

Integration between services and mediation

Oxford University
Software Engineering
Programme
December 2018



© 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

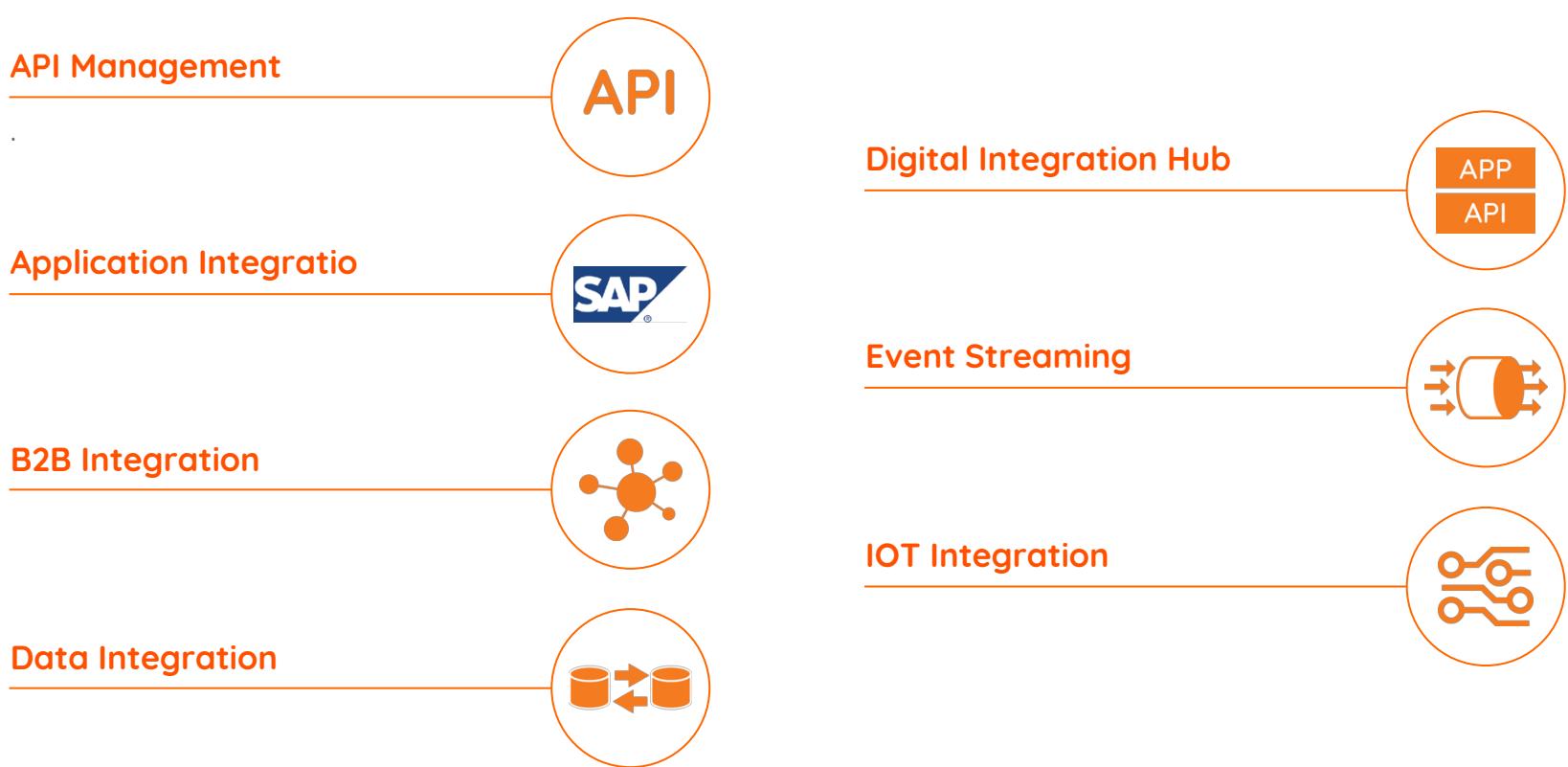
© 2018 Gartner, Inc. and/or its affiliates. All rights reserved.

