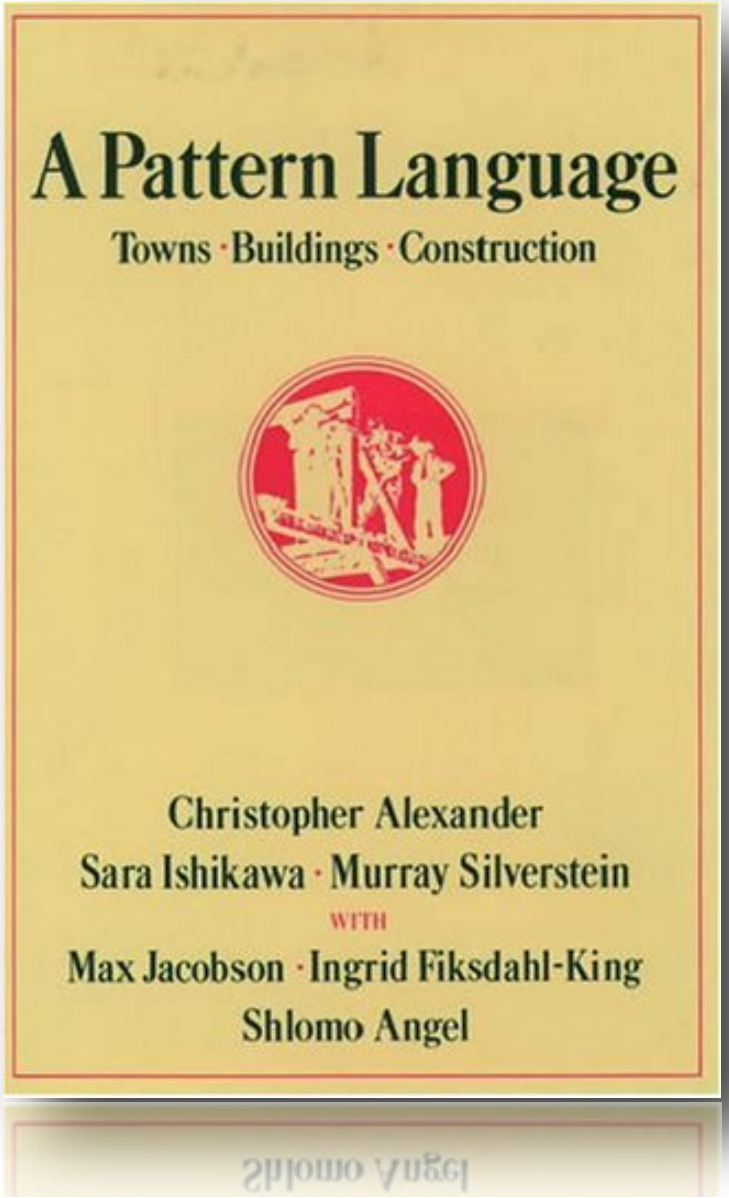
Architecture Patterns 1

Design Patterns in Architecture*

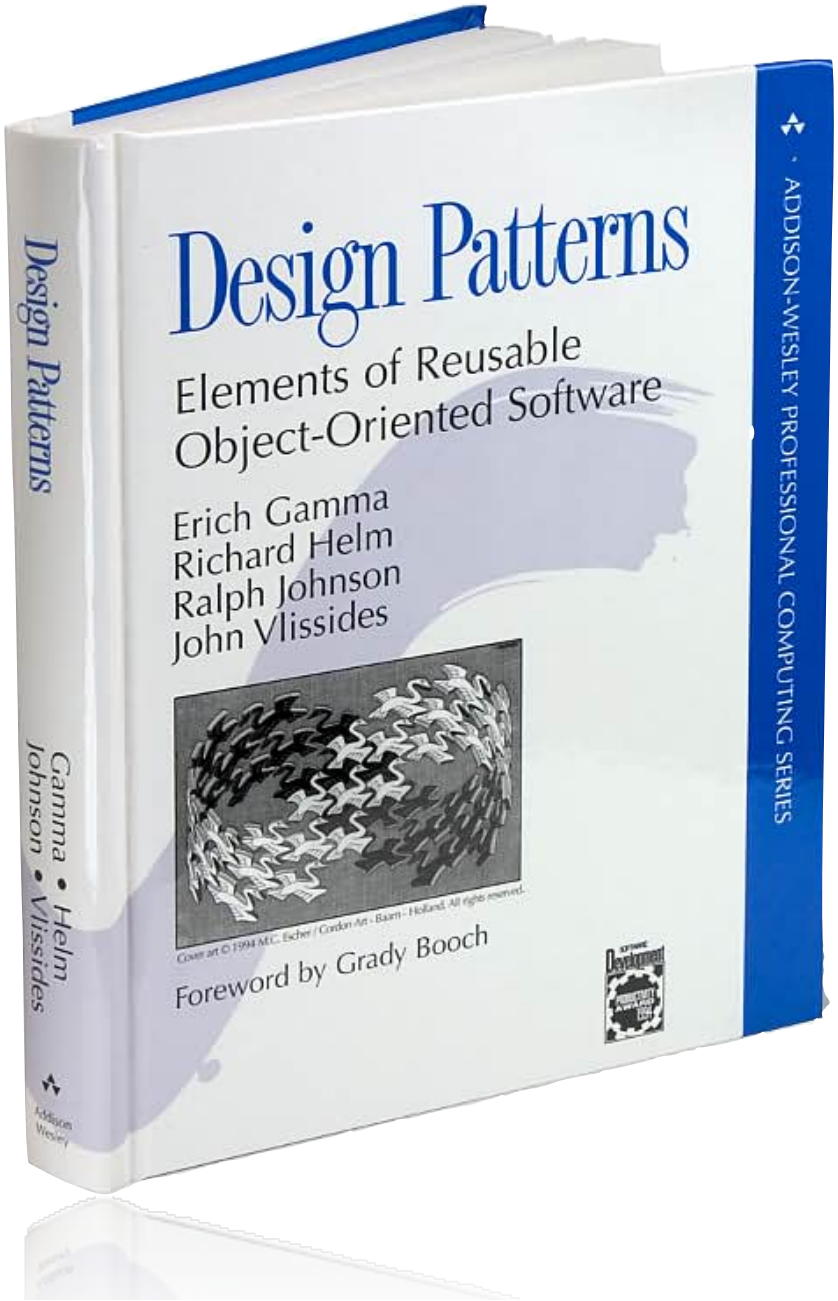


Christopher Alexander

Oxford University Press (1977)

*buildings, not software

Gang of Four



Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides

Published: 1984

patterns everywhere...

JAOO conference
- for Developers by Developers

JAOO Sydney 2009

See also JAOO Brisbane

Speakers
Schedule
Training
Social Events
Sponsors

Registration
Volunteers

Venue
Travel
Hotels

Check Us Out

About JAOO
Contact
Archives
Future events

Presentation: "Pattern Landscapes – or What Can We Learn From Dating Patterns?"

Time: Friday 15:15 - 16:00

Location: To be announced

Abstract:

Design patterns have been around in software development for more than a decade. Some use them, some don't. Given the vast number of patterns, it proves difficult to navigate in them. There are generic patterns, like the GoF patterns, there are patterns aimed at a specific programming language, there are patterns for a specific domain or for a specific phase in a development process.

