universität innsbruck

Increasing throughput of server applications by using asynchronous techniques

A case study on CoAP.NET

Philip Wille
Supervisors: Michael Felderer, Andreas Danek

1. **Programming paradigms**
2. **Synchronous and asynchronous server**
3. **Task-based Asynchronous Pattern (TAP)**
4. **Constrained Application Protocol (CoAP)**
5. **Bachelor thesis**

Programming paradigms

- Synchronous

Programming paradigms

- Synchronous

- Asynchronous

Programming paradigms

- Synchronous
  - Must **stop** program flow.
- Asynchronous

Programming paradigms

- Synchronous
  - Must **stop** program flow.

- Asynchronous
  - Can **go further** in program flow.

Programming paradigms

- Synchronous
  - Must **stop** program flow.
  - **Checks** periodically.
- Asynchronous
  - Can **go further** in program flow.

Programming paradigms

- Synchronous
  - Must **stop** program flow.
  - **Checks** periodically.
- Asynchronous
  - Can **go further** in program flow.
  - Will be **notified** by event.

Programming paradigms

- Synchronous
  - Must **stop** program flow.
  - **Checks** periodically.
  - Marked as **Blocked** (Linux) or **Waiting** (Windows).
- Asynchronous
  - Can **go further** in program flow.
  - Will be **notified** by event.

Programming paradigms

- Synchronous
  - Must **stop** program flow.
  - **Checks** periodically.
  - Marked as **Blocked** (Linux) or **Waiting** (Windows).
- Asynchronous
  - Can **go further** in program flow.
  - Will be **notified** by event.
  - **Free** for other tasks.

Ordering a book

- Synchronous way

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are sitting on the couch and waiting for the book.

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are sitting on the couch and waiting for the book.
  3. Postman is knocking on your door and giving you the book.

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are sitting on the couch and waiting for the book.
  3. Postman is knocking on your door and giving you the book.
  4. You start reading it.

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are sitting on the couch and waiting for the book.
  3. Postman is knocking on your door and giving you the book.
  4. You start reading it.
  5. You have finished it.

Ordering a book

- Synchronous way
    1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
    2. You are sitting on the couch and waiting for the book.
    3. Postman is knocking on your door and giving you the book.
    4. You start reading it.
    5. You have finished it.
    6. You are going outside, meeting friends, go hiking and so on.

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are sitting on the couch and waiting for the book.
  3. Postman is knocking on your door and giving you the book.
  4. You start reading it.
  5. You have finished it.
  6. You are going outside, meeting friends, go hiking and so on.
- Asynchronous way

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are sitting on the couch and waiting for the book.
  3. Postman is knocking on your door and giving you the book.
  4. You start reading it.
  5. You have finished it.
  6. You are going outside, meeting friends, go hiking and so on.
- Asynchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are sitting on the couch and waiting for the book.
  3. Postman is knocking on your door and giving you the book.
  4. You start reading it.
  5. You have finished it.
  6. You are going outside, meeting friends, go hiking and so on.
- Asynchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are going outside, meeting friends, go hiking and so on.

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are sitting on the couch and waiting for the book.
  3. Postman is knocking on your door and giving you the book.
  4. You start reading it.
  5. You have finished it.
  6. You are going outside, meeting friends, go hiking and so on.
- Asynchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are going outside, meeting friends, go hiking and so on.
  3. In the meanwhile the postman delivers the book to your home.

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are sitting on the couch and waiting for the book.
  3. Postman is knocking on your door and giving you the book.
  4. You start reading it.
  5. You have finished it.
  6. You are going outside, meeting friends, go hiking and so on.
- Asynchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are going outside, meeting friends, go hiking and so on.
  3. In the meanwhile the postman delivers the book to your home.
  4. You are coming home and picking up the book.

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are sitting on the couch and waiting for the book.
  3. Postman is knocking on your door and giving you the book.
  4. You start reading it.
  5. You have finished it.
  6. You are going outside, meeting friends, go hiking and so on.
- Asynchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are going outside, meeting friends, go hiking and so on.
  3. In the meanwhile the postman delivers the book to your home.
  4. You are coming home and picking up the book.
  5. You start reading it.

Ordering a book

- Synchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are sitting on the couch and waiting for the book.
  3. Postman is knocking on your door and giving you the book.
  4. You start reading it.
  5. You have finished it.
  6. You are going outside, meeting friends, go hiking and so on.
- Asynchronous way
  1. You are ordering *Clean Code (Robert C. Martin)* from amazon.com
  2. You are going outside, meeting friends, go hiking and so on.
  3. In the meanwhile the postman delivers the book to your home.
  4. You are coming home and picking up the book.
  5. You start reading it.
  6. You have finished it.