Smarter With 

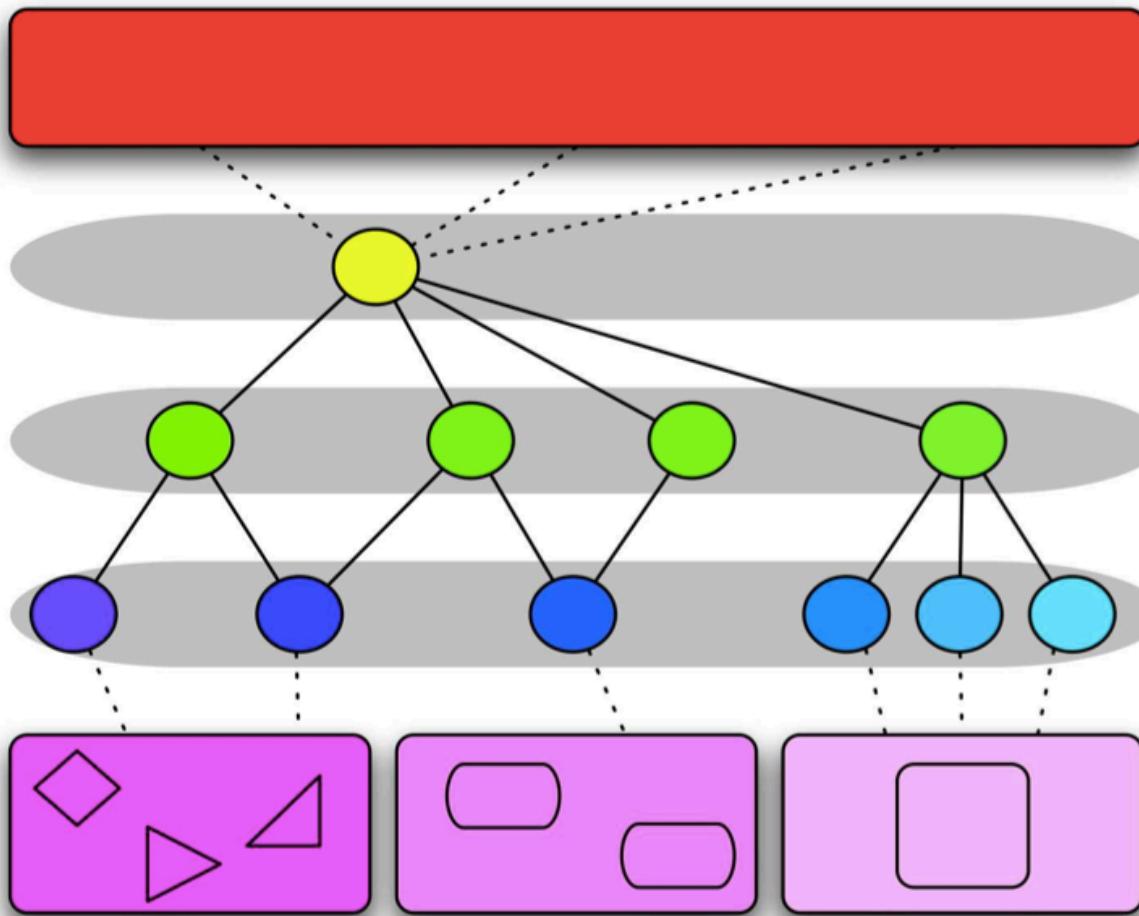


© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Seven Cases of Integration (from Gartner)



Recap on SOA model



business processes

orchestration service layer

business service layer

application service layer

application layer



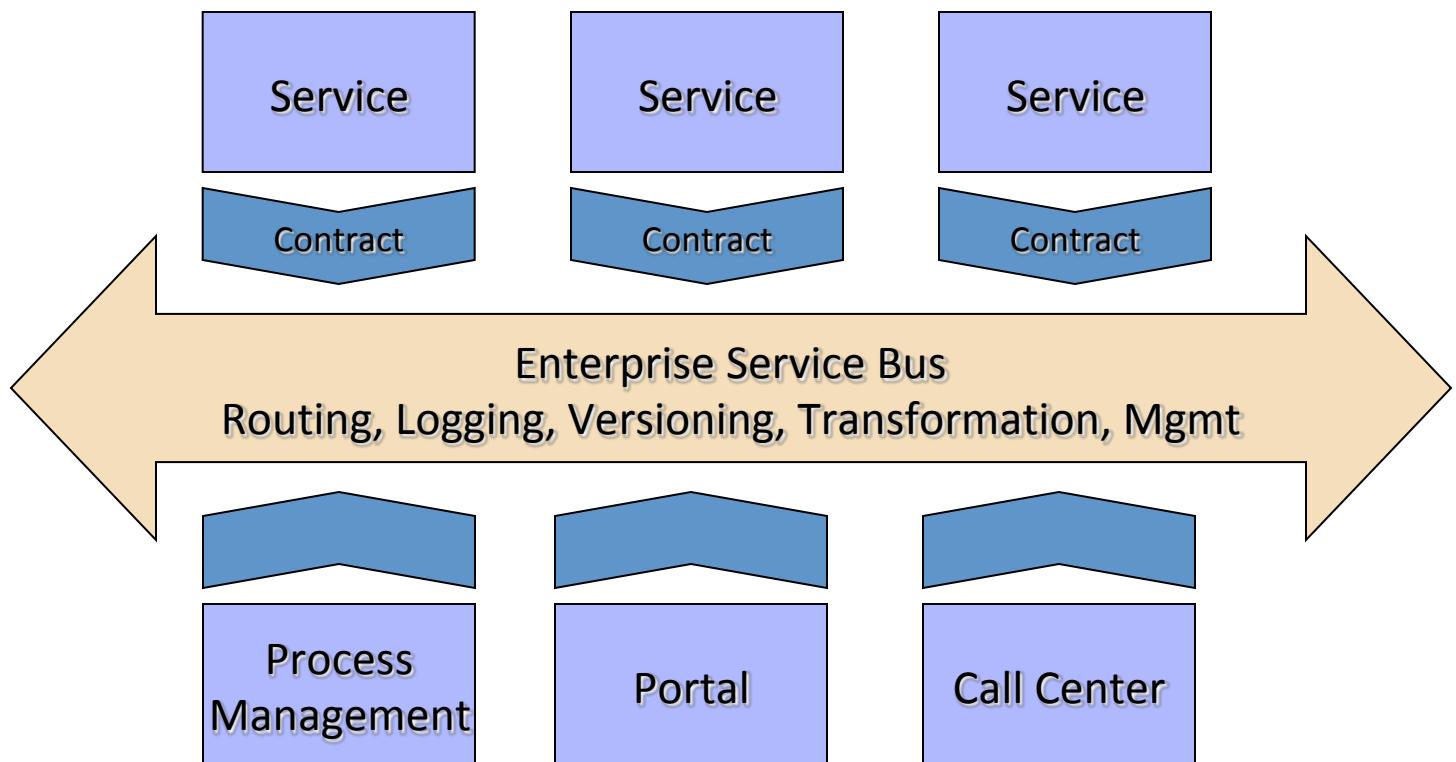
© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

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



Different approaches

- Point to Point
- Traditional EAI
- ESB
- Event Driven Architecture



© 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/>

Pros and Cons of an ESB

Pros

- Faster and cheaper accommodation of existing systems
- Increased flexibility: easier to change as requirements change
- Standards-based
- Scales to enterprise wide deployment
- Configuration rather than coding

Cons

- May end up with a proprietary solution
 - no common standards for the overall config and policies yet
- Requires more hardware to run
- New skills to learn to configure ESB
- Hard to get ROI on a small number of projects
- Centre of Excellence approach



ESB options

- Proprietary
 - IBM, Oracle, Tibco, SAP
- Open Source
 - Mule, Fuse, WSO2
 - Apache ServiceMix, Apache Synapse, Apache Camel