Attendees will take away a clear understanding of patterns and their importance for software design and development. They will also get an approach for introducing them into the organization. The concept and use of patterns will be illustrated with dating patterns.

Download slides

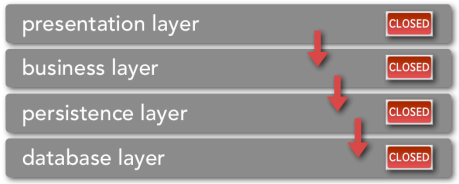
Retrospectives Facilitator Aino Vonge Corry, Trifork A/S

Aino Corry is technical conference editor and retrospectives facilitator at Trifork. She holds a masters degree and a ph.d. in computer science from the University of Aarhus, Denmark. She has 12 years of experience with Patterns in Software Development as a developer, architect and mentor. Aino was the architecture and coordinator for the EU project PalCom where she was responsible for the common architecture.

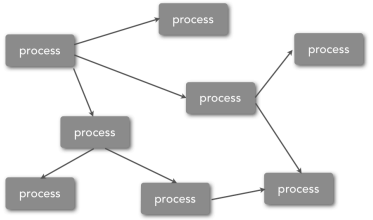
When she gets the chance, she teaches OO design and development.


Design Patterns
Elements of Reusable
Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES
PATTERNS
presentationzen
Simple Ideas on Presentation Design and Delivery
Garr Reynolds
slide:ology
NANCY DUARTE

