



# Microservices: Patterns and Applications

Designing distributed services  
and systems patterns

Lucas Krause

# MICROSERVICES: PATTERNS AND APPLICATIONS

*DESIGNING FINE-GRAINED SERVICES  
BY APPLYING PATTERNS*

*LUCAS KRAUSE*

**Copyright © 2014, 2015 by Lucas Krause.**

**This version was published on 4-1-2015**

**V1.0**

Get additional information, code samples  
and community support at:

[microservicesbook.io](http://microservicesbook.io)

Please help Lucas Krause by spreading the word about this book on Twitter!

The suggested hashtag for this book is #microservicespatterns.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on

Twitter:

<https://twitter.com/search?q=#microservicespatterns>

FOR LIZ AND LILY

# WHY I WROTE THIS BOOK

I began my own foray into Microservices many years ago and attended many conferences learning about fine-grained systems, benefits and pitfalls. As a software designer, developer and architect for many years, I believe that all architecture choices I'd made were always the most important choices effecting the outcome and the success of software creation.

Microservices is a major architectural shift and is a fundamentally different way of building software. When I started writing this book there weren't many resources except conferences and blogs. I was excited about the prospect of designing and building these systems and I wanted to share my experience with the world to aide other developers, architects, administrators and managers to better understand Microservices.

# WHY YOU SHOULD READ THIS BOOK

This book is to help you understand Microservices and how you can use them to solve problems, it outlines when to use Microservices and how to think about the Microservice architecture which is a departure from traditional software design paradigms. Microservices is creating and designing fine-grained, detailed, and focused services for new (or to replace existing) monolithic complex applications. Microservices adhere to a core tenet of the UNIX platform to “do one thing and do it well.”

This book explains Microservice architecture, patterns, and designs and how they allow your solution to scale, be easier to change and adapt to evolving needs. This book is for software designers (Architects, Engineers, Administrators, and Managers) and also for people who just want to learn more about what Microservices are and what makes them important.

# TABLE OF CONTENTS

Why I Wrote This Book

Why You Should Read This Book

Table of Contents

Chapter 1. Microservices

What are they and Why are they important?

Summary

Chapter 2. Approach to the Problem

How do Microservices solve these problems

Common Problems and how to identify these problems

Solutions

Summary

Chapter 3. Applied Microservices more than a choice, a philosophy

Microservice Philosophy

Emergence

Failure

Evolution

Platform as a Service (PaaS)

Root-Cause Analysis

Summary

Chapter 4. Walkthrough the process Monolithic App To Microservice

Breaking out Bounded Contexts

Another approach to the problem

Start small, incremental improvements

Summary

## Chapter 5. Walkthrough the process SOA Application To Microservice

Breaking out Bounded Contexts

Start small, incremental improvements

Summary

## Chapter 6. Walkthrough for creating Green Field Microservices

Breaking out Bounded Contexts

Start small, incremental improvements

Summary

## Chapter 7. Data

Relational Databases

No SQL

Sharing Data

Summary

## Chapter 8. Real-World Services – Problems and Solutions

Dependency

Versioning

Security

Exceptions, Time-outs and Logging

Summary

## Chapter 9. Scalability

Scale Cube

Caching

Containers

Virtualization

Summary

## Chapter 10. Microservice and Monolithic Architectures

Monolithic Architecture

Microservices Architecture

Solving complexity of distributed systems

Summary

## Chapter 11. Microservice Patterns

API Patterns

API Proxy Pattern

Orchestrated API Pattern

Discovery Patterns

Registration Patterns

Instantiation Patterns

System Patterns

Message Broker

Command Query Responsibility Segregation (CQRS)

Twelve – factor

Circuit Breaker

Bulkheads

Summary

## Chapter 12. Making your services successful

Simplicity

Organizational Alignment

What's Next?

Summary

References

# CHAPTER 1. MICROSERVICES

Microservices are the next big thing in modern software architecture and development. Since the inception of the computer, building software has been a complicated and difficult process. Many challenges may seem different but are all too common in the industry and the solutions spanning generations seem to change from centralized computing (mainframes) to distributed computing (pc) back and forth with each wave of new technologies and hardware capability, adding more choices and at the same time making things more confusing.

Distributed architecture is part of a sea-change in the industry and distributed systems have massively changed the way technology works, specifically:

- Git (distributed source control).
- Cloud Computing (AWS, Google Cloud, Azure).
- Bit torrent (distributed file sharing).
- Bitcoin (distributed virtual cryptocurrencies).

These technologies have changed the way computer systems are used, work and built. The advent of Microservices is a logical evolution of these disruptive changes.

The biggest reason these services are starting to gain popularity now is the intersection of operational automation and simplicity alongside a more centralized technology skillset and stack. If you review the recent innovations in cloud computing, virtualization and container technology, it makes the operational management of Microservices easier. Many of the recent advances in technology, specifically cloud infrastructure, containers, event processing, and service buses, all lower the barriers to accomplish what would have

been much more difficult with older technologies. Unified software languages, specifically JavaScript, are also a reason this makes sense. The availability of easy server-side and client-side technologies sharing the same language makes things easier to understand and also increases the rate of innovation and maturity.

## WHAT ARE THEY AND WHY ARE THEY IMPORTANT?

“The Microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

- (Fowler, 2014)

Microservices is an approach to developing a large or complex application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and are independently deployable using automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

## REFERENCES

- (1968). In F. Brooks, *The Mythical Man-Month*. Addison-Wesley.
- (1999). In D. T. Andres Hunt, *The Pragmatic Programmer: From Journeyman to Master*. The Pragmatic Programmers, LLC.
- Conway, M. E. (1968). How do Committees Invent? *Datamation*, 28-31.
- Einstein, A. (1933). On the Method of Theoretical Physics. *Philosophy of Science*.
- Fowler, M. (2014, March 25). *Microservices*. Retrieved from MartinFowler.com:  
<http://martinfowler.com/articles/microservices.html>
- Martin, R. C. (2009). *Getting a SOLID start*. Retrieved from <https://sites.google.com/site/unclebobconsultingllc/getting-a-solid-start>
- Nygard, M. T. (2007). *Release It!: Design and Deploy Production-Ready Software*. Pragmatic Bookshelf.
- Raymond, E. S. (2003). *The Art of UNIX Programming*. Addison-Wesley .
- Richardson, C. (2014, March 18). *Microservice architecture patterns and best practices*. Retrieved from Microservices.IO: <http://microservices.io/>

Wiggins, A. (2012, 1 30). Retrieved from The Twelve Factor  
App: <http://12factor.net/>

## About The Author

**Lucas Krause** has been a consultant and problem solver in the industry for over 15 years. He lives in Colorado with his family and enjoys the outdoors, all things technology, including technical challenges. Lucas loves educating and inspiring others to learn from his experience to solve their problems and make the world a better place.

Learn more about Lucas at

# ONE LAST THING...

If you enjoyed this book or found it useful I'd be very grateful if you'd post a short review on Amazon. Your support really does make a difference and I read all the reviews personally so I can get your feedback and make this book even better.

If you'd like to leave a review then all you need to do is click the review link on this book's page on Amazon here:

Thanks again for your support!