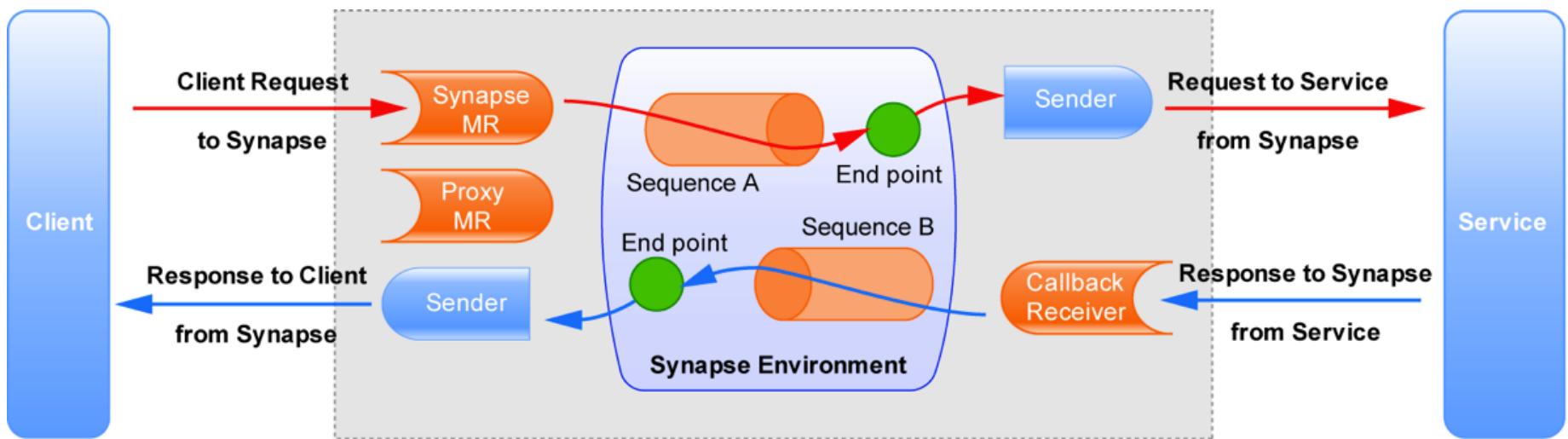
ESB models

- Almost all ESBs work on the same principle
- Message arrives
- Sequence of actions (Pipeline)
- Message is sent on