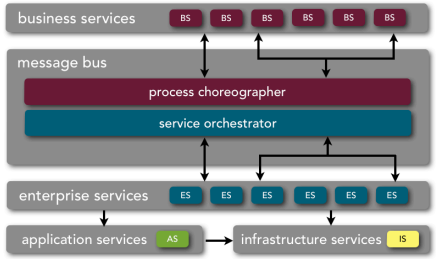
architecture patterns 1



traditional layered architecture

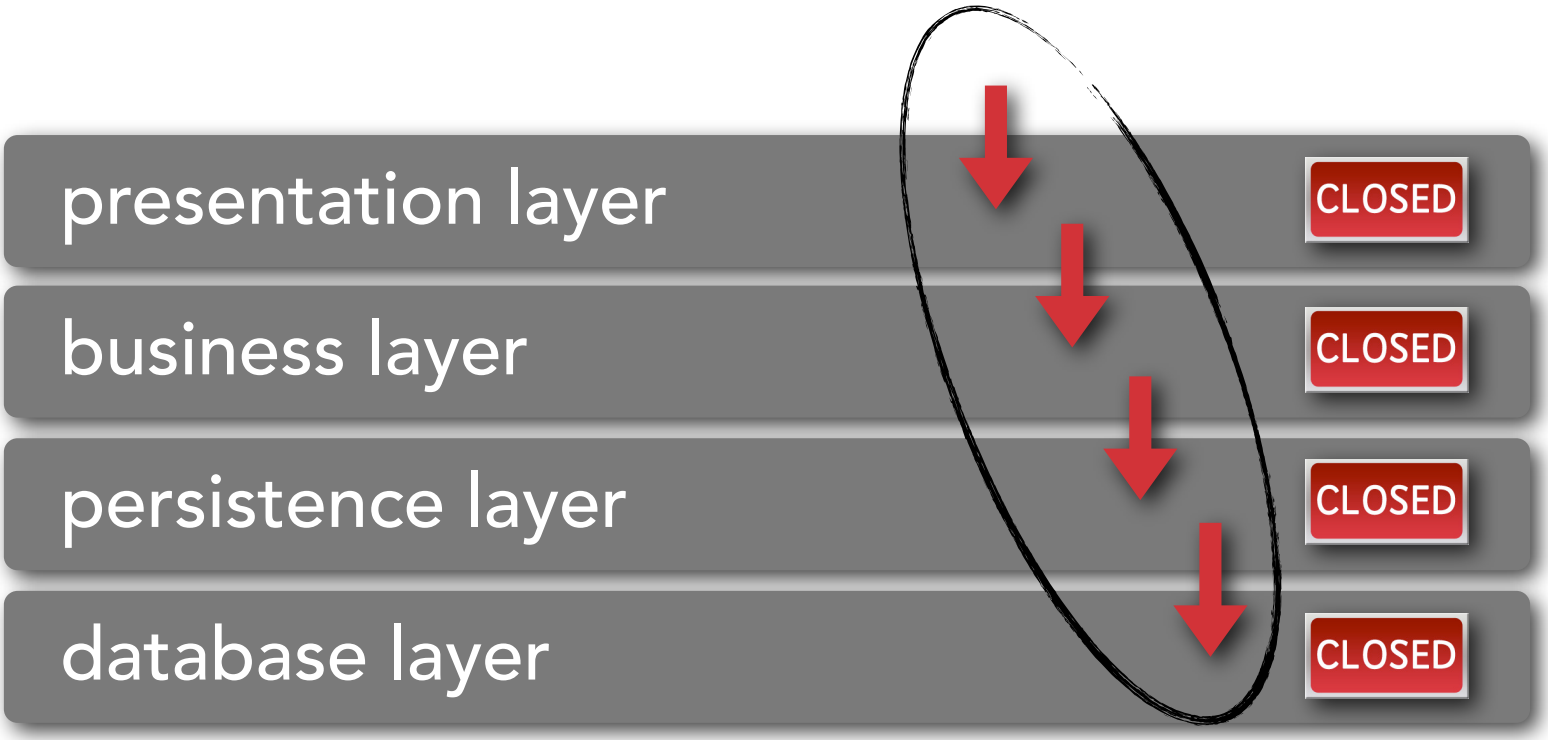


event-driven architecture



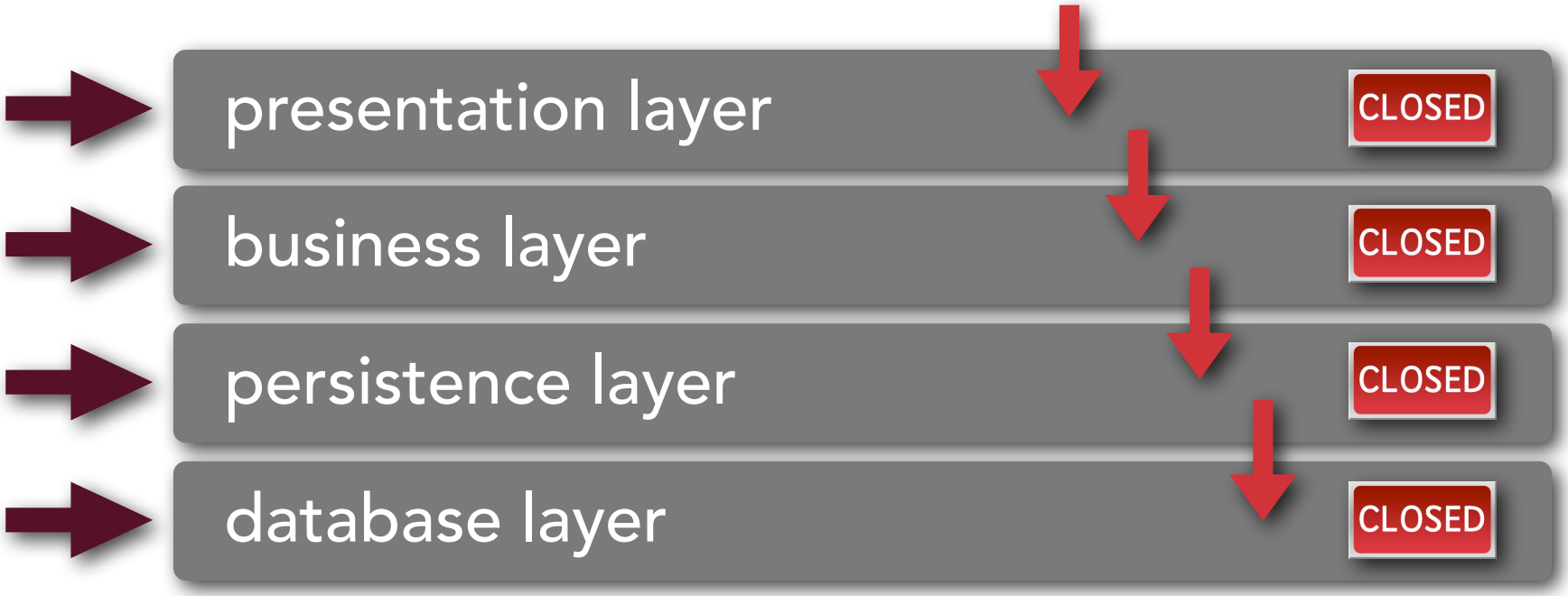
service-oriented architecture

