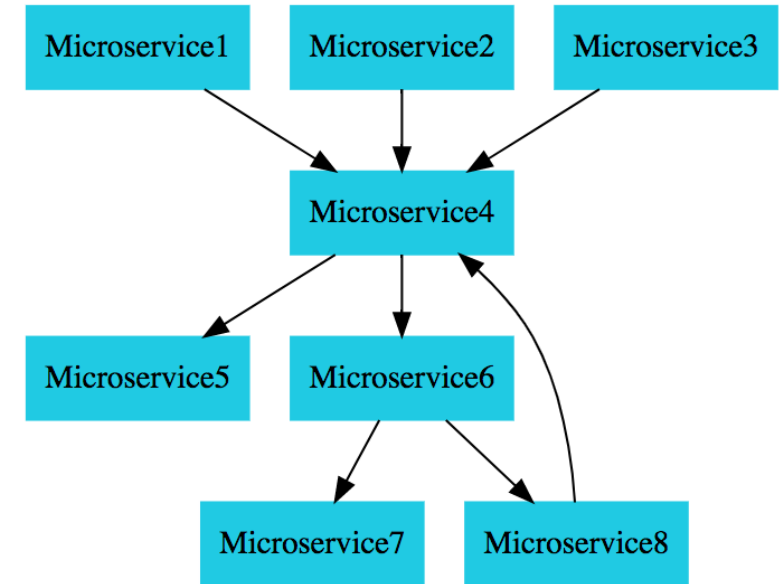


# Apache Camel



# Enterprise Integrations are Complex

- **Enterprises** have 100's of applications:
  - Complex communication patterns
  - Variety of transports - HTTP, Queues etc
  - Variety of protocols - HTTP, JMS, AMQP
- **Evolution of Cloud and Microservices** makes Enterprise Integration even more complex
- How can we **simplify Enterprise Integrations**?
  - Follow Enterprise Integration Patterns
- How to implement Enterprise Integration Patterns?
  - Use Apache Camel



# Apache Camel

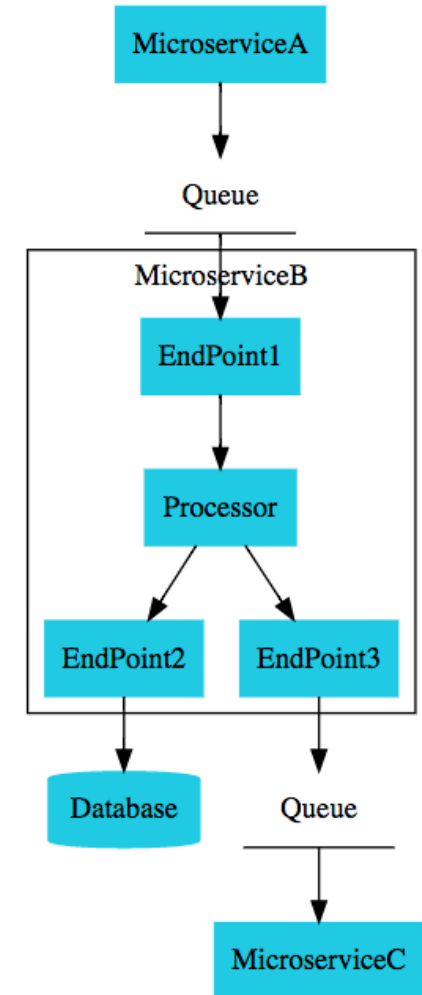
In 28  
Minutes

- Open Source **Enterprise** Integration framework
  - Easily integrate systems consuming or producing data
- Inspired by "**Enterprise Integration Patterns**" - **Gregor Hohpe and Bobby Woolf**
  - Evolved to Microservice Architectures and Cloud
- **Lean** - Lightweight and Extensible:
  - **Component architecture** keeps footprint low
  - Provides **100+** components for databases, message queues, APIs, cloud integration etc
  - Supports **200+** protocols, transports and data formats (& **300+** converters)
  - Provides **Domain Specific Language (DSL)** customized to suit needs of Application Integration



# Camel - Terminology

- Important Terminology:
  - **Camel Context** - (0..n) Routes + Components + ..
    - **Endpoint** - Reference to a queue, database or a file
    - **Route** - Endpoints + Processor(s) + Transformer(s)
    - **Components** - Extensions (Kafka, JSON, JMS etc)
    - **Transformation:**
      - Data format transformation - XML to JSON
      - Data type transformation - String to CurrencyConversionBean
  - **Message** - Body + Headers + Attachments
  - **Exchange** - Request + Response
    - Exchange ID
    - Message Exchange Pattern (MEP) - InOnly/InOut
    - Input Message and (Optional) Output Message



