FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis

A Framework for Distributed Systems Based on The Actor Programming Model and Dart language, Which Unifies Applications Across Devices, Clients and Servers, and Supports Features for Hot Deployment and Migration of Actors

Sushil Man Shilpakar

FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis

A Framework for Distributed Systems Based on The Actor Programming Model and Dart language, Which Unifies Applications Across Devices, Clients and Servers, and Supports Features for Hot Deployment and Migration of Actors

TODO: Titel der Abschlussarbeit

Author:	Sushil Man Shilpakar
Supervisor:	Prof. Hans Arno Jacobsen
Advisor:	Richard Billeci
Submission Date:	TODO: Submission date

I assure the single handed composition of this master's thesis only supported by declared resources.

Munich, TODO: Submission date Sushil Man Shilpakar

Acknowledgments

Abstract

Contents

Acknowledgments iii

Abstract iv

1 Introduction 1

1.1 Section 1

1.1.1 Subsection 1

1.2 Section 1

2 Literature Review 3

2.1 Actor Programming Model 3

2.2 Erlang 3

2.3 Akka Toolkit 3

2.4 The Dart Language 3

2.5 STOMP 3

2.6 Message Broker System - RabbitMQ 3

2.7 WebSockets 3

3 General Design Decisions 4

3.1 Section 4

4 Results 5

4.1 Section 5

5 Discussions 6

5.1 Section 6

6 Conclusion 7

6.1 Section 7

7 Recommendations 8

7.1 Section 8

8 Appendices **9**
8.1 Section 9

List of Figures **10**

List of Tables **11**

Bibliography **12**

1 Introduction

1.1 Section

Citation test [Lam94]. The purpose of this thesis is to build a framework based on the actor programming model and the Dart language. The framework unifies applications across devices, client and server and also supports migration of actors in a distributed system. So, first of all we should briefly overview the actor programming model and how it can be realized efficiently in the Dart language. Although the actor model was introduced in mid 1980s and there had been programming languages like Erlang that implemented it, only now it has started gaining wide popularity in distributed systems. Especially after the introduction of Scala and Akka, the actor model has been gaining good popularity. The Dart programming language provides a homogeneous system that encompasses both client as well as server as the Dart Virtual Machine runs in servers as well as in browsers. This particular nature of the Dart language makes it possible to create a fully distributed application in which isolates (the actor like entities of Dart language) may run everywhere — in servers, in desktop browsers and even in mobile browsers.

1.1.1 Subsection

See Figure 1.1.

Figure 1.1: An example for a figure.

1.2 Section

See Table 1.1, Figure 1.2, Figure 1.3, Figure 1.4.

Table 1.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

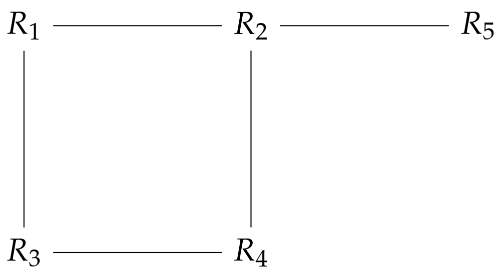

Figure 1.2: An example for a simple drawing.

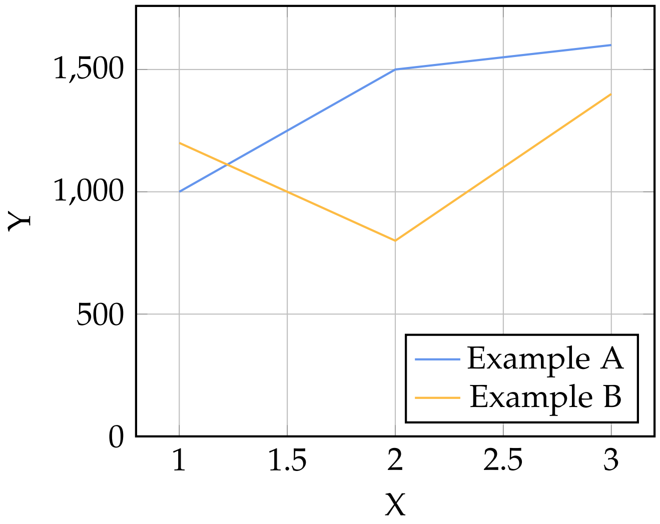

Figure 1.3: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 1.4: An example for a source code listing.

2 Literature Review

2.1 Actor Programming Model

Actor programming model is a programming paradigm designed for concurrent computation.

2.2 Erlang

2.3 Akka Toolkit

2.4 The Dart Language

2.5 STOMP

2.6 Message Broker System - RabbitMQ

2.7 WebSockets

3 General Design Decisions

3.1 Section

4 Results

4.1 Section

5 Discussions

5.1 Section

6 Conclusion

6.1 Section

7 Recommendations

7.1 Section

8 Appendices

8.1 Section

List of Figures

1.1 Example figure 1
1.2 Example drawing 2
1.3 Example plot 2
1.4 Example listing 2

List of Tables

1.1 Example table 2

Bibliography

[Lam94] L. Lamport. *LaTeX : A Documentation Preparation System User's Guide and Reference Manual*. Addison-Wesley Professional, 1994.