traditional layered architecture



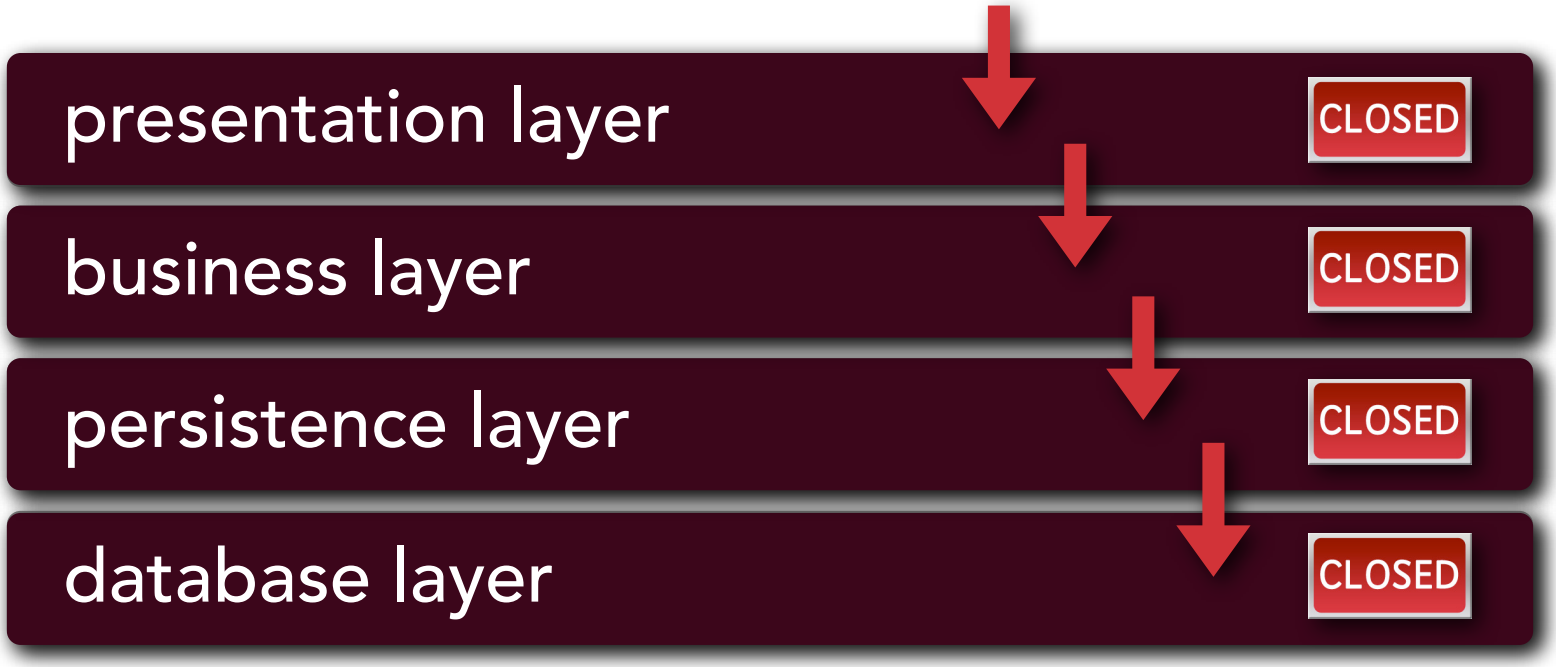
traditional layered architecture

separation of concerns



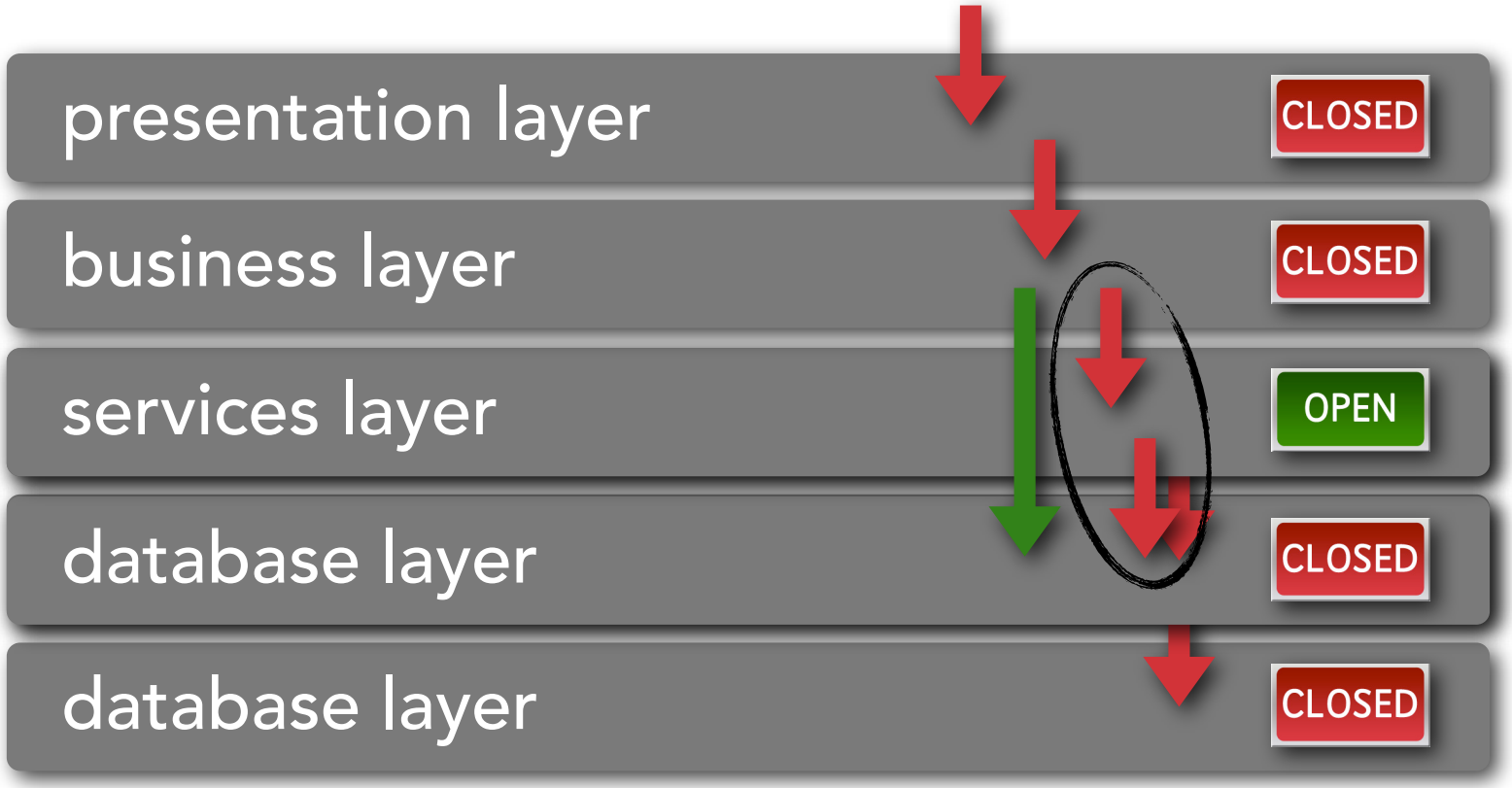
traditional layered architecture

layer isolation

