

Integration between services and mediation

Oxford University
Software Engineering
Programme
April 2021



© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>



Through 2020, integration work will account for 50% of the time and cost of building a digital platform.

Use a Hybrid Integration Approach for Digital Transformation

© 2018 Gartner, Inc. and/or its affiliates. All rights reserved.

Smarter With Gartner®



© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Seven Cases of Integration (from Gartner)

API Management

API

Application Integratio

SAP

B2B Integration



Data Integration



Digital Integration Hub

APP
API

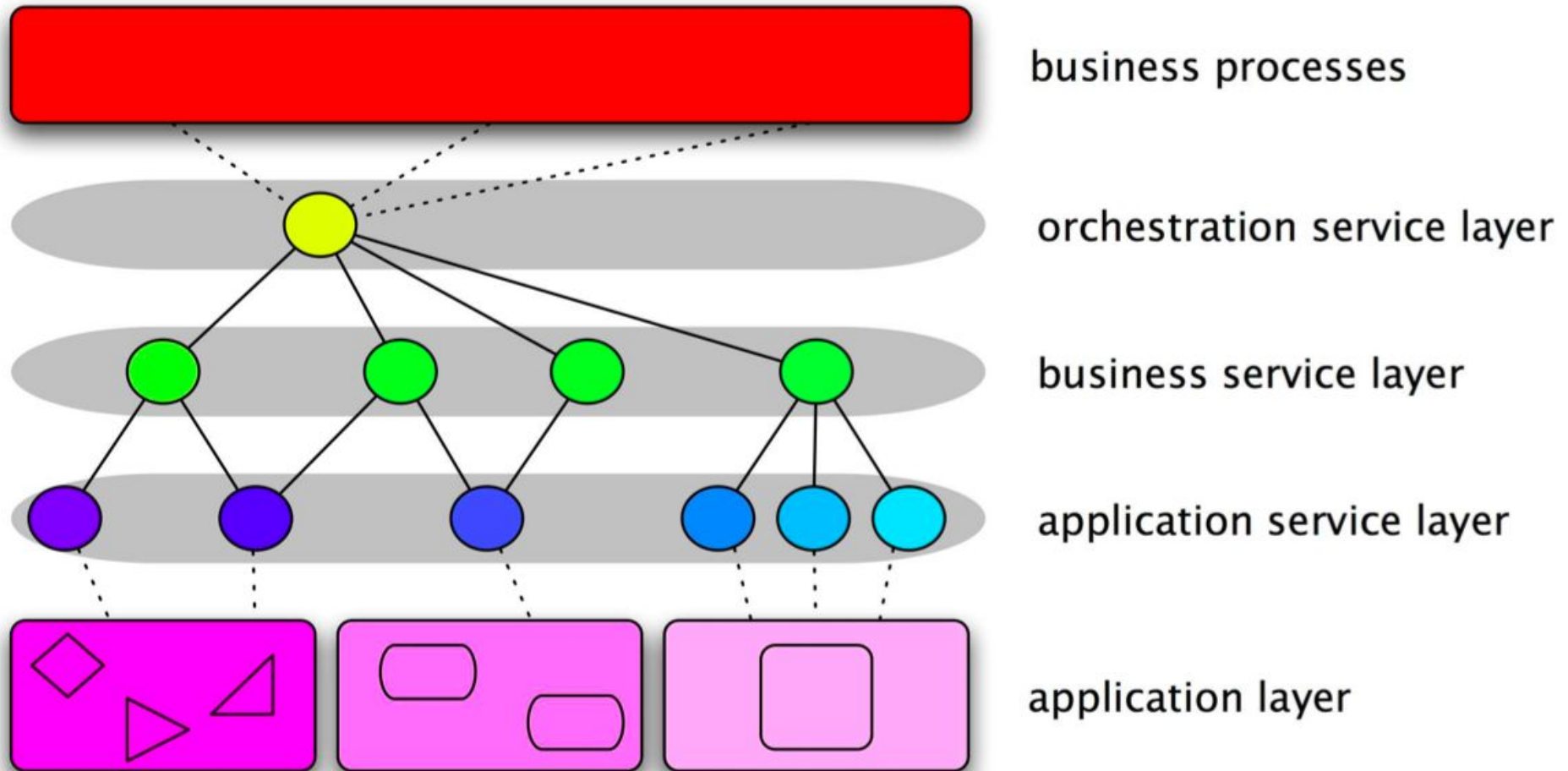
Event Streaming



IOT Integration



Recap on SOA model

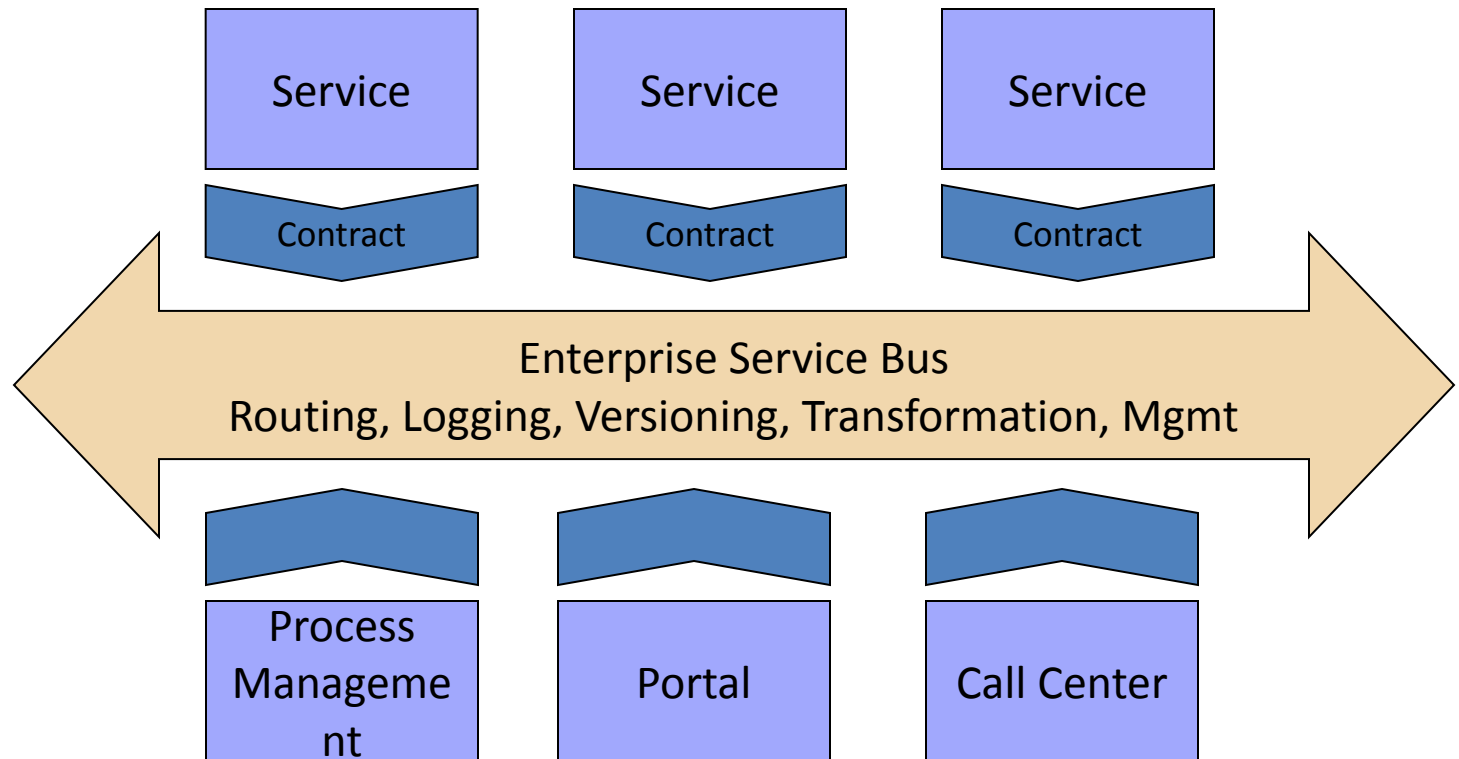


Enterprise Service Bus (ESB)

- A software architecture
 - A logical intermediary through which every message flows
 - Offers a policy based approach to decide what to do to each message or interaction
- The benefits of the gateway model
 - Without a physical hub and spoke
- Many vendors offer ESB products
 - Often a layer over an existing messaging framework



