



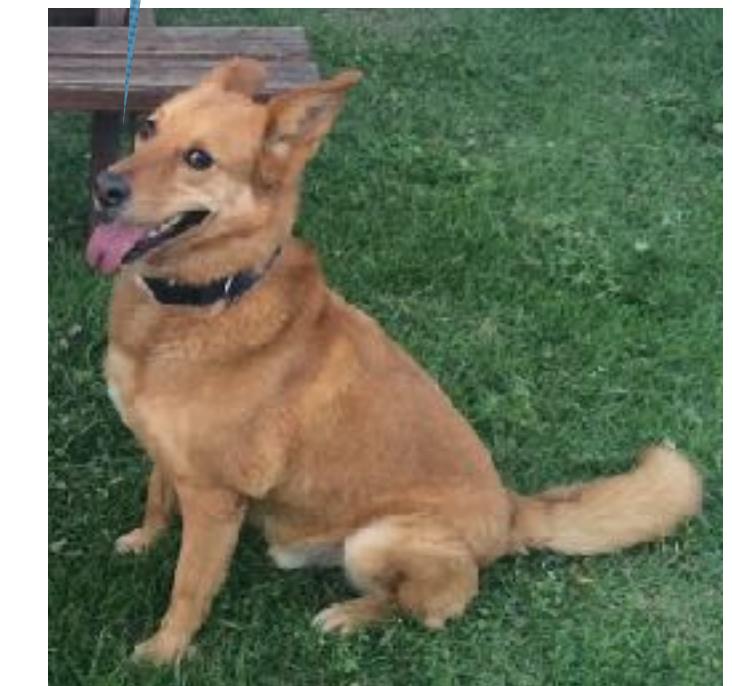
Building Microservices with Kotlin
Haim Yadid

Disclaimer

- The purpose of this talk is to share our experience and with Kotlin not to teach the language syntax. I will delve into some details for the basics just go to the documentation (<https://kotlinlang.org/docs/reference/>)
- While comparison between Kotlin and Scala is tempting this will not be the focus of the talk.

About Me

- Developing software since 1984 (Boy, am I getting old?)
- Basic -> Pascal -> C -> C++ -> Java -> Kotlin
- Developer , architect, group manager
- Independent performance expert for 8 years
- Head of backend engineering in 