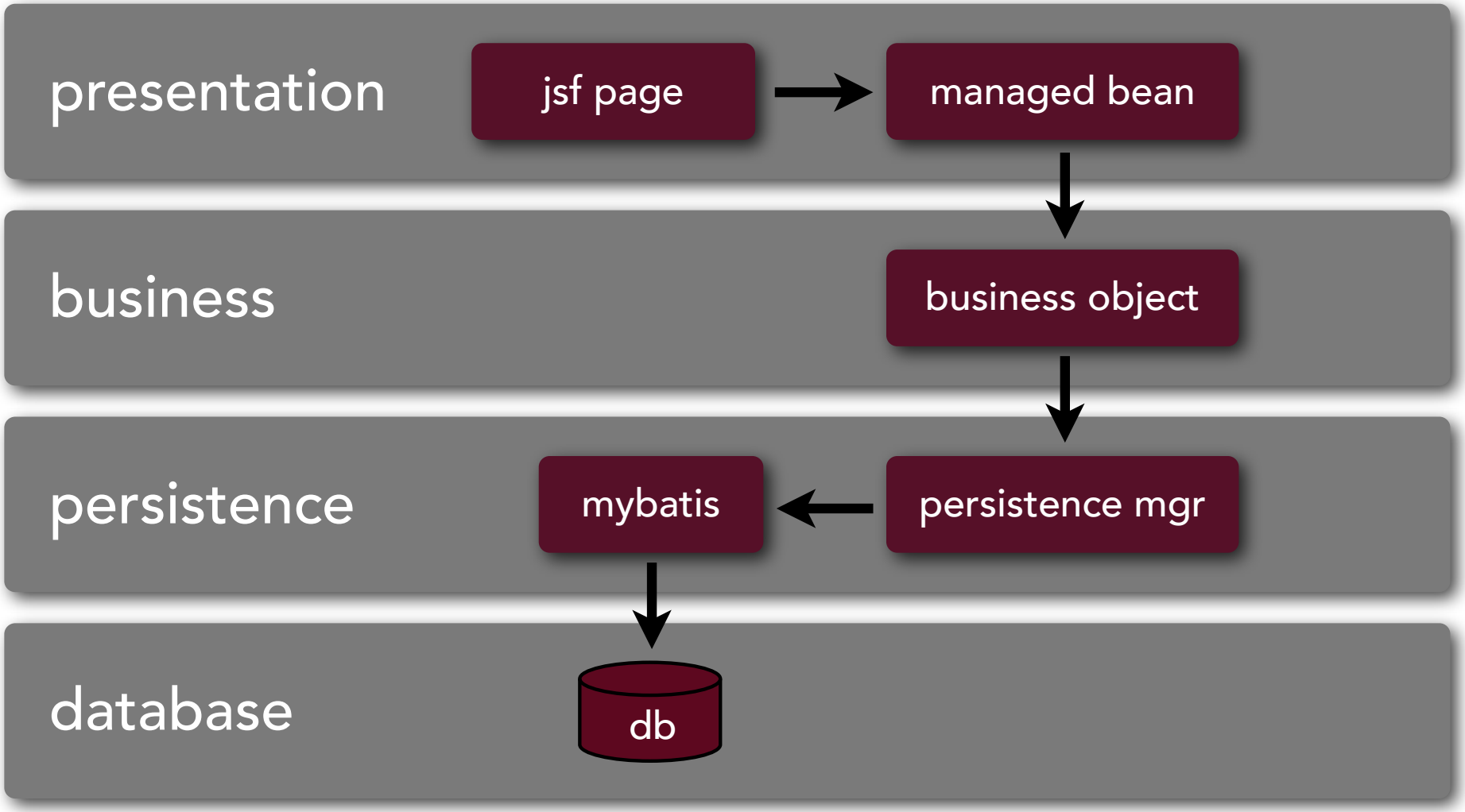


traditional layered architecture

variations and hybrids



traditional layered architecture



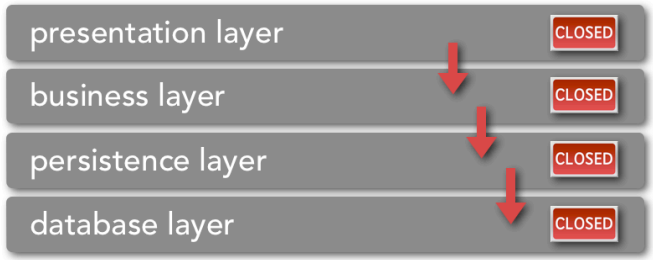
traditional layered architecture