ESB as an implementation of SOA



FUSE - LoanBroker/eip/credit_response.eip_diagram - FUSE

File Edit Diagram Navigate Search Project Run Window Help

Segoe UI 9 B I A 100%

Project Explorer

- LoanBroker
 - Spring Elements
 - src
 - JRE System Library [jre1.6]
 - build
 - eip
 - credit_response.eip
 - credit_response.eip.d
 - null

credit_response.eip_diagram

```
graph LR; Start(( )) --> Check[Check...]; Check --> Decision{ }; Decision --> Accept[Accept]; Decision --> Reject[Reject]; Accept --> ProcessReq[Process Req]; ProcessReq --> InformCust[Inform Cust]; InformCust --> Merge(( )); Reject --> InvalidReq[Invalid Req]; InvalidReq --> Merge; Merge --> End(( ))
```

Palette

- Patterns
 - Resequencer
 - Routing Slip
 - Splitter
 - Throttler
- Endpoints
 - CXF
 - Direct
 - File
 - Generic
- Camel Proces...
 - Bean
 - Catch
 - Convert Body To
 - Finally

Problems Tasks Properties Console Servers

Check Processor

Properties

General

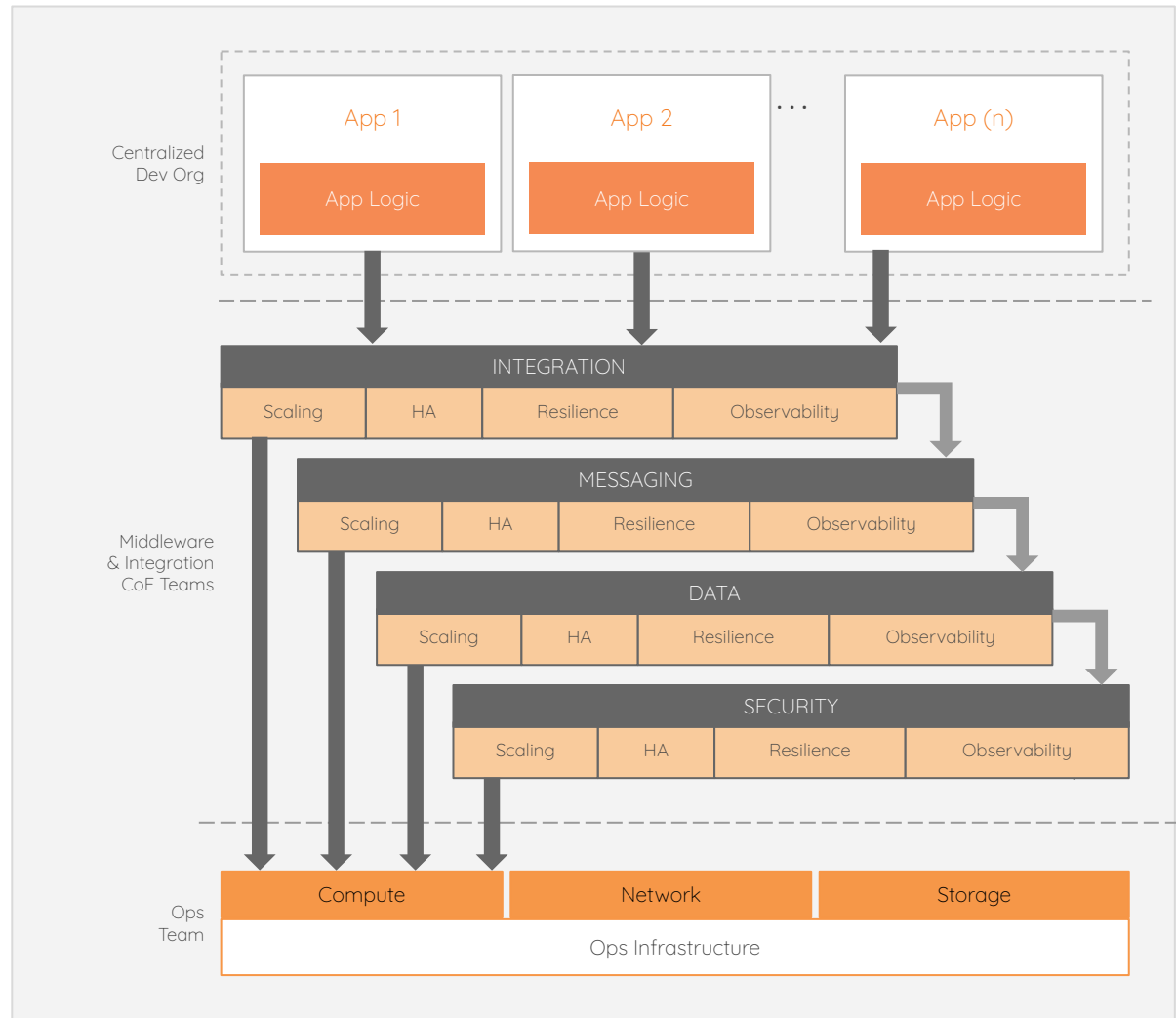
Destination Parameters

*Type queue

*Name checkResponse

Fast Waterfall

“Wagile”
“Fagile”



Enterprise Integration Patterns

Home - Enterprise Integratio x


www.eaipatterns.com

Offline Mail Inbox (124,820) - p 100+ WSO2 WSO2, Inc. - Calend + bitmark Shorten with bit.ly

**Enterprise
Integration
Patterns**

Home

[HOME](#) • [PATTERNS](#) • [RAMBLINGS](#) • [ARTICLES](#) • [TALKS](#) • [DOWNLOAD](#) • [LINKS](#) • [BOOKS](#) • [CONTACT](#)


 **Ramblings**

My ongoing thoughts about the present and future of integration, SOA and Web services. [\[see all\]](#)

[DDD - Diagram Driven Design](#)
(March 22, 2010)

[What Does It Mean to Use Messaging?](#)
(Feb 17, 2010)

[A Chapter a Day...](#)
(Feb 1, 2010)

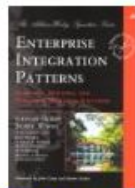
 **Upcoming Events**

Articles & Interviews

[Conversations Between Loosely Coupled Services](#)
(Video on [InfoQ](#))

[Developing in a Service-oriented World](#)
(Video on [InfoQ](#))

[SOA Patterns - New Insights or Recycled Knowledge?](#)
(Whitepaper)


