Camel 4

What to expect?

About

Otavio R. Piske

- PSE @Red Hat
- Committer+PMC @ Apache Camel

Online

- Github: @orpiske
- Twitter: @otavio021
- angusyoung@gmail.com

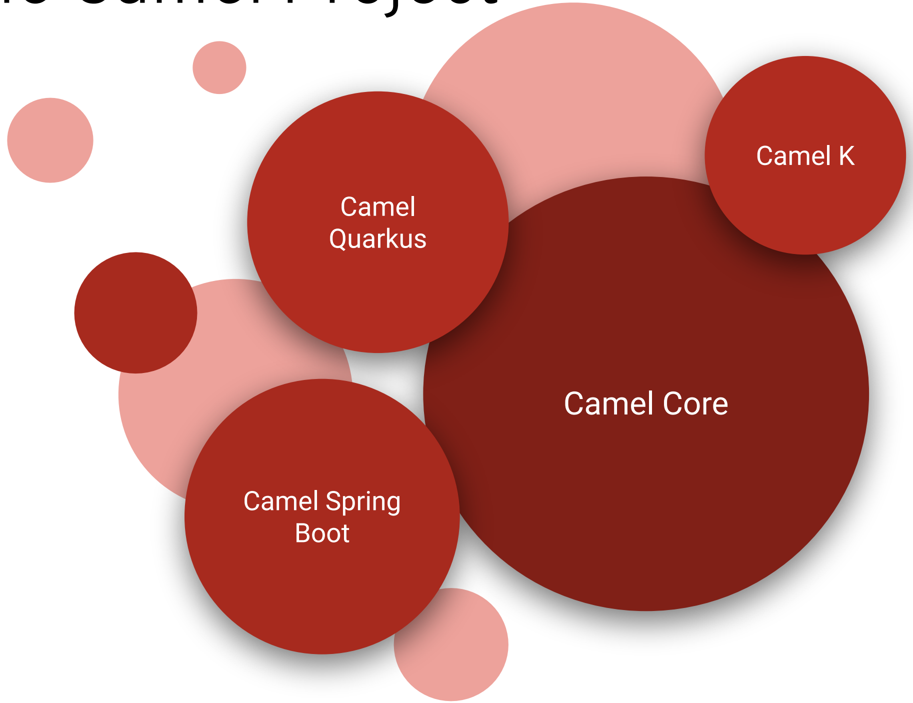
Real Life

- Brazilian @ Brno

Agenda

- Overview
- Motivations and objectives
- Under the hood
- Planning the upgrade
- Q&A

The Apache Camel Project



Camel Quarkus

Camel K

Camel Core

Camel Spring Boot

Apache Camel

```
public class KafkaToSQSRoute extends RouteBuilder {

    @Override
    public void configure() throws Exception {
       from("kafka:some-topic")
             .to("aws-sqs://someQueue");
    }
}
```