Synchronous server

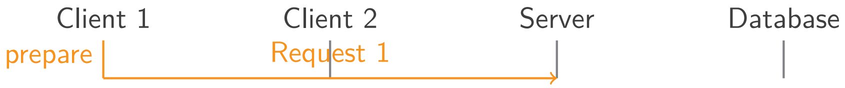

Figure: Sequence diagram of synchronous server

Synchronous server

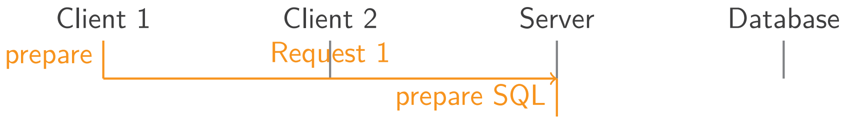

Figure: Sequence diagram of synchronous server

Synchronous server

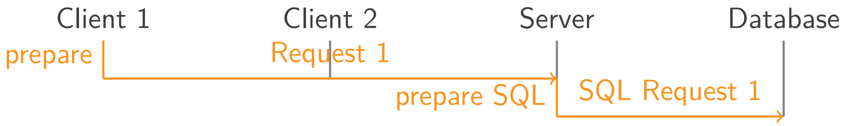

Figure: Sequence diagram of synchronous server

Synchronous server

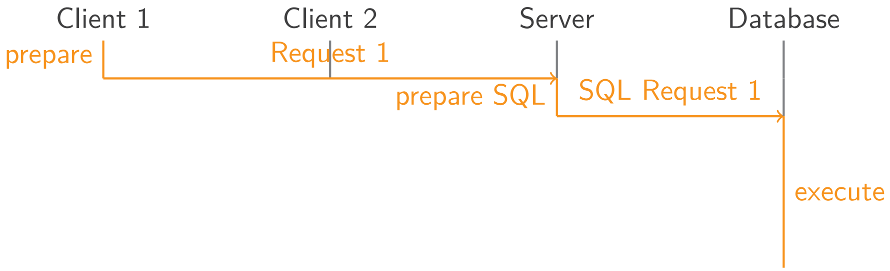

Figure: Sequence diagram of synchronous server

Synchronous server

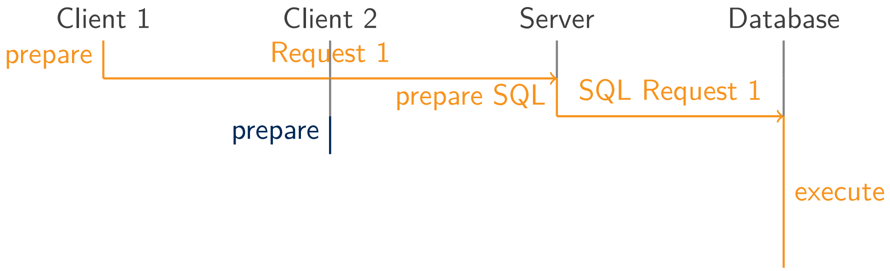

Figure: Sequence diagram of synchronous server

Synchronous server

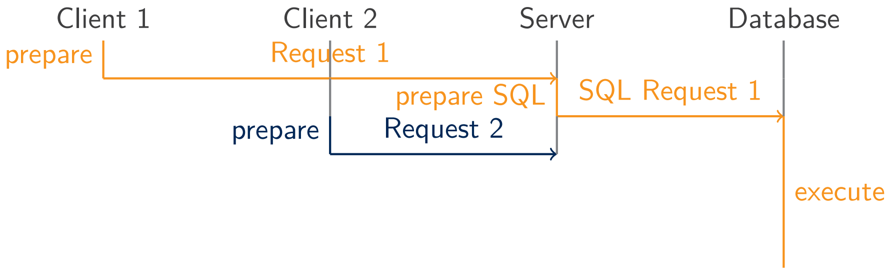

Figure: Sequence diagram of synchronous server

Synchronous server

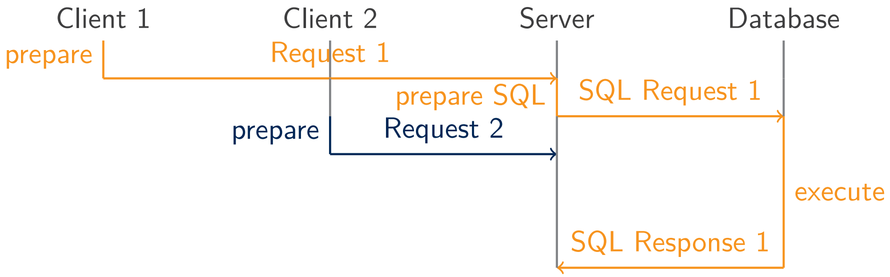