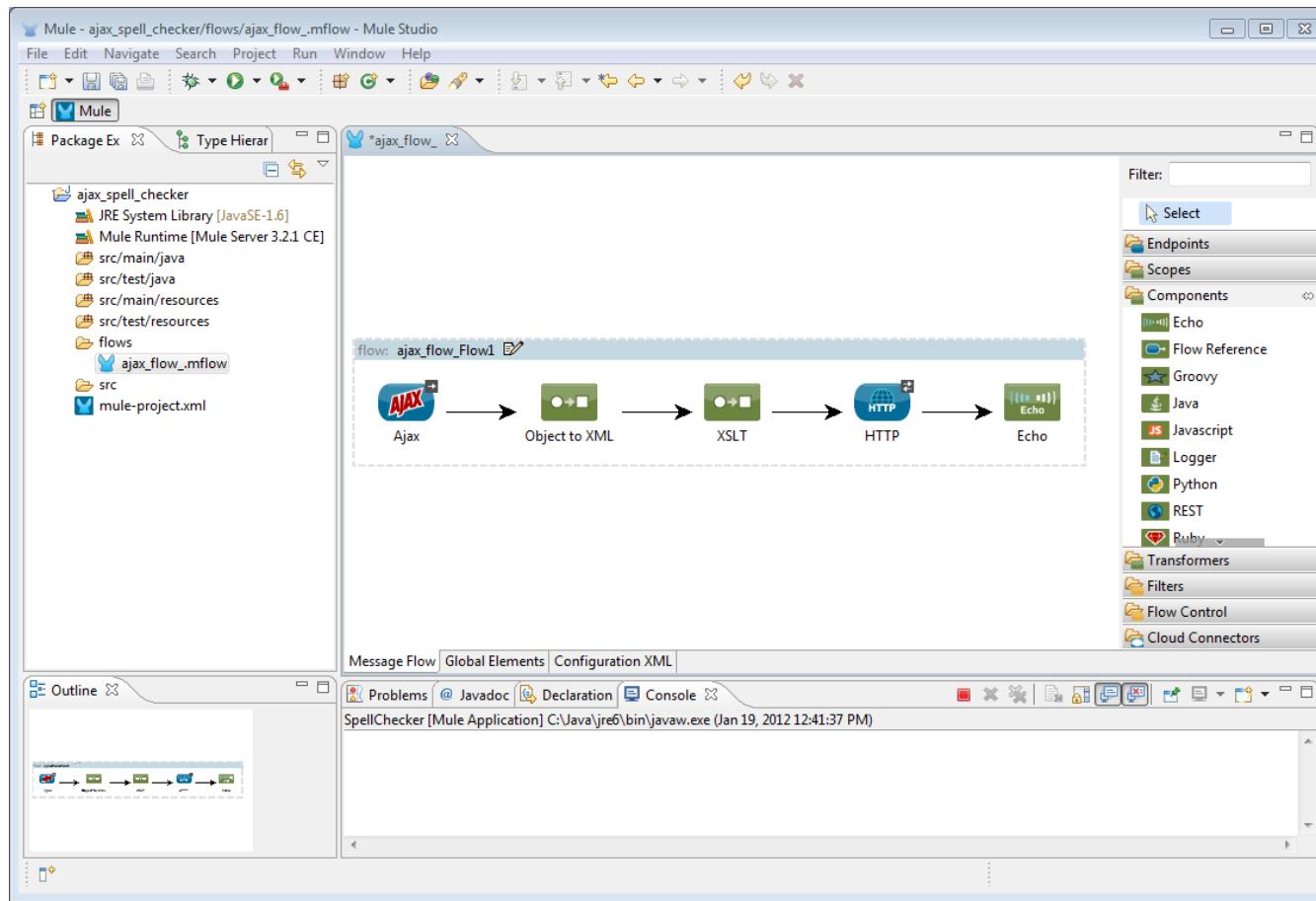


Graphically

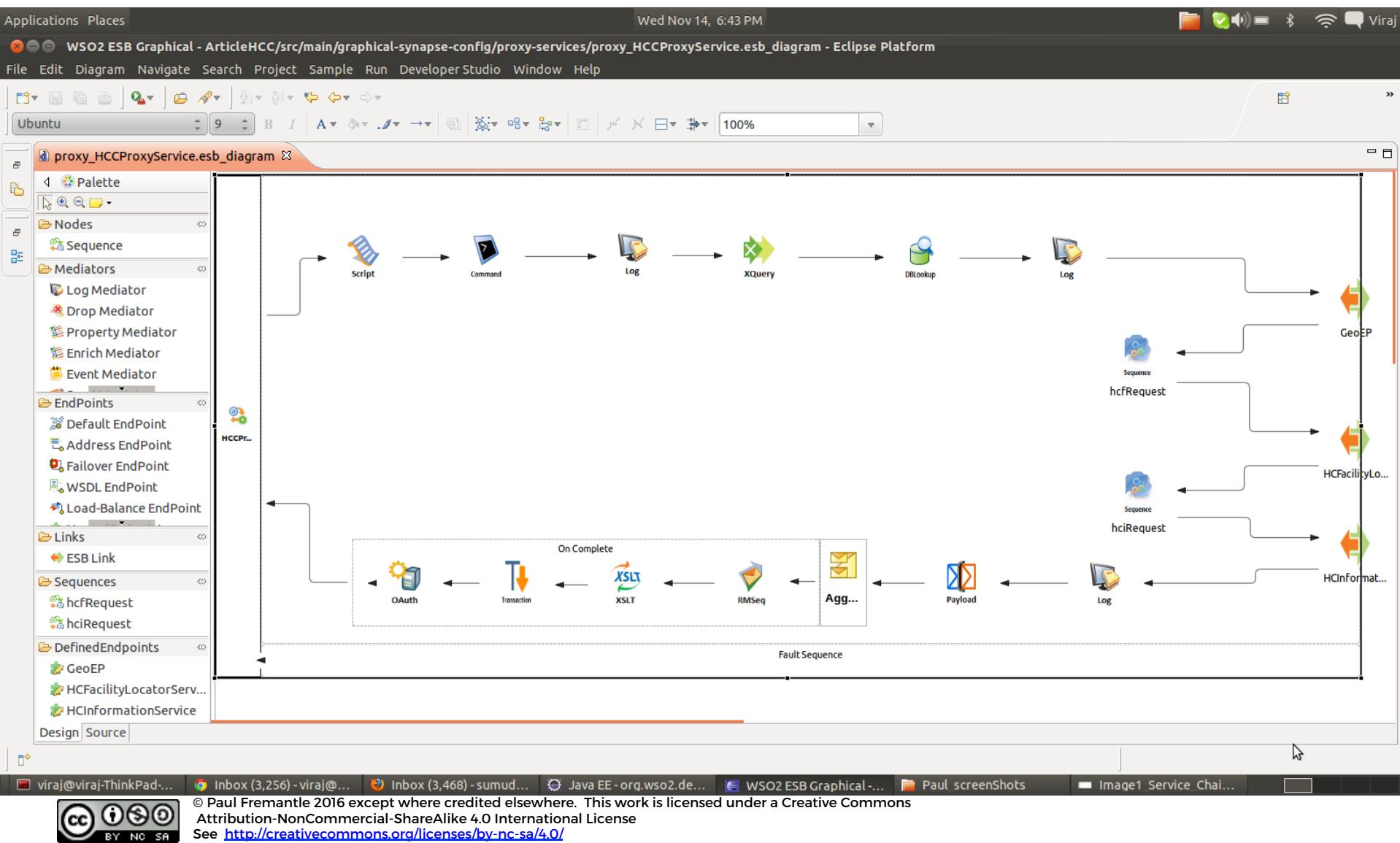
Apache Synapse terminology used

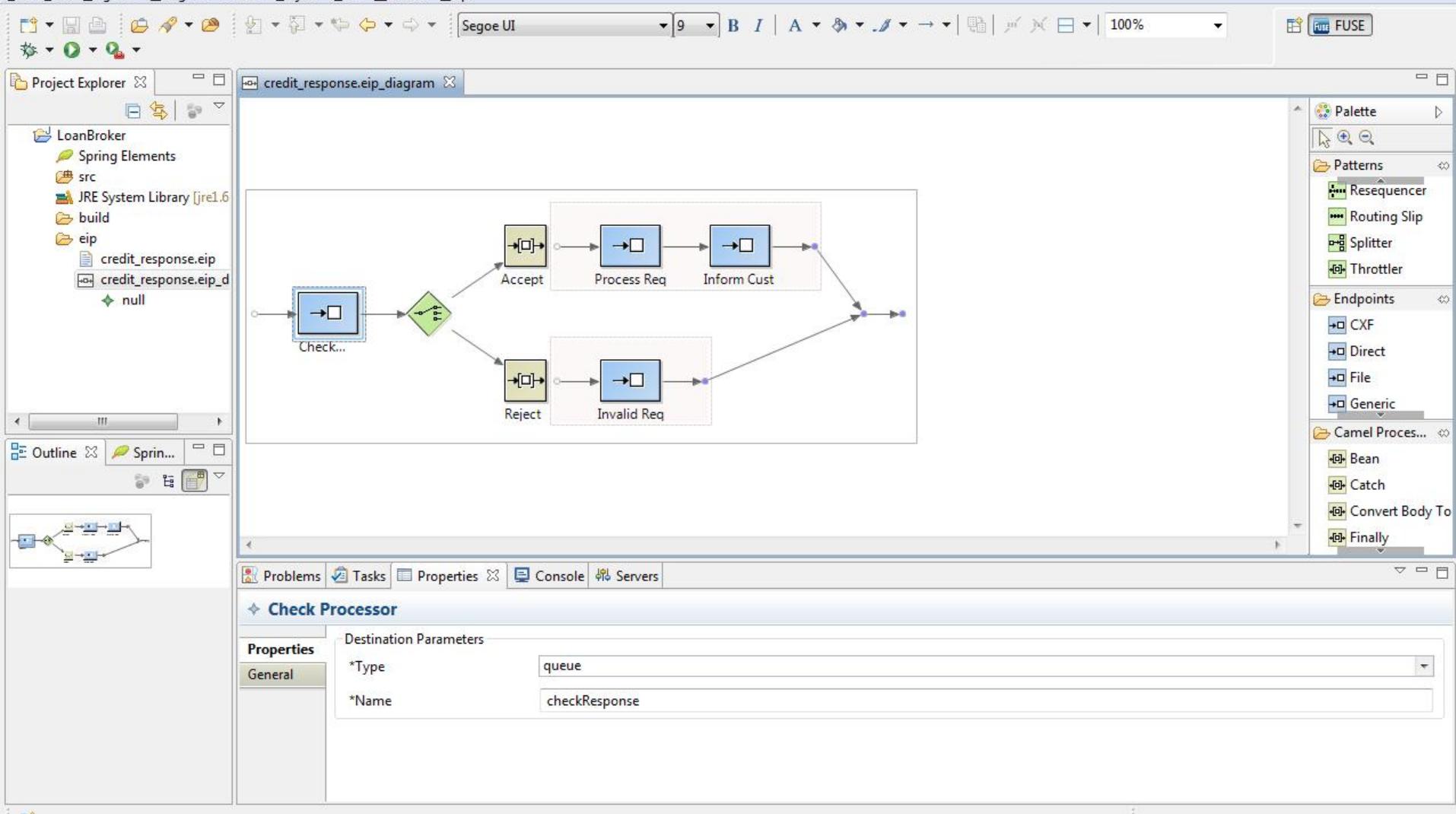


From some tooling



More Tooling





Enterprise Integration Patterns

Home - Enterprise Integratio × www.eipatterns.com

Offline Mail | Inbox (124,820) - pe | WSO2 | WSO2, Inc. - Calendar | + 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](#))

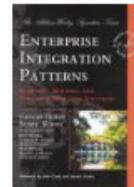
[SOA Patterns - New Insights or Recycled Knowledge?](#)
(Whitepaper)

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



The book *Enterprise Integration Patterns* provides a consistent vocabulary and visual notation to describe large-scale integration systems. It is authored by Gregor Hohpe, Bobby Woolf, and Martin Fowler, and published by Addison Wesley. The book is divided into several parts, each covering a different aspect of enterprise integration, such as messaging, shared databases, remote procedure invocation, and file transfer.

