High performance asynchronous transaction orchestration with Java Reactive frameworks

intralot

The Business

Wagering applications

- Multiple channels (retail, web, mobile, TV)
- Significant volumes of transactions
- Rapidly spiking transactions on events
- Low response time requirements
- Strict regulatory restrictions (auditing, logging)
- Penalties for lost or mishandled transactions

The Goal

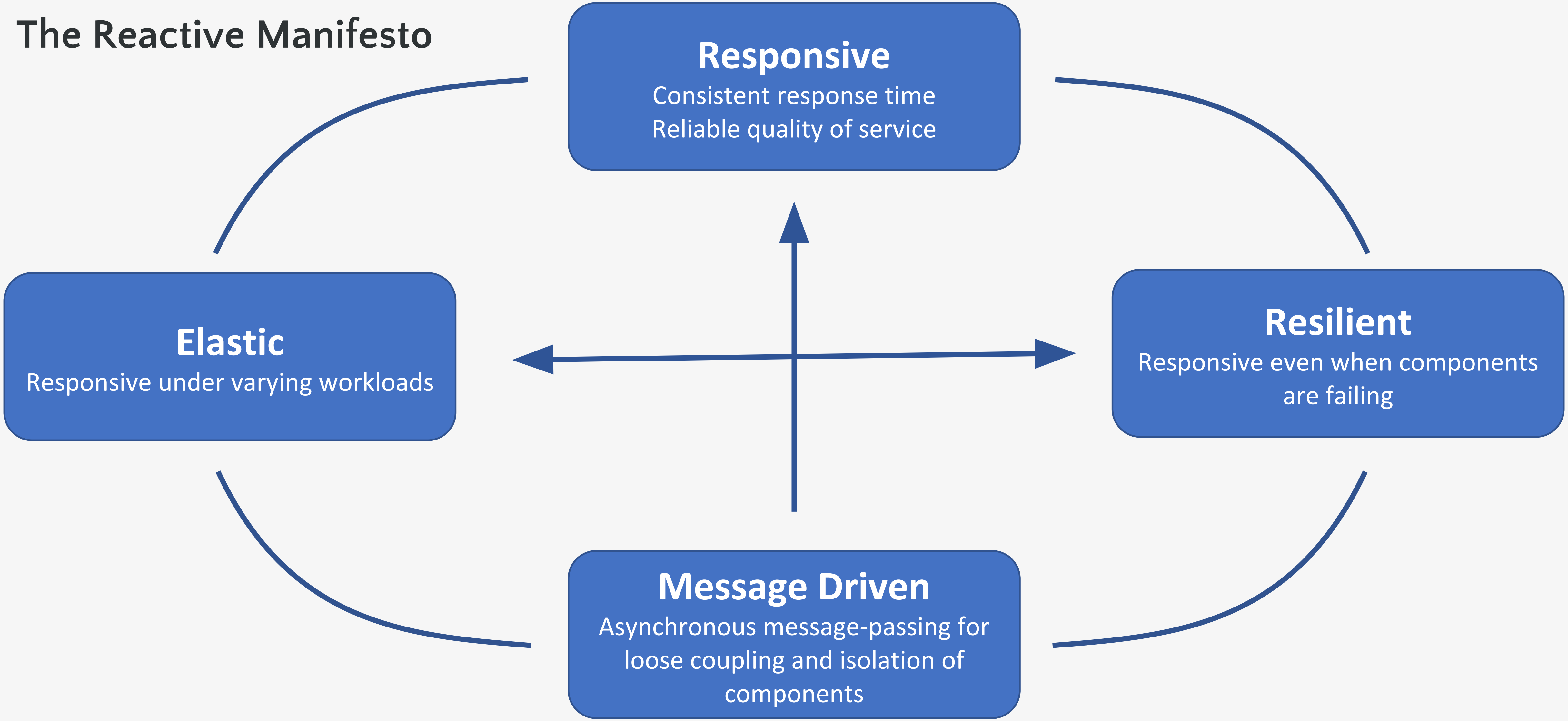
Create a gateway service that is able to