Figure: Sequence diagram of synchronous server

Synchronous server

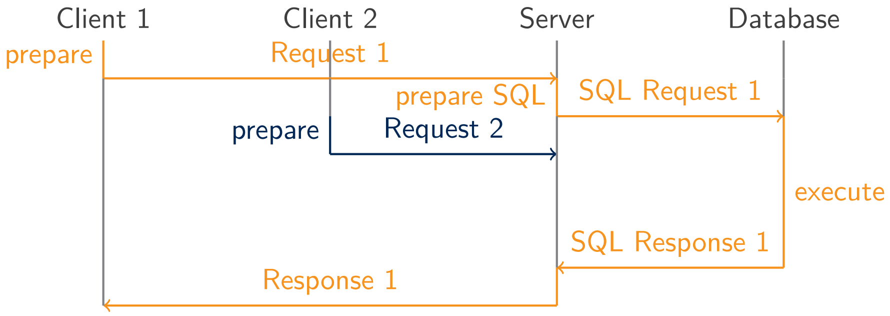

Figure: Sequence diagram of synchronous server

Synchronous server

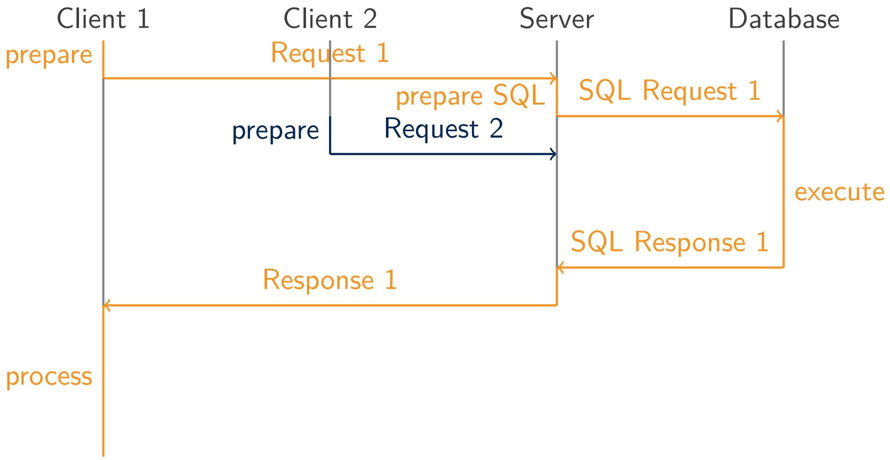

Figure: Sequence diagram of synchronous server

Synchronous server

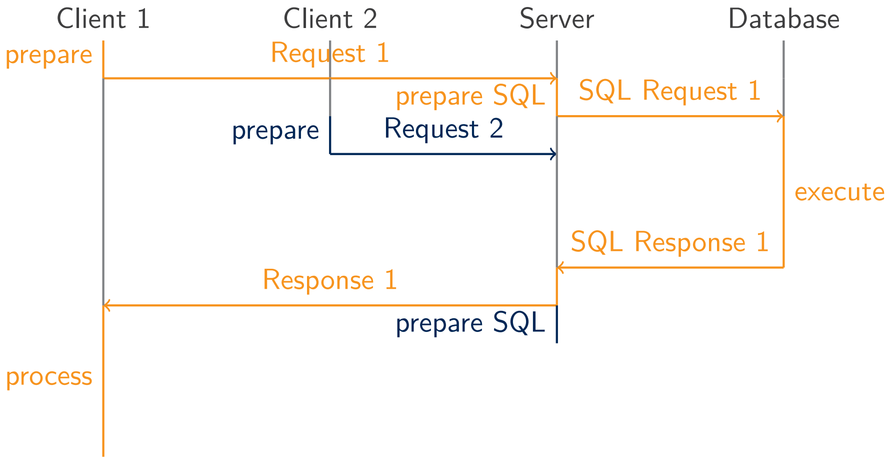

Figure: Sequence diagram of synchronous server

Synchronous server

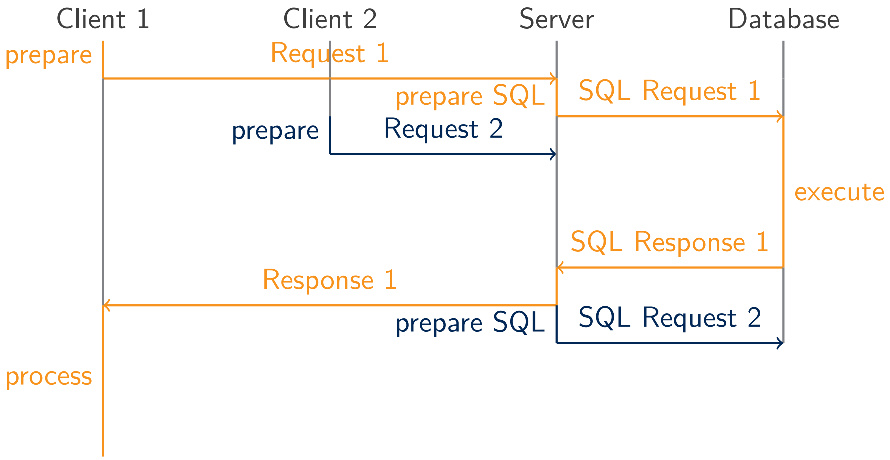

Figure: Sequence diagram of synchronous server

Synchronous server

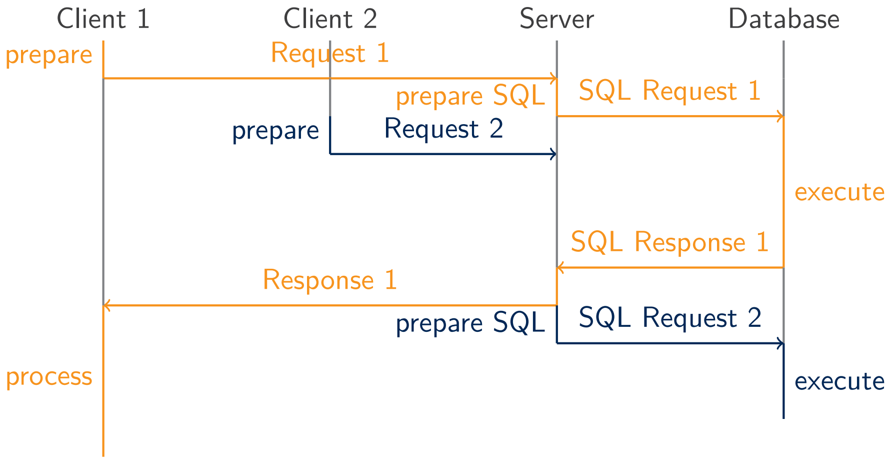

Figure: Sequence diagram of synchronous server

Synchronous server

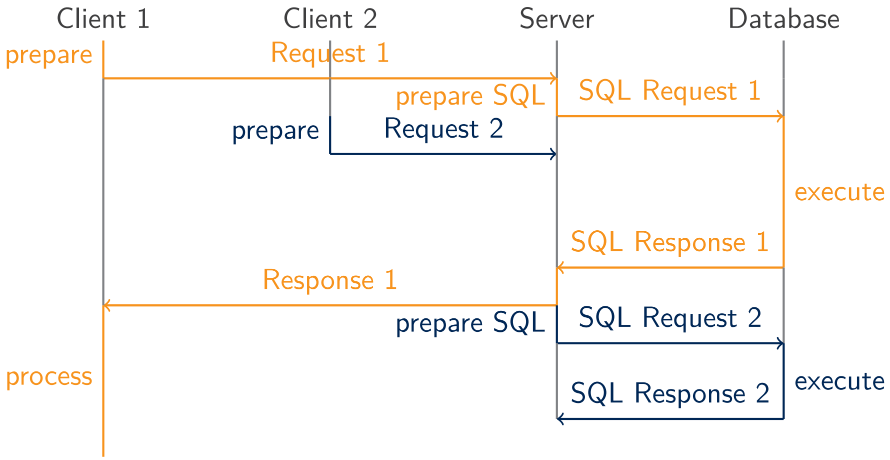

Figure: Sequence diagram of synchronous server

Synchronous server