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

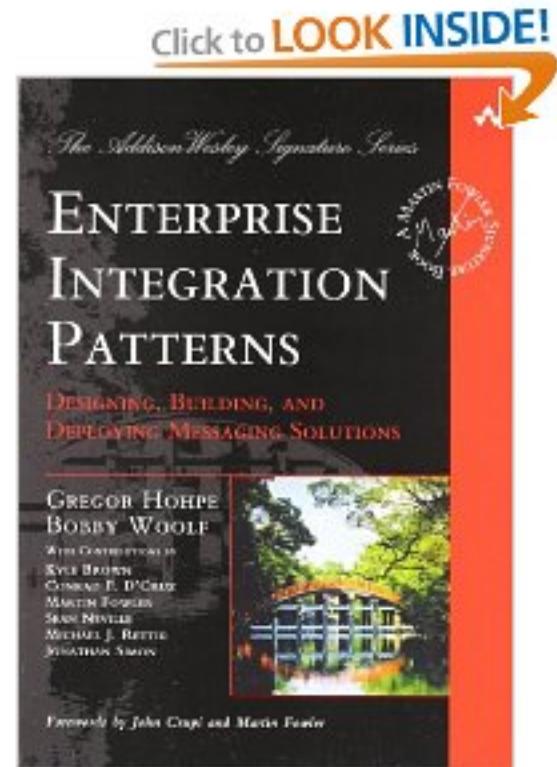
Messaging Channels

- Introduction
- Point-to-Point Channel

CC BY NC SA

Enterprise Integration Patterns

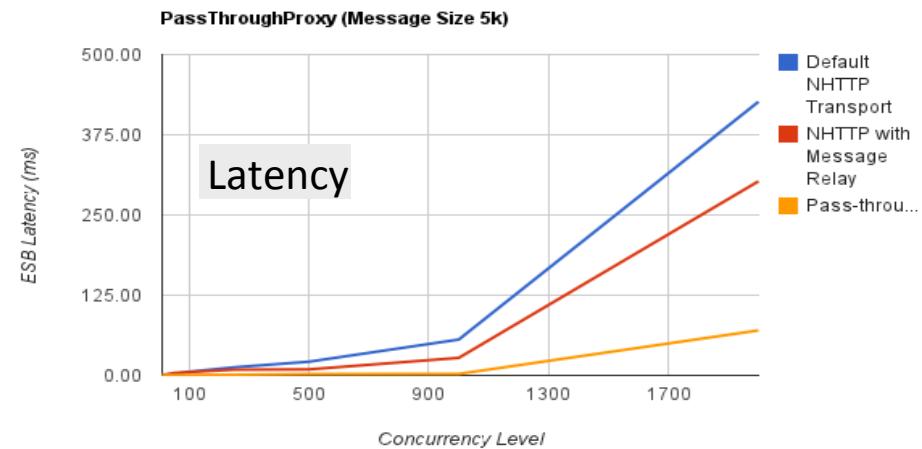
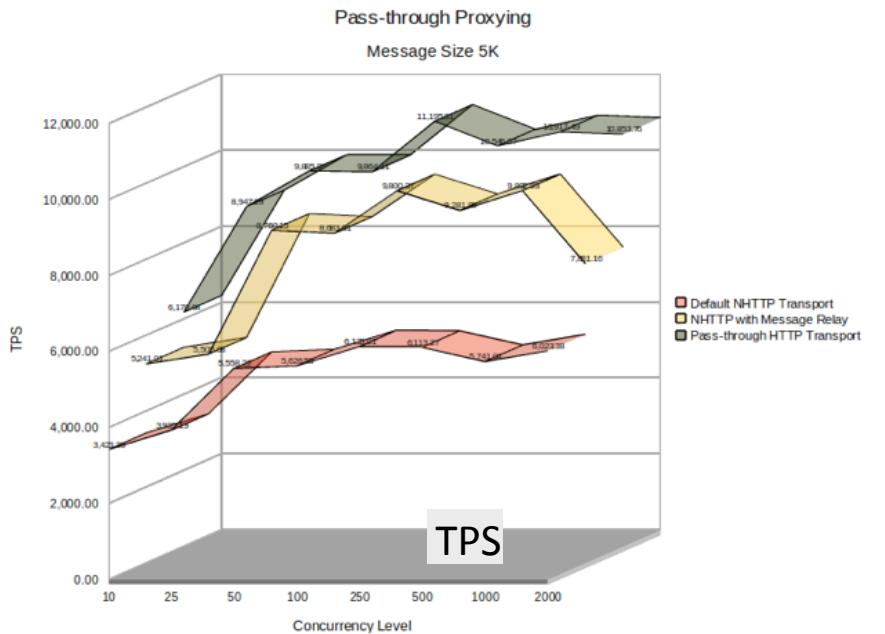
- [http://
www.eaipatterns.com](http://www.eaipatterns.com),
- The book
 - Enterprise Integration Patterns
 - Gregor Hohpe, Bobby Woolf



© 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/>

High Performance and Stability

- Supports 1000s of concurrent non-blocking HTTP transaction per server
- Pure streaming and Optimization using Message relay (on-demand processing of messages)
- Very Low latency (0.5 ms for Non-Blocking IO transport)
- Long Term Execution Stability with Low Resources Utilization
- Response Caching



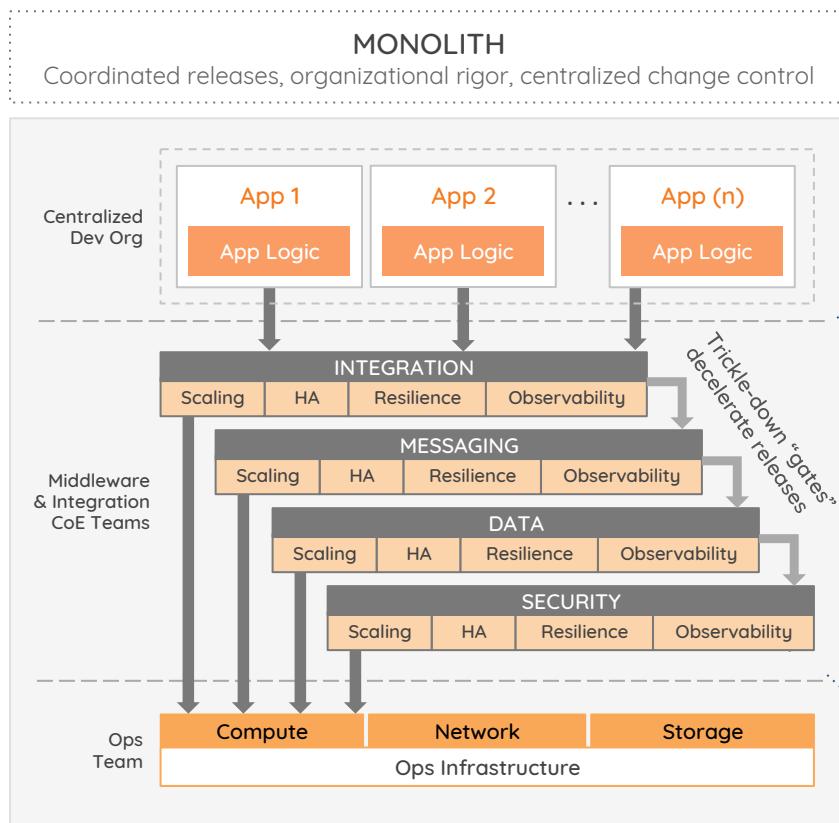
How does mediation / integration fit into Microservices / Containers?

“Smart Endpoints and Dumb Pipes”



© 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/>

ESB-led integration has created “gates” hindering agile development



PEOPLE

Center of Excellence teams add governance barriers to releases.



PROCESS

Middleware and integration systems create trickle-down “gates” decelerating releases.