# Camel - Architecture

# Camel Context Example

```
//To start with
CamelContext camelContext = new DefaultCamelContext();
camelContext.addRoutes(new Route1());
camelContext.addRoutes(new Route2());
camelContext.addComponent("xyz",
    XyzComponent.xyzComponentAutoAcknowledge(connectionFactory));

camelContext.start();

//At the end
camelContext.stop();
```







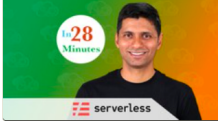
# References

Topic	Reference
Why Camel?	<a href="https://camel.apache.org/manual/latest/faq/why-the-name-camel.html">https://camel.apache.org/manual/latest/faq/why-the-name-camel.html</a>
Camel Examples	<a href="https://github.com/apache/camel-examples/tree/master/examples">https://github.com/apache/camel-examples/tree/master/examples</a>
Camel Spring Boot Configuration	<a href="https://camel.apache.org/camel-spring-boot/latest/spring-boot.html">https://camel.apache.org/camel-spring-boot/latest/spring-boot.html</a>
Complete Spring Boot Starters List	<a href="https://camel.apache.org/camel-spring-boot/latest/list.html">https://camel.apache.org/camel-spring-boot/latest/list.html</a>
Camel Spring Boot Examples	<a href="https://github.com/apache/camel-spring-boot-examples">https://github.com/apache/camel-spring-boot-examples</a>
Enterprise Integration Patterns	<a href="https://camel.apache.org/components/latest/eips/enterprise-integration-patterns.html">https://camel.apache.org/components/latest/eips/enterprise-integration-patterns.html</a>



# What Next?

- <https://github.com/in28minutes/learn>
  - Certifications
    - AWS, Azure and GCP
  - DevOps
    - Docker, Kubernetes, Azure DevOps, Terraform and Ansible
  - Serverless (AWS Lambda and Azure Functions)
  - Full Stack (with Angular and React)
  - Microservices (Spring Boot and Spring Cloud)

	<b>Master Docker with Java - DevOps for Spring Microservices</b> Create Docker Containers for Java Spring Boot Microservices. DevOps with Docker and Docker Compose for Java Developers. in28Minutes Official 4.6 ★★★★★ (970) 6.5 total hours • 62 lectures • All Levels
	<b>Master Kubernetes with Docker on Google Cloud, AWS &amp; Azure</b> Learn Kubernetes and Docker with Microservices ( Spring Boot + Java ) on Google Cloud GKE, AWS EKS & Azure AKS in28Minutes Official 4.6 ★★★★★ (1,455) 13 total hours • 137 lectures • All Levels
	<b>Go Java Full Stack with Spring Boot and React</b> Build Your First Java Full Stack Application with React & Spring Boot. Become a Java Full Stack Java Web Developer Now! in28Minutes Official 4.5 ★★★★★ (2,903) 12 total hours • 135 lectures • Beginner
	<b>Go Java Full Stack with Spring Boot and Angular</b> Become a Full Stack Java Developer. Build Your First Java Full Stack Application with Angular and Spring Boot. in28Minutes Official 4.5 ★★★★★ (6,318) 11 total hours • 125 lectures • All Levels
	<b>AWS Certified Developer Associate - AWS Certification</b> AWS Certified Developer Associate - AWS Certification. Achieve the AWS Developer Certification in in28minutes way! in28Minutes Official 4.6 ★★★★★ (1,195) 33.5 total hours • 521 lectures • All Levels
	<b>AWS Certified Solutions Architect - AWS Certification</b> AWS Certified Solutions Architect Associate - AWS Certification. Get AWS Solutions Architect Associate Certification. in28Minutes Official 4.6 ★★★★★ (2,706) 27.5 total hours • 419 lectures • All Levels
	<b>AWS Lambda &amp; Azure Functions - Go Serverless</b> Serverless with AWS Lambda, API Gateway & Azure Functions. Automate Serverless - SAM & Serverless Framework in node.js. in28Minutes Official 4.8 ★★★★★ (120) 12.5 total hours • 139 lectures • All Levels

