Architecture

Architecture

Camel uses a Java based Routing Domain Specific Language (DSL) or an Xml Configuration to configure routing and mediation rules which are added to a CamelContext to implement the various Enterprise Integration Patterns.

At a high level Camel consists of a CamelContext which contains a collection of Component instances. A Component is essentially a factory of Endpoint instances. You can explicitly configure Component instances in Java code or an IoC container like Spring or Guice, or they can be auto-discovered using URIs.

An Endpoint acts rather like a URI or URL in a web application or a Destination in a JMS system; you can communicate with an endpoint; either sending messages to it or consuming messages from it. You can then create a Producer or Consumer on an Endpoint to exchange messages with it.

The DSL makes heavy use of pluggable Languages to create an Expression or Predicate to make a truly powerful DSL which is extensible to the most suitable language depending on your needs. The following languages are supported

- Bean Language for using Java for expressions
- Constant
- the unified EL from JSP and JSF
- Header
- JSonPath
- JXPath
- Mvel
- OGNL
- Ref Language
- ExchangeProperty / Property
- Scripting Languages such as
    - BeanShell
    - JavaScript
    - Groovy
    - Python
    - PHP
    - Ruby
- Simple
    - File Language
- Spring Expression Language
- SQL
- Tokenizer
- XPath
- XQuery
- VTD-XML

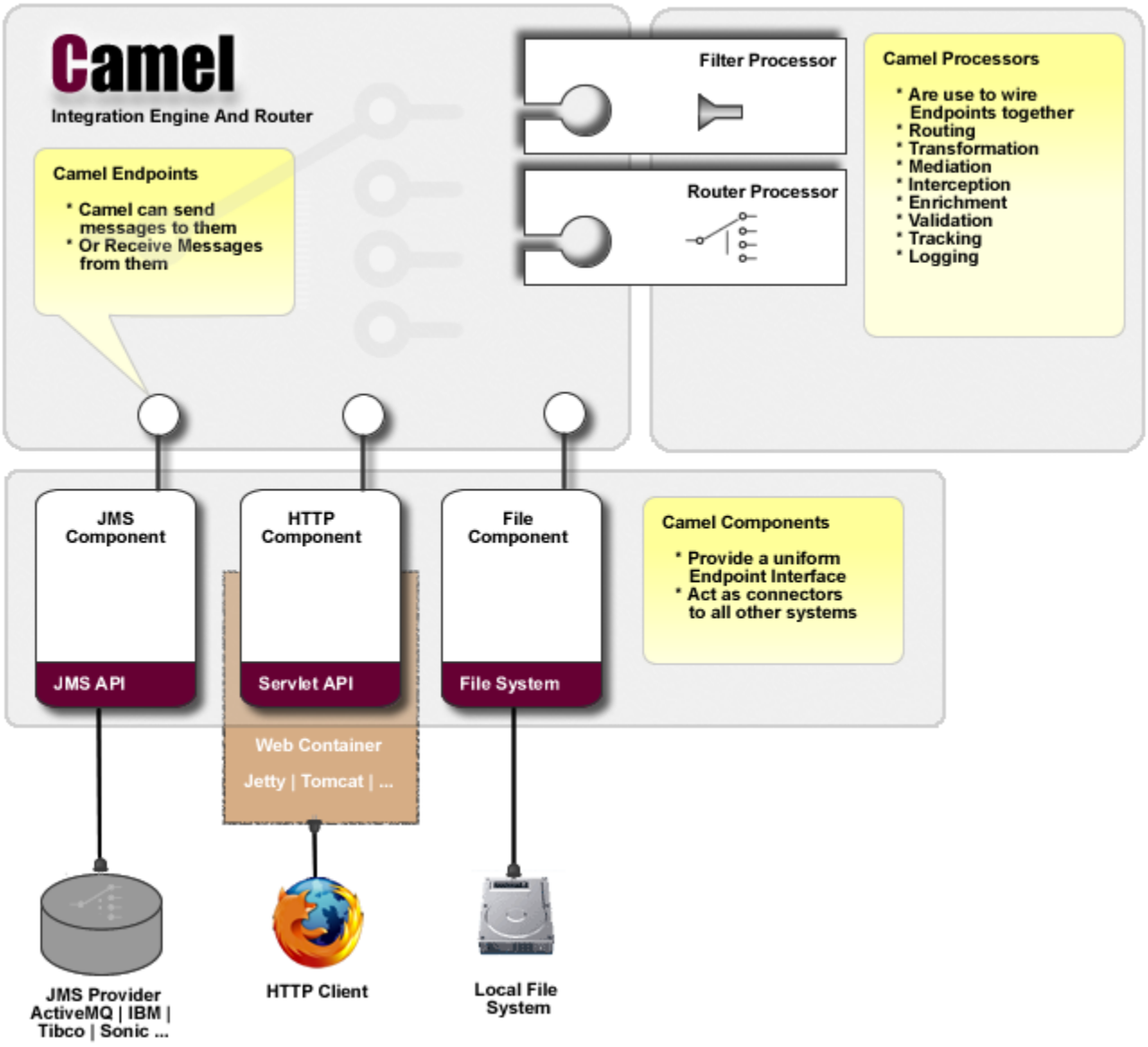
Most of these languages is also supported used as Annotation Based Expression Language.

Contents

The following links are to the individual parts of the Architecture.

- Expression
- HTTP-Session Handling
- Injector
- Intercept
- Inversion Of Control With Smart Defaults
- Languages
- Lifecycle
- OnCompletion
- Pluggable Class Resolvers
- Predicate
- Processor
- ProcessorFactory
- Registry
- RouteBuilder
- RoutePolicy
- Routes
- RX
- Security
- ServicePool
- Stream caching
- Threading Model
- ToAsync
- Tracer
- Transport
- Type Converter
- URIs
- UuidGenerator
- Xml Configuration

Diagram


Camel
Integration Engine And Router
Camel Endpoints
* Camel can send messages to them
* Or Receive Messages from them
Filter Processor
Router Processor
Camel Processors
* Are use to wire Endpoints together
* Routing
* Transformation
* Mediation
* Interception
* Enrichment
* Validation
* Tracking
* Logging
JMS Component
HTTP Component
File Component
JMS API
Servlet API
File System
Camel Components
* Provide a uniform Endpoint Interface
* Act as connectors to all other systems
Web Container
Jetty | Tomcat | ...
JMS Provider
ActiveMQ | IBM | Tibco | Sonic ...
HTTP Client
Local File System