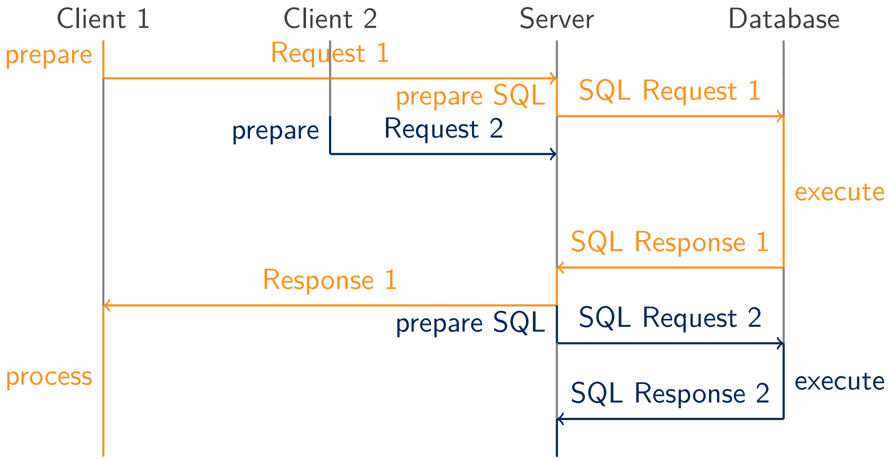

Figure: Sequence diagram of synchronous server

Synchronous server

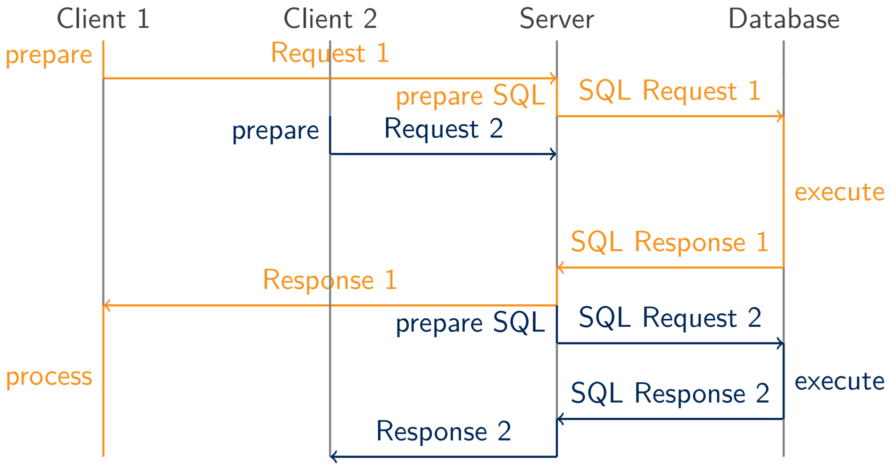

Figure: Sequence diagram of synchronous server

Asynchronous server

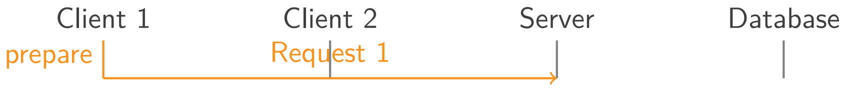

Figure: Sequence diagram of asynchronous server

Asynchronous server

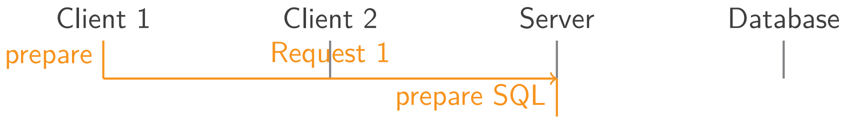

Figure: Sequence diagram of asynchronous server

Asynchronous server

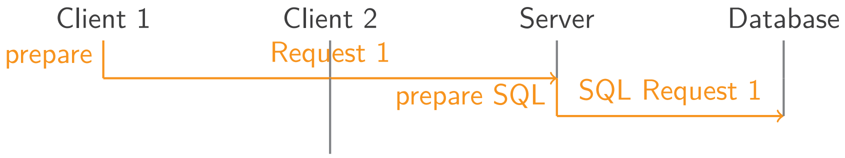

Figure: Sequence diagram of asynchronous server

Asynchronous server

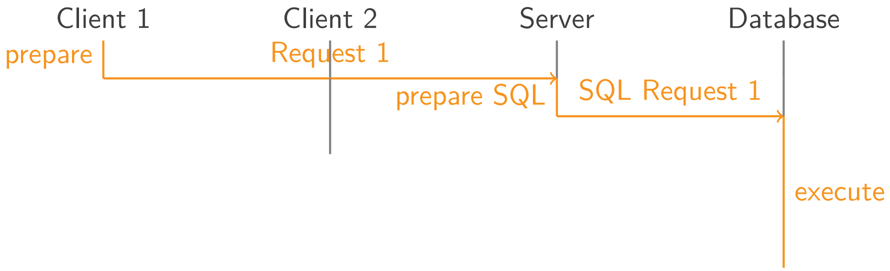

Figure: Sequence diagram of asynchronous server

Asynchronous server

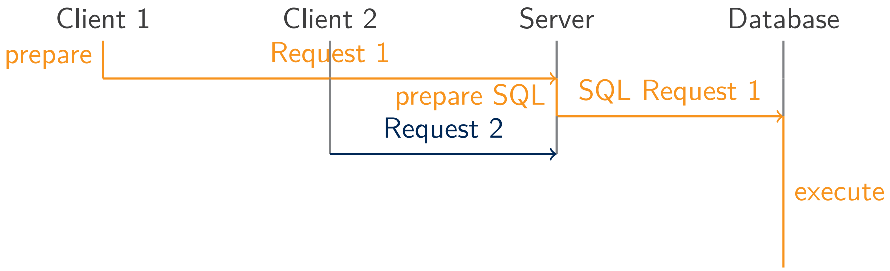

