# Building High Performance Applications with Spring Data and MongoDB

JPrime 2017

**Kiril Stefanov**
**Technical Lead @ VMware**

May 31, 2017

vmware

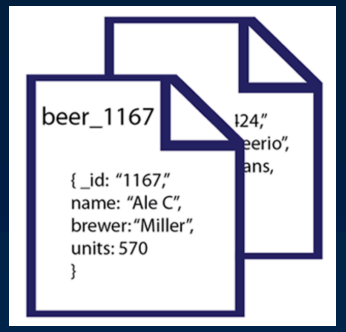
# Who Am I

- Java Software Engineer
  - Working on Java EE projects for more than 13 years
- Java Lecturer and Consultant
- Spring Certified Professional
- Technical Lead at VMware
  - Working on vRealize Cloud Automation Software

# Agenda

| | |
|---|---|
| 1 | **MongoDB Overview** |
| 2 | Spring Data MongoDB Overview |
| 3 | Self-Service Catalog Application Example |

# What is MongoDB?


Document Oriented Storage



Ad hoc queries, indexing and real time aggregation

Distributed Database

vmware

# Database Ranking

| Rank May 2017 | Rank Apr 2017 | Rank May 2016 | DBMS | Database Model | Score May 2017 | Score Apr 2017 | Score May 2016 |
|---|---|---|---|---|---|---|---|
| 1. | 1. | 1. | Oracle | Relational DBMS | 1354.31 | -47.68 | -107.71 |
| 2. | 2. | 2. | MySQL | Relational DBMS | 1340.03 | -24.59 | -31.80 |
| 3. | 3. | 3. | Microsoft SQL Server | Relational DBMS | 1213.80 | +9.03 | +70.98 |
| 4. | 4. | ↑ 5. | PostgreSQL | Relational DBMS | 365.91 | +4.14 | +58.30 |
| 5. | 5. | ↓ 4. | MongoDB | Document store | 331.58 | +6.16 | +11.36 |
| 6. | 6. | 6. | DB2 | Relational DBMS | 188.84 | +2.18 | +2.88 |
| 7. | 7. | ↑ 8. | Microsoft Access | Relational DBMS | 129.87 | +1.69 | -1.70 |
| 8. | 8. | ↓ 7. | Cassandra | Wide column store | 123.11 | -3.07 | -11.39 |
| 9. | 9. | 9. | Redis | Key-value store | 117.45 | +3.09 | +9.21 |
| 10. | 10. | 10. | SQLite | Relational DBMS | 116.07 | +2.27 | +8.81 |

# MongoDB Key Features

- High Performance
  - Support for embedded data models reduces IO activity
  - Indexes support for faster queries and can include keys from embedded documents and arrays
- Rich Query Language
  - Data aggregation, text search and geospatial queries
- High Availability
  - Automatic failover and data redundancy

