

Discovery & Test Driven Database Development

Daniel Hinojosa

https://github.com/dhinojosa/discovery_database_tdd

ARCHCONF | 2022

Software Architecture Conference

[DAYS]

23

[HOURS]

22

[MINUTES]

41

[SECONDS]

21

December 12 - 15, 2022 · Clearwater, FL

Attendance is limited to 350. [Register today!](#)

[REGISTER NOW](#)

<https://archconf.com/>





Venkat Subramaniam

Founder of Agile Developer, Inc.

[Creating Architectural Fitness Functions using ArchUnit](#)

[Creating Evolutionary Architecture and Design: From Problem to Solution by exploring patterns](#)

[Creating Flexible Design with Traits](#)

[Design Patterns in the Light of Lambda Expressions](#)

[Designing Microservices: From Architecting to Data Modeling](#)

[Designing in the World of Microservices: The Emphasis on Bounded Context](#)

[Software Design By Practice](#)

[Staying One Step Ahead of Java Evolution: Learning from Other Languages](#)

[The Journey from Monoliths to Microservices](#)



Daniel Hinojosa

Independent Consultant

[Hashicorp Vault](#)

[Java Design Patterns](#)

[Journey into Architectural Design Patterns](#)

[Kubernetes Security Concepts Part 1](#)

[Kubernetes Security Concepts Part 2](#)

[Machine Learning Data Pipelines](#)



Ken Sipe

Cloud Architect

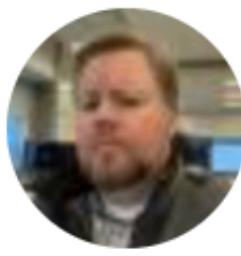
Architectural Awareness: Engineering Super-skill

Do You Know Da Wae

Micro-Services Day 2 Networking

Standing up Enterprise Architecture

To 99.99 and Beyond



Matt Stine

I Enable Early-Career Enterprise Software Engineers to Continuously Improve

Digital Knowledge Management for Architects

Egoless Architecture

Improving Enterprise Technical Architecture with Paved Paths



Michael Carducci

Software Architect

Mob Mentality - Collaborative coding and problem solving

Six Essential Soft Skills for the Software Architect

The Linked Data Revolution is here, are you ready?

The Secrets to Smart, Flexible, Self-Describing API

Understanding Resource-Oriented Architecture



Travis Gosselin

Principal Software Engineer, SPS Commerce

API Design First: Succeeding with API Governance

Compelling Code Reuse in the Enterprise

Unleashing Deploy Velocity with Feature Flags



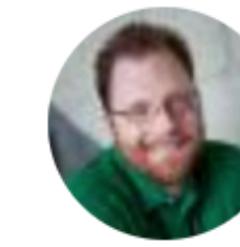
Dave Hendrickson

Senior Architect for Thomson Reuters, Author, and Mentor

[12 Essential Skills for Software Architects](#)

[AWS ML/AI Services Deep Dive](#)

[Cryptocurrency Architecture](#)



Nathaniel Schutta

Architect as a Service

[An Architect's Guide to Site Reliability Engineering](#)

[Architecting Cloud Native Applications](#)

[Developer To Architect](#)

[Modeling for Architects](#)

[Paved Roads - Architecting for Distributed Teams](#)

[Production Hardened Services](#)

[Spring For Architects](#)

[Succeeding With Microservices](#)

[Thinking Architecturally](#)



Brian Sletten

Forward Leaning Software Engineer

Knowledge Graphs: Architecture and Information

LLVM : Architecture and Language

Machine Learning Platforms : Architecture and Learning

Modern Software Systems : Succeeding at IT in the 21st Century

Reactive Spring : Modernizing Code and Architecture in the Spring Ecosystem

WebAssembly : Architecture and Heterogeneity



Jonathan Johnson

Software Architect

Architectures for Cloud Native

Chaos Experiment Design - Boom!

Delivery Models with Pipelines on Kates

Maturity Model Vitals

Traffic Routing Through a Kubernetes Cluster

Zero Trust Kubernetes



Kirk Knoernschild

Software Developer & Mentor

30 Years of Software Development: Lessons from the Trenches

Agile Architecture

Modularity: The Foundation of Modern Architectures



Tim Berglund

VP, Developer Relations at StarTree

An Introduction to Apache Pinot

Building a Real-Time Analytics Database: a Choose Your Own Adventure Journey

Decision Making in Software Teams

Event-Driven Architecture Done Right



Justin Reock

Field CTO of Gradle Enterprise

DevOps, 12-Factor, and Open Source

Improving Software Fuzzing with AI and ML



Kate Wardin

Senior Engineering Manager @ Netflix

Adaptive Technical Leadership

Foundations of Tech Leadership

ARCHCONF | 2022

Software Architecture Conference

[DAYS]