Drakaris

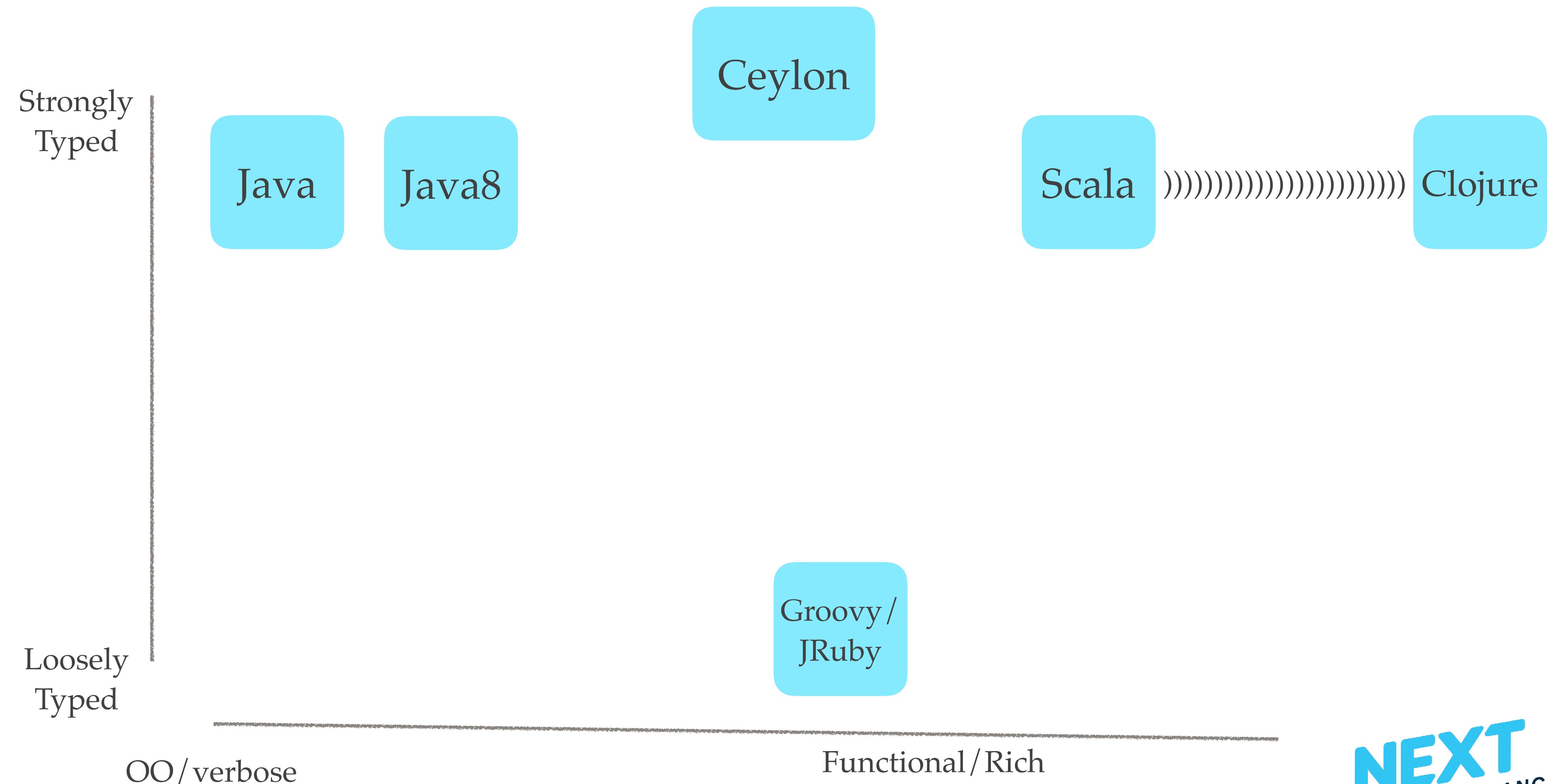


- Founded in the Beginning of 2016
- Disrupt the small businesses insurance field
- Providing online experience which is simple, fast and transparent
- HQ@Palo Alto RnD@Kfar Saba (Israel)
- We started to write real code on May 2016