- Handle many requests
- In a timely manner
- With assured responses
- While following prescribed flows
- Which can be easily reconfigured
- Even at run-time

Easy, right? ☺

The Inspiration

The Reactive Manifesto



Responsive
Consistent response time
Reliable quality of service

Elastic
Responsive under varying workloads

Resilient
Responsive even when components are failing

Message Driven
Asynchronous message-passing for loose coupling and isolation of components

The Inspiration

Microservices

- Clear separation of business concerns
- Improved fault isolation
- Independent scaling of mission-critical components
- Independent technology stacks and development teams
- Faster testing of isolated services

The Inspiration

Containers

- Isolation of software from runtime environment
- Assured execution uniformity
- Reduced resource requirements
- Secure by default
- Docker standard maturity

The Inspiration

Kubernetes

- Orchestration of containers on clusters
- Cluster self-heal capabilities
- Service discovery and load balancing
- Automation of container rollouts and rollbacks
- Horizontal scalability automated based on triggers
- Multiple ways to apply configuration to containers (ConfigMaps, Env, Volumes)

The Tools

Java

- Mature and proven
- Extensive tooling
- Easy prototyping
- Existing talent pool
- Community support

The Tools

Java

- Netty
- SpringBoot 2 - WebFlux
- Jersey (JAX-RS)
- jBPM (JBoss)

The Tools

Netty

- Unified API for various transport types - blocking and non-blocking
- Based on a flexible and extensible event model
- Highly customizable thread model - single thread, one or more thread pools
- Better throughput, lower latency
- Reduced resource consumption
- Complete SSL/TLS support

The Tools

SpringBoot 2 – WebFlux

- **SpringBoot 2**
  - Extremely easy application dependency configuration
  - Single executable JAR artifact simplifies CI/CD
- **Spring 5**
  - A complete ecosystem with multiple tools (OAuth2, JPA, JMS)
  - Mature and stable
  - Dependency injection

The Tools

SpringBoot 2 – WebFlux

- **Reactor (extends RxJava)**
  - Pub-Sub model
  - Combines with Netty events to asynchronously handle requests
  - Reacts to the requests by filtering, sorting, batching, delay, back-pressure
- **Actuator**
  - Provides access to runtime application settings management via API
  - Needed for health and liveness checks in Kubernetes
  - Live application monitoring

The Tools

Jersey (JAX-RS)

- REST provider with specifications-compliant implementation
- Backwards compatibility with existing code-base (CXF)
- Fast and reliable
- Swagger (API presentation and testing)

The Tools

jBPM (JBoss)

- BPM objects help conceptualize the service flow
- Combines BPM flows with Reactive object implementations
- Provides run-time reconfiguration of flows
- Ease of design and development (Eclipse BPMN2 Modeler)

