JAVA
USER
GROUP
CH

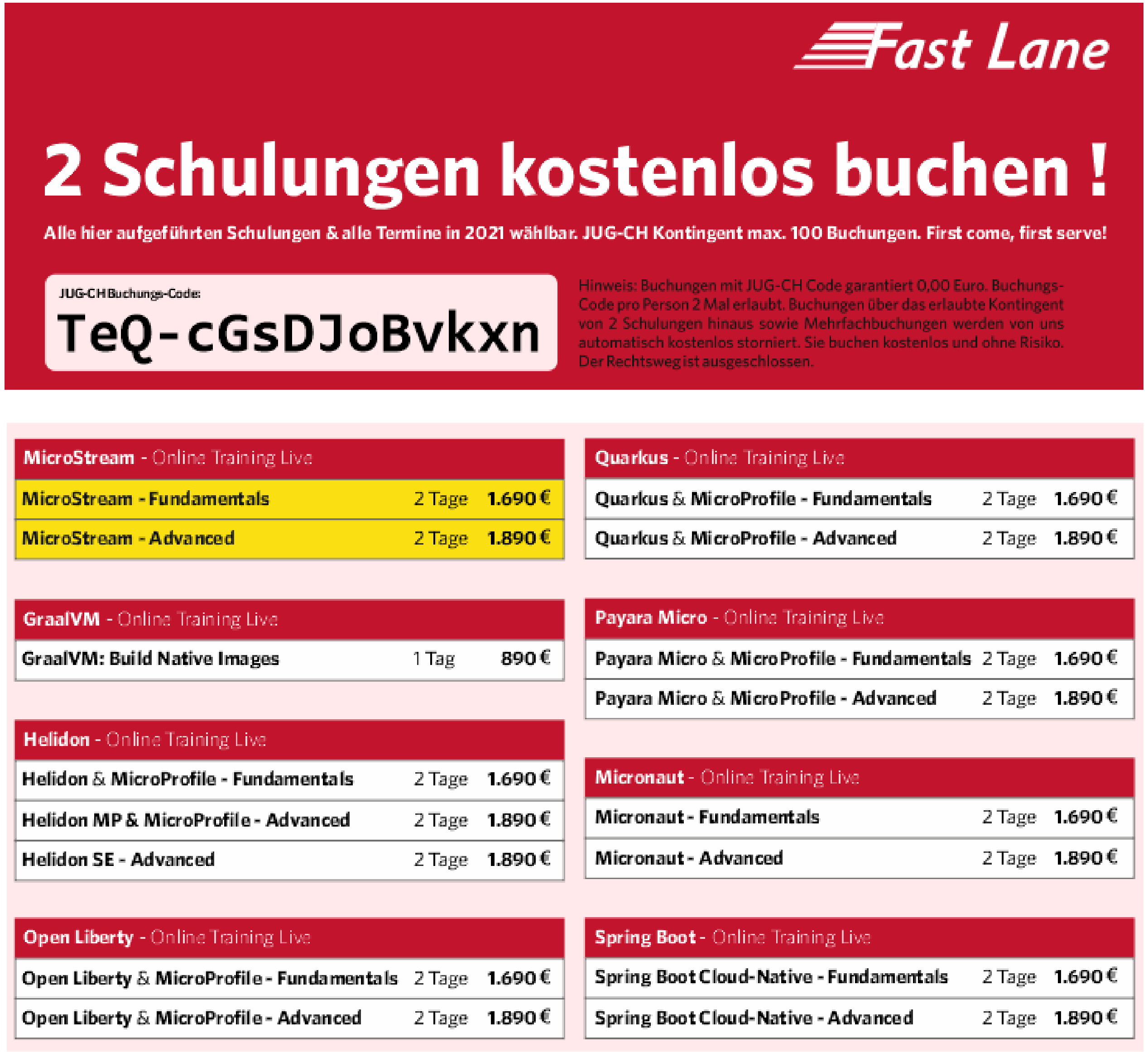
# ENTWICKLE IN-MEMORY DATENBANK-APPLIKATIONEN & MICROSERVICES MIT JAVA UND MICROSTREAM

MARKUS KETT & CHRISTIAN KUEMMEL

MicroStream

@MarkusKett markuskett

JAVA
USER
GROUP
CH



Fast Lane

# 2 Schulungen kostenlos buchen !

**Alle hier aufgeführten Schulungen & alle Termine in 2021 wählbar. JUG-CH Kontingent max. 100 Buchungen. First come, first serve!**

JUG-CH Buchungs-Code:

**TeQ-cGsDJoBvkxn**

Hinweis: Buchungen mit JUG-CH Code garantiert 0,00 Euro. Buchungs-Code pro Person 2 Mal erlaubt. Buchungen über das erlaubte Kontingent von 2 Schulungen hinaus sowie Mehrfachbuchungen werden von uns automatisch kostenlos storniert. Sie buchen kostenlos und ohne Risiko. Der Rechtsweg ist ausgeschlossen.

| **MicroStream** - Online Training Live | | |
|---|---|---|
| MicroStream - Fundamentals | 2 Tage | 1.690 € |
| MicroStream - Advanced | 2 Tage | 1.890 € |

| **GraalVM** - Online Training Live | | |
|---|---|---|
| GraalVM: Build Native Images | 1 Tag | 890 € |

| **Helidon** - Online Training Live | | |
|---|---|---|
| Helidon & MicroProfile - Fundamentals | 2 Tage | 1.690 € |
| Helidon MP & MicroProfile - Advanced | 2 Tage | 1.890 € |
| Helidon SE - Advanced | 2 Tage | 1.890 € |

| **Open Liberty** - Online Training Live | | |
|---|---|---|
| Open Liberty & MicroProfile - Fundamentals | 2 Tage | 1.690 € |
| Open Liberty & MicroProfile - Advanced | 2 Tage | 1.890 € |

| **Quarkus** - Online Training Live | | |
|---|---|---|
| Quarkus & MicroProfile - Fundamentals | 2 Tage | 1.690 € |
| Quarkus & MicroProfile - Advanced | 2 Tage | 1.890 € |

| **Payara Micro** - Online Training Live | | |
|---|---|---|
| Payara Micro & MicroProfile - Fundamentals | 2 Tage | 1.690 € |
| Payara Micro & MicroProfile - Advanced | 2 Tage | 1.890 € |

| **Micronaut** - Online Training Live | | |
|---|---|---|
| Micronaut - Fundamentals | 2 Tage | 1.690 € |
| Micronaut - Advanced | 2 Tage | 1.890 € |

| **Spring Boot** - Online Training Live | | |
|---|---|---|
| Spring Boot Cloud-Native - Fundamentals | 2 Tage | 1.690 € |
| Spring Boot Cloud-Native - Advanced | 2 Tage | 1.890 € |

**JUG-CH CODE:**

**TeQ-cGsDJoBvkxn**

**FRAGEN ZUR AKTION:**

**Sebastian Bock**
**m.bock@microstream.one**

**TERMINE & BUCHUNG:**

**www.flane.de/microservice-development**

## Kostenlos für alle JUG-Mitglieder!

1.000 JUG-Tickets insgesamt verfügbar. First come, first server!

Jetzt gleich Ticket sichern: www.jcon.one

# About us

@MarkusKett

markuskett

- Markus Kett, CEO
- Living in Weiden, Germany
- Grew up with Atari & C64
- Building Java developer tools since 2001 – Xpage, XDEV IDE, JPA-SQL, RapidClipse
- RapidClipse Contributor
- External Editor in Chief, JAVAPRO Magazine

- Christian Kümmel, Senior Developer Advocate, Project Lead RapidClipse, Project Manager
- Living in Weiden, Germany
- Building Java developer tools since 2001 – Xpage, XDEV IDE, JPA-SQL, RapidClipse

# Agenda

1. What‘s the problem with Java today?
2. Java is changing
3. What‘s the problem with data storage in Java?
4. Building ultra-fast in-memory database applications with pure Java and MircroStream
5. Live-Demo
6. Some Code
7. Q&A

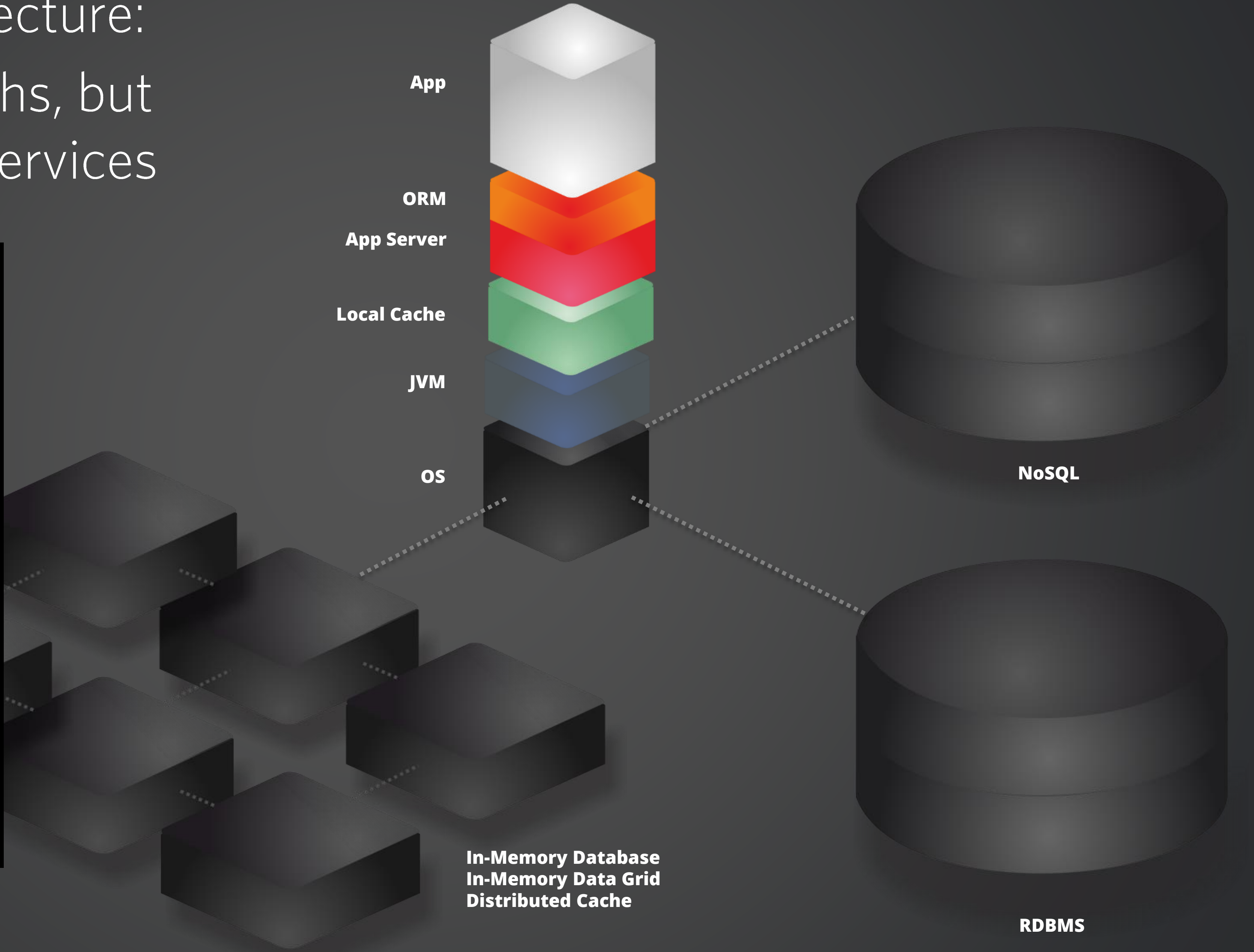
# Java Standard Architecture: Designed for Monoliths, but not Suited for Microservices

What's the Problem?