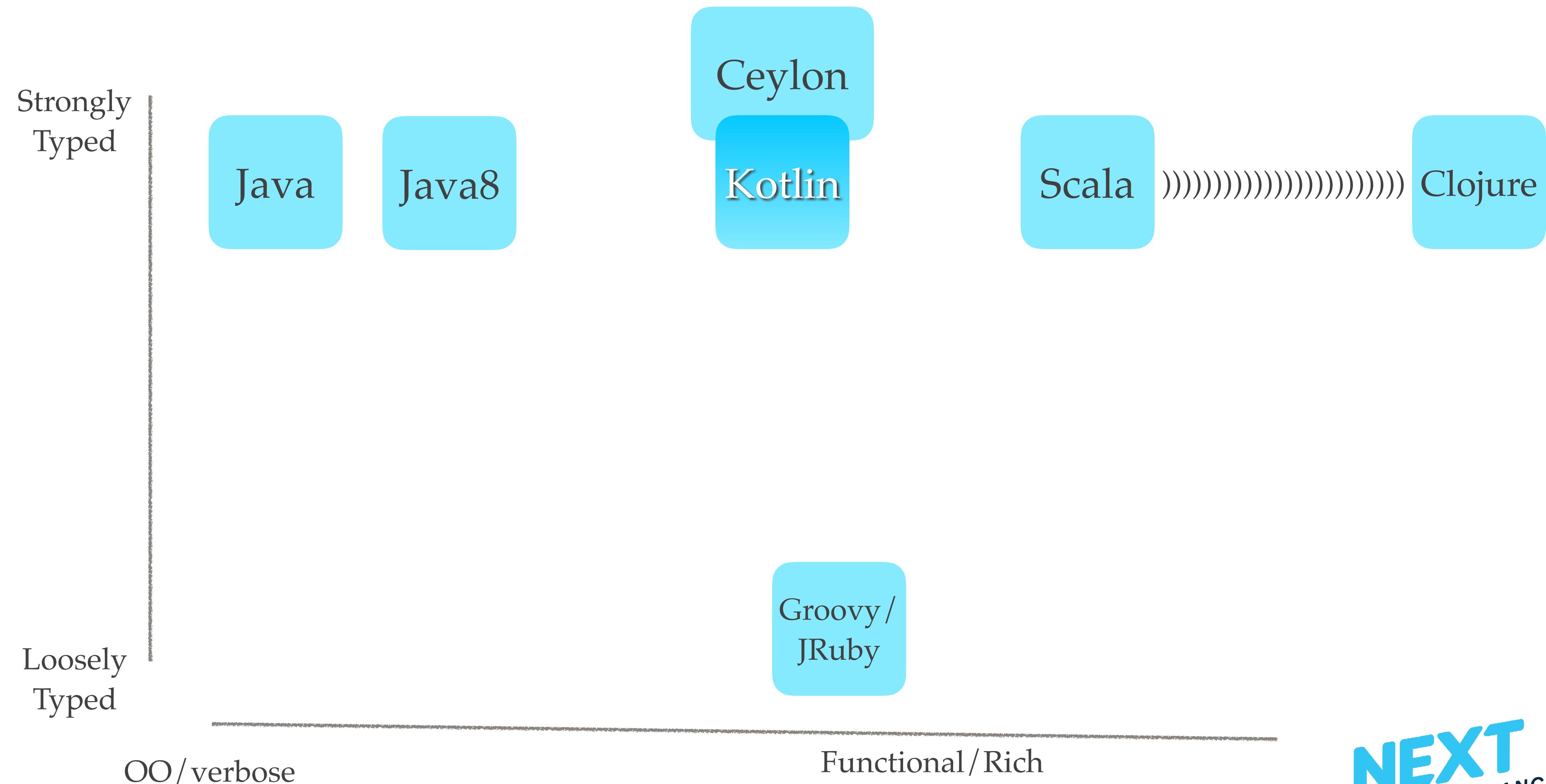
Looking for a language



JVM Languages



JVM Languages



What is Kotlin ?

- Strongly typed programming language
- For the JVM, Android and the browser (JS)
- 100% interoperable with Java™ (well almost)
- Developed by JetBrains
- Revealed as an open source in July 2011
- v1.0 Release on Feb 2016
- 1.1.51 current stable version (as of 28-Sep-2017)
- 1.2 is in EA



Kotlin Design Goals

- Concise
- Safe
- Versatile
- Practical
- Interoperable



NEXT
INSURANCE

Bottom Line

Huge Success



Kotlin Adoption



- Android official language
- 9 talks in JavaOne 2017
- Community enthusiasm (hype ?)

We use Kotlin for



- Building our backend micro-services over DropWizard (deployed to AWS)
- Building serverless endpoints (AWS Lambda)

120K lines of
Kotlin code

5K lines of
Java code

8 micro
services

12 Lambda
functions

Kotlin version upgrade



- Started at 1.0.2
- Upgraded to every release of Kotlin immediately
- Migration to 1.1.0 (Java8 support) was smooth
- No breaking changes so far (for us)
- Now on 1.1.51

Onboarding



- Onboarding of new Java developers proved to be smooth
- Java developers are capable to developing in Kotlin on the same pace they are able to understand the architecture



NEXT
INSURANCE

Java Ecosystem

Java Ecosystem

- Java open source libraries works well with Kotlin.
- Just add the dependency to you build file and you are done

Kotlin Primitives

- kotlin.Int => int
- kotlin.Double => double
- kotlin.String => java.lang.String

Collections

- kotlin.HashMap = java.util.LinkedHashMap
- Underlying Java collections
- Maps and lists can be either mutable and immutable
- “Immutability” = immutable view (Compromise)