The Challenges

Making it all work together

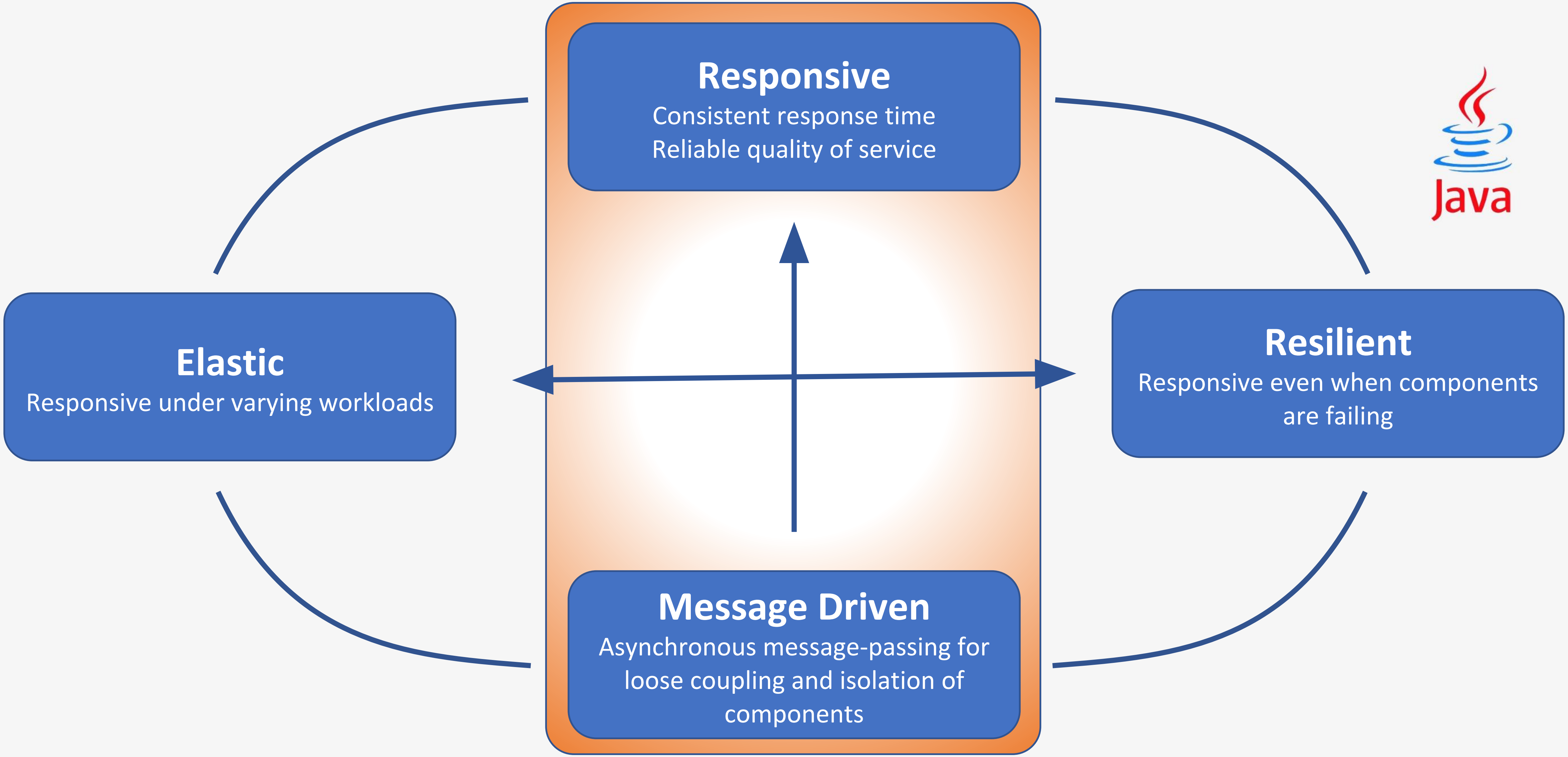
- Asynchronous design
  - Developer paradigm shift (Imperative to Reactive programming)
  - Flux objects to support back-pressure
  - Finalizers to assure asynchronous completion
  - Write-ahead of request information allows to recover the request
- Features
  - Configurable REST endpoints proxying
  - Replacing Spring REST implementation with Jersey
  - Integrating jBPM objects with Reactor

The Challenges

Making it all work together

- Performance considerations
  - Separate thread pools for events and operations
  - Reactor worker threads tuning
  - CPU and memory requirements are much lower (Tomcat)
- Java in containers
  - Security defaults
  - Minimizing packaging
  - Externalizing configuration