- Slow JVM startup time
- High JVM memory footprint
- Many layers, complex architecture
- One central data storage for everything
- Database performance

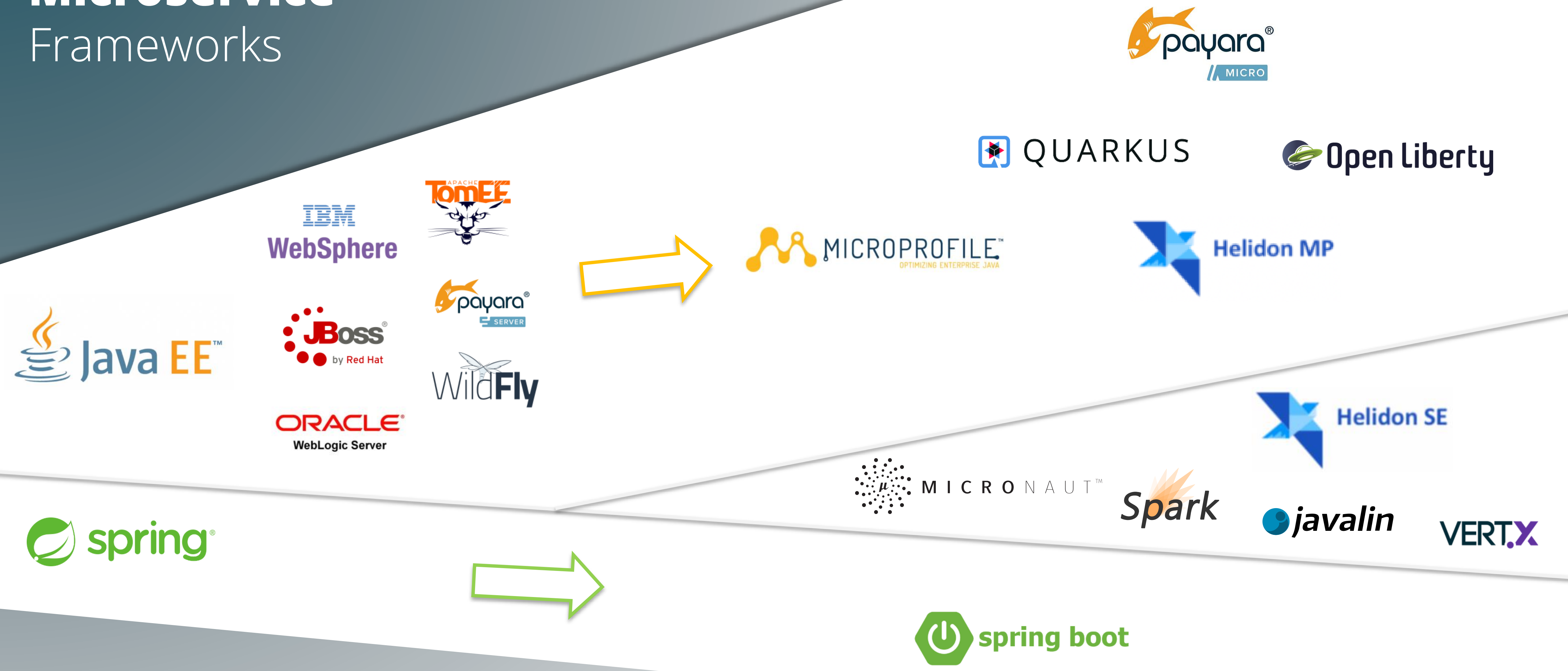
Some Requirements for Microservices:

- Responsible for only one specific task
- Small size
- Low memory Footprint
- Low startup time
- Deployable as container
- Stateless & Scaleable
- Fast to serve requests