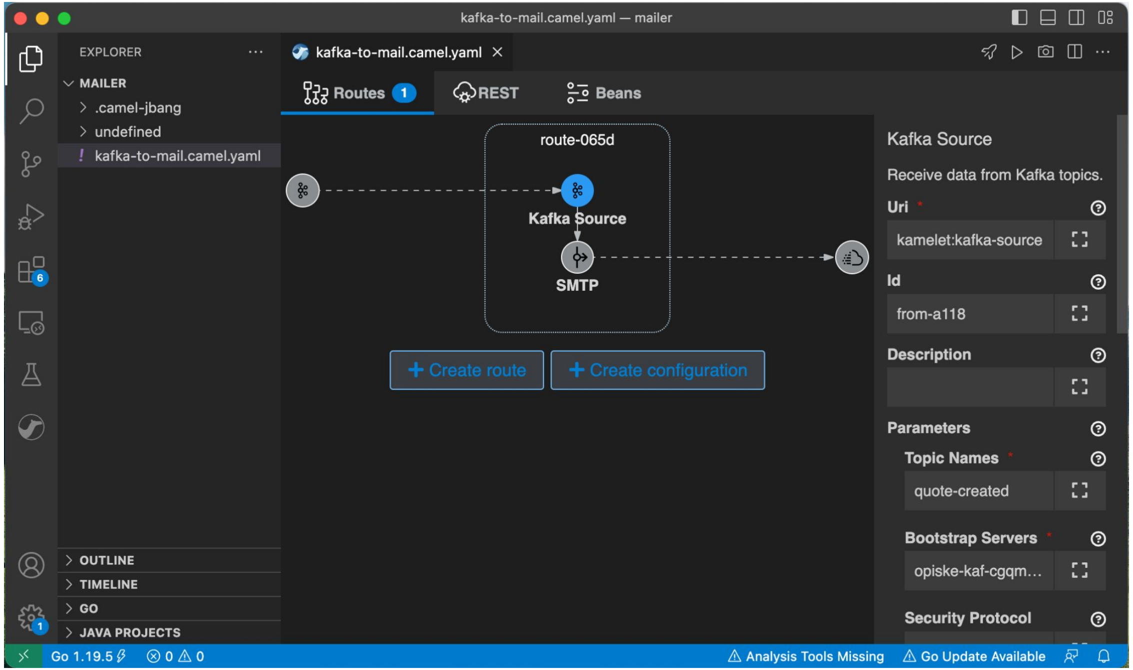
Camel JBang

- Quick way to start running integration code
- Great for prototyping
- Simplifies bootstrapping new project
- Provides functionality for other projects

```
$ camel run my-integration.yaml
```

Camel Karavan

- Toolkit for designing routes
- Visual designer for routes
- Extension for VS Code



Kafka Source

Receive data from Kafka topics.

Uri *: kamelet:kafka-source

Id: from-a118

Description:

Parameters

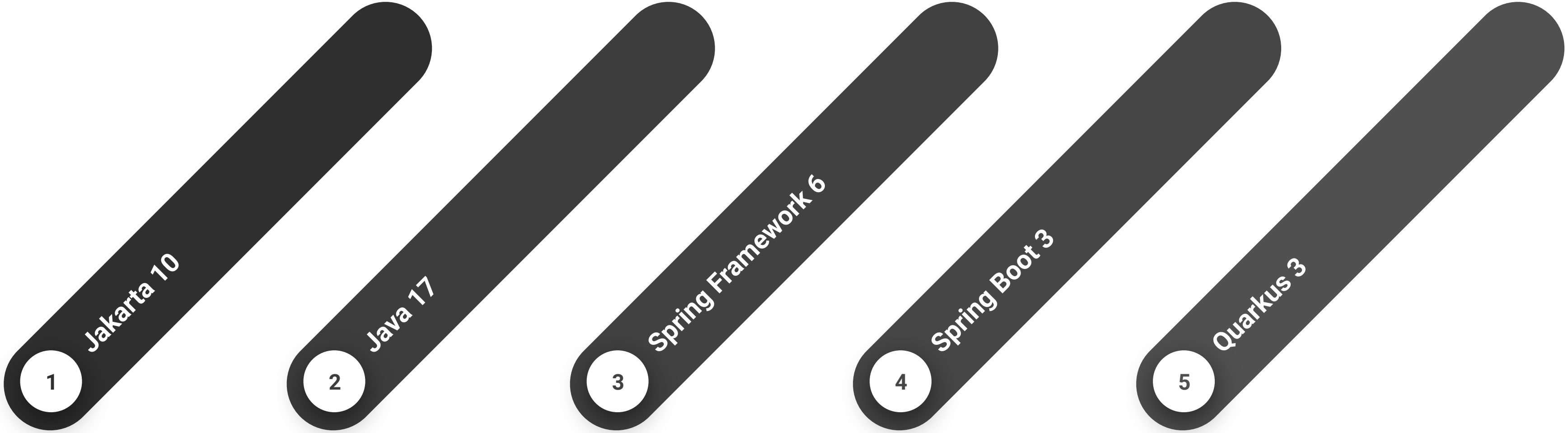
Topic Names *: quote-created

Bootstrap Servers *: opiske-kaf-cgqm…

Security Protocol

Motivations & Gogals

Motivations for Camel 4



1. Jakarta 10
2. Java 17
3. Spring Framework 6
4. Spring Boot 3
5. Quarkus 3

Build a foundation for the future

- Suport Java 17 features
  - What about Java 11?
- Forward-looking
  - Java 21 LTS in September

Simplify our build and release

- Camel Karaf (OSGI) as a best effort
- Decouple Camel Karaf releases from Camel Core

Reduce our maintenance effort

- Internal cleanups
  - Cleanup deprecated internal APIs
- Remove MDC support
- Drop legacy features
  - Drop JUnit 4 support
- Remove deprecated components

Deprecated component

- Temporarily or permanently removed
  - 33 components affected
  - 20 have an easy alternative (i.e; camel-activemq → camel-jms)
  - Most have shown little usage

Under the hood

Internal changes

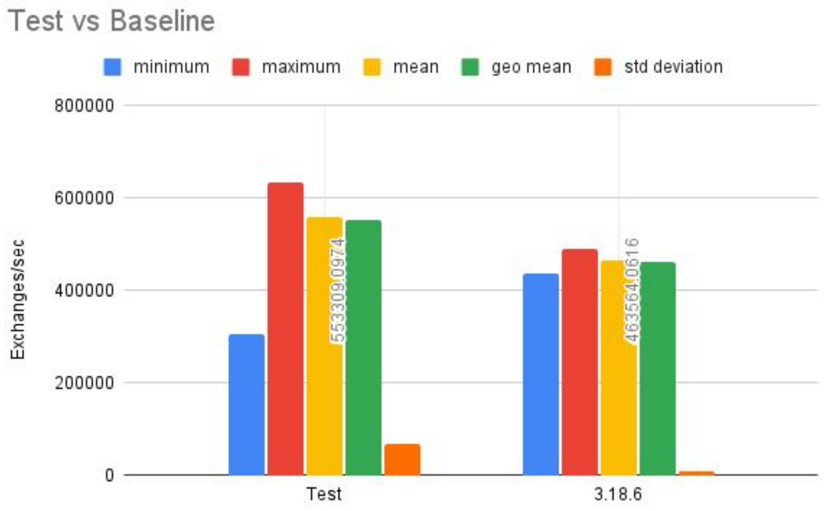
- Internal Plugin Manager
  - `CamelContext`
  - `ExtendedCamelContext`
- Internalized the `IntrospectionSupport`
- Tests cleanups
  - Aiming for a clean build on Power and s390x
  - JUnit extension for the CamelContext
- Software Bill of Materials (SBOM)

Performance improvements

- JDK-8180450
  - To learn more about
    - Quarkus Runtime: A Peek Into JVM Internals
    - Seeing through hardware counters
    - Cracking the Scalability Wall
- Optimizations
  - Seda
  - Disruptor
  - Core (in progress)
- Other micro-optimizations