Collections

```
val heros = mapOf("Terion" to "Lannister", "John" to "Snow")  
heros["Terion"]
```

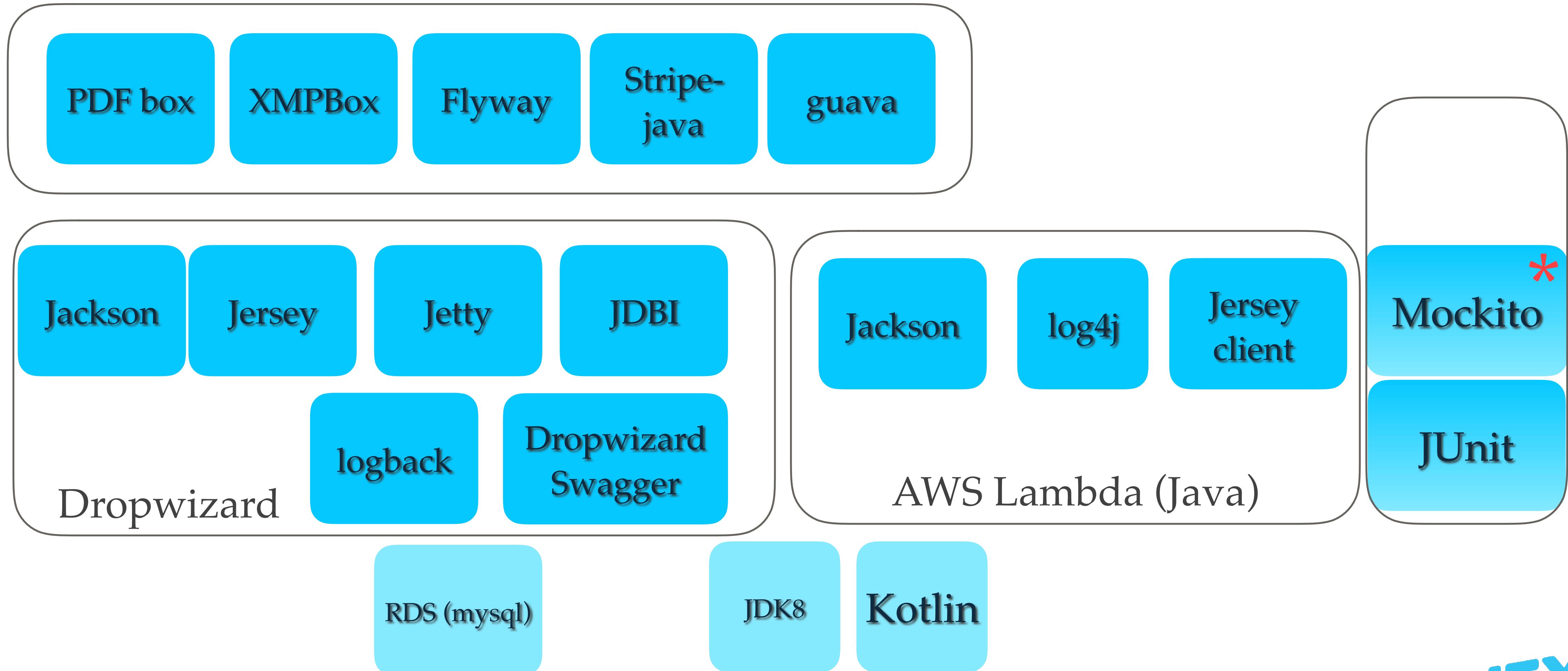


```
heros["Aria"] = "Stark"
```

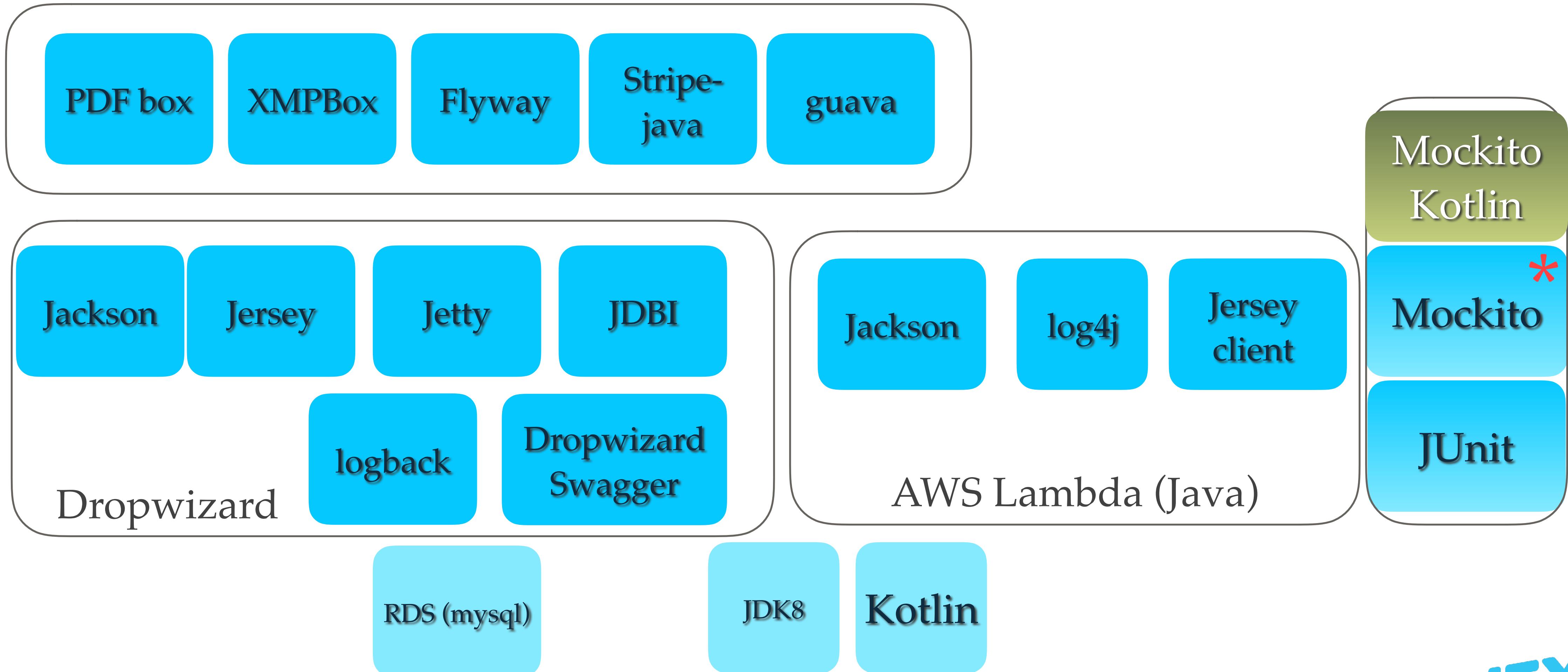
```
val chars2 = mutableMapOf<String, String>()  
chars2["A girl"] = "has no name"
```

```
val sigils = ["Direwolf", "Lion", "Three headed Dragon",  
"Flower"]  
println(sigils[0])
```

Third Party Libraries



Third Party Libraries



Mockito Kotlin

- when -> `when` -> whenever
- Solve Null safety issues with any()
- DSL like syntax using Lambda expressions

Project organization

- Build with *kotlin-maven* plugin
- Same source paths as java
 - src/main/java
 - src/test/java
- Dependency management :
 - *kotlin-stdlib*
 - *kotlin-reflect*
 - *kotlin-stdlib-jre7*
 - *kotlin-stdlib-jre8*



NEXT
INSURANCE

Extension Functions

Extension functions

- Add functionality to a class w/o inheritance
- Only extend functionality cannot override members
- Not part of the class
- Static methods with the receiver as first a parameter
- (Not like ruby monkey patching)

```
fun String.paperWrap = "[${this}]"
"hello".paperWrap
-> "[hello]"
```

ResultSet null values

- When querying a java.sql.ResultSet for a value that is nullable
- `getLong` will return 0 when the value is null and you are expected to invoke `wasNull` afterwards

```
fun ResultSet.getNullableLong(colName: String): Long? {  
    val value = this.getLong(colName)  
    return if (this.wasNull()) null else value  
}
```

Measuring endpoint duration

- Write endpoint duration to log entries
- ES/Kibana (we use logz.io)
- When we report our data to we have duration field for every endpoint invocation.