Figure: Sequence diagram of asynchronous server

Asynchronous server

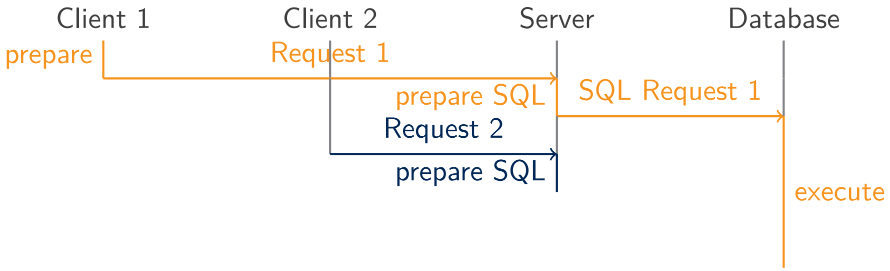

Figure: Sequence diagram of asynchronous server

Asynchronous server

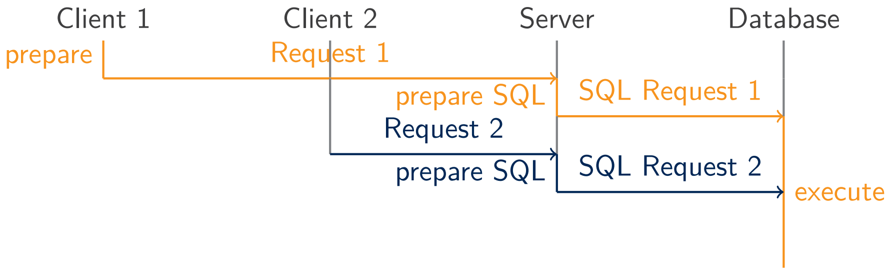

Figure: Sequence diagram of asynchronous server

Asynchronous server

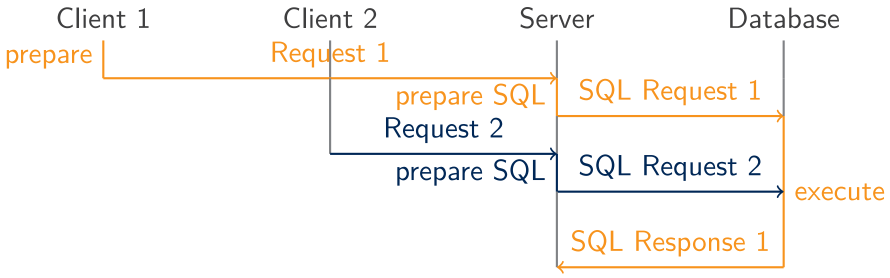

Figure: Sequence diagram of asynchronous server

Asynchronous server

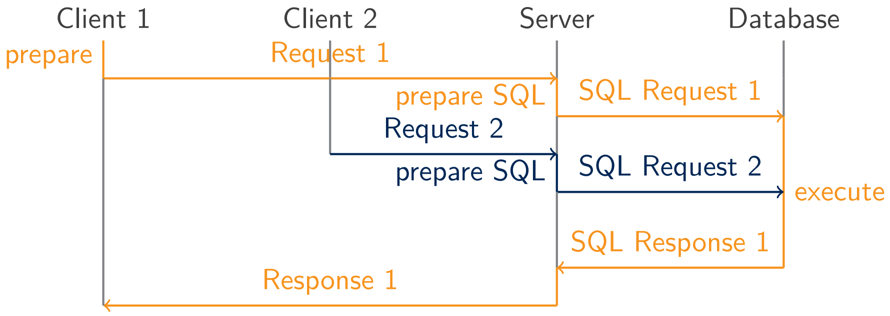

Figure: Sequence diagram of asynchronous server

Asynchronous server

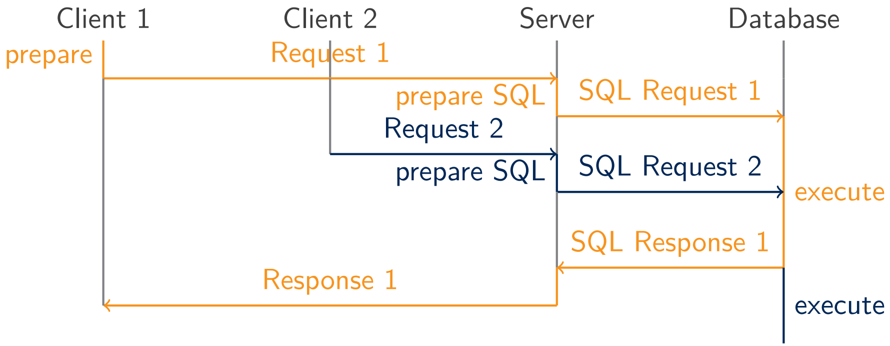

Figure: Sequence diagram of asynchronous server

Asynchronous server

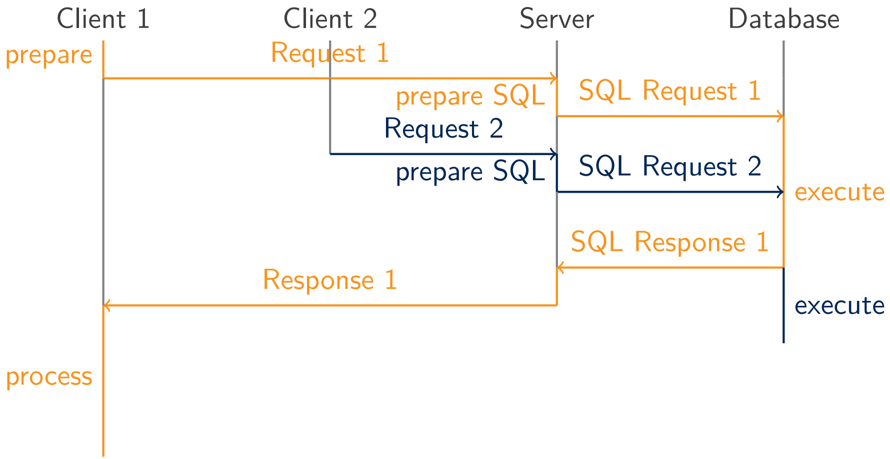

Figure: Sequence diagram of asynchronous server