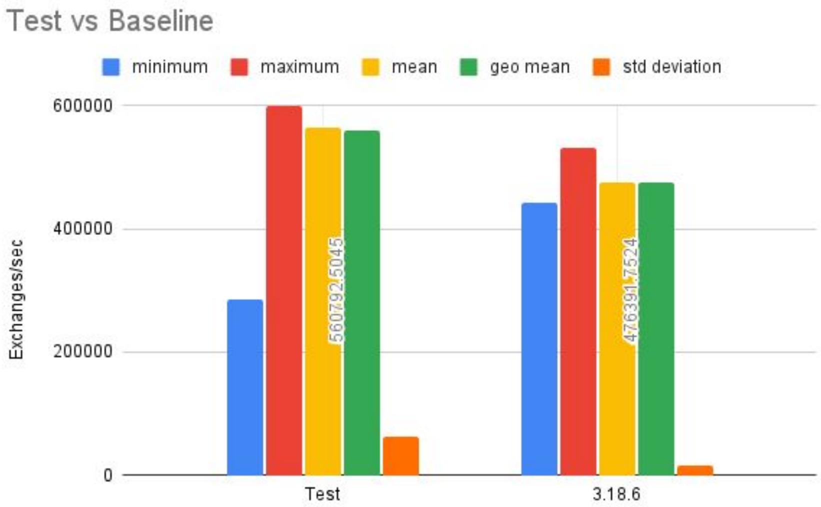
Performance improvements: SEDA

- Low-contention scenario
  - 2 consumers
  - 1 producer
- Heterogeneous data types
- Results
  - ~19% faster



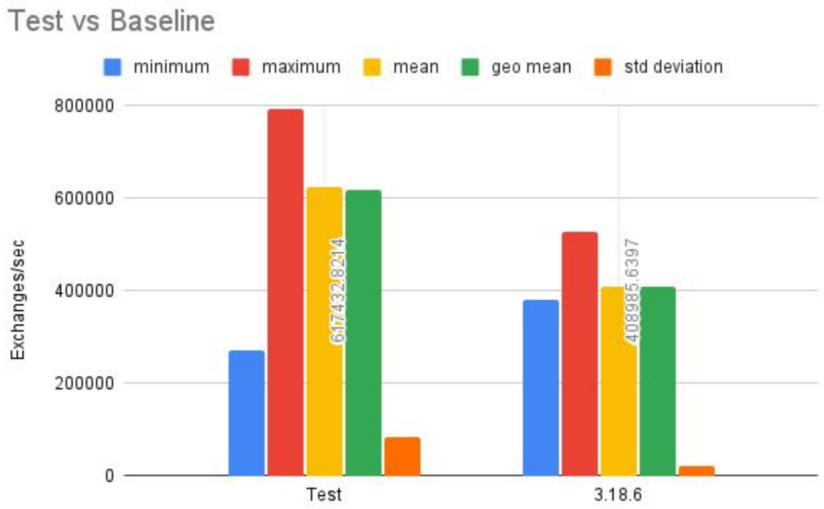
Performance improvements: SEDA

- Low-contention scenario
  - 4 consumers
  - 1 producer
- Heterogeneous data types
- Results
  - ~18% faster



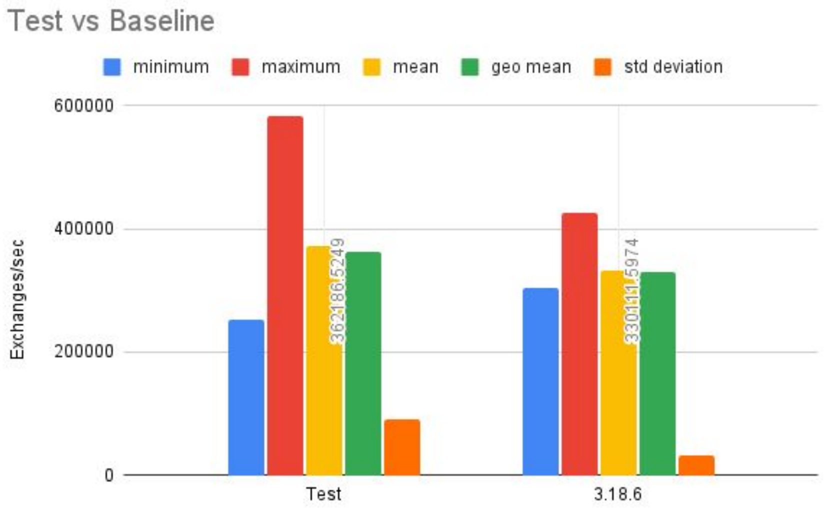
Performance improvements: SEDA

- Medium-contention scenario
  - 8 consumers
  - 1 producer
- Heterogeneous data types
- Results
  - ~51% faster