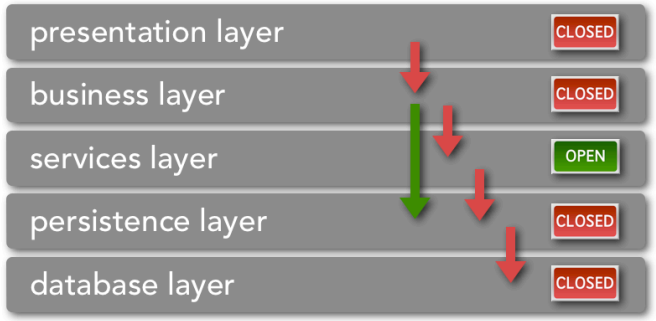
good general purpose architecture

easy to implement, test, and govern

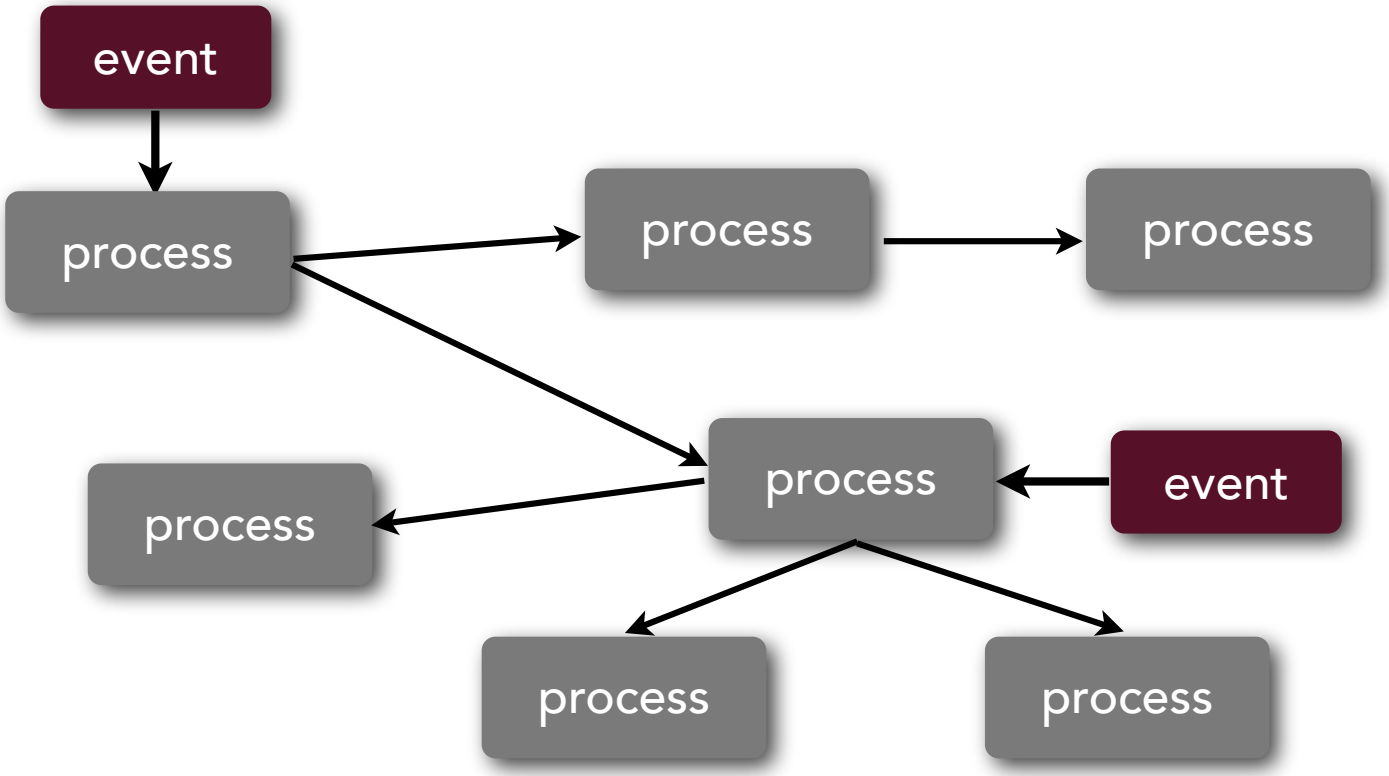
good starting point for most systems

not always optimized for specific business drivers