23

[HOURS]

22

[MINUTES]

41

[SECONDS]

21

December 12 - 15, 2022 · Clearwater, FL

Attendance is limited to 350. [Register today!](#)

[REGISTER NOW](#)

<https://archconf.com/>





My Database Testing Journey

The Adam

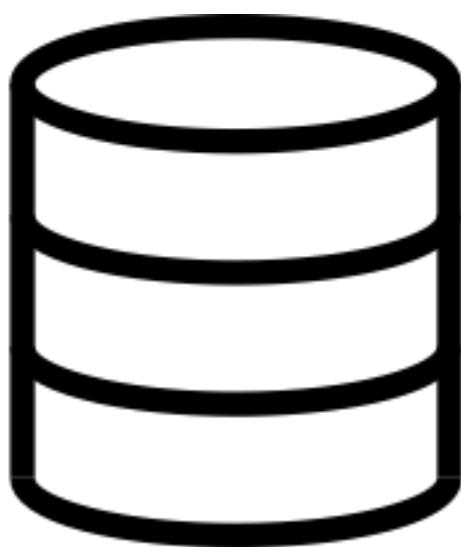
TEST-DRIVEN
DEVELOPMENT

BY EXAMPLE

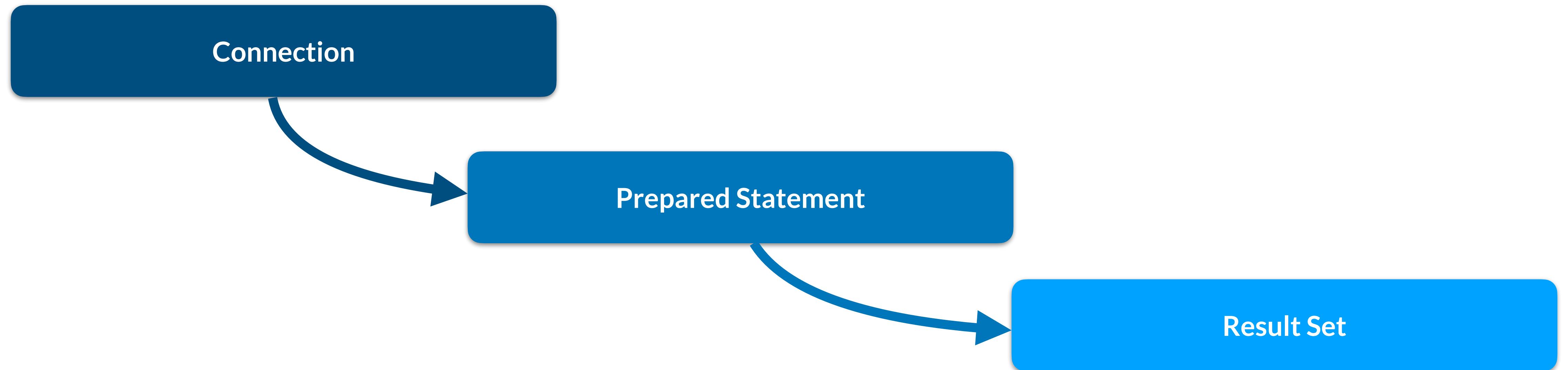
KENT BECK

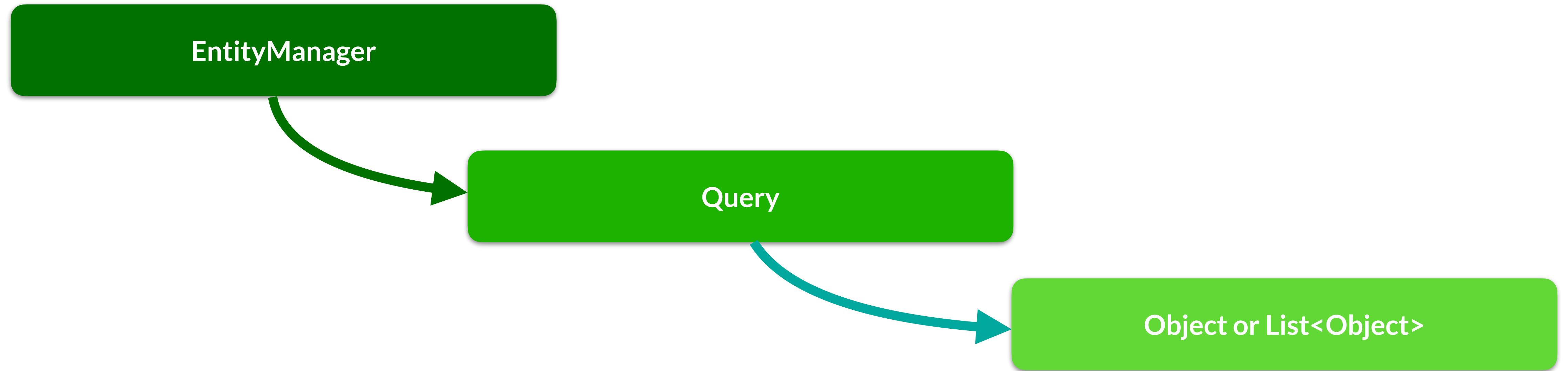


But when it came to databases?









**Mocking (can be/is) a
code smell**



Miško Hevery

The Testability Explorer Blog

HOME ABOUT MIŠKO HEVERY ASK! GUIDE: WRITING TESTABLE CODE PRESENTATIONS

Guide: Writing Testable Code

To keep our code at Google in the best possible shape we provided our software engineers with these constant reminders. Now, we are happy to share them with the world.

Many thanks to these folks for inspiration and hours of hard work getting this guide done:

- [Jonathan Wolter](#)
- [Russ Ruffer](#)
- [Miško Hevery](#)

Also thanks to Blaine R Southam who has turned it into a [pdf book](#).

[Entries](#) [Comments](#)

To search, type and hit enter

RECENT POSTS

-  Video Recording & Slides:
Psychology of Testing at
Wealthfront Engineering
-  RTAC2010: All hands on testing
-  What Flex taught me about data-binding
-  <angular/>: A Radically Different Way of Building AJAX Apps

Flaw #2: Digging into collaborators

- Objects are passed in but never used directly (only used to get access to other objects)
- Law of Demeter violation: method call chain walks an object graph with more than one dot (.)
- Suspicious names: context, environment, principal, container, or manager