Measuring endpoint duration

```
fun Logger.infoWithDuration(message: String,  
                           vararg additionalArgs: Any){  
    loggerActionWithDuration {  
        if (additionalArgs.isNotEmpty())  
            info(message,*additionalArgs)  
        else  
            info(message)  
    }  
}
```

loggerActionWithDuration

```
inline fun Logger.loggerActionWithDuration(action: () -> Unit){  
    updateDurationMDCValue()  
    action.invoke()  
    MDC.remove(MDC_DURATION_KEY)  
}
```

Calculating endpoint duration

- store on MDC the transaction start time
- in this method we store the the duration from start on MDC as well

```
fun Logger.updateDurationMDCValue() {  
    val startTimestampMDCValue = MDC.get(MDC_TRANSACTION_START_TS_KEY)  
  
    if(startTimestampMDCValue!=null){  
        val startTimestamp = startTimestampMDCValue.toLong()  
        val requestDuration = now() - startTimestamp  
        MDC.put(MDC_DURATION_KEY, requestDuration.toString())  
    }  
}
```

Request Filter

```
class DurationStartFilter : ContainerRequestFilter {  
    override fun filter(requestContext: ContainerRequestContext) {  
        val transStartTS = now()  
  
        MDC.put(MDC_TRANSACTION_START_TS_KEY, transStartTS.toString())  
    }  
}
```

Response Filter

```
class DurationFilter : ContainerResponseFilter {  
    val logger: Logger = LoggerFactory.getLogger(DurationFilter::class.java)  
  
    override fun filter(containerRequestContext: ContainerRequestContext?,  
                       containerResponseContext: ContainerResponseContext?) {  
        logger.infoWithDuration("Request processing finished.")  
    }  
}
```



NEXT
INSURANCE

Null Safety

Null Safety

- Nullability part of the type of an object
- Option[T] Optional<T>
- Swift anyone ?



```
var msg : String = "Welcome"  
msg = null  
  
val nullableMsg : String? = null
```

Safe operator ?.

- The safe call operator ?. will result null on null receiver
- Elvis operator for default

```
fun funny(funnier: String?): Int? {  
    println(x?.length ?: "")  
    return x?.length  
}
```

Bang Bang !!

- The bang bang !! throws an NPE if object is null

```
fun funny(funnier: String?): String {  
    return funnier!!  
}
```

Null Pollution

- It is really easy to pollute our code with nulls
- Java code is not handling nulls properly
- map.get() or the [] shortcut possibly return null
- map["key"]!! with throw KotlinNullPointerException

Require

- Our own implementation
- Extension function
- Throws a clearer exception

```
fun <T> Map<String, T>.require(key: String, allowEmpty: Boolean = false): T {  
    val value = this[key] ?:  
        throw IllegalArgumentException("Required aMap[$key] is missing. aMap.size = ${this.size}")  
    if (!allowEmpty) {  
        if (value is String) {  
            if (value.isEmpty()) {  
                throw IllegalArgumentException("Required aMap[$key] is empty. aMap.size = ${this.size}")  
            }  
        }  
    }  
    return value  
}
```

getValue

- Since Kotlin 1.1
- Throws a Descriptive NoSuchElementException which includes key name

```
val value = mappy.getValue("key")
```

Delegate by map

```
data class Person(val firstName: String,  
                 val lastName: String,  
                 var props: MutableMap<String, String>) {  
  
    var businessname: String by props  
  
    val emailaddress: String by props  
}
```