Asynchronous server

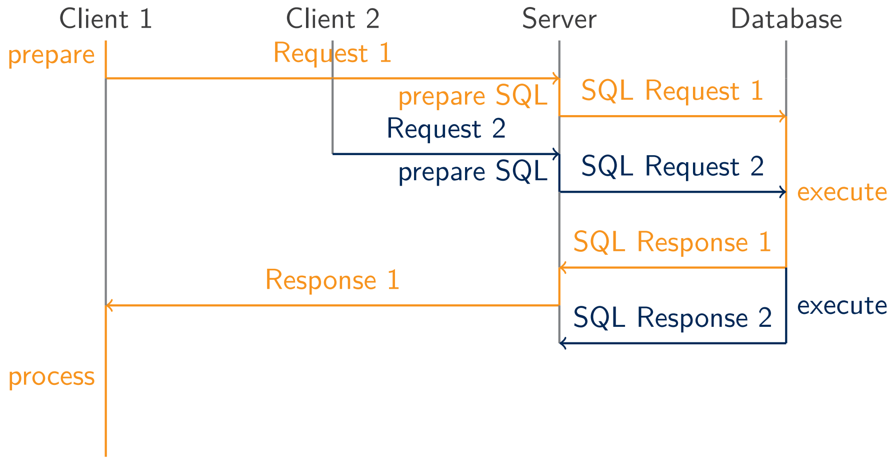

Figure: Sequence diagram of asynchronous server

Asynchronous server

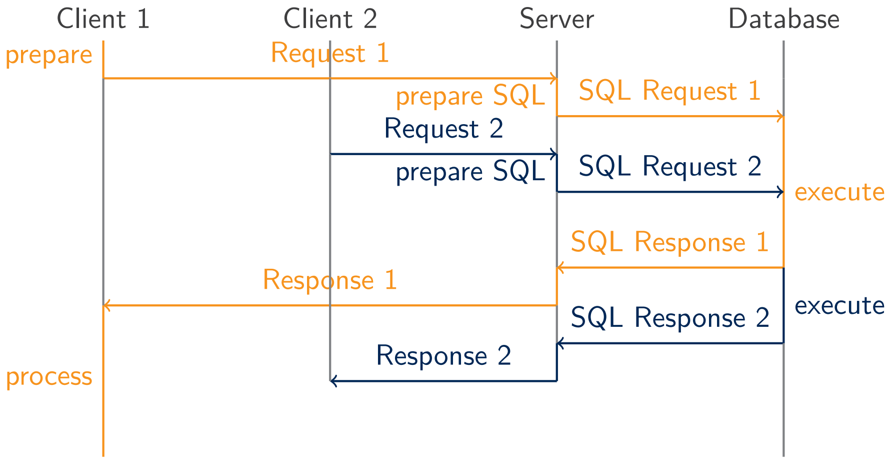

Figure: Sequence diagram of asynchronous server

Asynchronous server

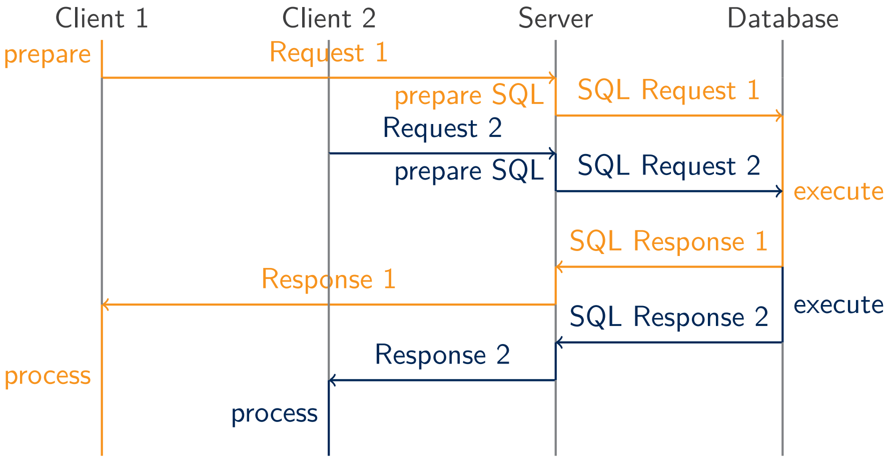

Figure: Sequence diagram of asynchronous server

Task-based Asynchronous Pattern (TAP)

- Developed by **Microsoft**.

Task-based Asynchronous Pattern (TAP)

- Developed by **Microsoft**.
- Easy transformation **Synchronous Code** $\rightarrow$ **Asynchronous Code**.

Task-based Asynchronous Pattern (TAP)

- Developed by **Microsoft**.
- Easy transformation **Synchronous Code** $\rightarrow$ **Asynchronous Code**.
- **Built-in** in C#.