- <http://misko.hevery.com/code-reviewers-guide/flaw-digging-into-collaborators/>



Try-Catch

```
public static void viewTable(Connection con) throws SQLException {  
    String query = "select COF_NAME, SUP_ID, PRICE, SALES, TOTAL from COFFEES";  
    try (Statement stmt = con.createStatement()) {  
        ResultSet rs = stmt.executeQuery(query);  
        while (rs.next()) {  
            String coffeeName = rs.getString("COF_NAME");  
            int supplierID = rs.getInt("SUP_ID");  
            float price = rs.getFloat("PRICE");  
            int sales = rs.getInt("SALES");  
            int total = rs.getInt("TOTAL");  
            System.out.println(coffeeName + ", " + supplierID + ", " + price +  
                ", " + sales + ", " + total);  
        }  
    } catch (SQLException e) {  
        JDBCUtilities.printSQLException(e);  
    }  
}
```

Try-With-Resources

```
public List<User> getUser(int userId) {  
    String sql = "SELECT id, username FROM users WHERE id = ?";  
    List<User> users = new ArrayList<>();  
    try (Connection con = DriverManager.getConnection(myConnectionURL);  
        PreparedStatement ps = con.prepareStatement(sql)) {  
        ps.setInt(1, userId);  
        try (ResultSet rs = ps.executeQuery()) {  
            while(rs.next()) {  
                users.add(new User(rs.getInt("id"), rs.getString("name")));  
            }  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    return users;  
}
```

<https://stackoverflow.com/questions/8066501/how-should-i-use-try-with-resources-with-jdbc>