TECHNOLOGY

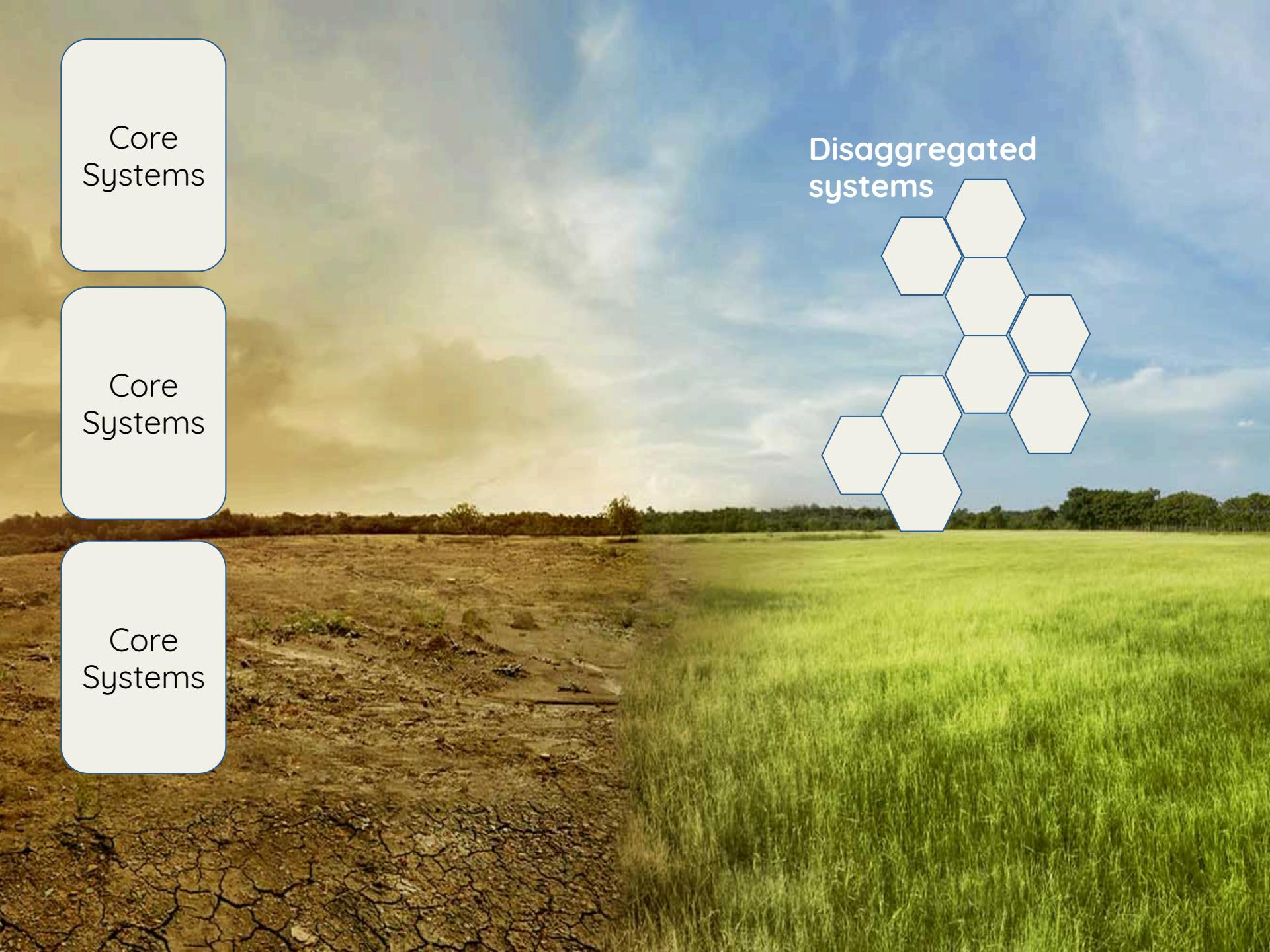
Middleware is a dependency slowing app dev teams.





Brownfield

Greenfield



Core
Systems

Core
Systems

Core
Systems

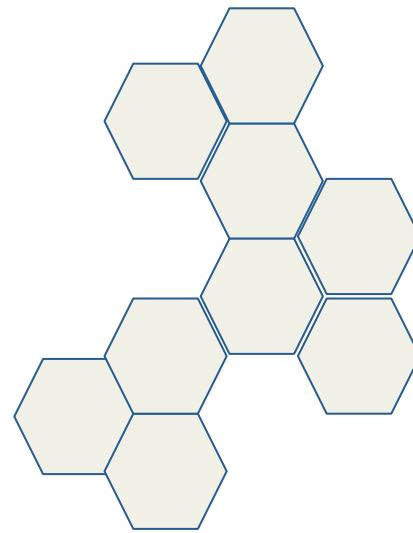
Disaggregated
systems

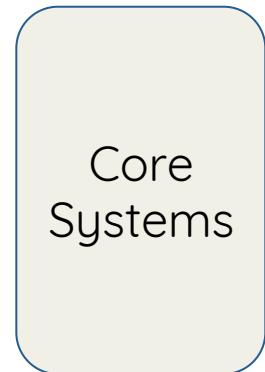
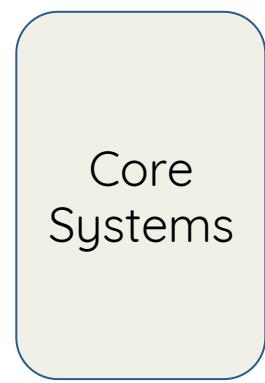
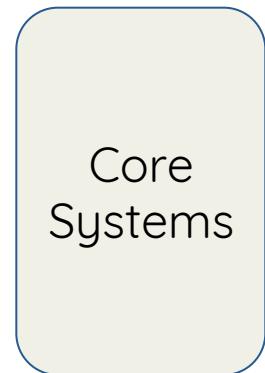