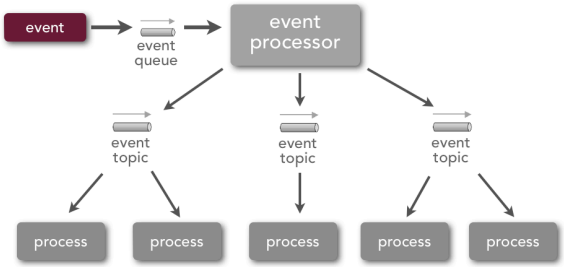




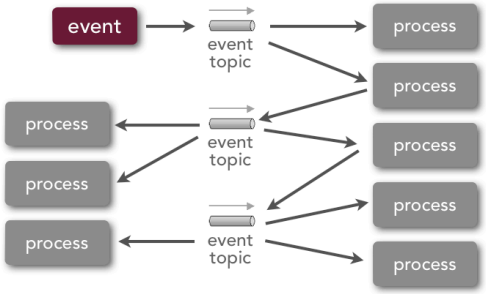
event-driven architecture



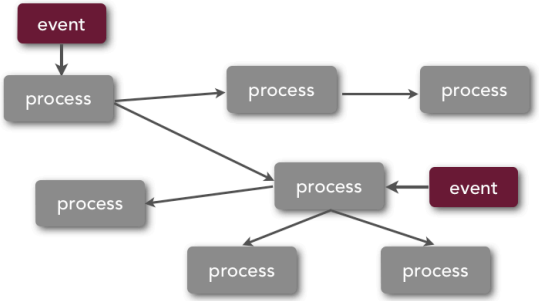
event-driven architecture



event processor topology



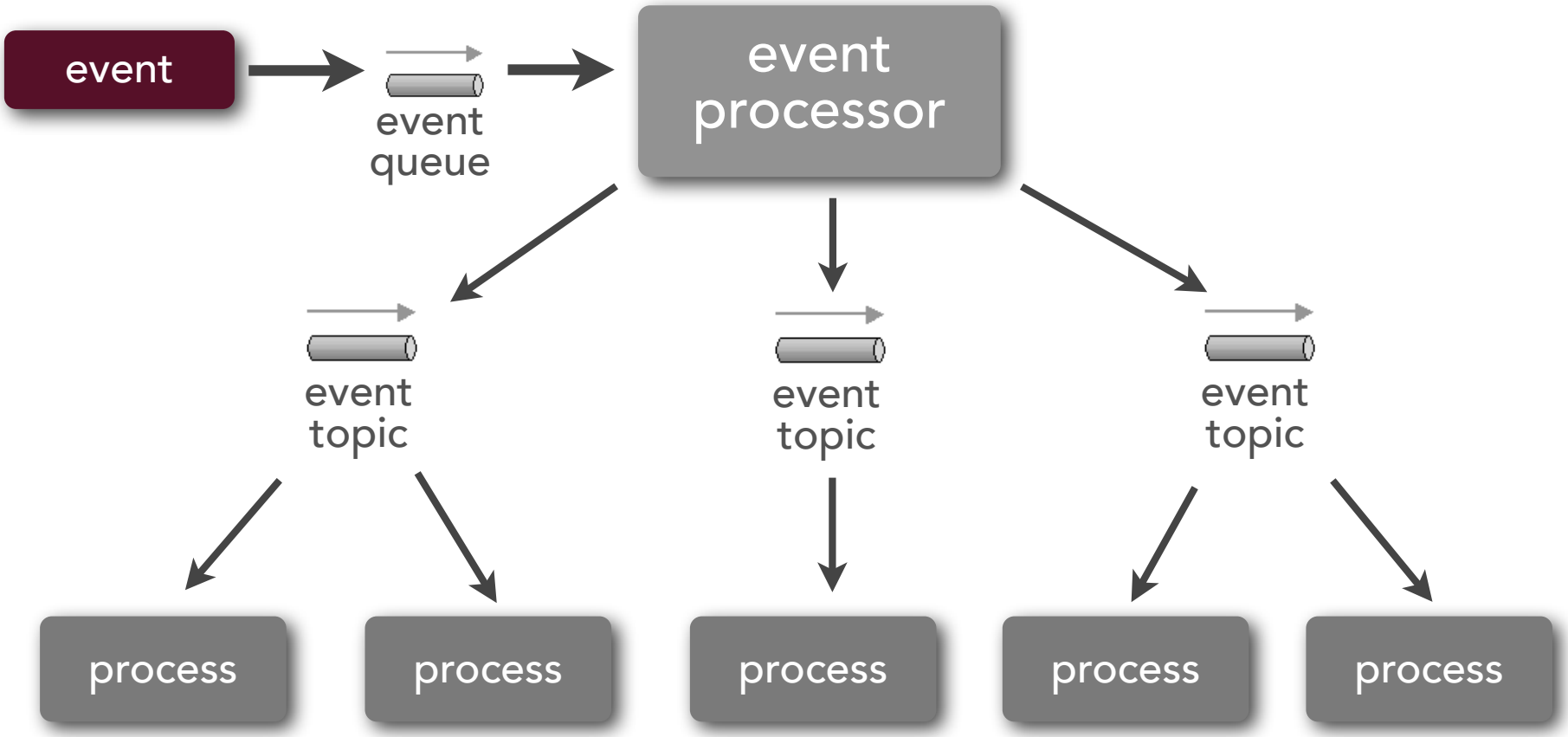