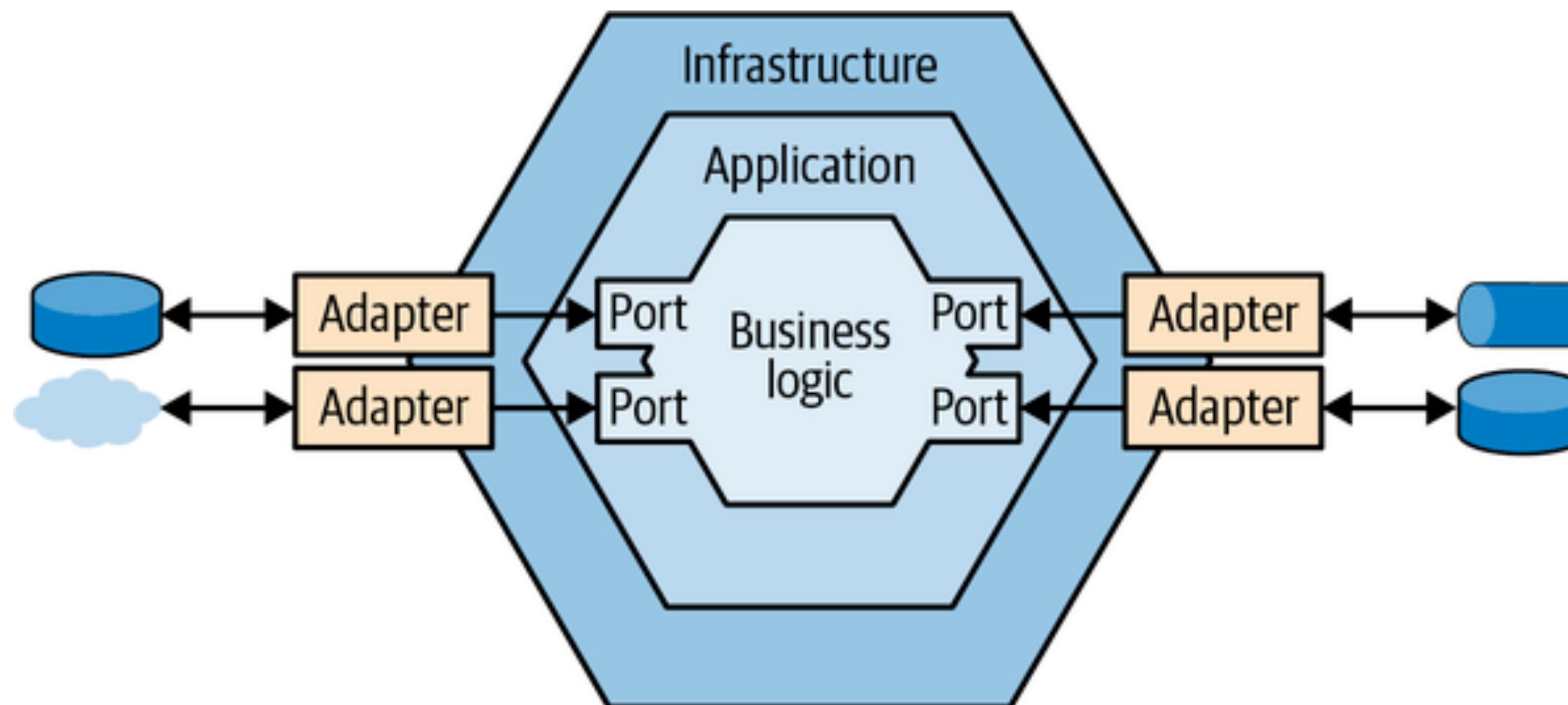
Lambdas did make it
easier, and you can do
some clever tricks

Hexagonal Architecture

Hexagonal Architecture

- Ports and Adapters Architecture
- Decouple the system's business logic from its infrastructural components
- The infrastructure layer implements “adapters”, which are concrete implementations
- Originally conceptualized by Alistair Cockburn
- Compliments Domain Driven Design, which we discuss next
- Separation of Concerns. Domain is encapsulated and interacts through the ports and adapters

Hexagonal Architecture



Learning Domain Driven Design - Vlad Khononov

Interfaces

- Data Access Objects are defined by interfaces from the domain
- Domain Determines what is required, and the DAO should adhere to the contract
- The DAO can have many implementations