Enterprise Integration Patterns

Gregor Hohpe, Bobby...

Best Price \$23.99 or Buy New \$50.74

Patterns and Best Practices for Enterprise Integration

This site is dedicated to making the design and implementation of integration solutions easier. The solutions and approaches described here are relevant for integration tools and platforms such as IBM WebSphere MQ, TIBCO, Vitria, SeeBeyond, WebMethods, or BizTalk, messaging systems such as JMS, WCF, or MSMQ, ESB's such as Sonic, Fiorano, ServiceMix, Mule, Apache Synapse, or WSO2, and SOA and Web-service based solutions.

All content on this site is original and is maintained by [Gregor Hohpe](#). I have been building integration solutions for large clients for many years and enjoy sharing my findings with the community. I hope you find this material insightful and useful. Please [contact me](#) if you have suggestions or feedback.

Enterprise Integration Patterns - The Book

Enterprise integration remains harder than it really should be. While integration is inherently complex, I felt that one of the major stumbling blocks is the lack of a common vocabulary and body of knowledge around asynchronous messaging architectures used to build integration solutions. Under the guidance of Martin Fowler and Kyle Brown, I teamed up with Bobby Woolf to create such a language in the form of 65 [integration patterns](#) (see the pattern links on the right).

The book [Enterprise Integration Patterns](#) provides a consistent vocabulary and visual notation to describe large

Integration Patterns

[Integration Patterns Overview](#)

[Table of Contents](#)

[Revision History](#)

Introduction

[Preface](#)

[Introduction](#)

[Solving Integration Problems using Patterns](#)

Integration Styles

[Introduction](#)

[File Transfer](#)

[Shared Database](#)

[Remote Procedure Invocation](#)

[Messaging](#)

Messaging Systems

[Introduction](#)

[Message Channel](#)

[Message](#)

[Pipes and Filters](#)

[Message Router](#)


[Message Translator](#)

[Message Endpoint](#)

