Scaling Go Applications Horizontally

INTRODUCTION

Michael Van Sickle

@vansimke

Overview

The Problem

Possible Solutions

Target Audience

Demo Scenario

Course Outline

The Problem

How to optimize the use of resources to most economically deliver valuable services to customers.

Resources

Network Bandwidth

Processing Power

Available Memory

Data Storage

Possible Solutions

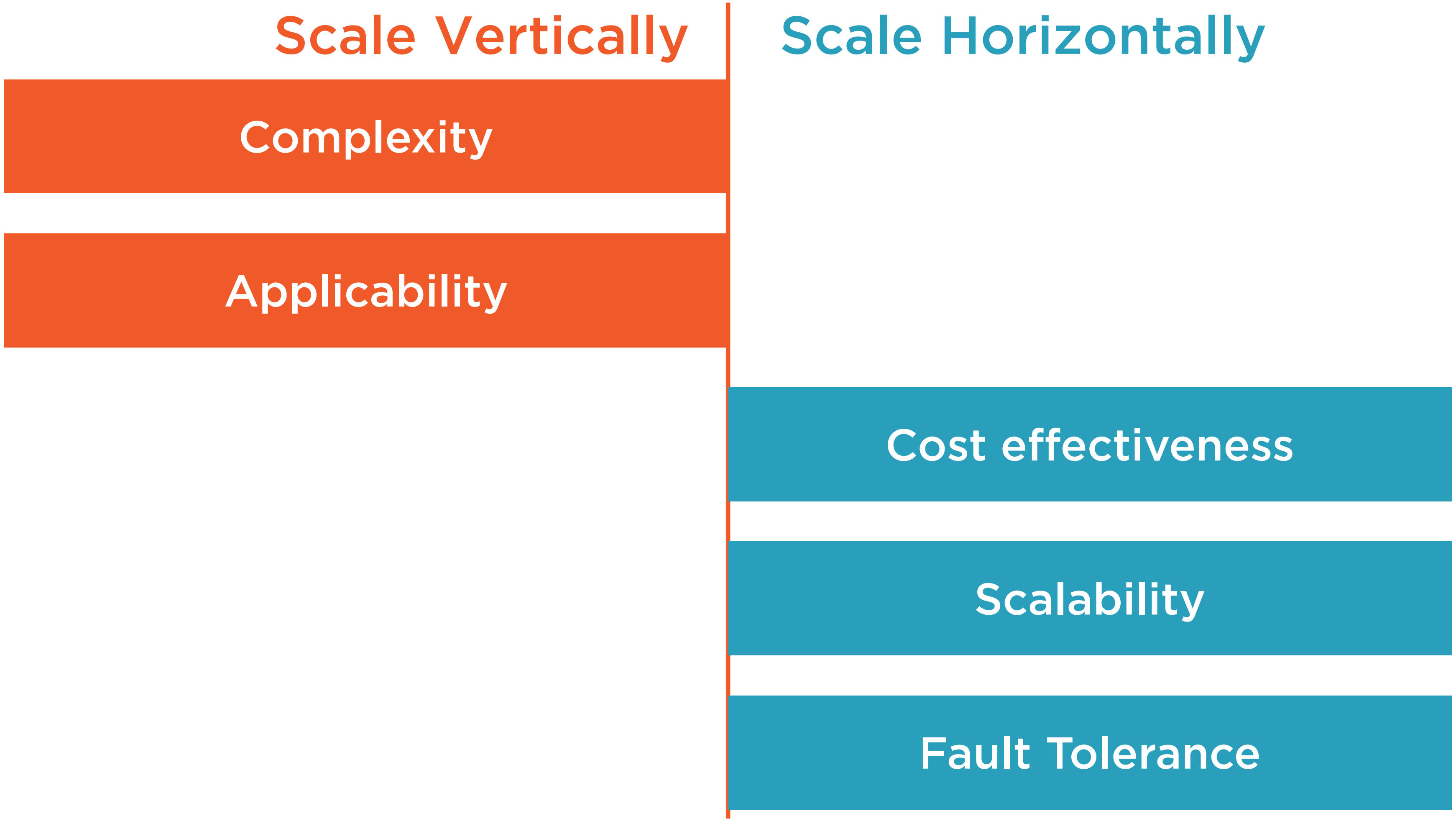
Better Hardware

Scale Vertically

More Hardware

Scale Horizontally

Possible Solutions



Scale Vertically

- Complexity
- Applicability

Scale Horizontally

- Cost effectiveness
- Scalability
- Fault Tolerance

Horizontal Scaling Technologies

Physical Machines

Virtual Machines

Containers

Horizontal Scaling Technologies

	Physical	Virtual	Container

Horizontal Scaling Technologies

	Physical	Virtual	Container
Network			

Horizontal Scaling Technologies

	Physical	Virtual	Container
Network	+	-	-

Horizontal Scaling Technologies

	Physical	Virtual	Container
Network	+	-	-
CPU			

Horizontal Scaling Technologies

	Physical	Virtual	Container
Network	+	-	-
CPU	++	--	-

Horizontal Scaling Technologies

	Physical	Virtual	Container
Network	+	-	-
CPU	++	--	-
Memory			

Horizontal Scaling Technologies

	Physical	Virtual	Container
Network	+	-	-
CPU	++	--	-
Memory	++	--	-

Horizontal Scaling Technologies

	Physical	Virtual	Container
Network	+	-	-
CPU	++	--	-
Memory	++	--	-
Storage			

Horizontal Scaling Technologies

	Physical	Virtual	Container
Network	+	-	-
CPU	++	--	-
Memory	++	--	-
Storage	0	0	-

