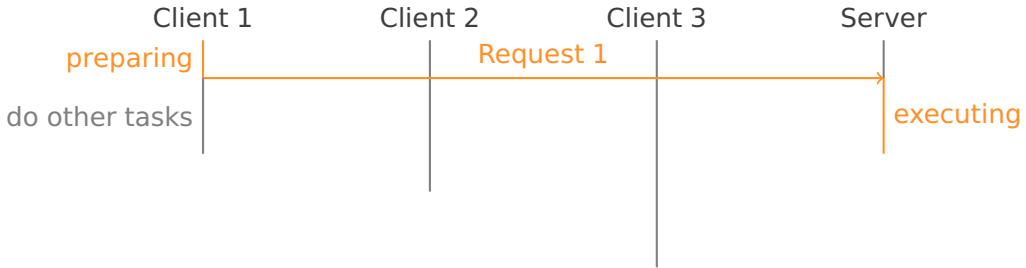
Synchronous execution



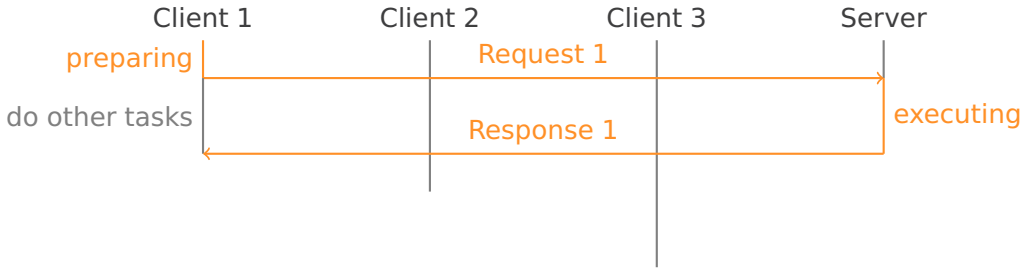
Asynchronous execution



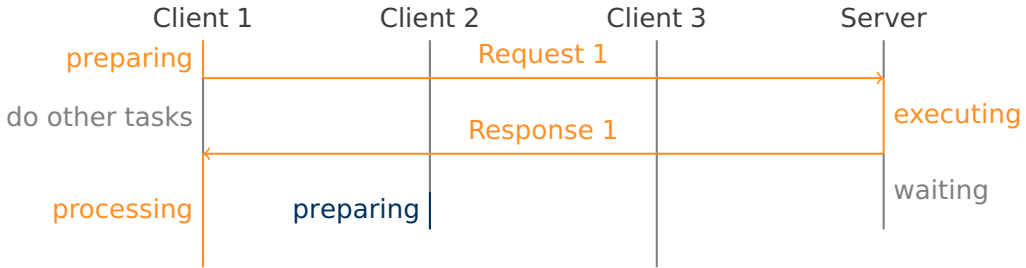
Asynchronous execution



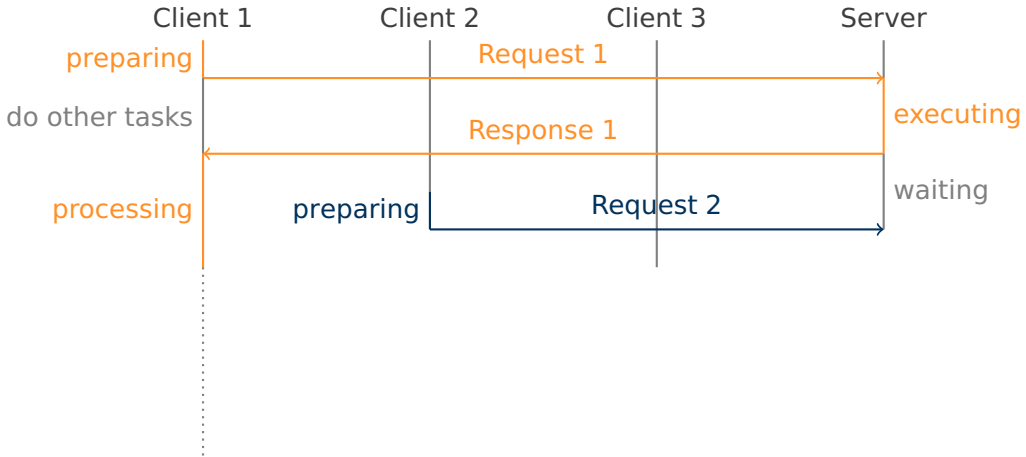
Asynchronous execution



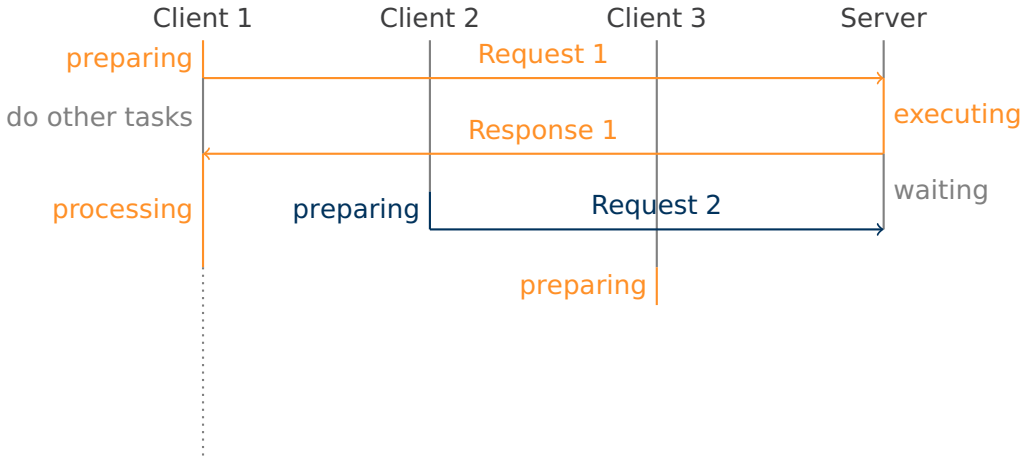