Core
Systems

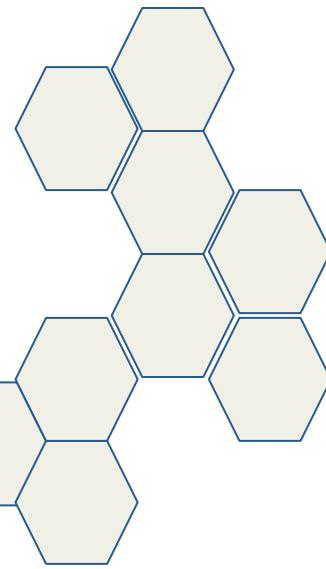
Core
Systems

Core
Systems





Governance
Self-Service
Identity
Dependency Management
Notification
Versioning



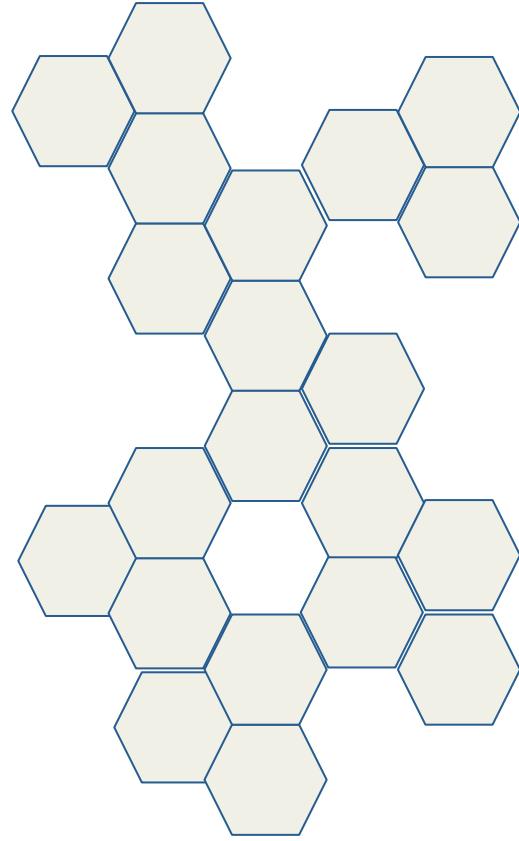
Core
Systems

Core
Systems

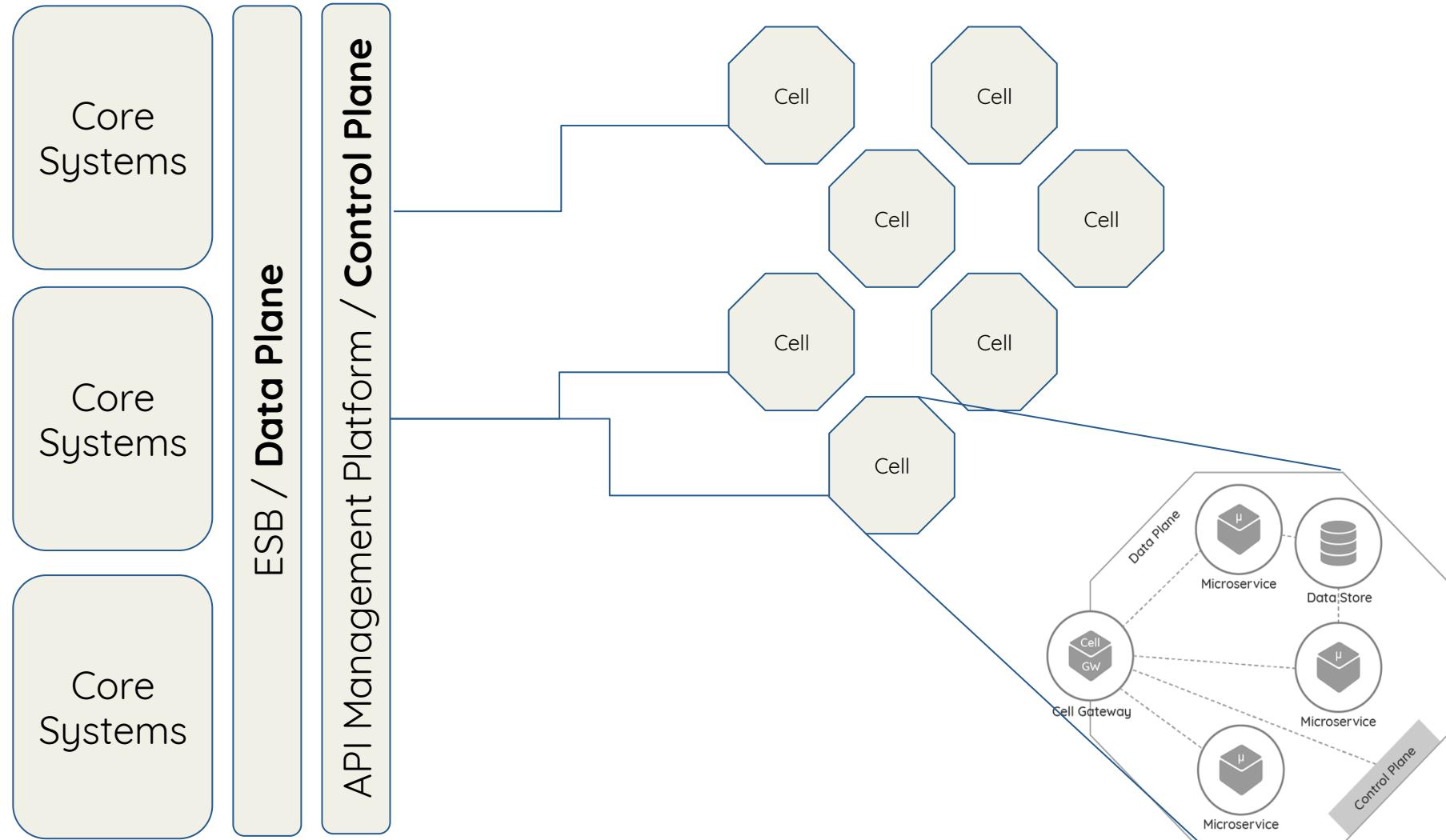
Core
Systems

ESB

API Management Platform



© 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/>



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>



© 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/>

Ballerina diagram and language

The screenshot shows a Ballerina development environment with two main panes. The left pane displays the Ballerina source code for a file named `demo.bal`. The right pane shows a `Ballerina Diagram` for a service named `hello`.

Code (demo.bal):

```
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         hi (endpoint caller, http:Request request, json js) {
29             string title = check <string>js.title;
30             string content = check <string>js.content;
31             github4:Issue iss = check
32                 git->createIssue("pzfro", "btest", title,
33             json resp = {
34                 key: "value",
35                 id: untaint iss.id
36             };
37             _ = caller -> respond(resp);
38         }
39     }
40 }
```

Ballerina Diagram:

The diagram illustrates the interaction between a `caller` and a `git` service. A `default` participant handles the `hi` endpoint. It receives a `request(js)` from the `caller`, performs a `createtissue("pzfro", "btest", title, content, [])` action on the `git` service, and then sends a `respond(resp)` back to the `caller`.

```
sequenceDiagram
    participant caller
    participant default
    participant git
    caller->>default: request(js)
    activate default
    default->>git: createtissue("pzfro", "btest", title, content, [])
    activate git
    git-->>default: respond(resp)
    deactivate default
    deactivate git
    caller-->>default: respond(resp)
```



Resources

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