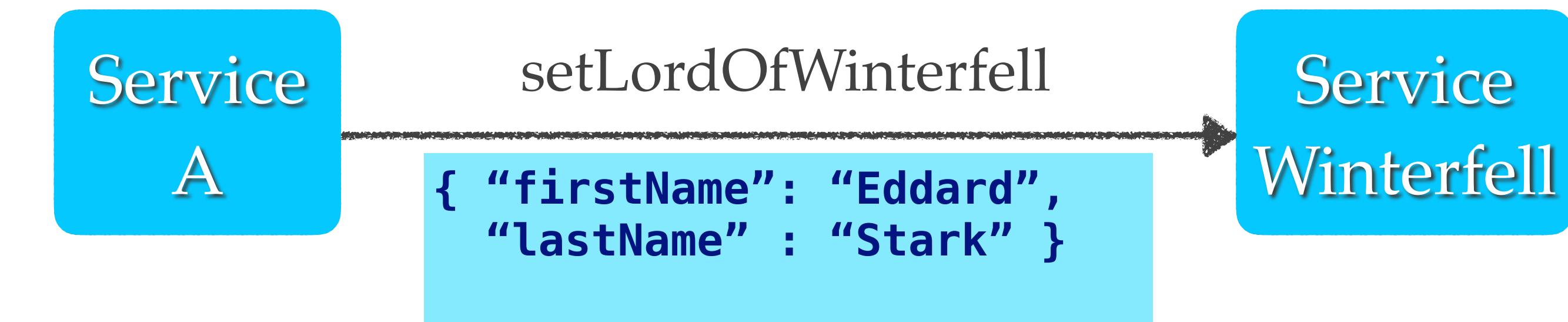


NEXT
INSURANCE

JSON Serialization

Microservices Talk

- Over HTTP
- JSON serialization



DTOs

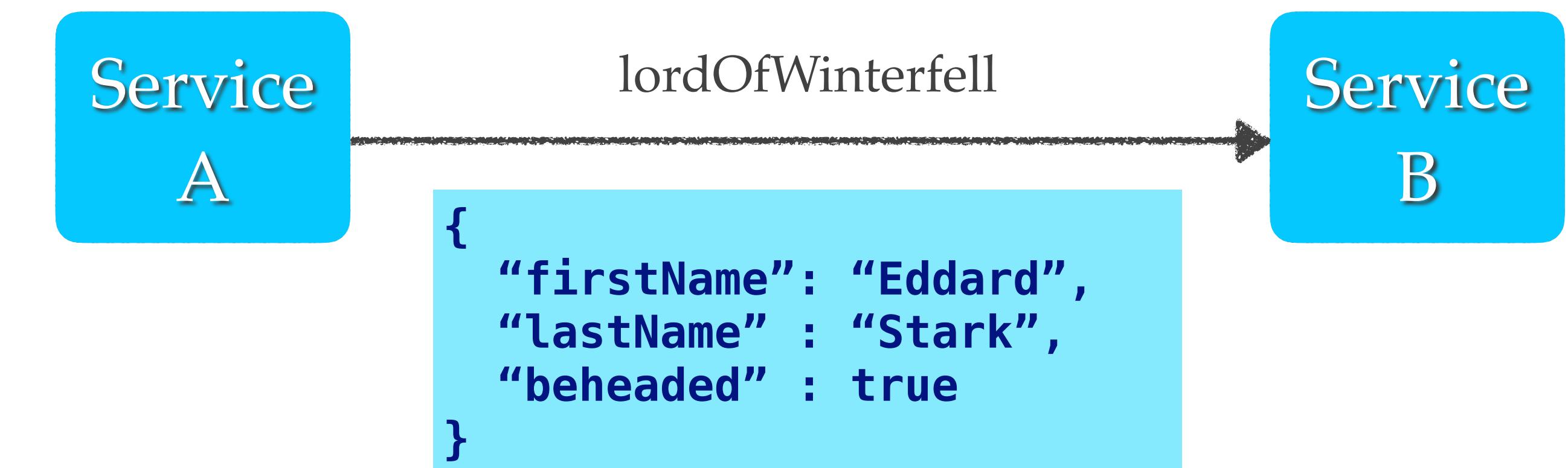
- Getter and setters for all fields
- Implementation of
 - Implemented hashCode() equals()
 - copy()
 - destructuring (component1 component2 ...)

```
data class Lord(val firstName: String, val lastName: String)
```

```
val starkInWinterfellS1 = Lord("Eddard", "Stark")
val starkInWinterfellS2to3 = starkInWinterfell.copy(firstName = "Robb")
val (first, last) = starkInWinterfell
```