MicroStream

# From **Full Stack to Microservice** Frameworks



Java EE™

IBM WebSphere

Apache TomEE

JBoss by Red Hat

payara® SERVER

ORACLE® WebLogic Server

WildFly

MICROPROFILE™ OPTIMIZING ENTERPRISE JAVA

Helidon MP

QUARKUS

Open Liberty

payara® MICRO

spring®

spring boot

MICRONAUT™

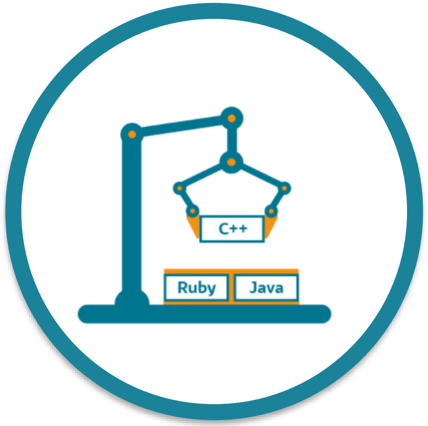
Spark

javalin

VERT.X

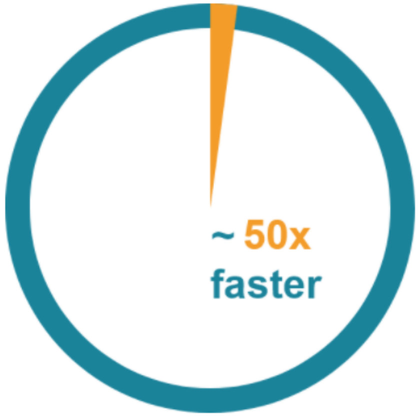
Helidon SE

# GraalVM™

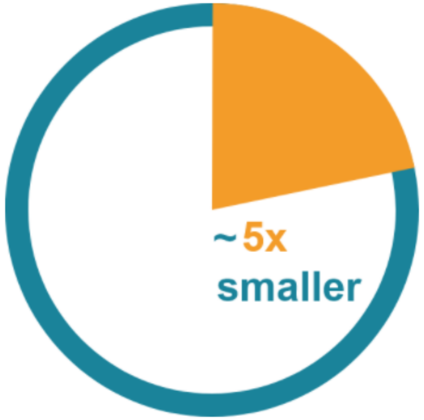


Polyglot Programming

Native Images

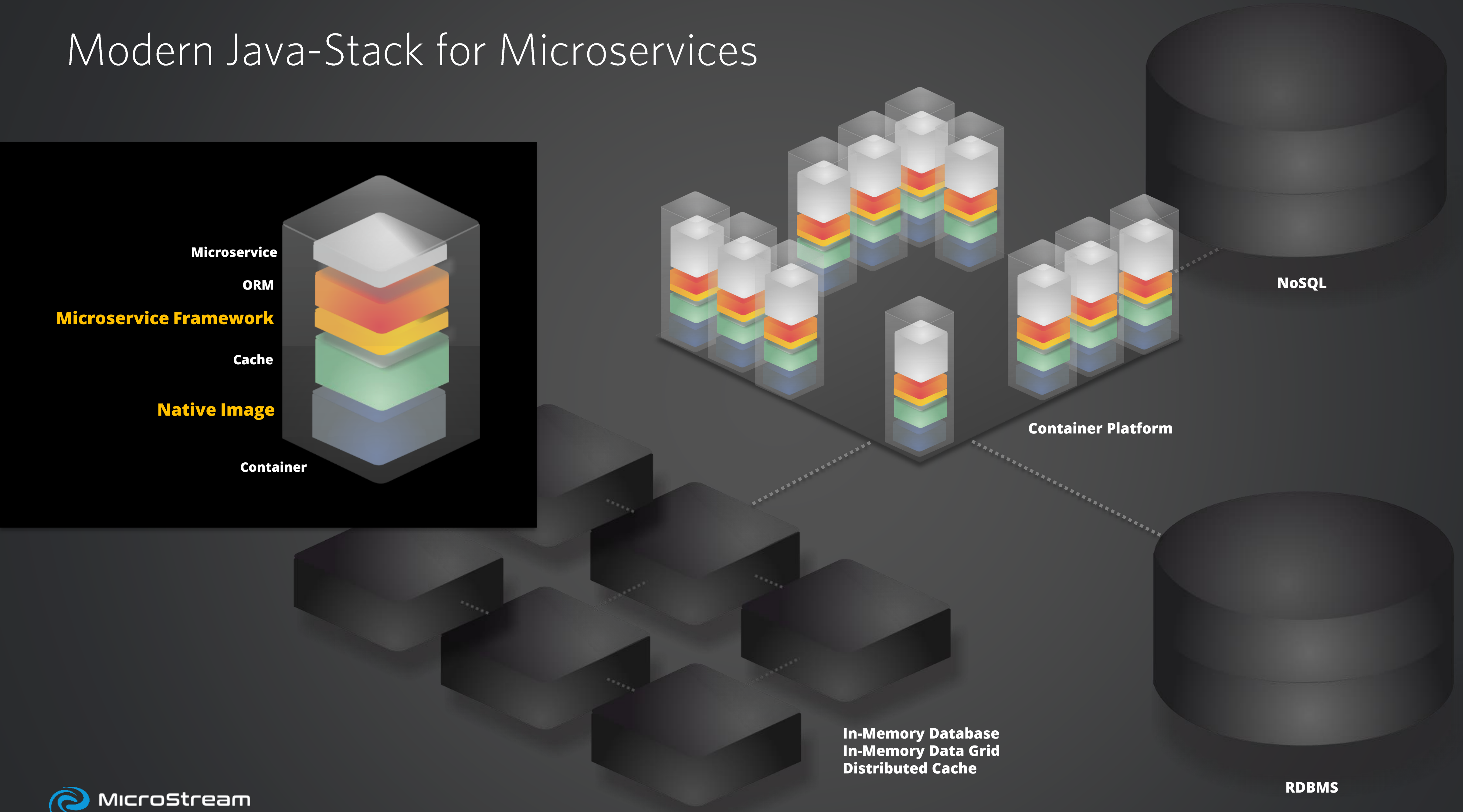


Startup Time

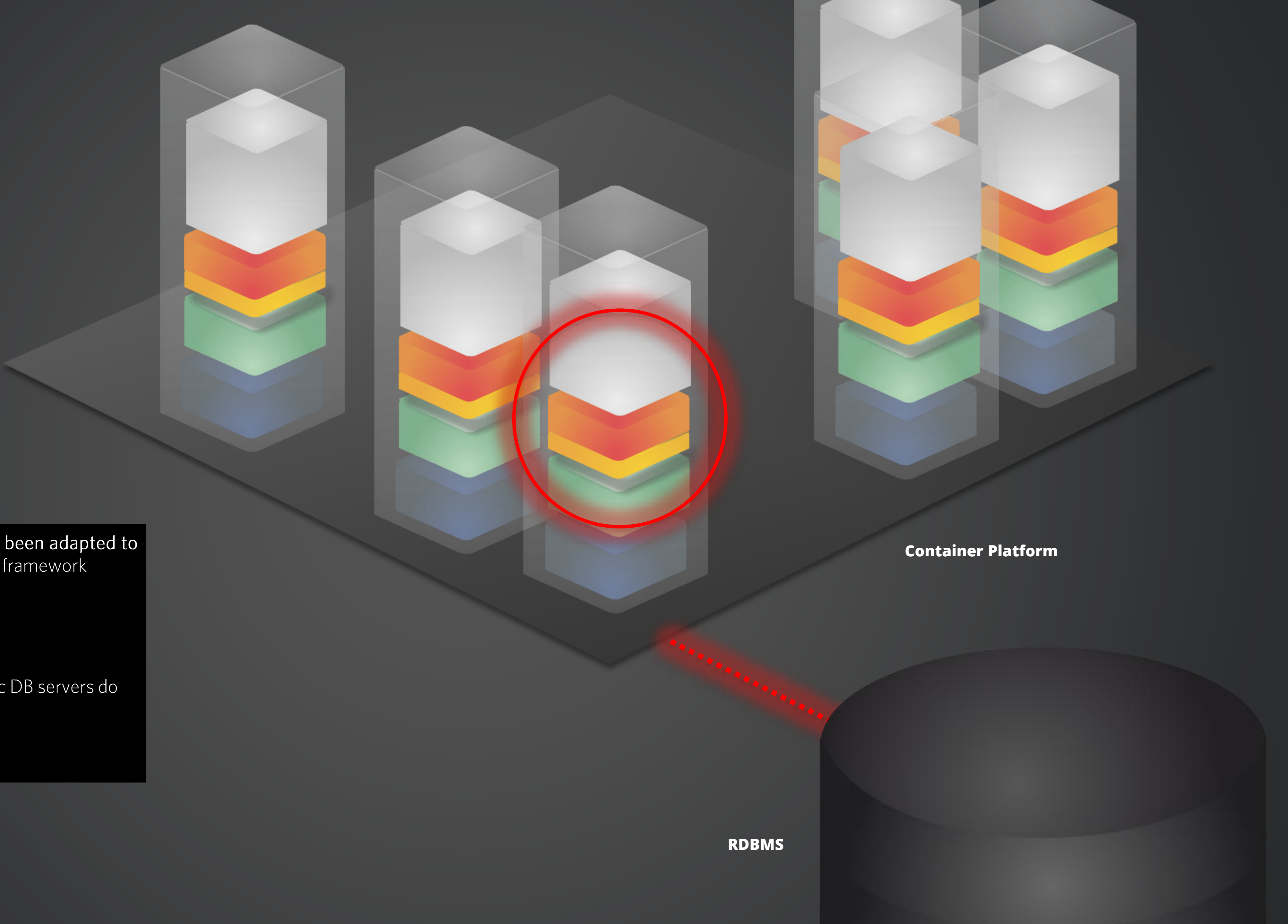


Memory Footprint

# Modern Java-Stack for Microservices



# What's about Persistence?

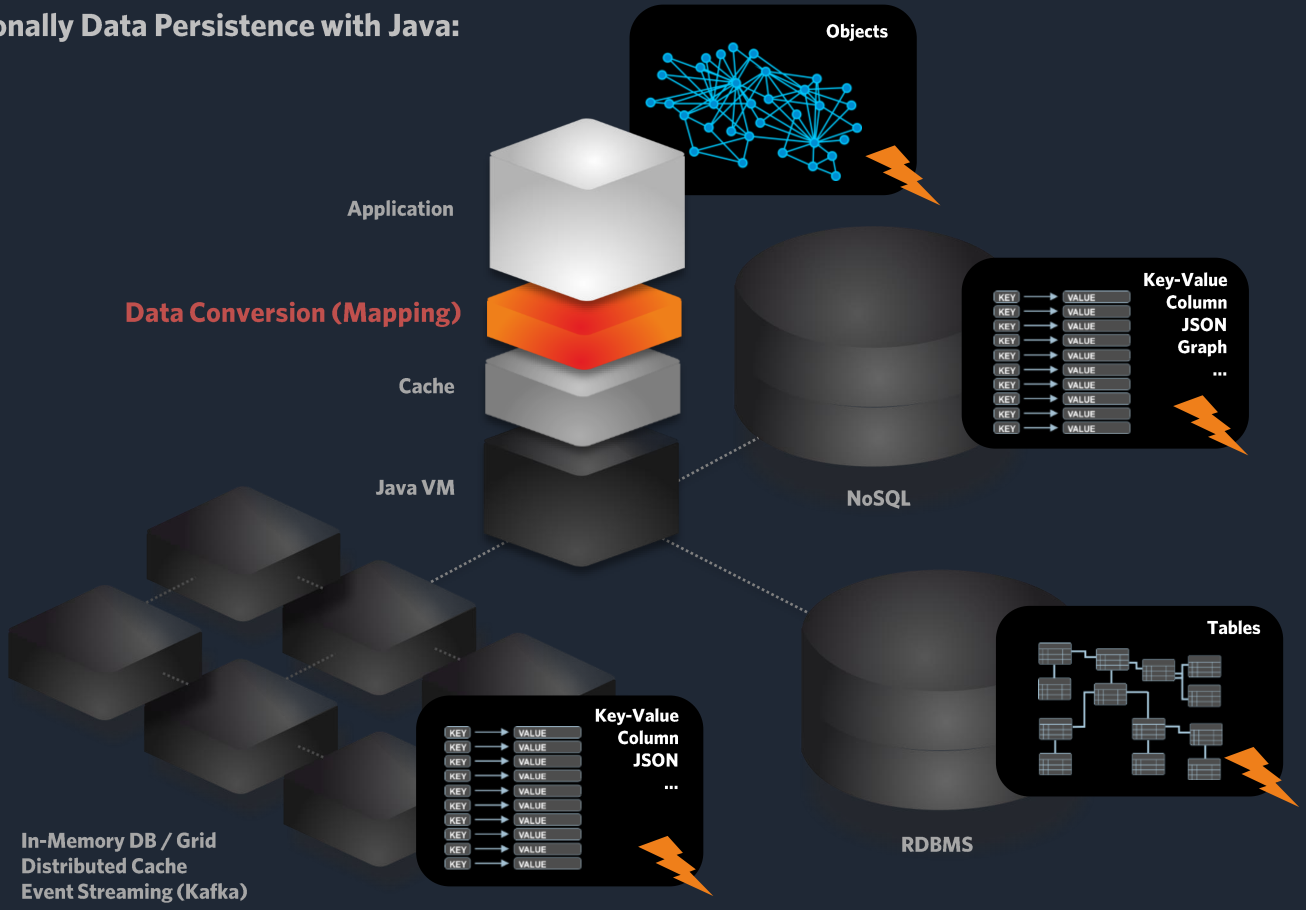


The Persistence strategy has not been adapted to microservicesHigh complex ORM framework required

- Complex architecture
- Expensive latencies
- Microservices and monolithic DB servers do not fit together
- Mostly too low performance

# Data Conversions For Handling Incompatible Data Structures Kill Your Performance and Let Your Costs Explode

Traditionally Data Persistence with Java:



Challenge: Storing Objects into Tables / JSON / Key Value Stores / Graphs

**Data Conversion Through Every Single Read & Write !**

- Requires lots of vCPU power
- Performance killer no. 1
- Complex architecture
- Expensive development process
- High network latencies
- Performance issus require huge cluster infrastructure
- Infrastructure costs will explode

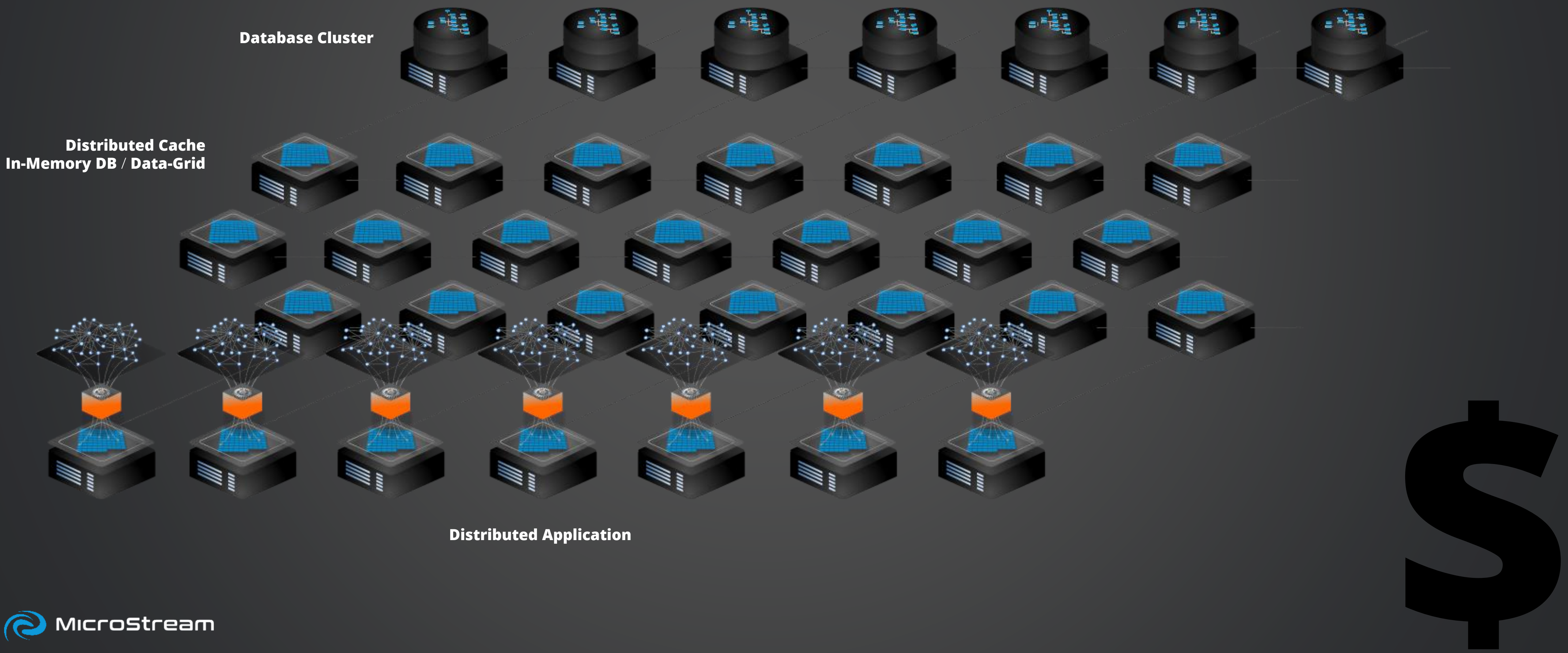
# Impedance Mismatch

- Granularity mismatch
- Inherintance mismatch
- Identity mismatch
- Associations mismatch
- Data navigation mismatch
- Data type mismatch

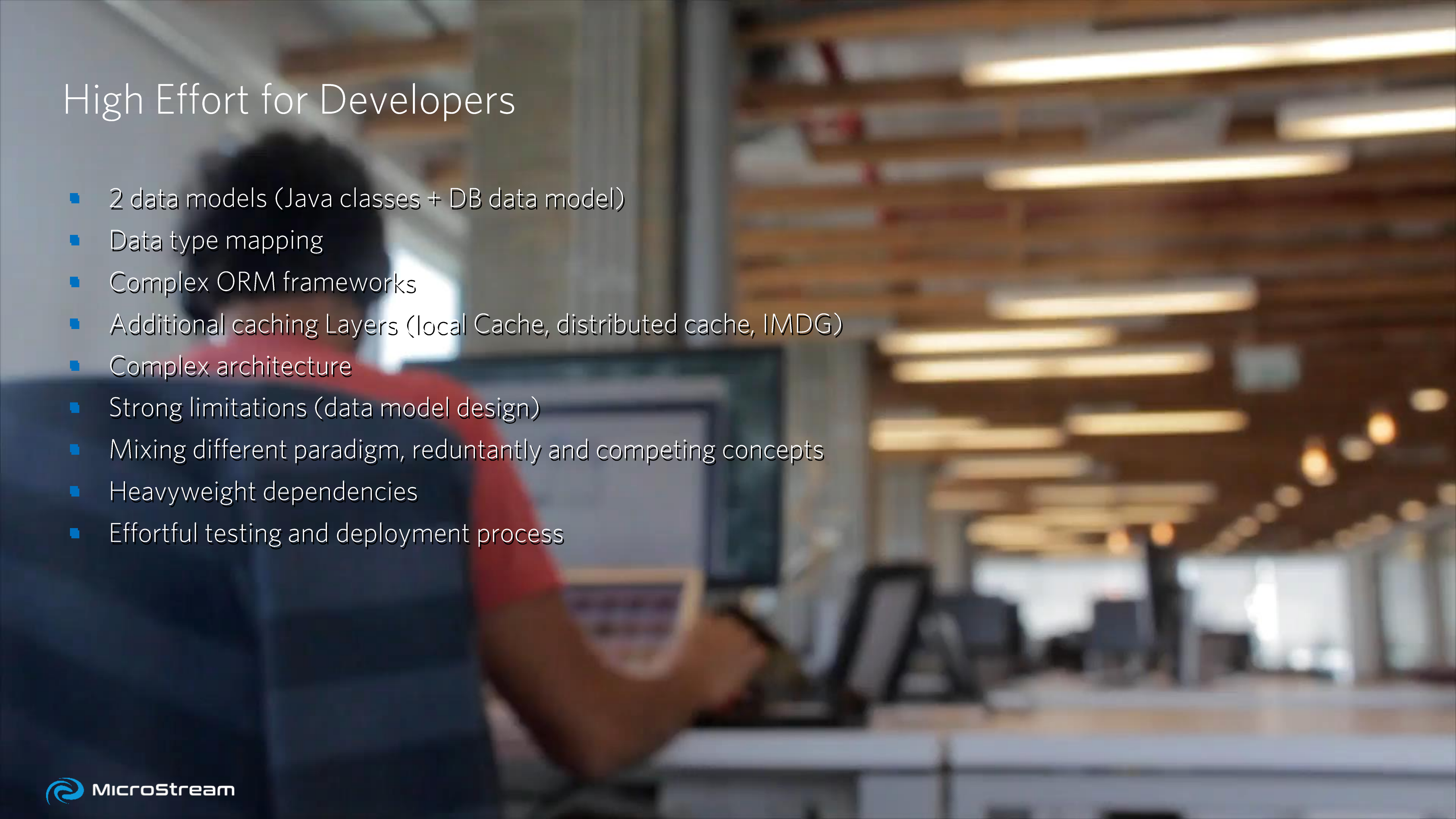
WIKIPEDIA
The Free Encyclopedia

There are various solutions, but they are only a **more or less elegant way around the problem. No matter which solution you choose** - as long as the systems are different, every developer will **sooner or later** get to the point where his solution **no longer** meets one or more of the following points: **Maintainability, performance, intelligibility**.