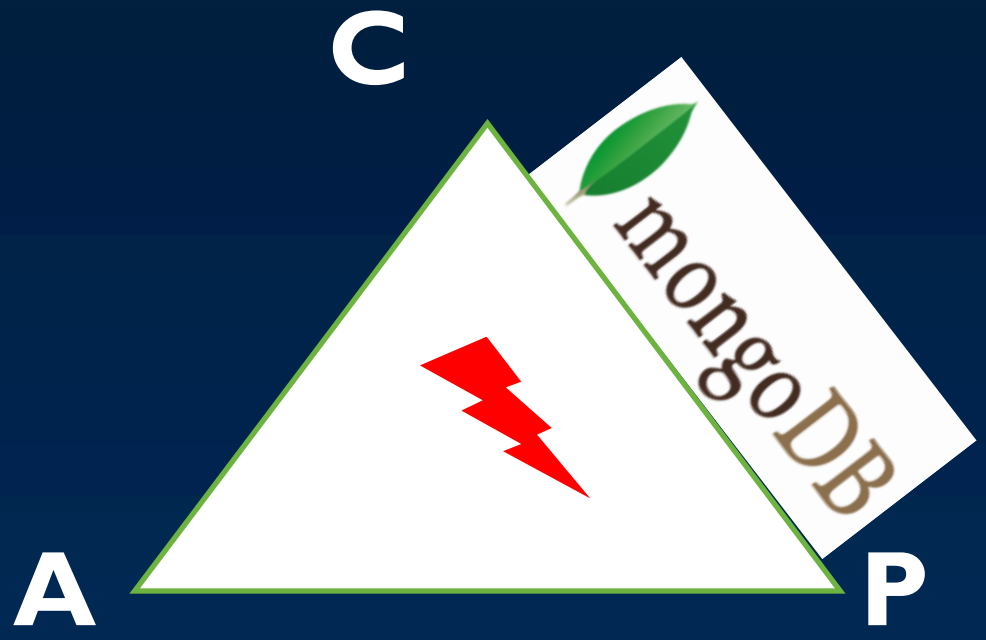
# MongoDB Key Features

- Horizontal Scalability
  - Sharding distributes data across a cluster of machines
  - Mongo 3.4 supports zones of data based on shard key
- Support for Multiple Storage Engines
  - WiredTiger, MMAPv1, In-Memory

# MongoDB: CAP Approach

## Focus on Consistency and Partition tolerance

- **C**onsistency
  - all replicas contain the same version of data
- **A**vailability
  - system remains operational on failing nodes
- **P**artition tolarence
  - multiple entry points
  - system remains operational on system split




CAP Theorem:
Satisfying all three at the same time is impossible!

vmware

# Why use MongoDB?

- Build applications faster
- MongoDB documents maps naturally to modern OOP
- Flexible data model
- Scale within and across multiple distributed datacenters easily with no downtime

# What is MongoDB great for?

- RMDS for Web Applications
- Common use cases include:
  - Single View
  - Internet of Things
  - Real-time Analytics
  - Catalog
  - Content Management

# What is MongoDB not great for?

- Highly translational applications
- Legacy applications built around the relational data model and SQL

# RDB Concepts to NoSQL

| RDBMS | | MongoDB |
|---|---|---|
| Database | → | Database |
| Table, View | → | Collection |
| Row | → | Document (JSON, BSON) |
| Column | → | Field |
| Index | → | Index |
| Join | → | Embedded Document |
| Foreign Key | → | Reference |
| Partition | → | Shard |

# MongoDB Document Model

```
{
        "_id" : ObjectId("5114e0bd42…"),
        "name" : ”Ivan Petrov",
        "age" : 34,
        "address" : {
                "street" : ”Vitosha",
                "city" : ”Sofia",
                ”country” : ”Bulgaria” }
        "expertise" : [ "MongoDB", "Java", "Spring" ],
        "location" : [ 42.69, 23.32 ]
}
```

Primary Key (internal)

String

Integer

Nested Document

Array

Geo-spatial Coordinates

vmware

# Agenda

1 MongoDB Overview

2 **Spring Data MongoDB Overview**

3 Self-Service Catalog Application Example

vmware

# Spring Data Basics

- **Repository**
  - Convert retrieved data into POJOs
  - Convert POJOs into saved data
- **Mapping**
  - Inferred from conventions
  - Annotations to help automate field mapping
- **Template**
  - Provide simplified direct-access to database
  - Automatically managing internal resources
- **Query**
  - Apply provided native queries directly
  - Convert QueryDSL into native queries

# Spring Data MongoDB: Key Features

- Configuration support for a Mongo driver instance and replica sets
- MongoTemplate
  - Allows simplified direct access to Mongo database
  - Provides high-level methods: load(), save(), find()
- Repository support
  - Automatic implementation of Repository interfaces including support for custom finder methods
  - Queries are derived from method signatures
  - Supports native mongodb queries (JSON syntax)
- Annotation based mapping
  - @Document, @Id, @Field, @Indexed, @TextIndexed, etc.