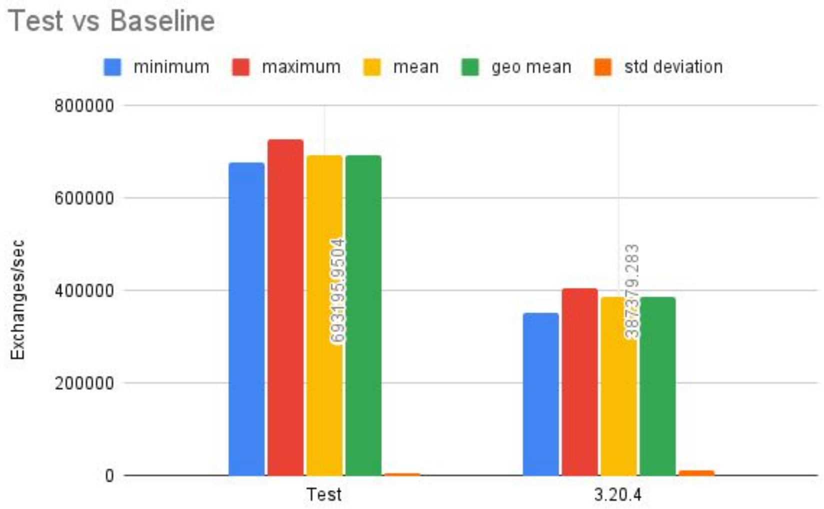
Performance improvements: SEDA

- High-contention scenario
  - 16 consumers
  - 1 producer
- Heterogeneous data types
- Results
  - ~10% faster



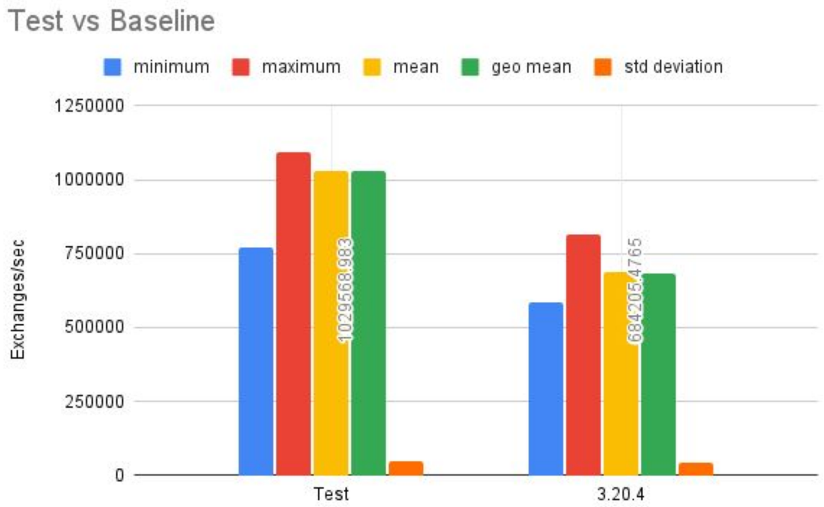
Performance improvements: disruptor

- Low-contention scenario
  - 3 consumers
  - 1 producer
- Heterogeneous data types
- Results
  - ~36% faster