The Solution



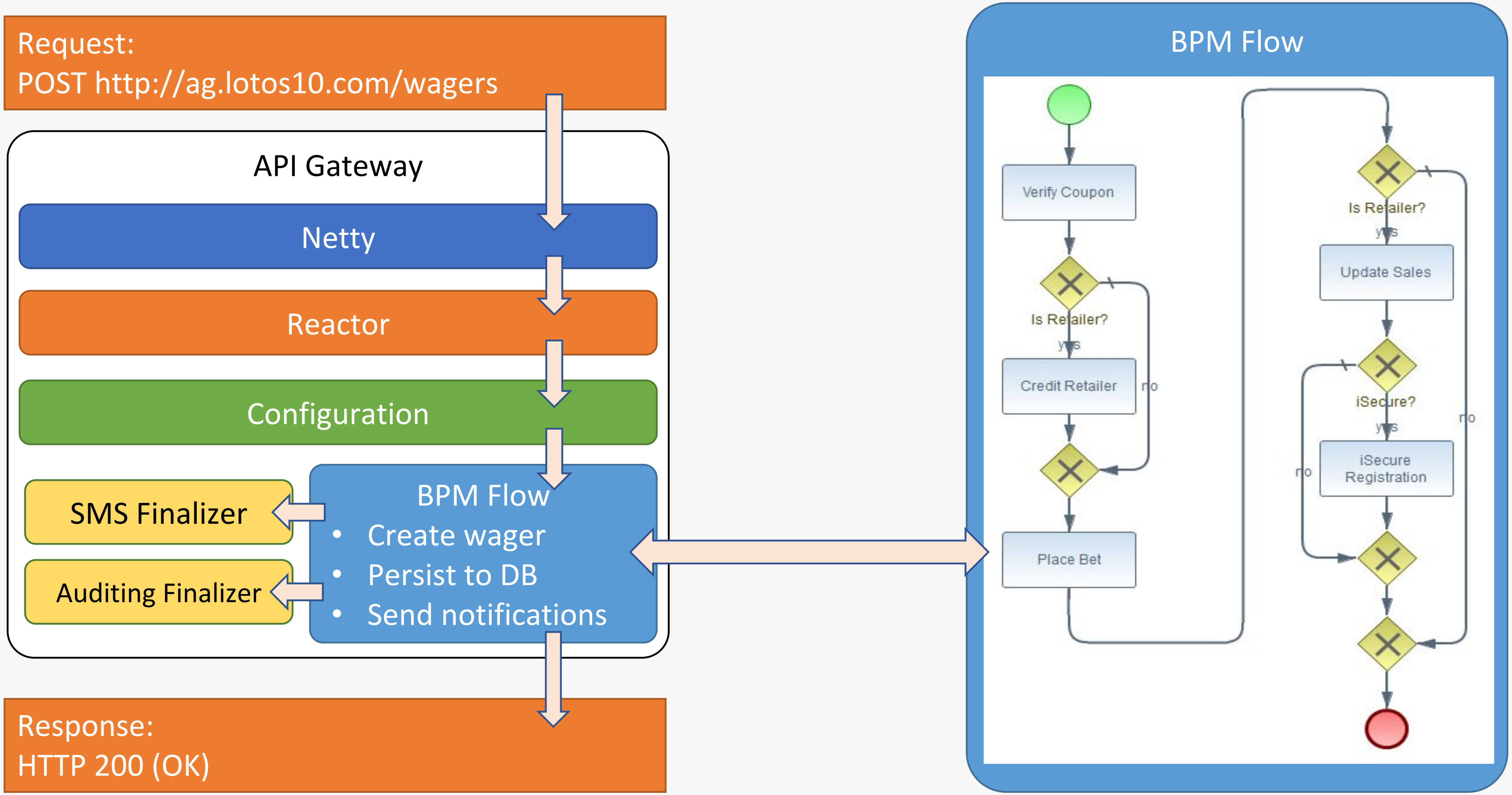
Responsive
Consistent response time
Reliable quality of service

Elastic
Responsive under varying workloads

Resilient
Responsive even when components are failing

Message Driven
Asynchronous message-passing for loose coupling and isolation of components