# Today's Database Application Infrastructures are Highly Inefficient, Complex, and Expensive



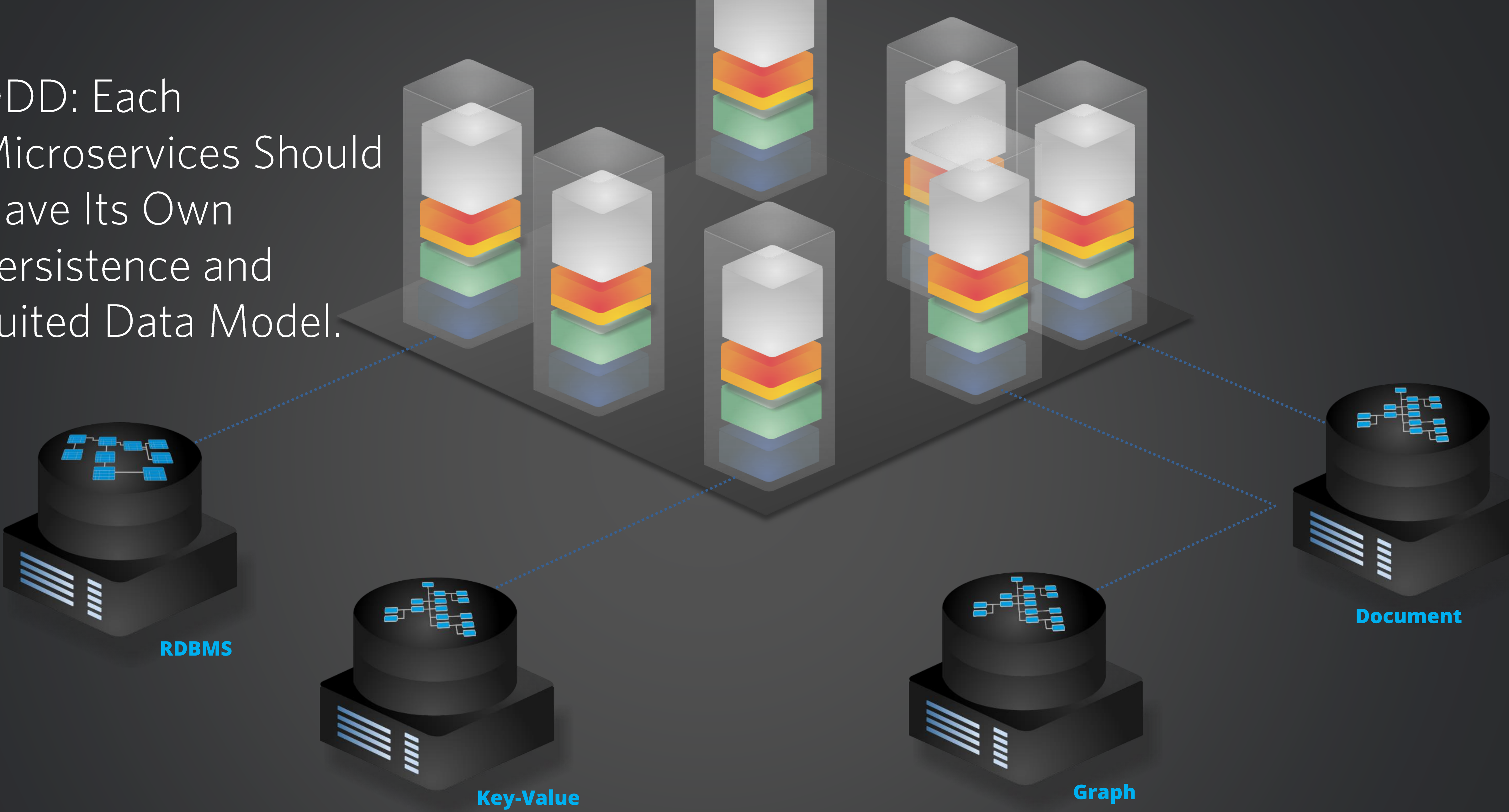
MicroStream



# High Effort for Developers

- 2 data models (Java classes + DB data model)
- Data type mapping
- Complex ORM frameworks
- Additional caching Layers (local Cache, distributed cache, IMDG)
- Complex architecture
- Strong limitations (data model design)
- Mixing different paradigm, reduntantly and competing concepts
- Heavyweight dependencies
- Effortful testing and deployment process

# DDD: Each Microservices Should Have Its Own Persistence and Suited Data Model.



MicroStream

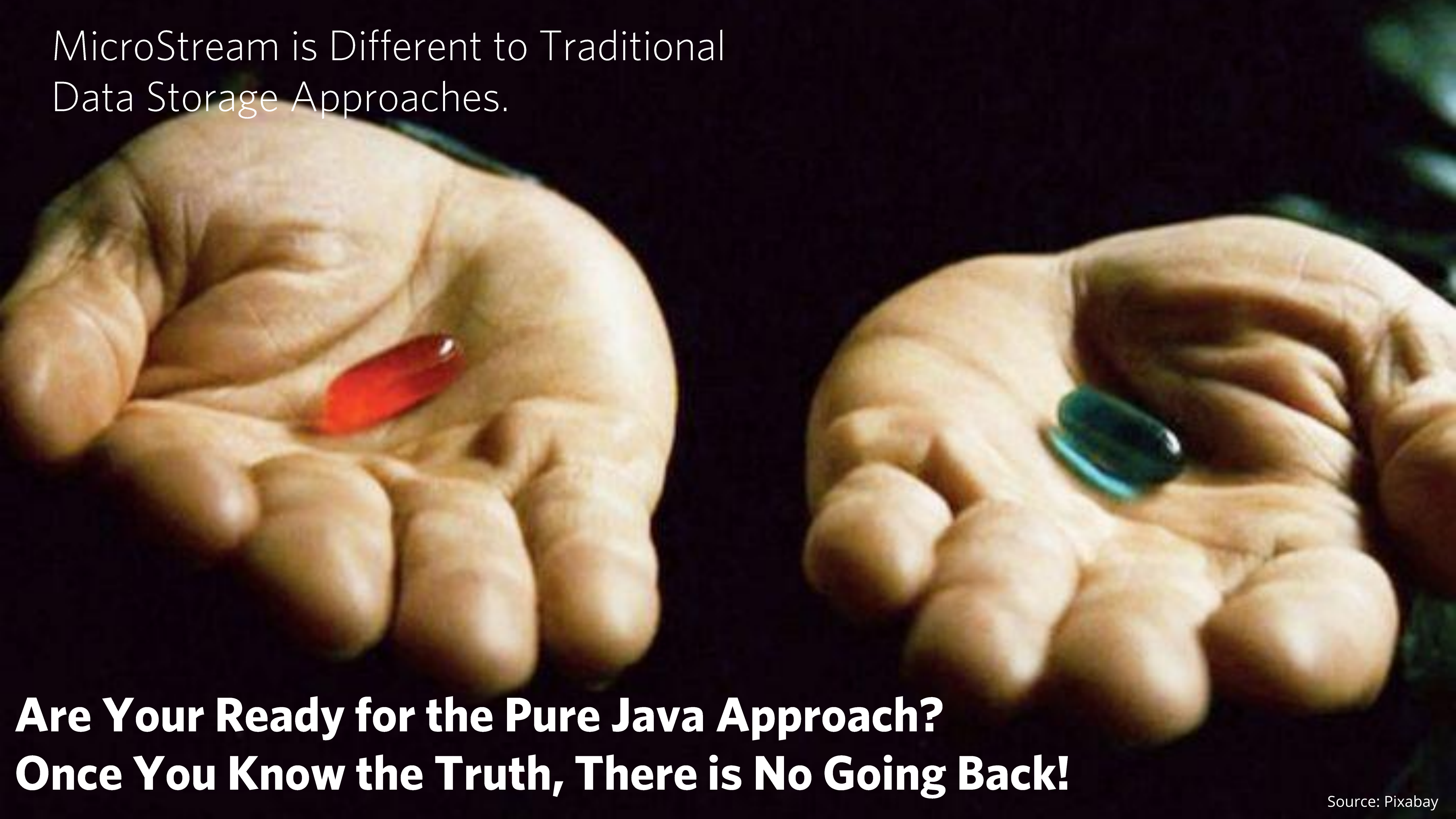
# MicroStream

## High-Performance Persistence

Store any Java Object Graph and Subgraphs into any Database.
Microsecond Query Time. Ultra-high Throughput. Minimum of Latencies.
Create Ultra-fast In-Memory Database Applications and Microservices.