Evolution Season 1 Episode 9

- Adding a field should not be a breaking change
- Ability to deserialize
 - With missing field
 - With additional field



jackson-module-kotlin

- Introduces an introspection which do not need annotations for most cases
- Nullable represents optional

```
data class Lord(  
    val firstName: String,  
    val lastName: String,  
    val beheaded: Boolean?,  
)
```

```
ObjectMapper()  
    .registerModule(KotlinModule())  
    .configure(DeserializationFeature.FAIL_ON_UNKNOWN_PROPERTIES, false)  
    .configure(DeserializationFeature.FAIL_ON_NULL_FOR_PRIMITIVES, true)
```

Delegate by map

```
data class Person(val firstName: String,  
                 val lastName: String,  
                 var props: MutableMap<String, String>) {  
    @get:JsonIgnore  
    val businessname: String by props  
    @get:JsonIgnore  
    val emailaddress: String by props  
}
```

```
println(person.businessname)
```



NEXT
INSURANCE

Inline / Reified

execution_statuses Table

id	name
1	success
2	referral
3	technical_error
4	decline
6	renewal

Enum

```
enum class ExecutionStatuses(  
    override val id: Int,  
    override val dbName: String) : DBEnum {  
  
    SUCCESS(1, "success"),  
    REFERRAL(2, "referral"),  
    TECHNICAL_ERROR(3, "technical_error"),  
    DECLINE(4, "decline"),  
    RENEWAL(5, "renewal");  
}
```

DBEnum

```
interface DBEnum {  
    val id: Int  
    val dbName: String  
  
    companion object {  
  
        inline fun <reified T> fromId(id: Int): T where T : Enum<T>, T : DBEnum {  
            return enumValues<T>().first { it.id == id }  
        }  
  
        inline fun <reified T> fromName(name: String): T where T : Enum<T>, T :  
DBEnum {  
            return enumValues<T>().first { it.dbName == name }  
        }  
    }  
}
```

Integration Test

```
inline fun <reified T> verifyValuesAreConsistent(dbIdToName:  
MutableMap<Int, String>) where T : Enum<T>, T : DBEnum {  
    val enumValues = enumValues<T>()  
    expect(dbIdToName.size).toBe(enumValues.size)  
    dbIdToName.forEach {  
        val dbId = it.key  
        val dbName = it.value  
        expect(DBEnum.fromId<T>(dbId).dbName).toBe(dbName)  
        expect(DBEnum.fromName<T>(dbName).id).toBe(dbId)  
    }  
}
```



NEXT
INSURANCE

Integration Test DSL

TestStage

```
abstract class TestStage<out T> {  
    lateinit var testIT: UnderwritingCycleTest  
    lateinit var testDataSet: DataSet  
  
    abstract fun construct(testIT: UnderwritingCycleTest,  
                          testDataSet: DataSet): T  
}
```

Test

```
@Ignore  
@Test  
fun e2eFlowWithCarrierTennesseeState() {  
    val testDataSet = DataSet.build(this, PartnerTestData) {  
        overrideParam("state", "TN")  
    }  
    startFlow(testDataSet)  
        .sendBusinessDetails()  
        .screenRequest()  
        .createQuoteSync()  
        .preSelect()  
        .select()  
        .paySuccessWithoutBind()  
        .bindFailure{  
            assertEquals(CarrierBindStatuses.MANUAL_STEPS_REQUIRED.dbName,  
                         this.carrierBindStatus)  
        }  
}
```

NEXT
INSURANCE



Questions?