The Solution



Request:
POST http://ag.lotos10.com/wagers

API Gateway

Netty

Reactor

Configuration

SMS Finalizer

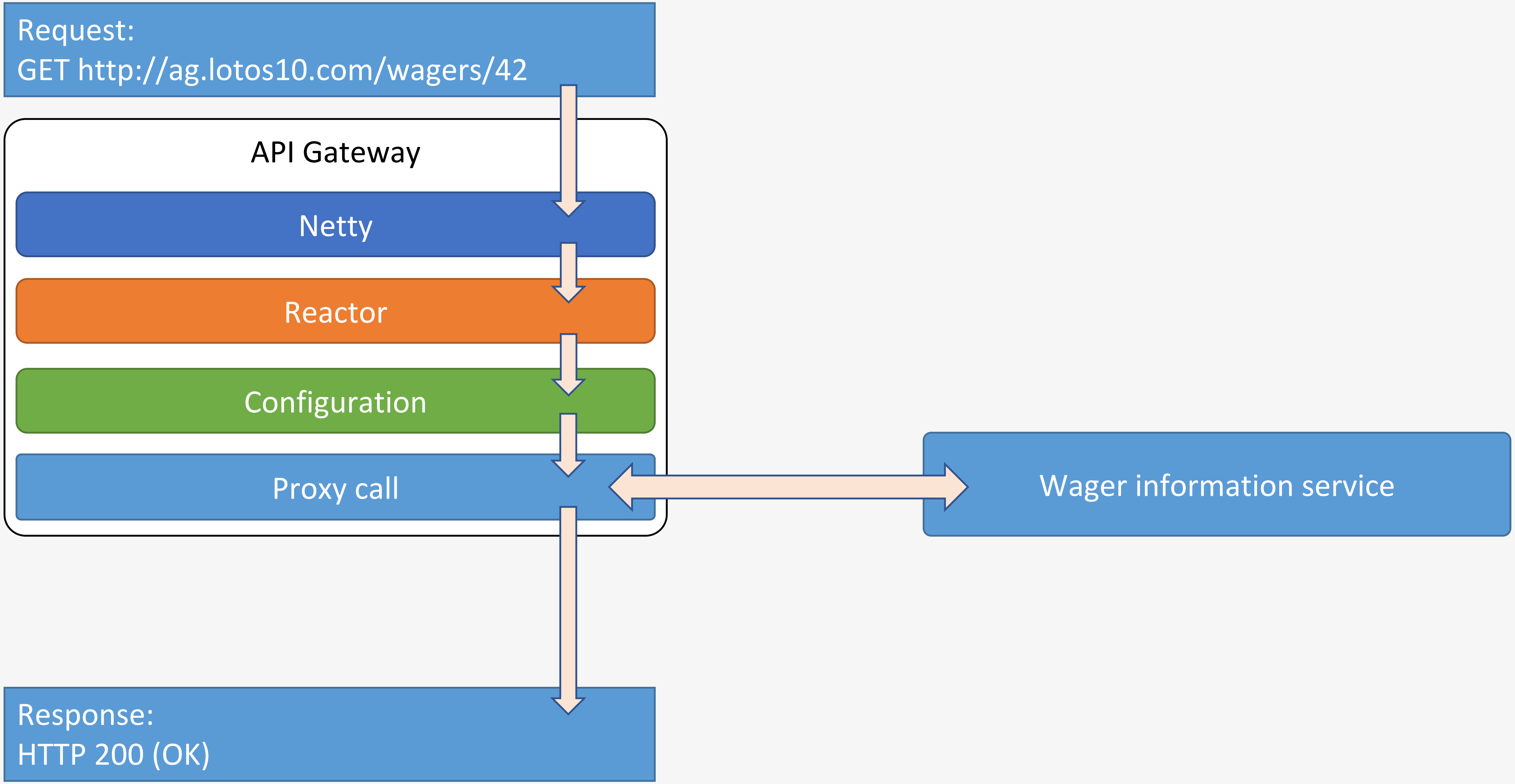
Auditing Finalizer

BPM Flow
- Create wager
- Persist to DB
- Send notifications

Response:
HTTP 200 (OK)