**www.microstream.one**

# MicroStream is Different to Traditional Data Storage Approaches.

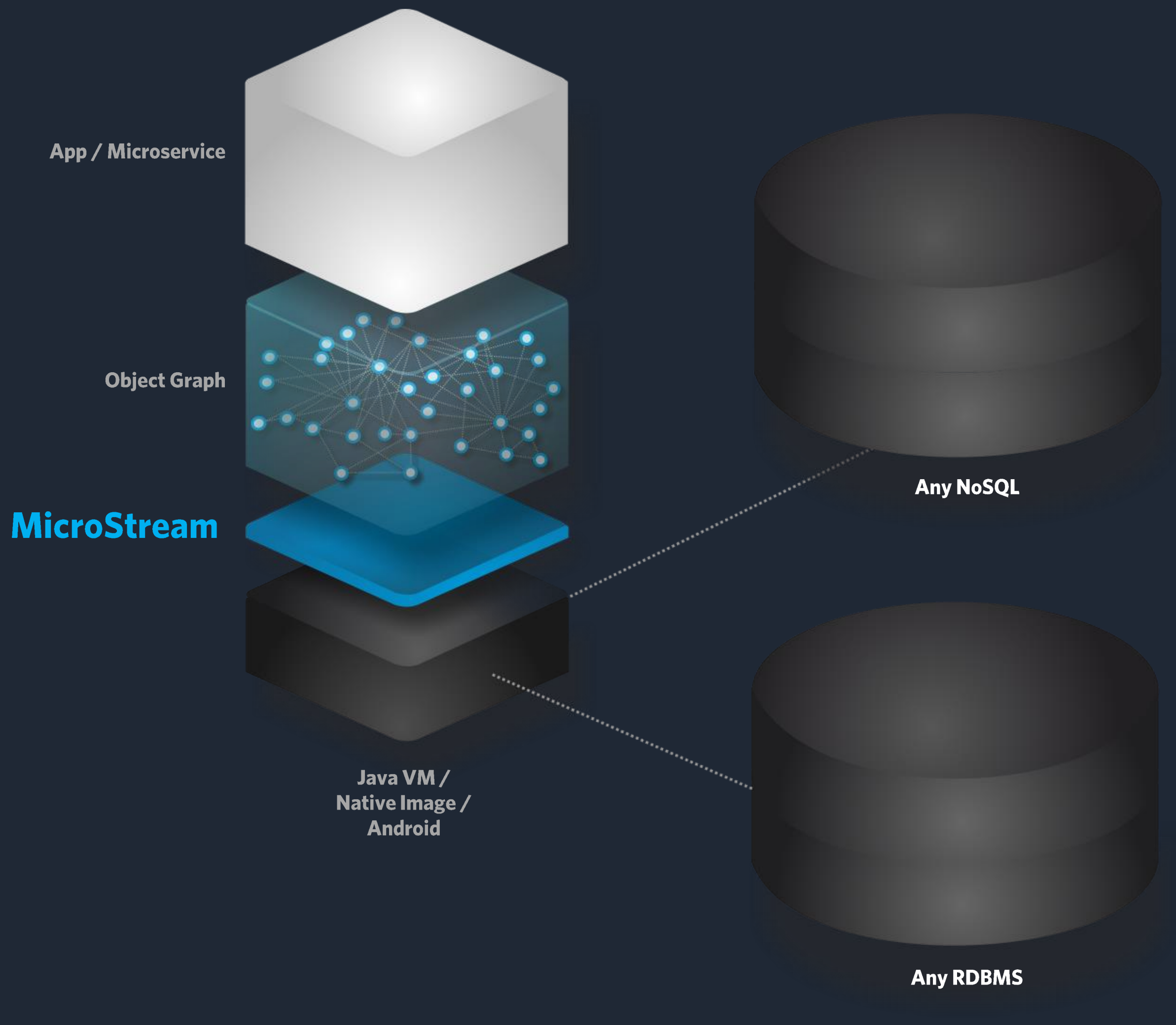


**Are Your Ready for the Pure Java Approach?**
**Once You Know the Truth, There is No Going Back!**

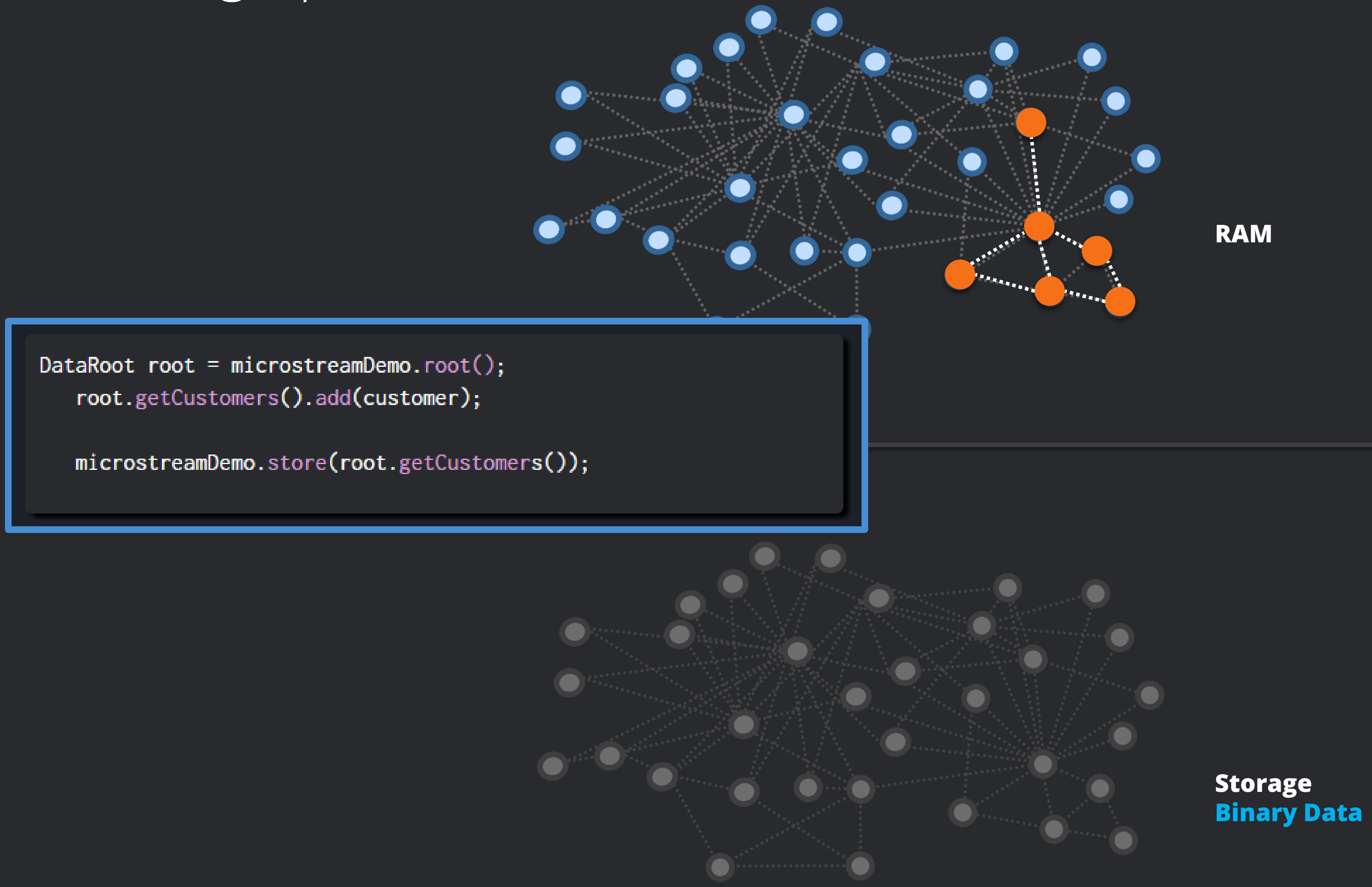
Source: Pixabay

# MicroStream Turns Your App Into a High-Performance In-Memory Data Processing Reactor

- **Object graph: multi-model data structure that supports any Java type**
- **Data model: Java classes only – database-specific data models are not needed at all**
- **Query language: searching object graphs in-memory with Java Streams or GraphQL**
- **Incredible in-memory high-performance enables microsecond query time and gigantic throughput**
- **Pure Java, fully object-oriented, typesafe, elegant programming model**
- **MicroStream: persist any object graph into any storage solution**



# Storing Objects and Subgraphs



```
DataRoot root = microstreamDemo.root();
   root.getCustomers().add(customer);

   microstreamDemo.store(root.getCustomers());
```

- Store any single Object or Subgraph explicitly
- Using inheritance is Trouble-free
- Strong Consistency
- Store any Java Type
- ACID Transactions
- High Secure
- Custom-tailored Type Handling for best Performance
- Multithreaded super-fast Write Operations
- Covers 3 CRUD Operations: Create, Update & Delete

MicroStream

# Loading Objects and Subgraphs Lazy

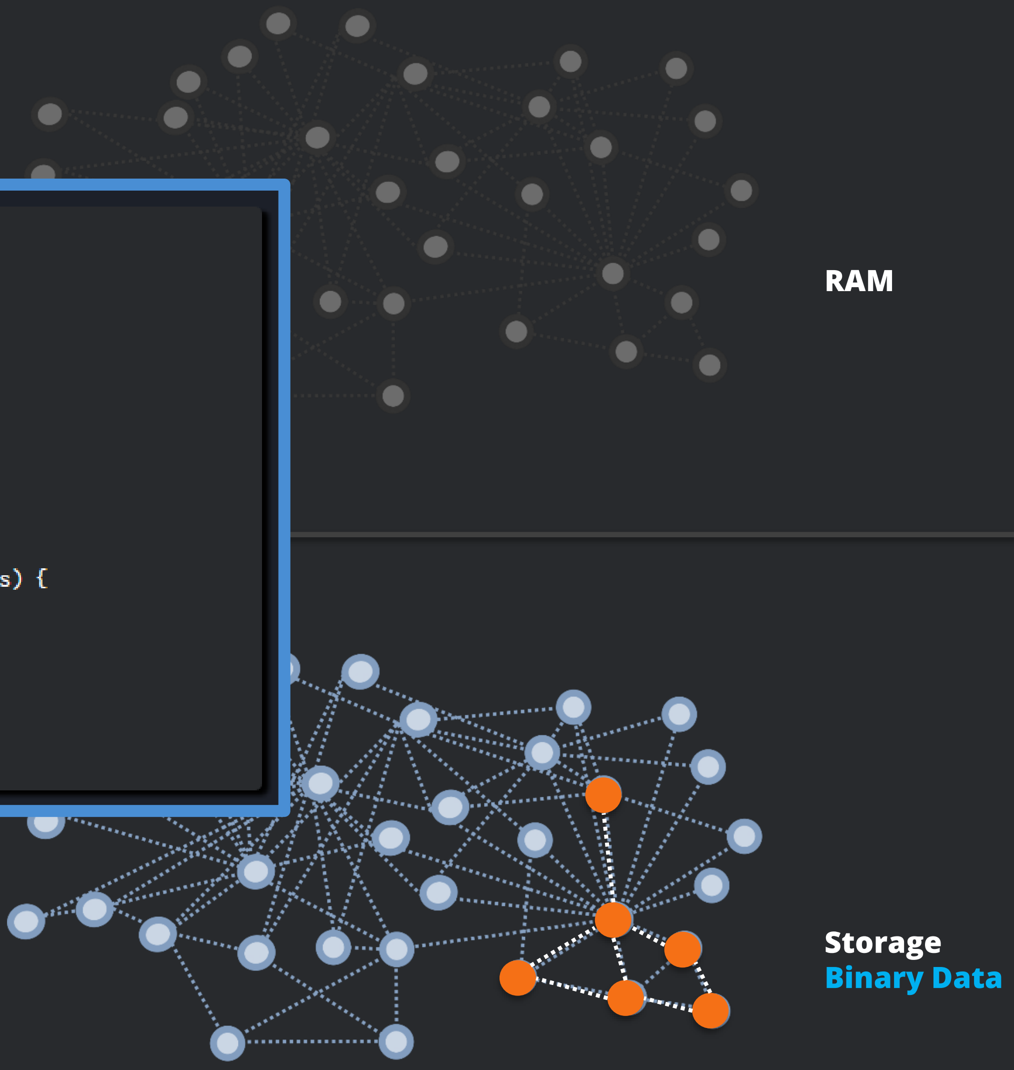
```java
public class Customer {

  ...
  private Lazy<Set<Order>> orders;
  ...

  public Set<Order> getOrders() {
     return Lazy.get(this.orders);
  }

  public void setOrders(final Set<Order> orders) {
     this.orders = Lazy.Reference(orders);
  }

  ...
}
```

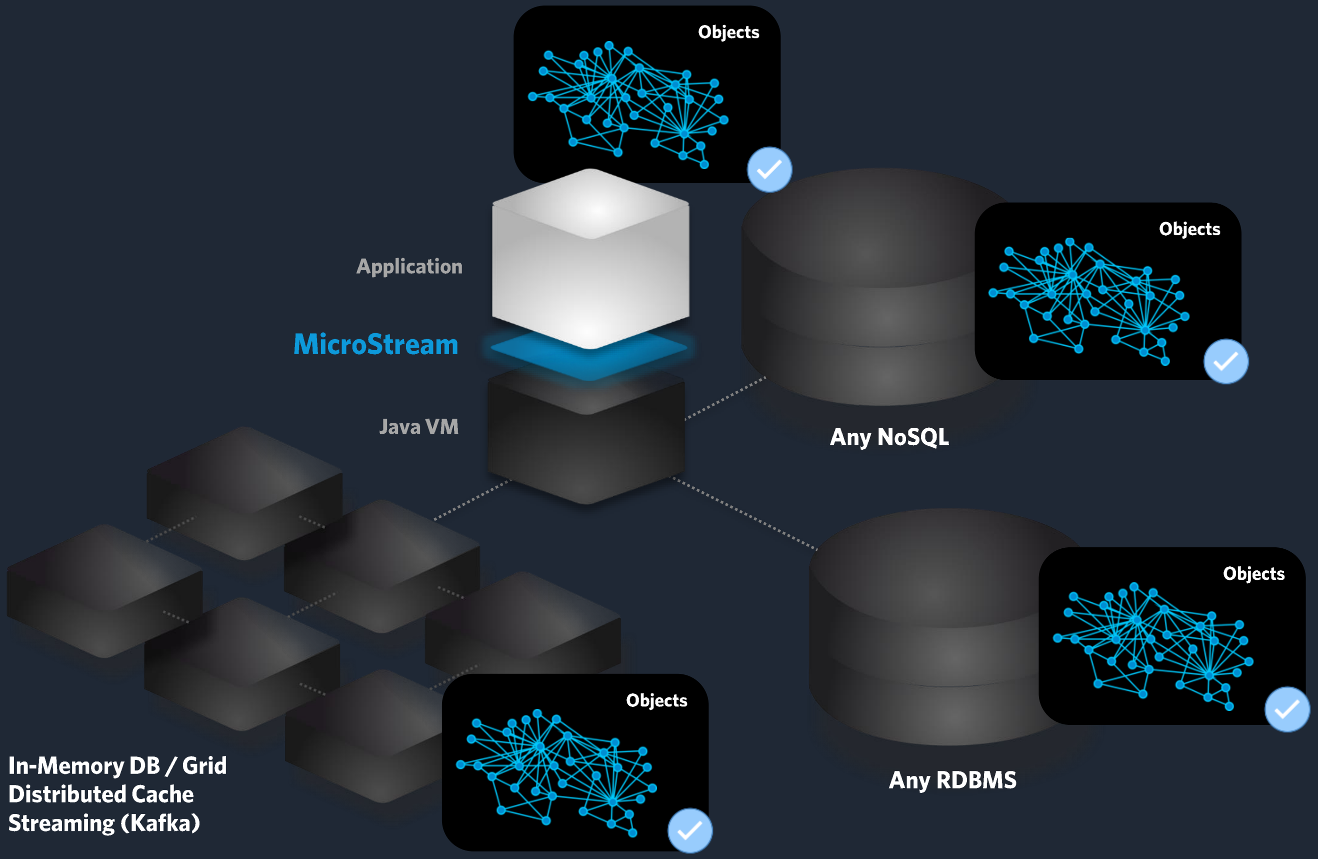


- Load any single Subgraph on Demand
- No more classic Selects, simply call Getter
- Loaded Objects are merged into the Object Graph
- Memory Management by MicroStream or explicit
- No inconvenient Object Copies
- Multithreaded super-fast Read Operations