Target Audience

Go

- Go: Getting Started
- Go Fundamentals

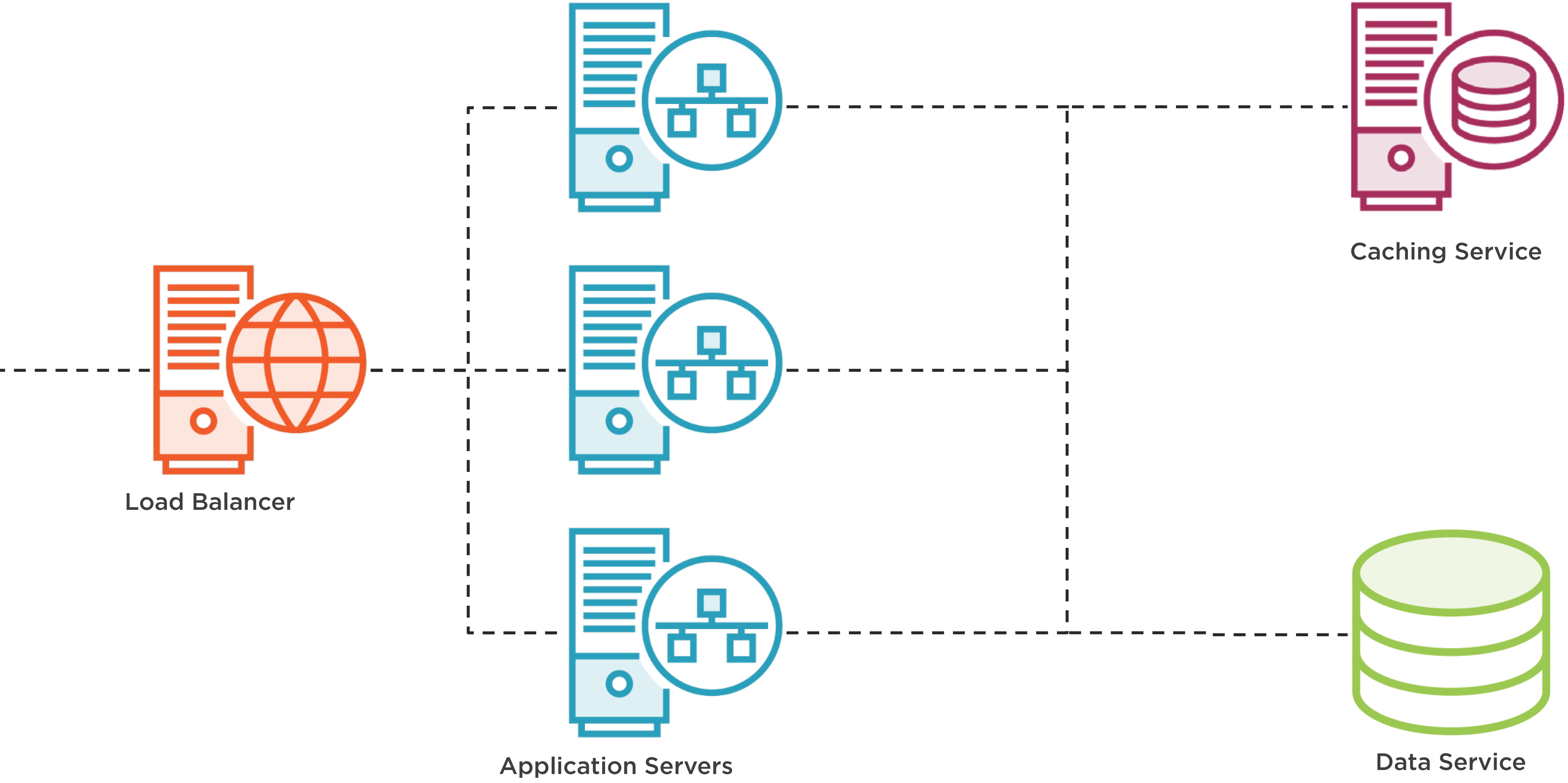
Web Applications

- Creating Web Applications with Go

Docker

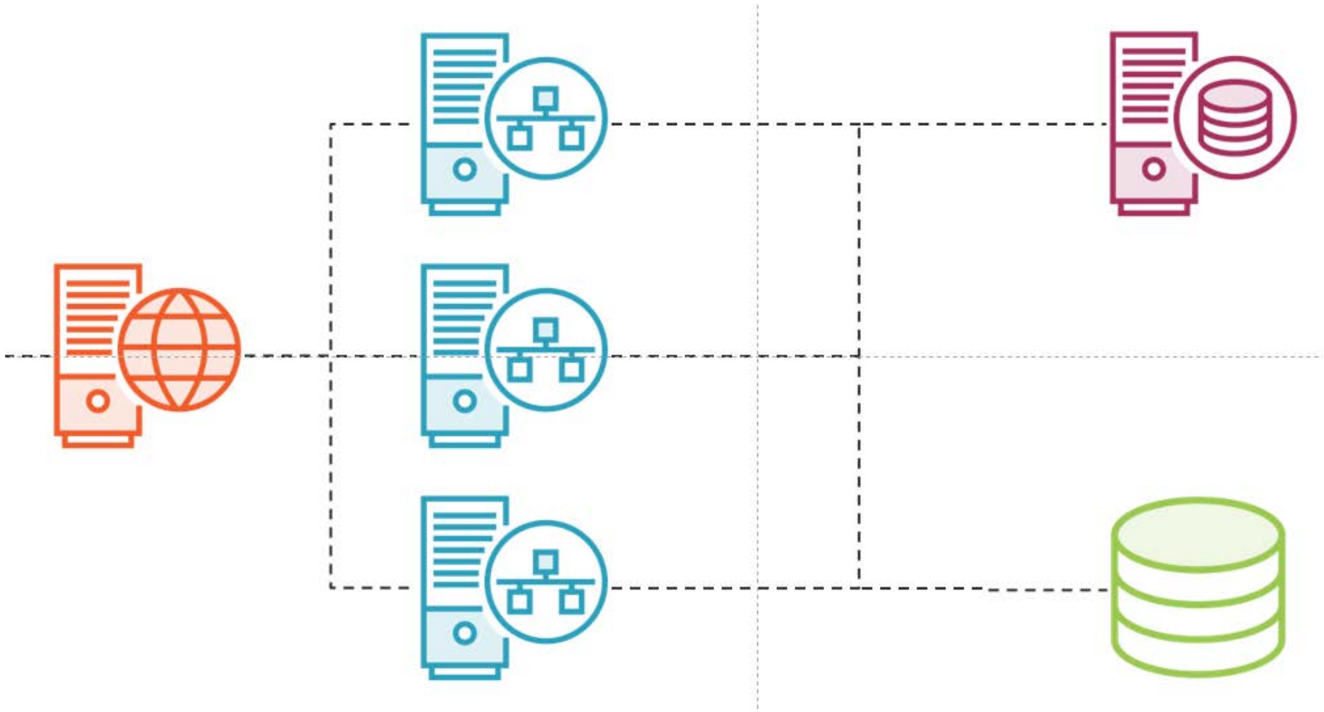
- Docker and Containers: The Big Picture

The Plan


Caching Service
Load Balancer
Application Servers
Data Service

Pure Go

Docker for Development



Outline

Initial Optimizations

Creating a Load Balancer

Caching

Centralized Logging