BPM Flow

The Solution



Request:
GET http://ag.lotos10.com/wagers/42

API Gateway

Netty

Reactor

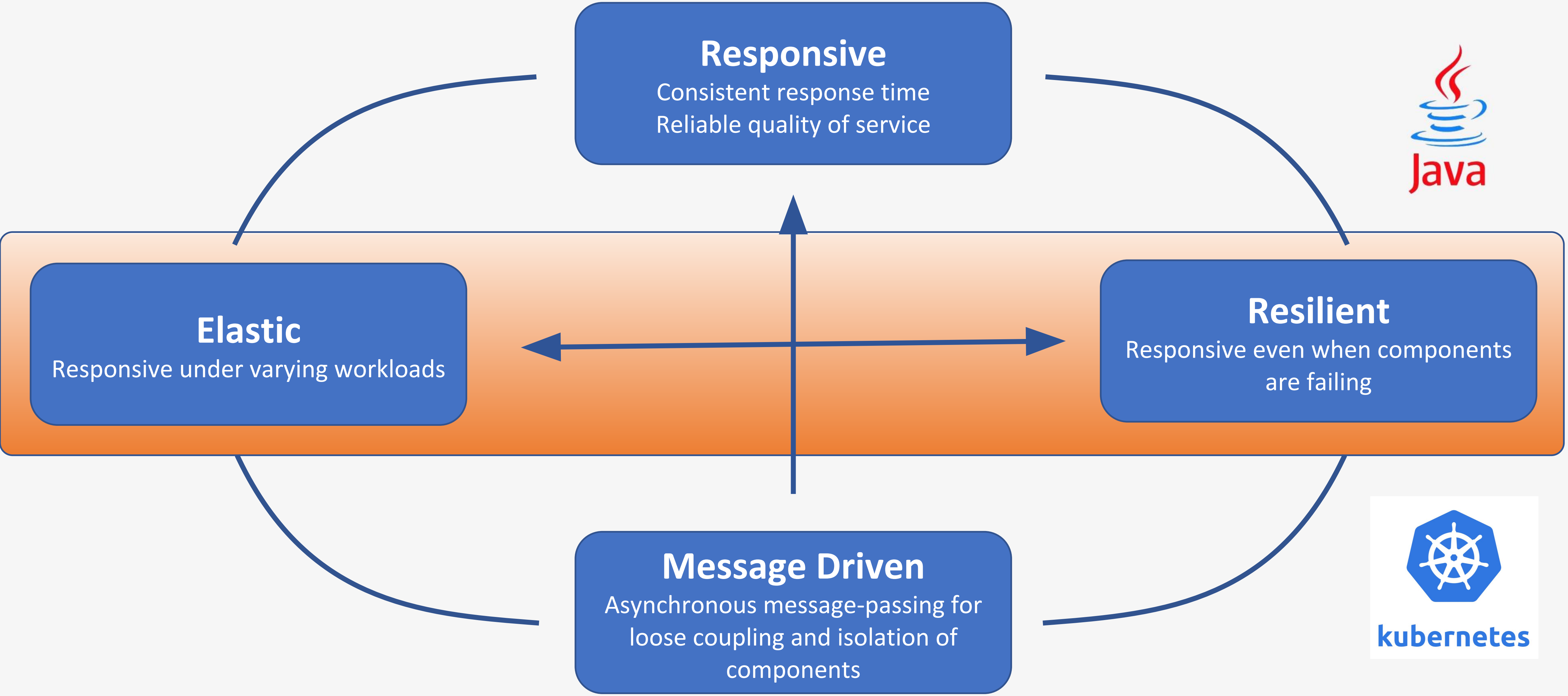
Configuration

Proxy call

Wager information service

Response:
HTTP 200 (OK)

The Solution



Responsive

Consistent response time
Reliable quality of service

Elastic

Responsive under varying workloads

Resilient

Responsive even when components are failing

Message Driven

Asynchronous message-passing for loose coupling and isolation of components

Java

kubernetes

The Solution

Elastic against varying workloads

- Flux uses delay, back-pressure to handle early overloads
- Horizontal Pod Autoscaler triggered by metrics
- Differential component scaling
- Node provisioning and decommissioning via Cluster Autoscaler

Resilient against failures

- Finalizers retry failed operations with optional back-off
- Write-ahead of request information allows to recover the request
- Kubernetes kills and reschedules pods that fail to respond in a timely manner
- Kubernetes reschedules pods from lost nodes
- Requests are directed to live services instances only

Your Questions

?

Thank you!

Thank you!

9:41 AM
100%
£100,00
mobile lottery
4:24
LIVE
KENO
FRENZY
€300 Bonus
A new draw every 5 minutes
PLAY OTHER LOTTERY GAMES
Lotto MONDO
TRIPLETS
€300K Prize Fund
FRENZY
€300 Bonus
Home
Favourites
Scan Ticket
Account
Menu

intralot