London vs. Chicago Style

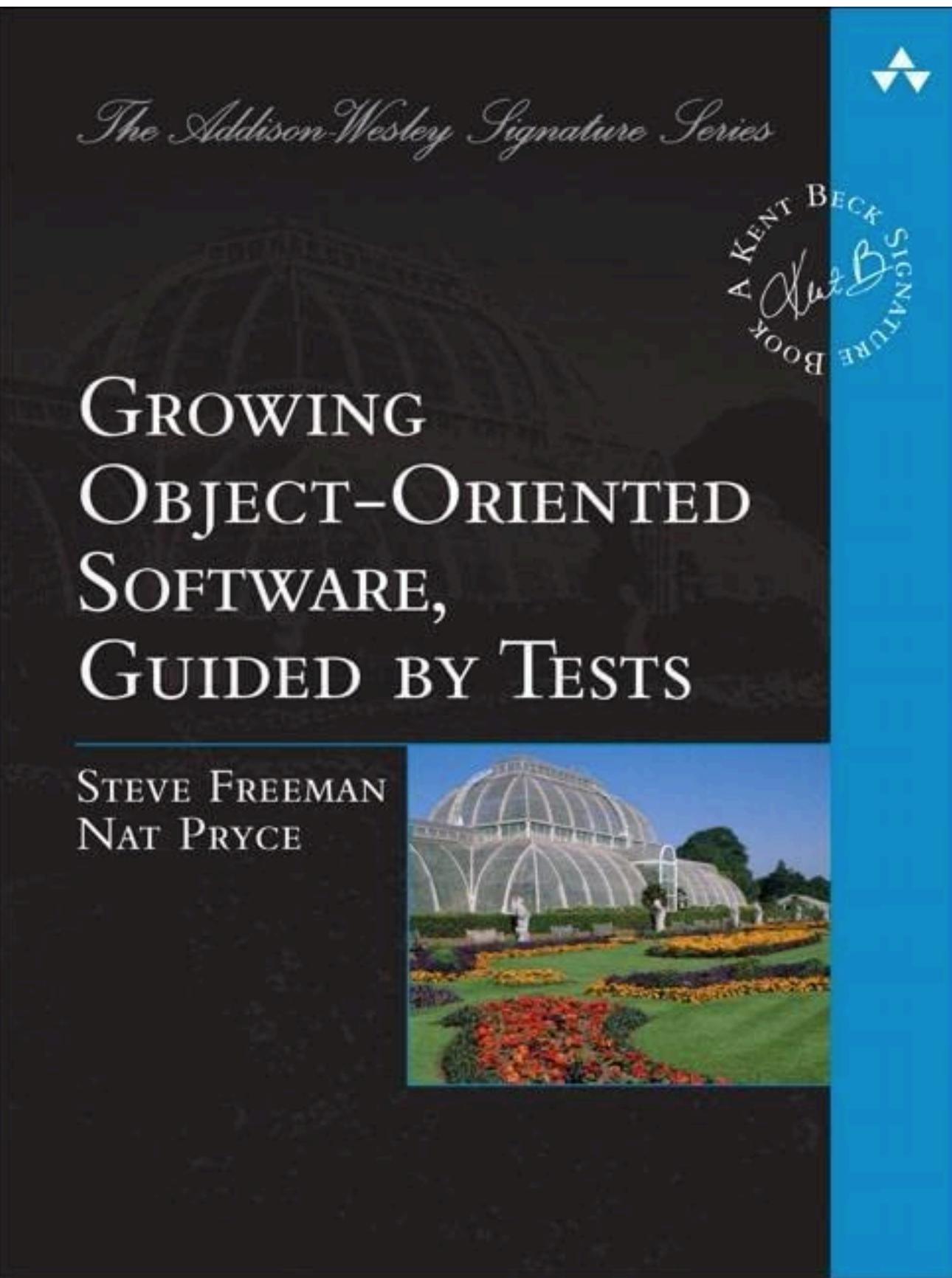
TDD



London vs. Chicago Style TDD

- Chicago TDD - State-based behavior testing and triangulation
- London TDD - Interaction testing, mocking and end-to-end TDD





**“Don’t test code that
you don’t own”**



<https://dbunit.org/>



<https://www.h2database.com>



<https://hsqldb.org/>



Testcontainers

What are test containers?