Asynchronous execution



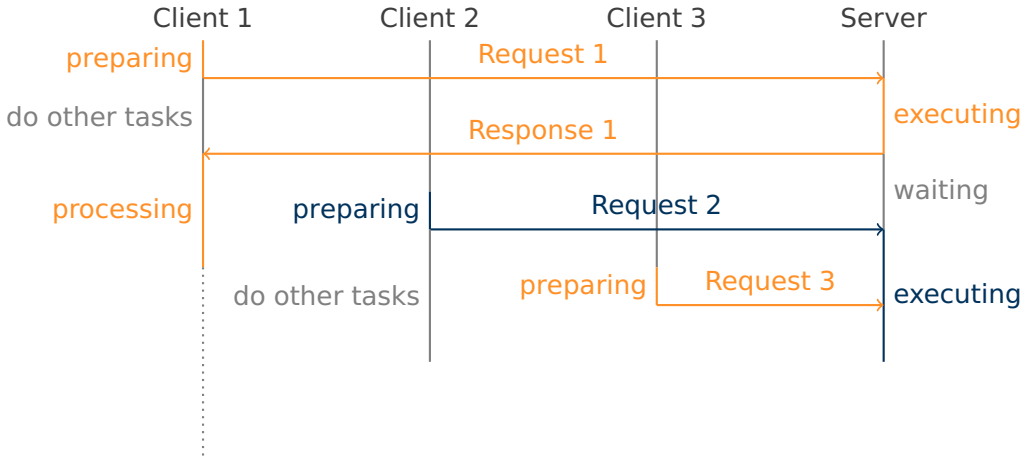
Asynchronous execution



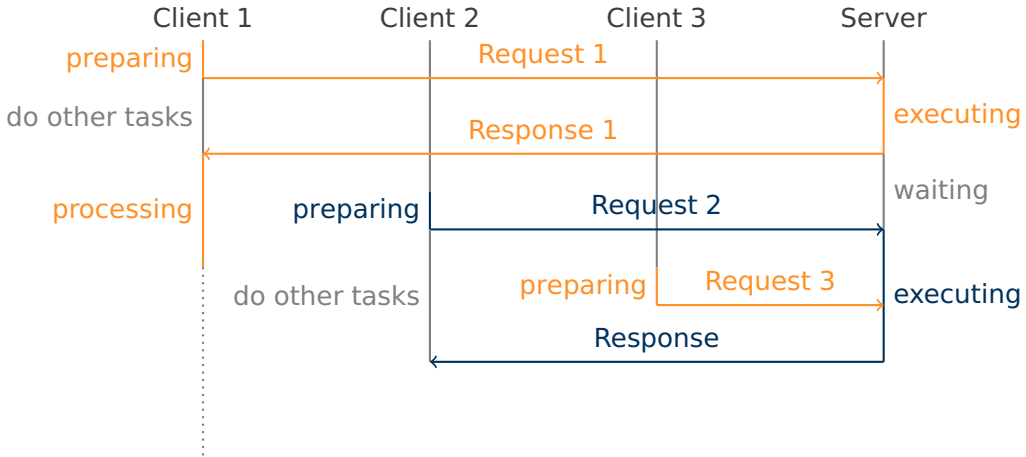
Asynchronous execution



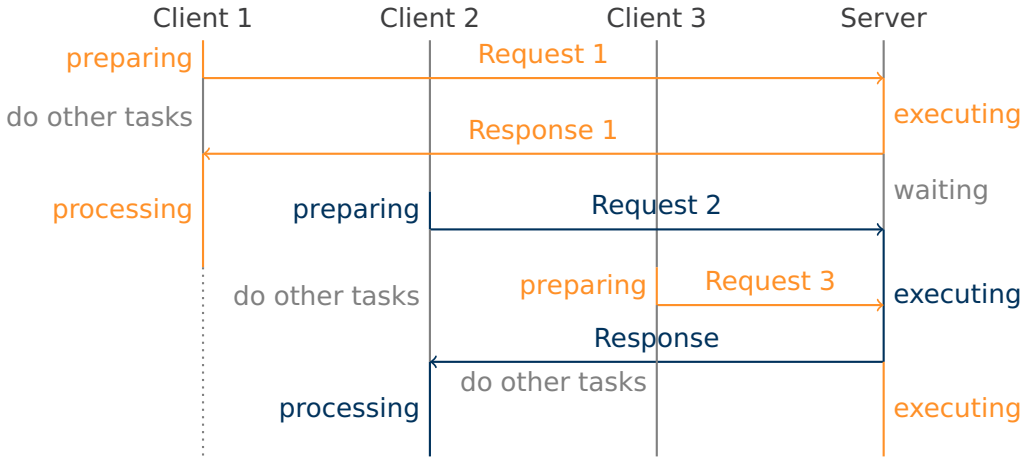
Asynchronous execution



Asynchronous execution



Asynchronous execution



Asynchronous execution