MicroStream

# Accelerating Database Queries 1000x and Saving up to 92% Costs

MicroStream Persistence:



**Streaming Objects**
**Directly Into the Database**

**Conversion Eliminated !**

- **Simple architecture**
- **Faster time to market**
- **Saves lots of vCPU power**
- **Minimizes latencies**
- **In-memory queries executed in microseconds**
- **Saves up to 92% costs of infrastructure**

# Supported Storages

ORACLE DATABASE | ORACLE TIMESTEN | MySQL | PostgreSQL | MariaDB | amazon DynamoDB | SQLite

NoSQL

kafka | SAP HANA | ORACLE NOSQL DATABASE | ORACLE Coherence | hazelcast | Experimental mongoDB | Experimental redis

Cloud Object Store

amazon web services S3 | Microsoft Azure Blob Storage | ORACLE CLOUD

# More Core Features

### Storage Garbage Collector

Legacy and corrupt data are removed from the storage by MicroStream automatically.

### Class Version Handling

Different versions of your classes are handled automatically. Custom mappings are also supported.

### Backups

Reliable and fully individual configurable data backup processes.

### REST Interface

REST interface allows remote access to the storage data.

### Storage Browser

Graphical user interface for browsing through your storage data remotely.

### Highly Configurable

MicroStream is highly configurable through its API.

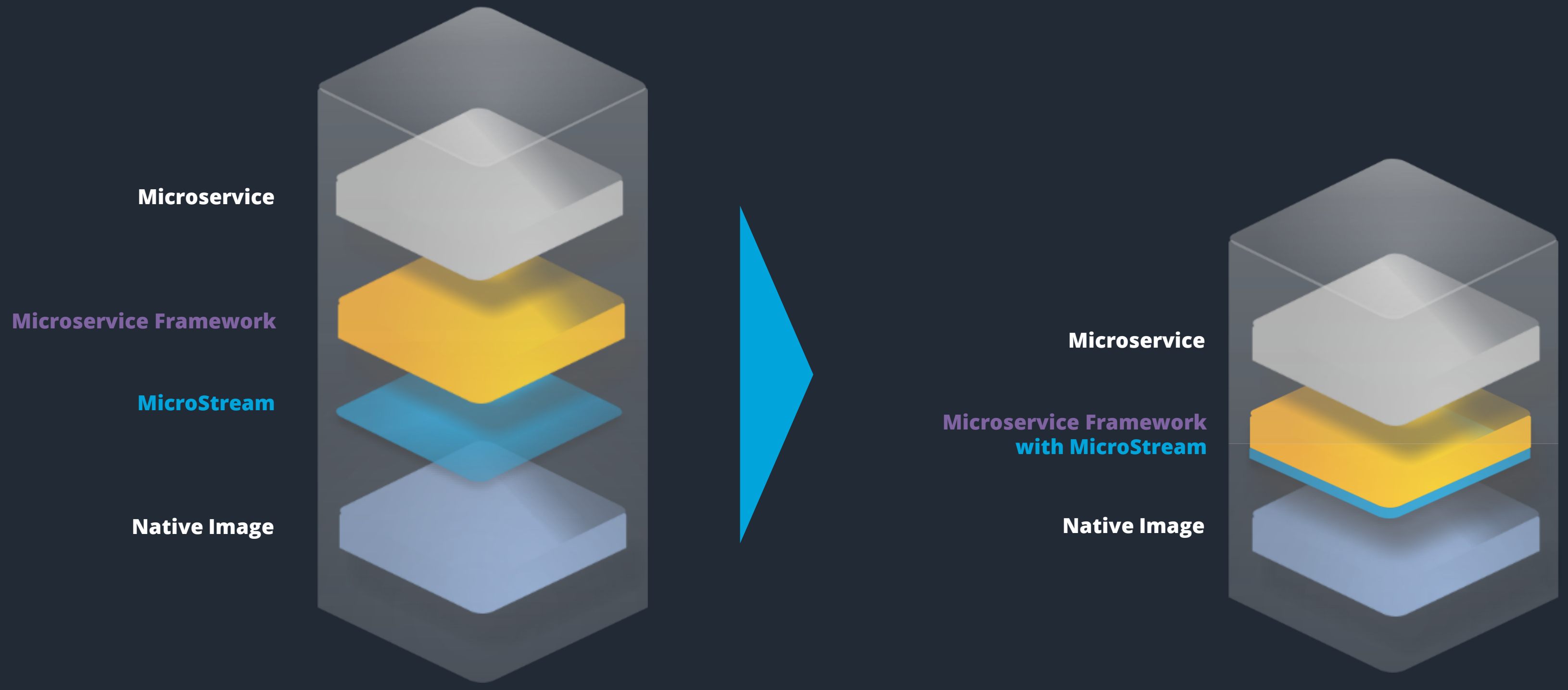
### High Quality API

MicroStream is a high-quality tiny Java API.

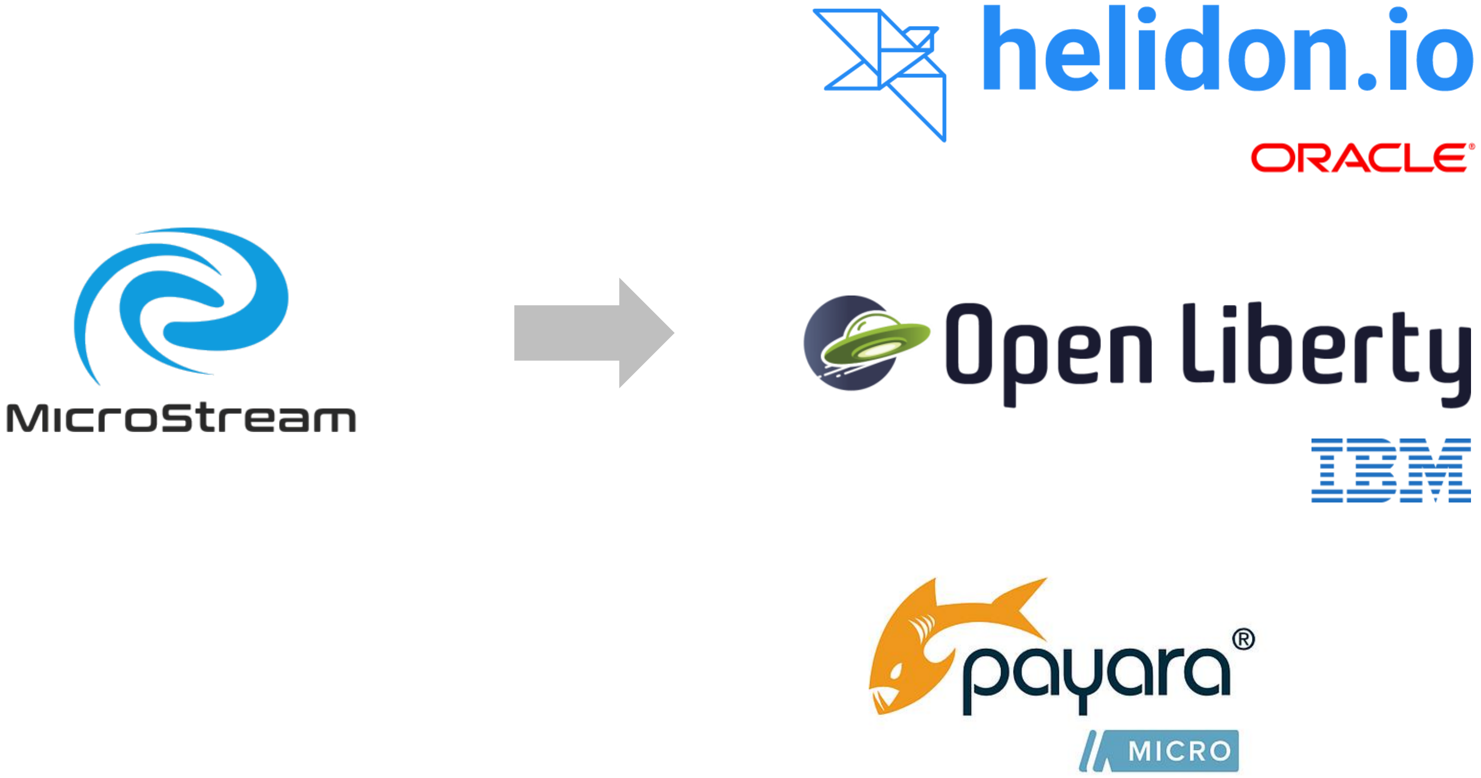
### Simple Implementation

MicroStream can be included in any Java project via Maven.