Test Containers

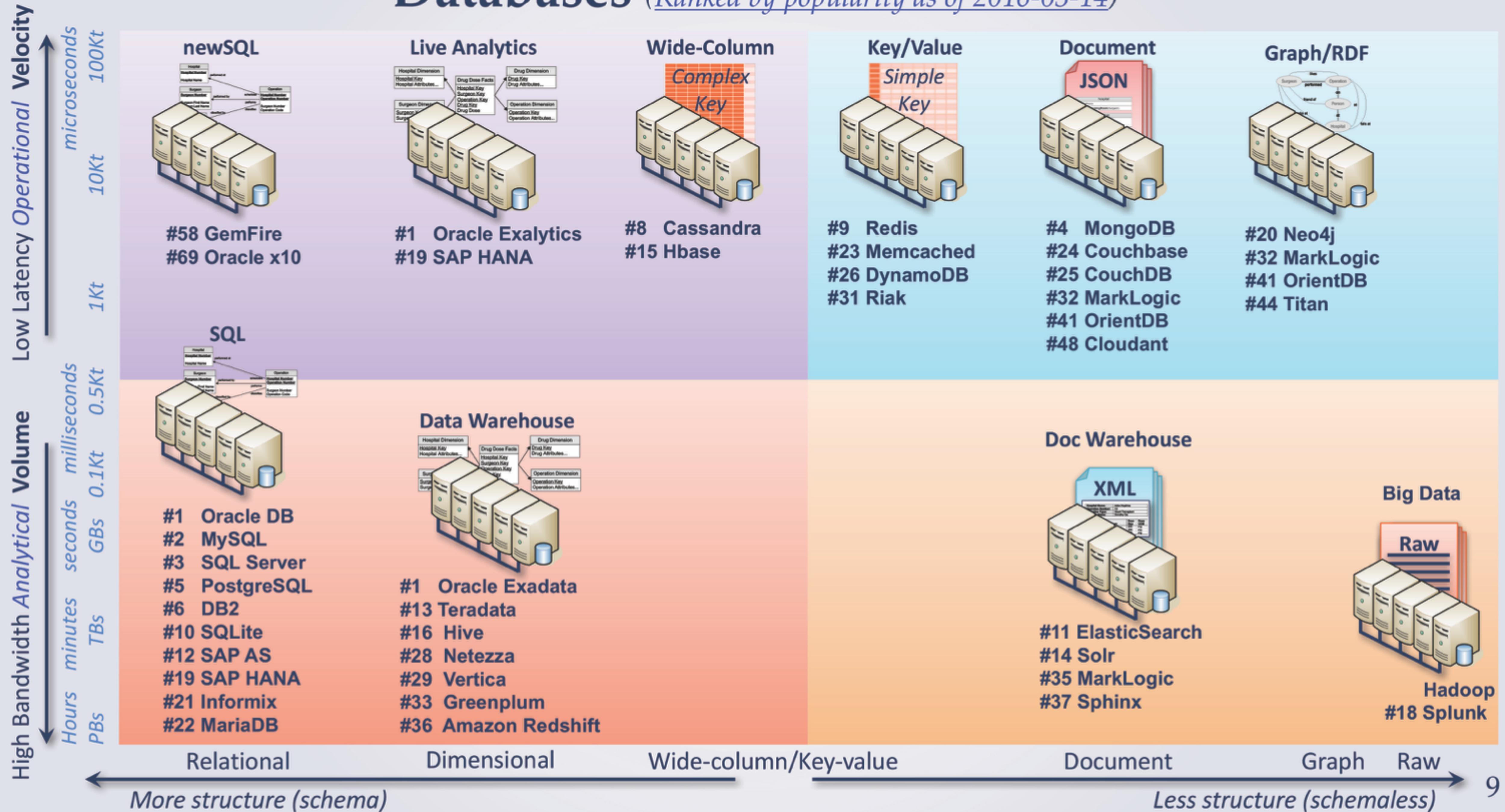
- TestContainers is a Java library that uses containers to perform testing against common systems
 - <https://www.testcontainers.org/>
 - Supports JUnit, Spock testing
 - Lightweight
 - Ephemeral
 - Docker-based, Possible with Podman
 - Test against the databases that you actually use in productions

What about discovery?

**Sometimes you don't
know the database you
are programming against.**

Databases

(Ranked by popularity as of 2016-03-14)