Messaging Channels

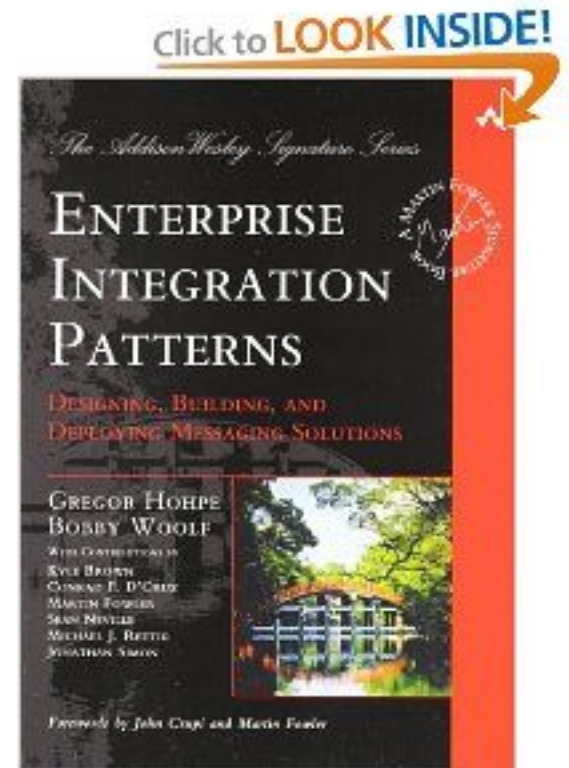
[Introduction](#)

[Point-to-Point Channel](#)



Enterprise Integration Patterns

- <http://www.eaipatterns.com/>
- The book
 - Enterprise Integration Patterns
 - Gregor Hohpe, Bobby Woolf



How does mediation / integration fit into Microservices / Containers?

“Smart Endpoints and Dumb Pipes”





Brownfield

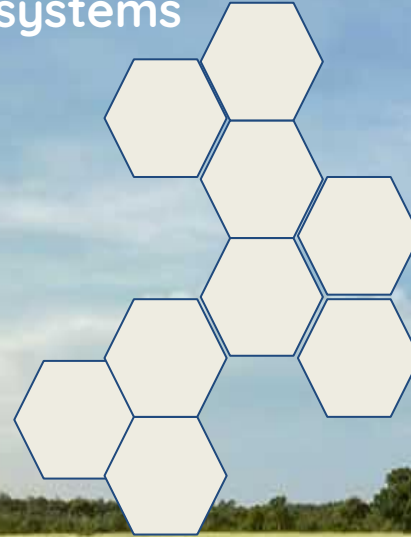
Greenfield

Core
Systems

Core
Systems

Core
Systems

Disaggregated
systems



Ballerina Language

- A new integration language and framework for Microservices, RESTful SOA
- Based on Swagger and Sequence Diagrams
- Textual and graphical are 100% interchangeable
- <https://ballerina.io>



Ballerina diagram and language

The screenshot displays the Ballerina IDE interface. The left pane shows the source code for `demo.bal`, which defines an HTTP service `hello` with a `listener` binding. The service has a `POST` endpoint `hi` that interacts with a `git` resource. The right pane shows the corresponding Ballerina Diagram, a sequence diagram illustrating the interaction between the `caller`, the `default` service endpoint, and the `git` resource. The diagram shows a `request.js` message from the caller to the service, followed by a `createIssue` call to the `git` resource, and a `respond(resp)` message from the service back to the caller.

```
18
19 @http:ServiceConfig {
20     basePath: "/"
21 }
22 service<http:Service> hello bind listener {
23     @http:ResourceConfig {
24         path: "/",
25         methods: ["POST"],
26         body: "js"
27     }
28 }
29 hi (endpoint caller, http:Request request, json js
30     string title = check <string>js.title;
31     string content = check <string>js.content;
32     github4:Issue iss = check
33     |git->createIssue("pzfreo", "btest", title
34     json resp = {
35         key: "value",
36         id: untaint iss.id
37     };
38     _ = caller -> respond(resp);
39 }
40 }
```

hello <http:Service>

hi

Sequence diagram illustrating the interaction between the caller, the default service endpoint, and the git resource.

1. The caller sends a `request.js` message to the default service endpoint.

2. The default service endpoint sends a `createIssue("pzfreo", "btest", title, content, [], [])` message to the git resource.

3. The git resource returns a response (indicated by a dashed arrow) to the default service endpoint.

4. The default service endpoint sends a `respond(resp)` message back to the caller.

Resources

- Wikipedia!
 - http://en.wikipedia.org/wiki/Enterprise_bus
- Books
 - David Chappell: ESB
 - Open Source ESBs in Action
- Open Source
 - synapse.apache.org
 - wso2.com/products/enterprise-service-bus
 - servicemix.apache.org