Task-based Asynchronous Pattern (TAP)

- Developed by **Microsoft**.
- Easy transformation **Synchronous Code** $\rightarrow$ **Asynchronous Code**.
- **Built-in** in C#.
- Main components

Task-based Asynchronous Pattern (TAP)

- Developed by **Microsoft**.
- Easy transformation **Synchronous Code** $\rightarrow$ **Asynchronous Code**.
- **Built-in** in C#.
- Main components
    - Task

Task-based Asynchronous Pattern (TAP)

- Developed by **Microsoft**.
- Easy transformation **Synchronous Code** → **Asynchronous Code**.
- **Built-in** in C#.
- Main components
  - Task
  - Task<TResult>

Task-based Asynchronous Pattern (TAP)

- Developed by **Microsoft**.
- Easy transformation **Synchronous Code** → **Asynchronous Code**.
- **Built-in** in C#.
- Main components
  - Task
  - Task<TResult>
  - CancellationToken

Task-based Asynchronous Pattern (TAP)

- Developed by **Microsoft**.
- Easy transformation **Synchronous Code** $\rightarrow$ **Asynchronous Code**.
- **Built-in** in C#.
- Main components
  - Task
  - Task<TResult>
  - CancellationToken
  - async/await keyword

Synchronous execution in C#

```
public string Download(Uri uri) {
    var client = new DownloadClient();
    var result = client.Download(uri);

    return result;
}
```

Listing 1: Synchronous usage in C#

Event-based execution in C#

```csharp
public DownloadResult Download(Uri uri) {
    var client = new DownloadClient();
    var result = new DownloadResult();

    client.DownloadComplete += (content) => result.SetComplete(content);
    client.StartDownload(uri);

    return result;
}
```

Listing 2: Usage of events in C#

Asynchronous execution in C#

```csharp
public async Task<string> DownloadAsync(Uri uri, CancellationToken ct) {
    var client = new DownloadClient();
    var result = await client.DownloadAsync(uri, ct).ConfigureAwait(false);

    return result;
}
```

Listing 3: Asynchronous usage in C#

Constrained Application Protocol (CoAP)

- Defined in RFC 7252.

Constrained Application Protocol (CoAP)

- Defined in RFC 7252.
- Designed for **constrained** environments.

Constrained Application Protocol (CoAP)

- Defined in RFC 7252.
- Designed for **constrained** environments.
- **Request/response** interaction model.

Constrained Application Protocol (CoAP)

- Defined in RFC 7252.
- Designed for **constrained** environments.
- **Request/response** interaction model.
- Uses **U**ser **D**atagram **P**rotocol (UDP).

Constrained Application Protocol (CoAP)

- Defined in RFC 7252.
- Designed for **constrained** environments.
- **Request/response** interaction model.
- Uses **U**ser **D**atagram **P**rotocol (UDP).
- Implementation for several programming languages.

Example of CoAP message

0101010001000101110111110001100100000000000000001110111110001110
....1111111100000010

- **Version**: 1 (01......)
- **Type**: Non-Confirmable (..01....)
- **Token Length**: 4 (....0100)
- **Code**: 2.05 Content (01000101)
- **Message ID**: 51773 (11011111 00011001; Big endian)
- **Token**: 61326 (00000000 00000000 11101111 10001110)
- **Options**: Set of options
- **Payload marker**: 255 (11111111)
- **Payload**: 2 (00000010)

CoAP.NET

- Implementation of CoAP for C#.

CoAP.NET

- Implementation of CoAP for C#.
- Development inactive.

CoAP.NET

- Implementation of CoAP for C#.
- Development inactive.
- Partially asynchronous.

CoAP.NET

- Implementation of CoAP for C#.
- Development inactive.
- Partially asynchronous.
- Offers a client and server implementation.

Bachelor thesis

- *»Has an asynchronous implementation of a server an impact on its throughput?«*

Bachelor thesis

- *»Has an asynchronous implementation of a server an impact on its throughput?«*
- Fully rewrite CoAP.NET library

Bachelor thesis

- *»Has an asynchronous implementation of a server an impact on its throughput?«*
- Fully rewrite CoAP.NET library **except retransmission and block-wise transfer**.

Bachelor thesis

- *»Has an asynchronous implementation of a server an impact on its throughput?«*
- Fully rewrite CoAP.NET library **except retransmission and block-wise transfer**.
- Implement tests for measuring throughput.

Bachelor thesis

- *»Has an asynchronous implementation of a server an impact on its throughput?«*
- Fully rewrite CoAP.NET library **except retransmission and block-wise transfer**.
- Implement tests for measuring throughput.
- Compare synchronous with asynchronous version.

Bachelor thesis

- *»Has an asynchronous implementation of a server an impact on its throughput?«*
- Fully rewrite CoAP.NET library **except retransmission and block-wise transfer**.
- Implement tests for measuring throughput.
- Compare synchronous with asynchronous version.
- Source code freely available at GitHub.

Bachelor thesis

- *»Has an asynchronous implementation of a server an impact on its throughput?«*
- Fully rewrite CoAP.NET library **except retransmission and block-wise transfer**.
- Implement tests for measuring throughput.
- Compare synchronous with asynchronous version.
- Source code freely available at GitHub.
- Collaboration with World-Direct eBusiness solutions GmbH.

Bachelor thesis

- *»Has an asynchronous implementation of a server an impact on its throughput?«*
- Fully rewrite CoAP.NET library **except retransmission and block-wise transfer**.
- Implement tests for measuring throughput.
- Compare synchronous with asynchronous version.
- Source code freely available at GitHub.
- Collaboration with World-Direct eBusiness solutions GmbH.