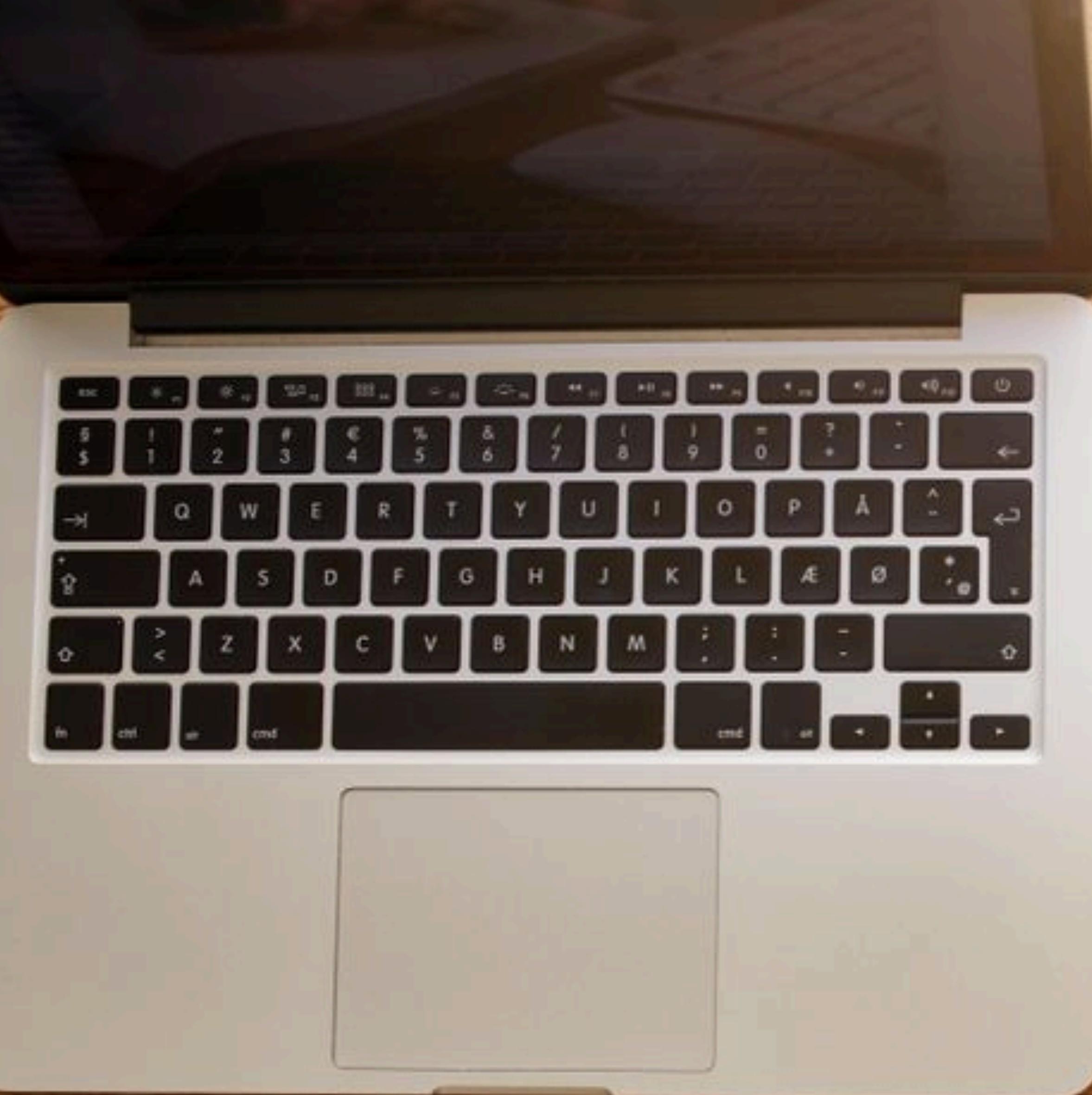




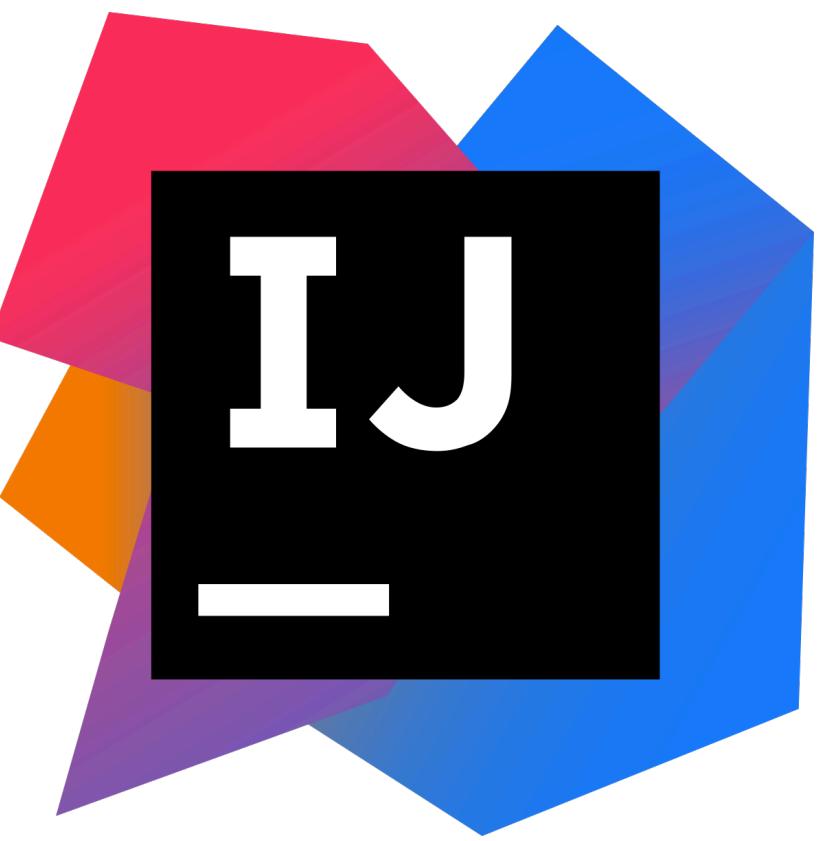
Google Cloud

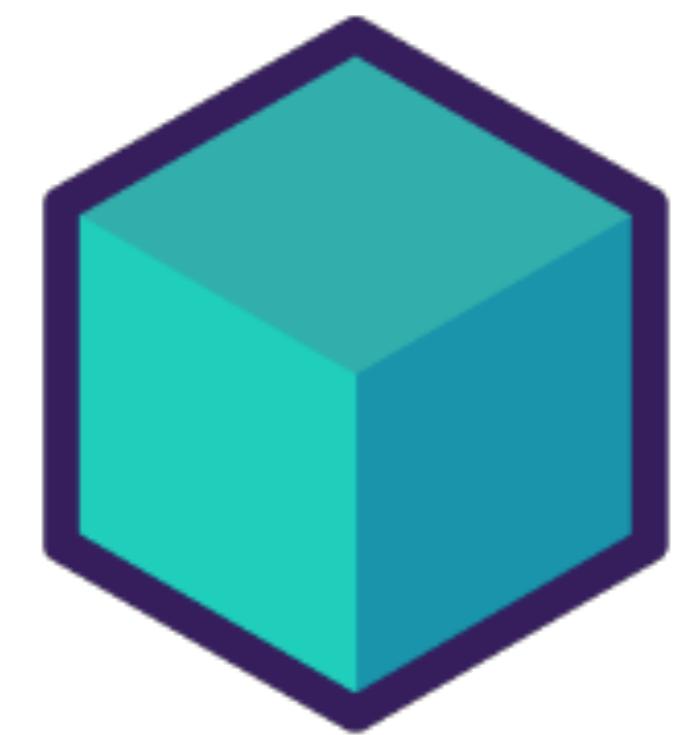
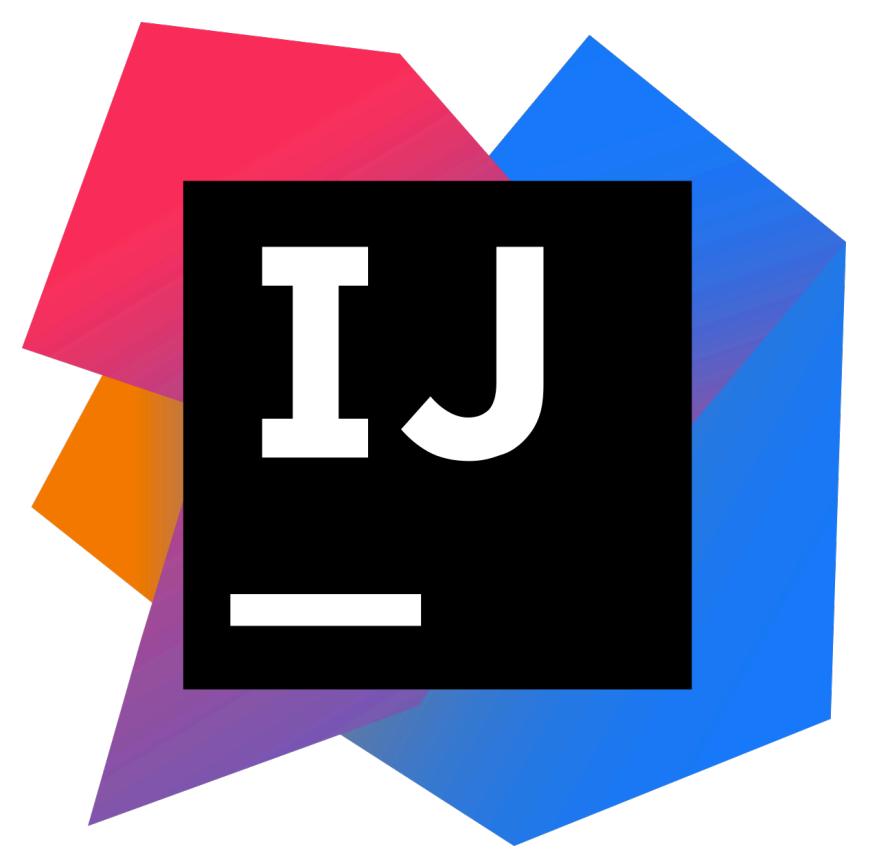


<https://github.com/dhinojosa/language-matrix>



Refactoring Skills with







Key Command	What it does
^ T	Invoke Refactor This (Mac)
Ctrl+Alt+Shift+T	Invoke Refactor This (Windows)
\ \⌘ M	Extract Method (Mac)
Ctrl+Alt+M	Extract Method (Windows)
\ \⌘ P	Extract Parameters (Mac)
Ctrl+Alt+P	Extract Parameter (Windows)



Let's try it!