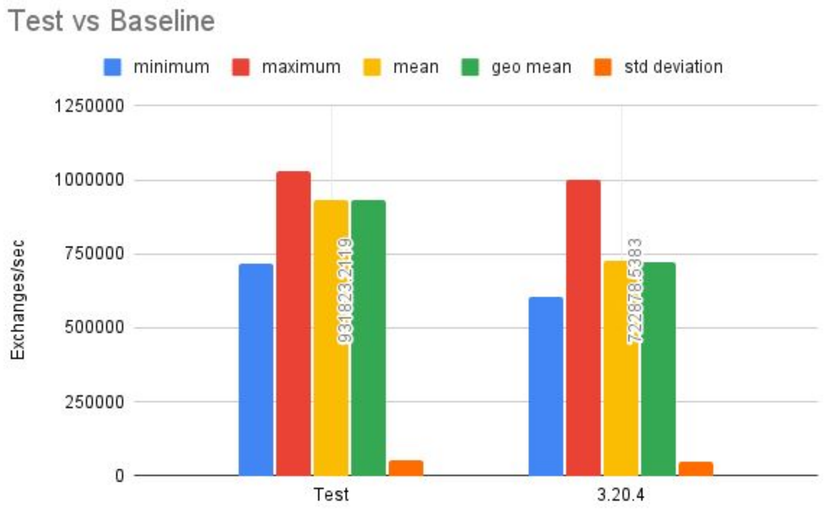
Performance improvements: disruptor

- Medium-contention scenario
  - 6 consumers
  - 5 producer
- Heterogeneous data types
- Results
  - ~36% faster



Performance improvements: disruptor

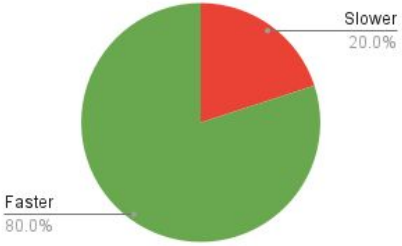
- High-contention scenario
  - 9 consumers
  - 9 producer
- Heterogeneous data types
- Results
  - ~29% faster



Performance improvements: overall

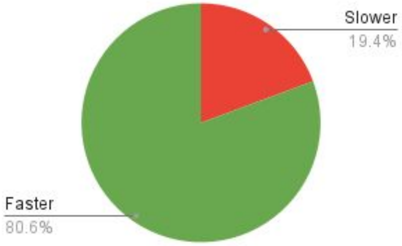
SEDA

- 4.x vs 3.20.4



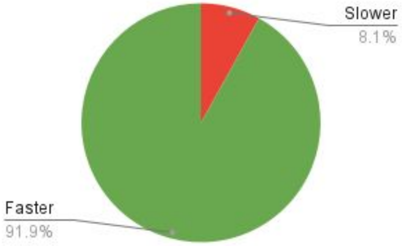
Disruptor

- 4.x vs 3.18.6



Disruptor

- 4.x vs 3.20.4



Performance Improvements

To learn more about the performance improvements check our blog post Hunting performance monsters on the back of a Camel.