# MicroStream will be Integrated and Delivered with Microservices Frameworks



# Cooperation & Integration



# Runs Wherever Java Runs

Desktops

On-Premise

Cloud

Container

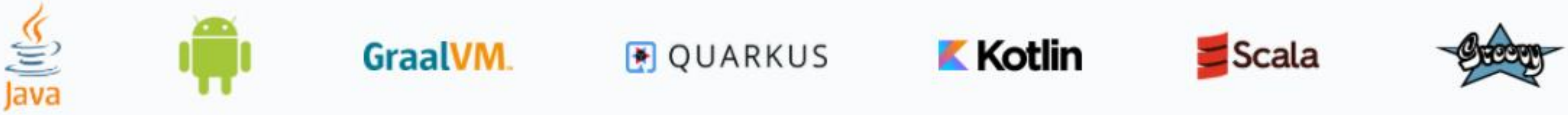
Native Image

Microservices

Android

JDK 8+

# Use any JVM Technology



# MicroStream - Android-Native High-Performance Persistence

MicroStream

RDBMS

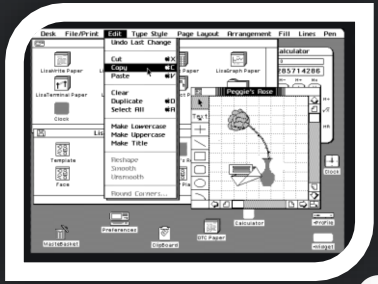
Binaries

# 3 Main Benefits …

# Performance Enables Revolutionary New Innovations, Features and Products

**1960s – The Main Frame**

**1976 – The Personal Computer**



**1983 – The Graphical User Interace**

**2007 – The Smartphone**

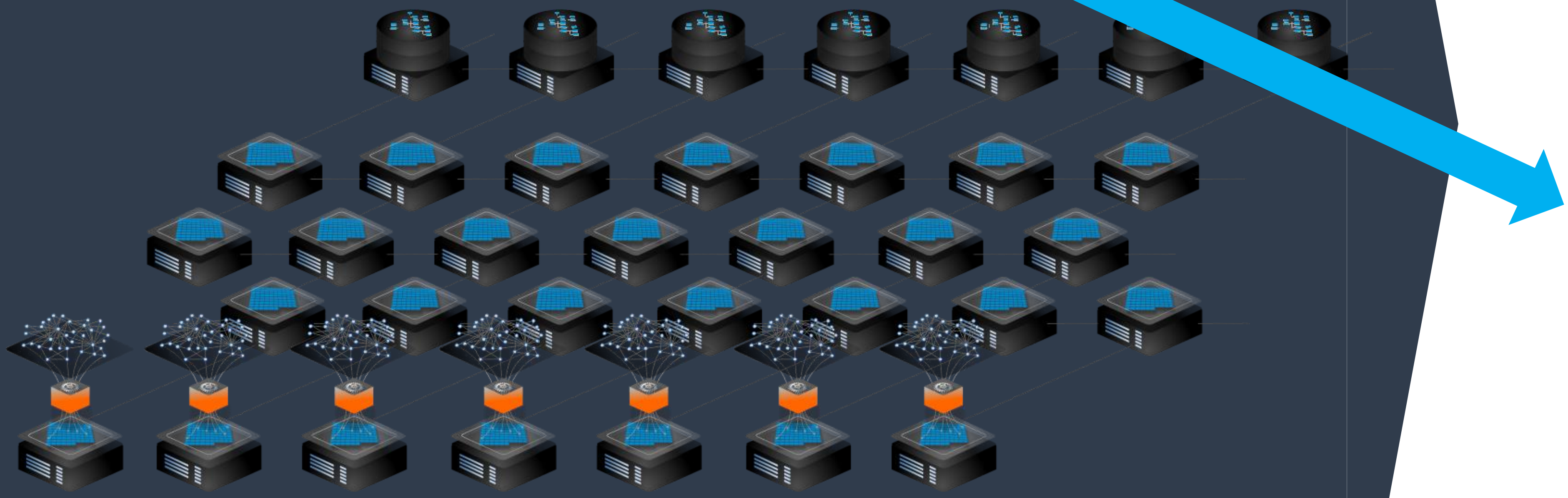
**Today – AI, ML, IoT, Automotive**

# Revolutionizing Cloud Infrastructure

**320 vCPUs**

**$134,400**

**Per year**

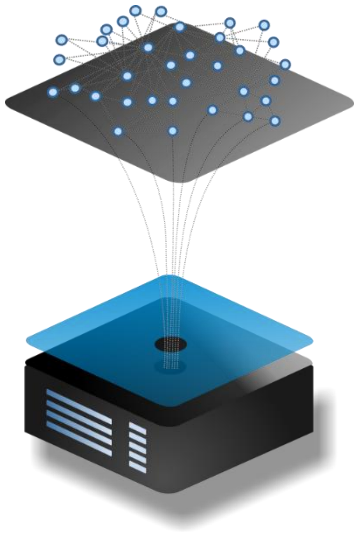


**Conventional Cluster for Running a globally App**

**Today with MicroStream:**

**- 87.5 %**

**Costs of Infrastructure annually**



**Only 1 Machine !**

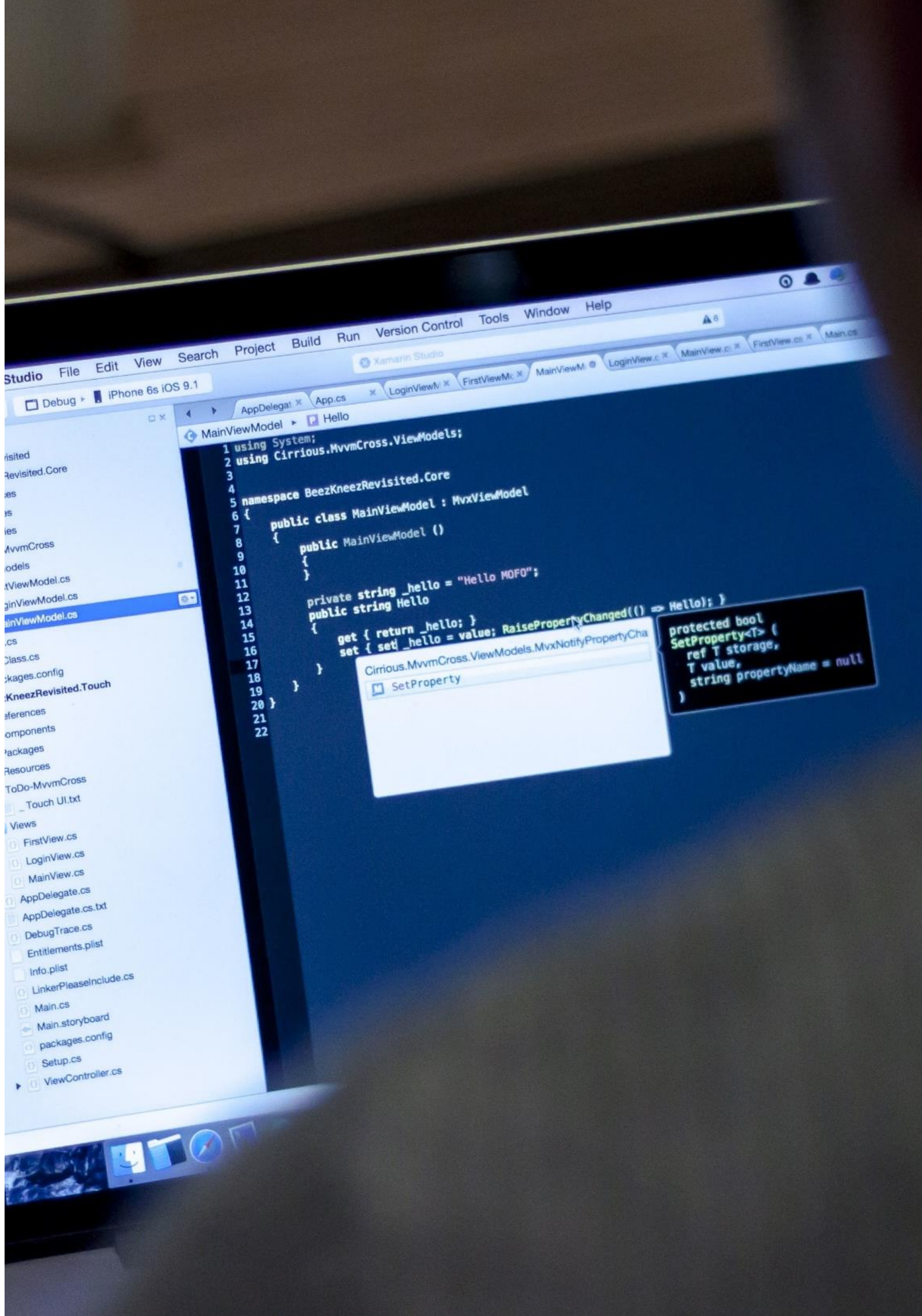
**40 vCPUs**

**$18,800**

**Per year**

# Simplifies your Development Process

- 1 data structure
- 1 data model - Java classes only
- No mapping, no impedance mismatch
- No JPA
- Query language: Java Streams API
- No local cache needed
- No dependencies, no special superclass or interfaces, no annotations, just POJOs
- Freely design of your Java object-model
- Core Java only



```
using System;
using Cirrious.MvvmCross.ViewModels;


namespace BeezKneezRevisited.Core
{
    public class MainViewModel : MvxViewModel
    {
        public MainViewModel ()
        {
        }

        private string _hello = "Hello MOFO";
        public string Hello
        {
            get { return _hello; }
            set { set _hello = value; RaisePropertyChanged(() => Hello); }
        }
    }
}
```

```
protected bool
SetProperty<T> (
   ref T storage,
   T value,
   string propertyName = null
)
```