# Spring Data MongoDB: Additional Features

- Exception translation into Spring's portable Data Access Exception hierarchy
- Feature Rich Object Mapping integrated with Spring's Conversion Service
- Persistence and mapping lifecycle events
- Java based Query, Criteria, and Update DSLs
- Cross-store persistence - support for JPA Entities with fields transparently persisted/retrieved using MongoDB
- GeoSpatial integration

# Agenda

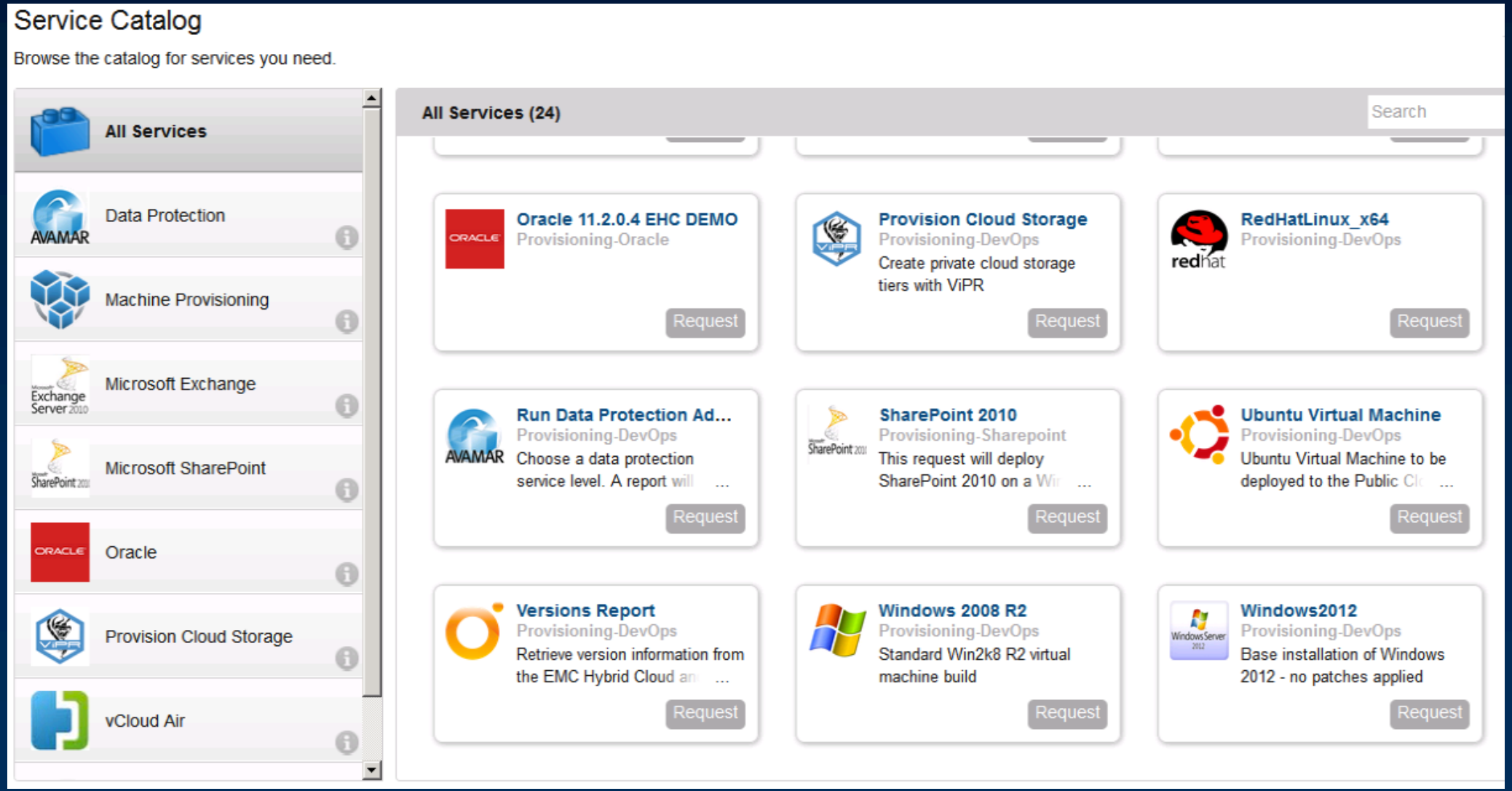
1 MongoDB Overview

2 Spring Data MongoDB Overview

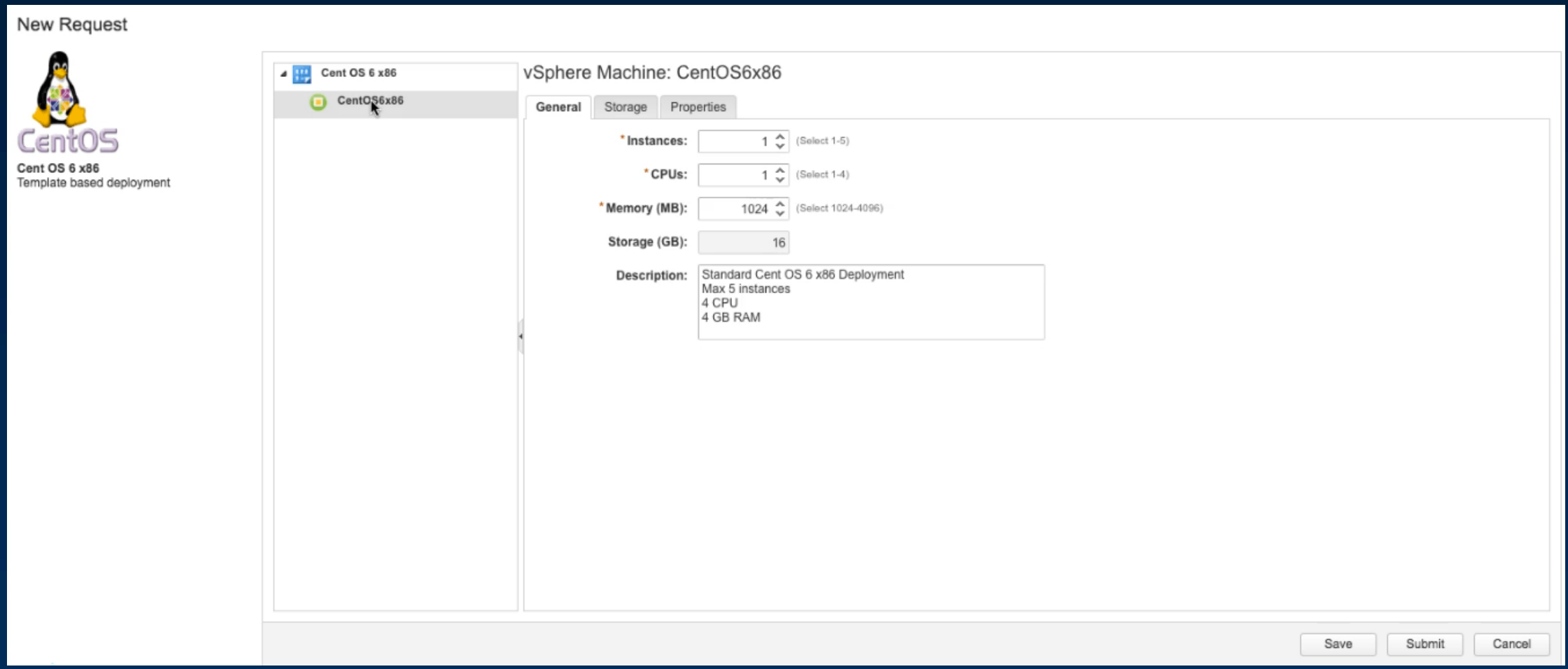
3 **Self-Service Catalog Application Example**

vmware

# Self-Service Catalog Application



# Self Service Catalog Item Request



# References

- www.mongodb.com
- projects.spring.io/spring-data-mongodb
- spring.io/guides/gs/accessing-data-mongodb

# Thank you

# Questions?

Kiril Stefanov

vmware®