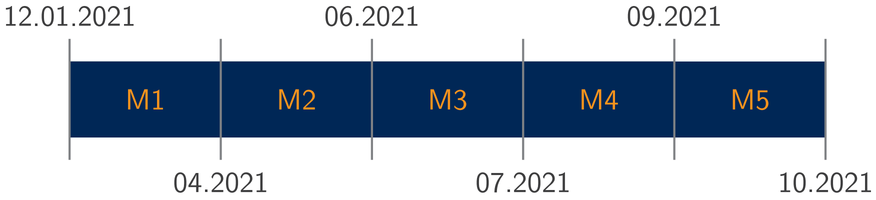

Figure: Phase-Milestone plan

Bachelor thesis

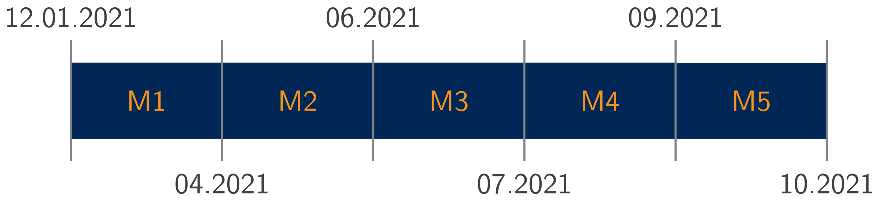

Figure: Phase-Milestone plan

Bachelor thesis

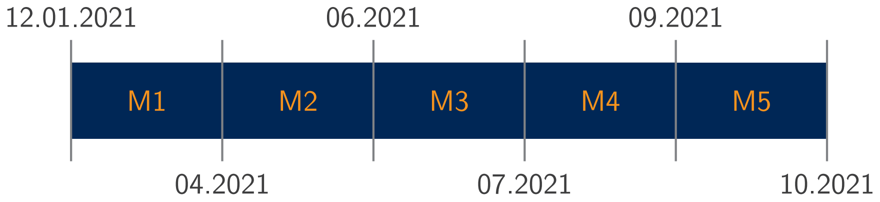

Figure: Phase-Milestone plan

1. 12.01.2021: Initial presentation.

Bachelor thesis

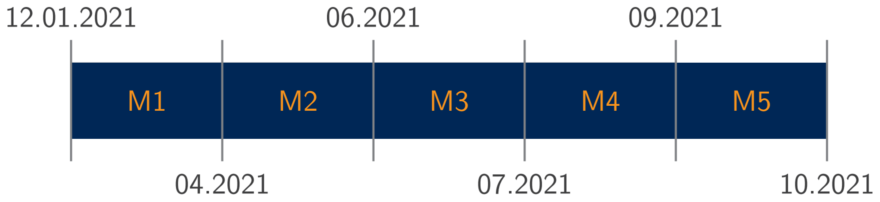

Figure: Phase-Milestone plan

1. 12.01.2021: Initial presentation.
2. 04.2021: Finish asynchronous implementation.

Bachelor thesis

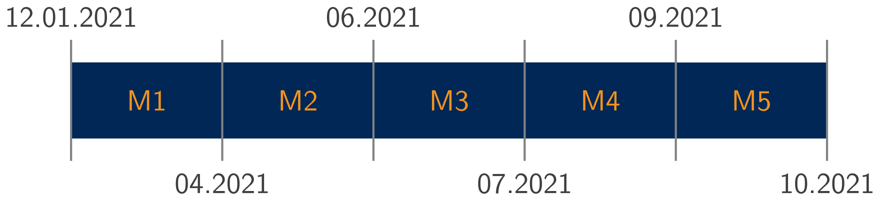

Figure: Phase-Milestone plan

1. 12.01.2021: Initial presentation.
2. 04.2021: Finish asynchronous implementation.
3. 06.2021: Finish measurements.

Bachelor thesis

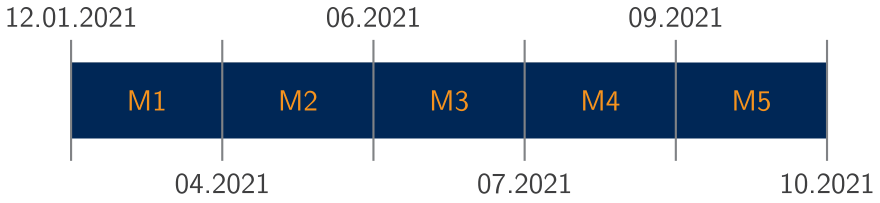

Figure: Phase-Milestone plan

1. 12.01.2021: Initial presentation.
2. 04.2021: Finish asynchronous implementation.
3. 06.2021: Finish measurements.
4. 07.2021: Finish comparison.

Bachelor thesis

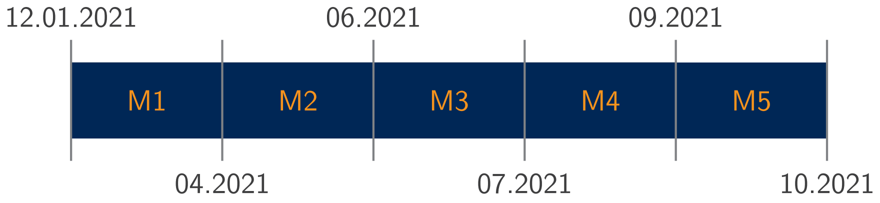

Figure: Phase-Milestone plan

1. 12.01.2021: Initial presentation.
2. 04.2021: Finish asynchronous implementation.
3. 06.2021: Finish measurements.
4. 07.2021: Finish comparison.
5. 09.2021: Finish writing of bachelor thesis.

Bachelor thesis

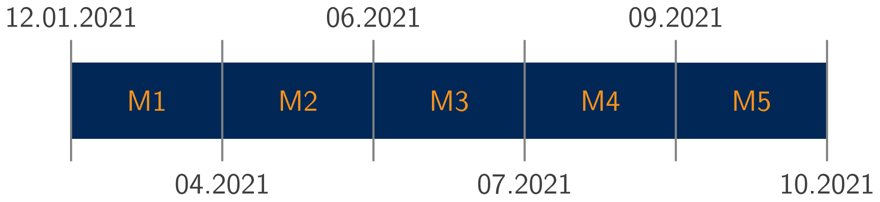

Figure: Phase-Milestone plan

1. 12.01.2021: Initial presentation.
2. 04.2021: Finish asynchronous implementation.
3. 06.2021: Finish measurements.
4. 07.2021: Finish comparison.
5. 09.2021: Finish writing of bachelor thesis.
6. 10.2021: Final presentation.