# Anything missing in MicroStream?

**Just have a look at the great Java Ecosystem**

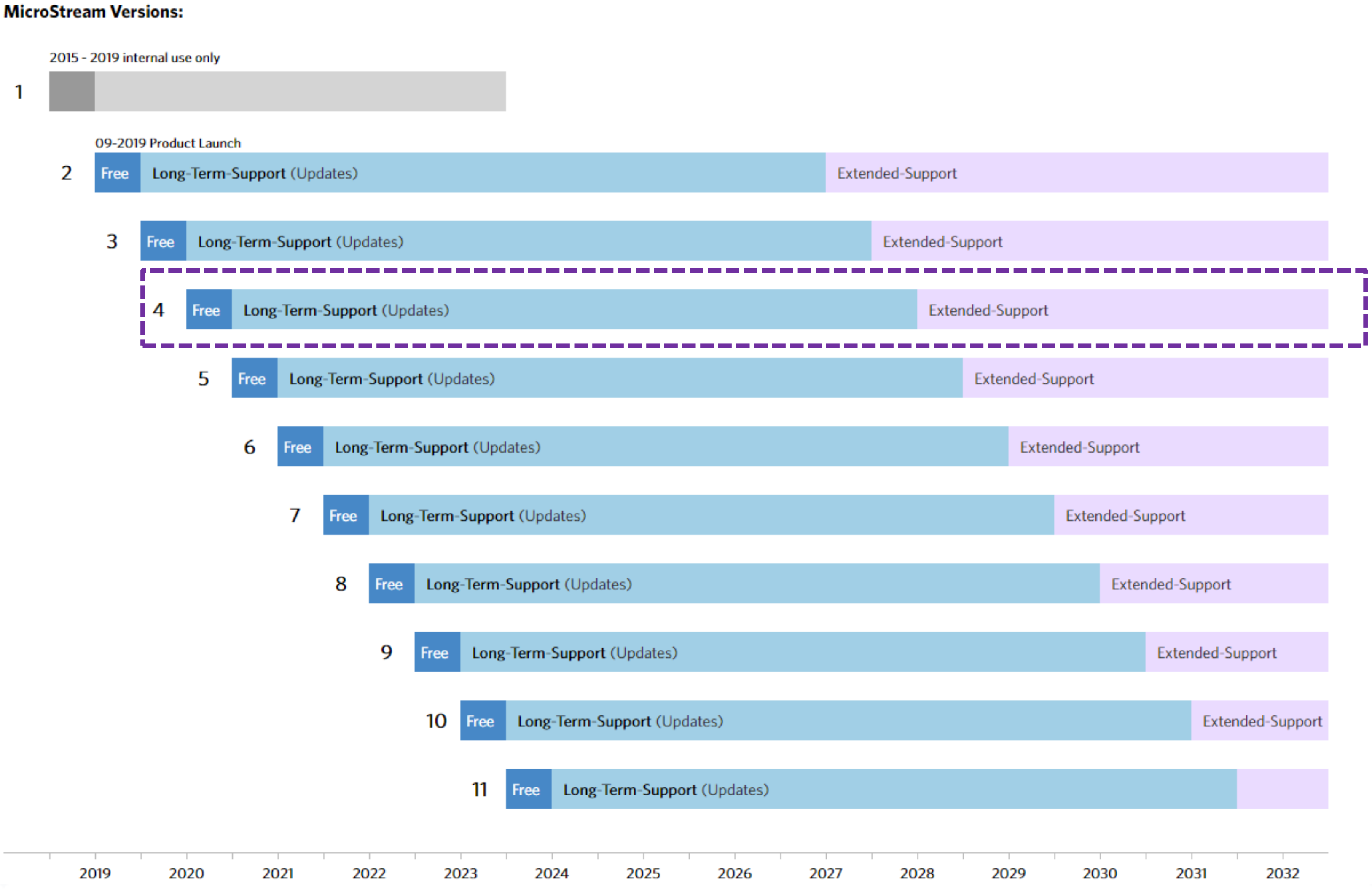
GraphQL

Lucene™

OpenJ9

RAPIDclipse™ X
Next Generation RapidClipse

# Release Plan



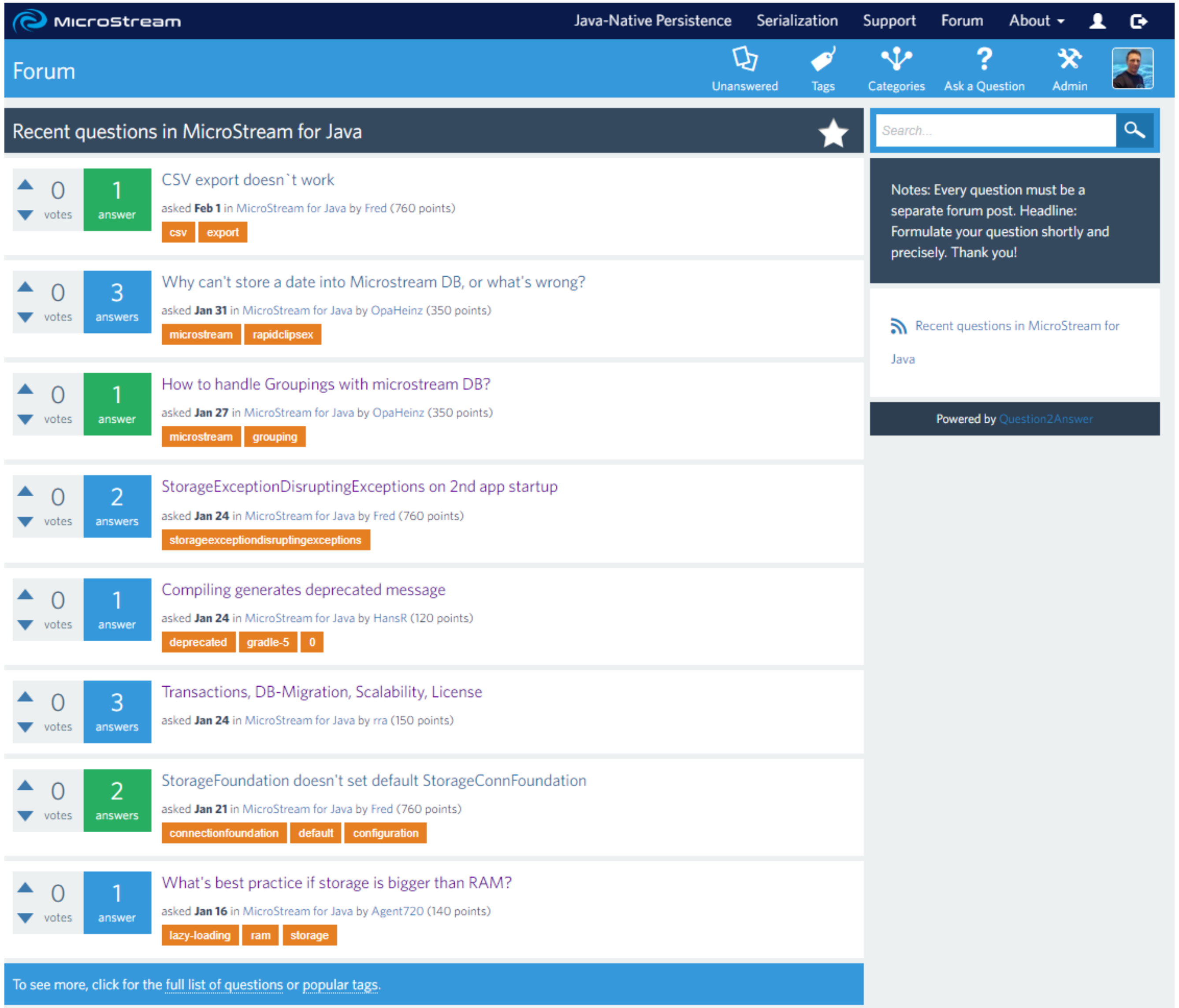
New major release every 6 months.

6 months updates free.

**Latest Version 4.1**

# MicroStream Support

- MicroStream Forum
- StackOverflow
- Professional Support
- Videos
- Training



Recent questions in MicroStream for Java

0 votes | 1 answer
CSV export doesn`t work
asked **Feb 1** in MicroStream for Java by Fred (760 points)
csv export

0 votes | 3 answers
Why can't store a date into Microstream DB, or what's wrong?
asked **Jan 31** in MicroStream for Java by OpaHeinz (350 points)
microstream rapidclipsex

0 votes | 1 answer
How to handle Groupings with microstream DB?
asked **Jan 27** in MicroStream for Java by OpaHeinz (350 points)
microstream grouping

0 votes | 2 answers
StorageExceptionDisruptingExceptions on 2nd app startup
asked **Jan 24** in MicroStream for Java by Fred (760 points)
storageexceptiondisruptingexceptions

0 votes | 1 answer
Compiling generates deprecated message
asked **Jan 24** in MicroStream for Java by HansR (120 points)
deprecated gradle-5 0

0 votes | 3 answers
Transactions, DB-Migration, Scalability, License
asked **Jan 24** in MicroStream for Java by rra (150 points)

0 votes | 2 answers
StorageFoundation doesn't set default StorageConnFoundation
asked **Jan 21** in MicroStream for Java by Fred (760 points)
connectionfoundation default configuration

0 votes | 1 answer
What's best practice if storage is bigger than RAM?
asked **Jan 16** in MicroStream for Java by Agent720 (140 points)
lazy-loading ram storage

To see more, click for the full list of questions or popular tags.

Notes: Every question must be a separate forum post. Headline: Formulate your question shortly and precisely. Thank you!

Recent questions in MicroStream for Java

Powered by Question2Answer

# Support Plans

**Perfect for evaluating and first steps**

## Community

- Community online forum
- Stack Overflow

**Free !**

**Fantastic for getting started**

## Gold

- Community support incl.
- Response time guaranteed
- Next business day
- Online ticket system
- 10 support hours incl.

**$119** **/ Developer / Month**

Annual billing

**Must have for enterprises**

## Platinum

- Gold support incl.
- Response time 4 hrs
- Enhanced issue priority
- Phone / Online support
- Remote login
- 20 support hours incl.

**$599** **/ Developer / Month**

Annual billing

**Best choice for large enterprises**

## Diamond

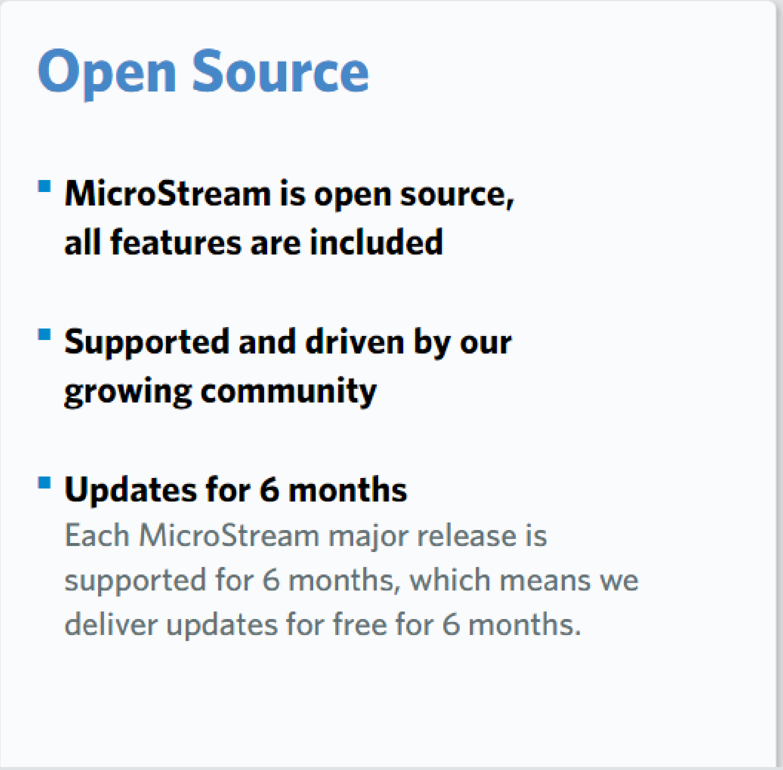
- Platin support incl.
- Hotfixes
- 24/7
- Escalation support
- Training
- Expert on demand
- Personal mentor
- Unlimited support hours
- Customized agreement

**Contact Sales**

Note: All prices on this website are net prices plus VAT and evt. local tax!

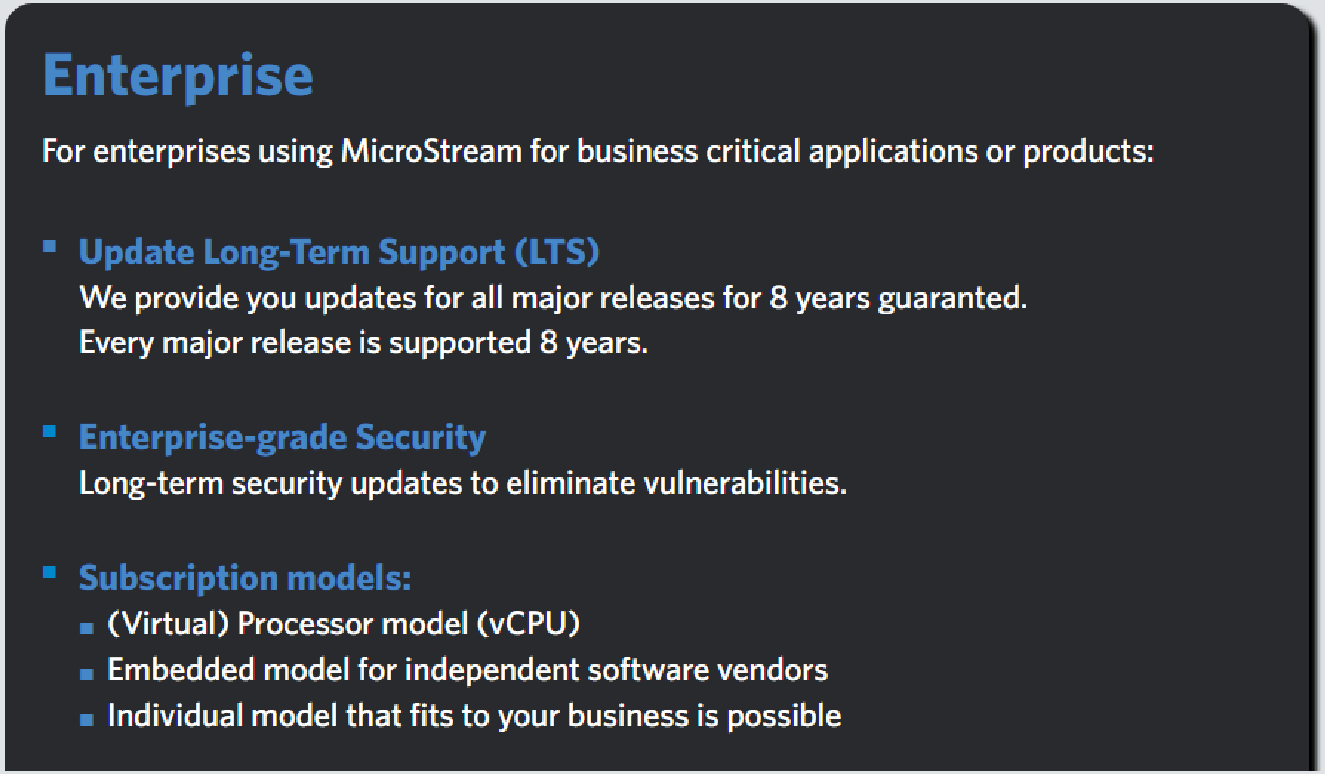
Do you have any questions about our services and prices, please contact sales.

# Update Warranty



## Open Source

- **MicroStream is open source, all features are included**
- **Supported and driven by our growing community**
- **Updates for 6 months**
  Each MicroStream major release is supported for 6 months, which means we deliver updates for free for 6 months.



## Enterprise

For enterprises using MicroStream for business critical applications or products:

- **Update Long-Term Support (LTS)**
  We provide you updates for all major releases for 8 years guaranted.
  Every major release is supported 8 years.
- **Enterprise-grade Security**
  Long-term security updates to eliminate vulnerabilities.
- **Subscription models:**
  - (Virtual) Processor model (vCPU)
  - Embedded model for independent software vendors
  - Individual model that fits to your business is possible

MicroStream

# High-Performance Persistence

Store any Java Object Graph and Subgraphs into any Database.
Microsecond Query Time. Ultra-high Throughput. Minimum of Latencies.
Create Ultra-fast In-Memory Database Applications and Microservices.

**www.microstream.one**

JAVA USER GROUP CH

# THANK YOU !