Planning the upgrade

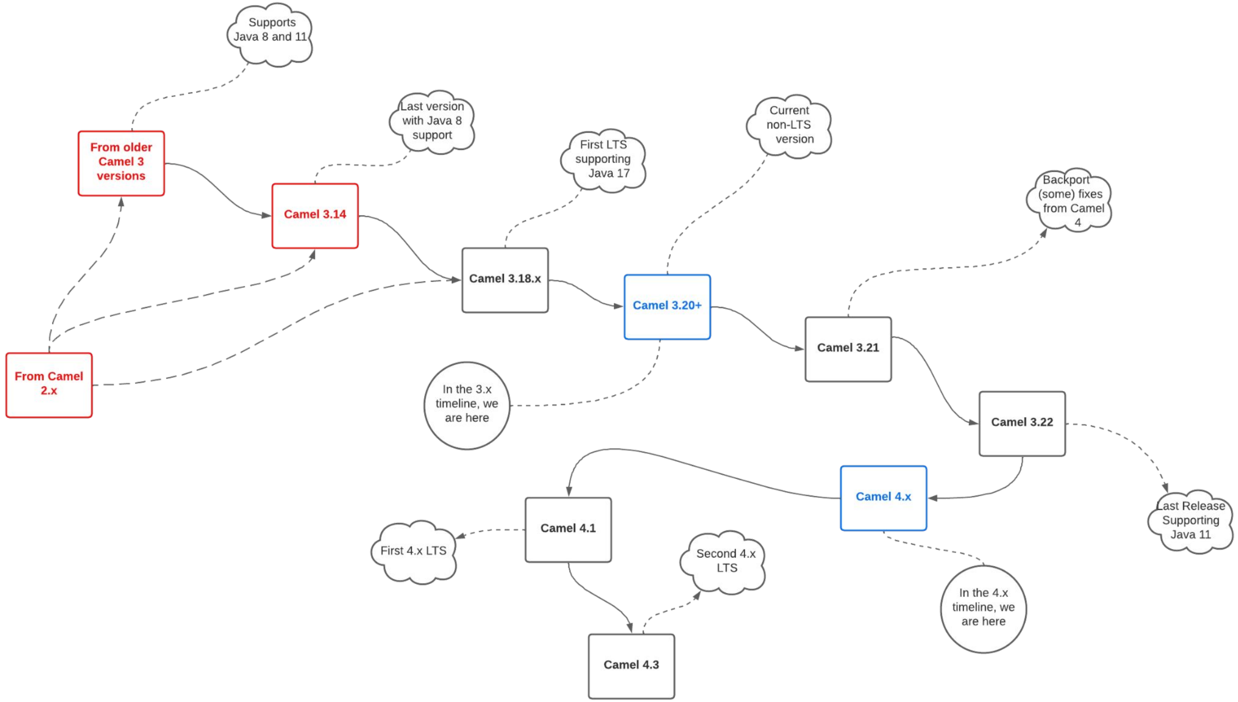
What to expect?

- Easier migration from Camel 3 to Camel 4
- No DSL incompatibilities

Planning the upgrade

- Downstream / commercial distributions
    - Talk to your vendor!
- Upstream / open source
    - Save time for testing
    - Provide feedback
    - Share your knowledge

The path to Apache Camel 4



Supports Java 8 and 11
From older Camel 3 versions
Last version with Java 8 support
Camel 3.14
First LTS supporting Java 17
Camel 3.18.x
Current non-LTS version
Camel 3.20+
From Camel 2.x
In the 3.x timeline, we are here
Backport (some) fixes from Camel 4
Camel 3.21
Camel 3.22
Last Release Supporting Java 11
Camel 4.x
Camel 4.1
First 4.x LTS
Second 4.x LTS
Camel 4.3
In the 4.x timeline, we are here

Migrating from Camel 3

- Should be easy if:
    - Using maintained technologies/components
    - Using one of:
        - Camel Spring Boot
        - Camel Quarkus
        - Camel K
    - If using Camel Core:
        - Relatively newer LTS of 3.x

Migrating from Camel 3

- May require a bit more effort
    - Using an older LTS version of 3.x (i.e.; older than 3.14)
- Will require code changes if:
    - Using internal APIs (ExtendedCamelContext, InternalPlugins, etc)
    - Using OSGI

User-impacting changes

- Logging
  - Slf4j-api upgraded to 2.x
- Exchange Patterns
  - `InOptionalOut` was removed
- API cleanups on `MainListener`
  - The method `configure()` was removed

Migrating from Camel 2

- Camel 3 was modularized
- Multiple contexts vs single context
- Java 11
  - Already old!
- Java 8 as a best effort

Closing comments

- Plan now
- Avoid unmaintained versions
- Re-evaluate usage of old technologies/standards
- Share the knowledge

Links

- Apache Camel Project
- Upgrade guides
  - Camel 2.x to 3.x migration guide
  - Camel 3.x to 4.x migration guide
  - Camel 3.x upgrade guides
- Release Notes
  - Camel 4.0.0-M2
  - Camel 4.0.0-M3