broker topology



broker-less topology

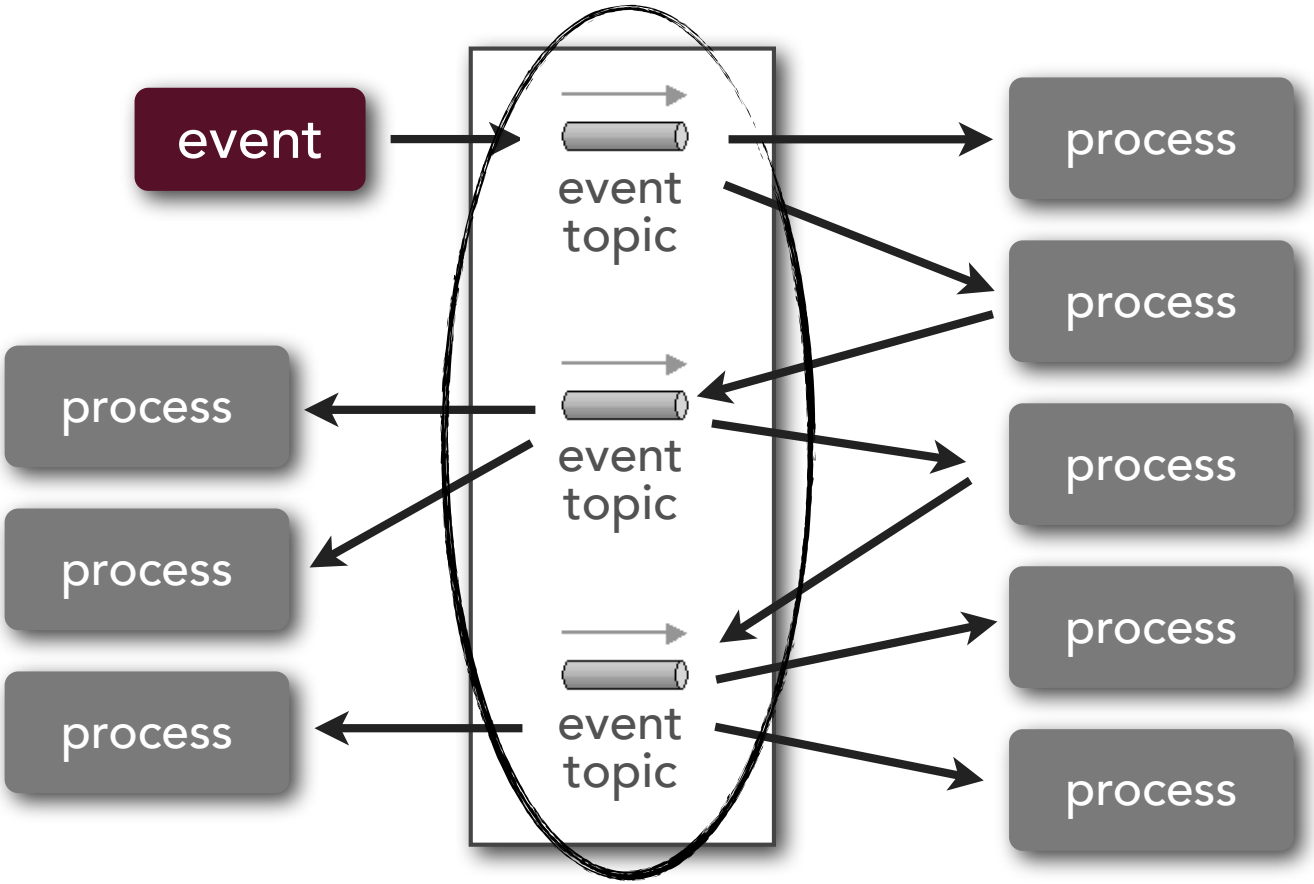
event-driven architecture

event processor topology



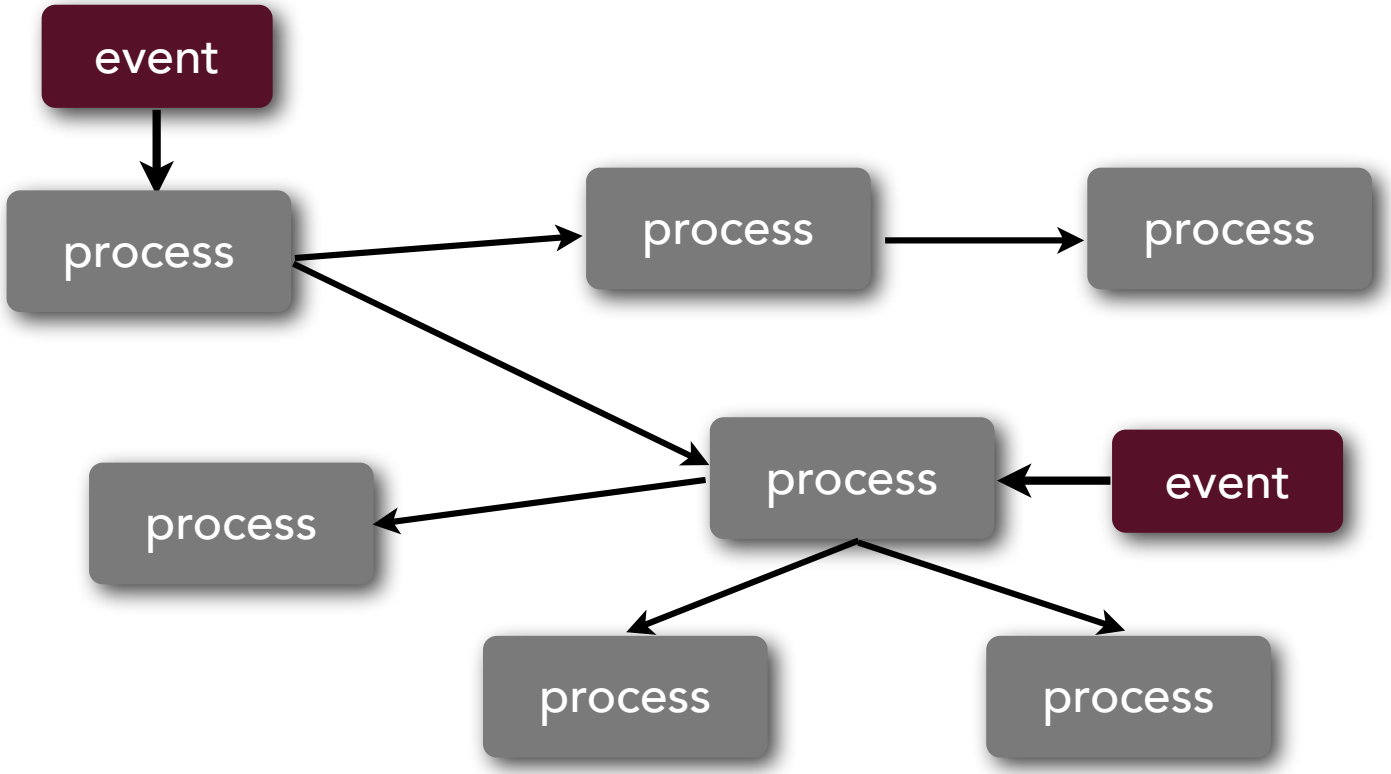
event-driven architecture

broker topology

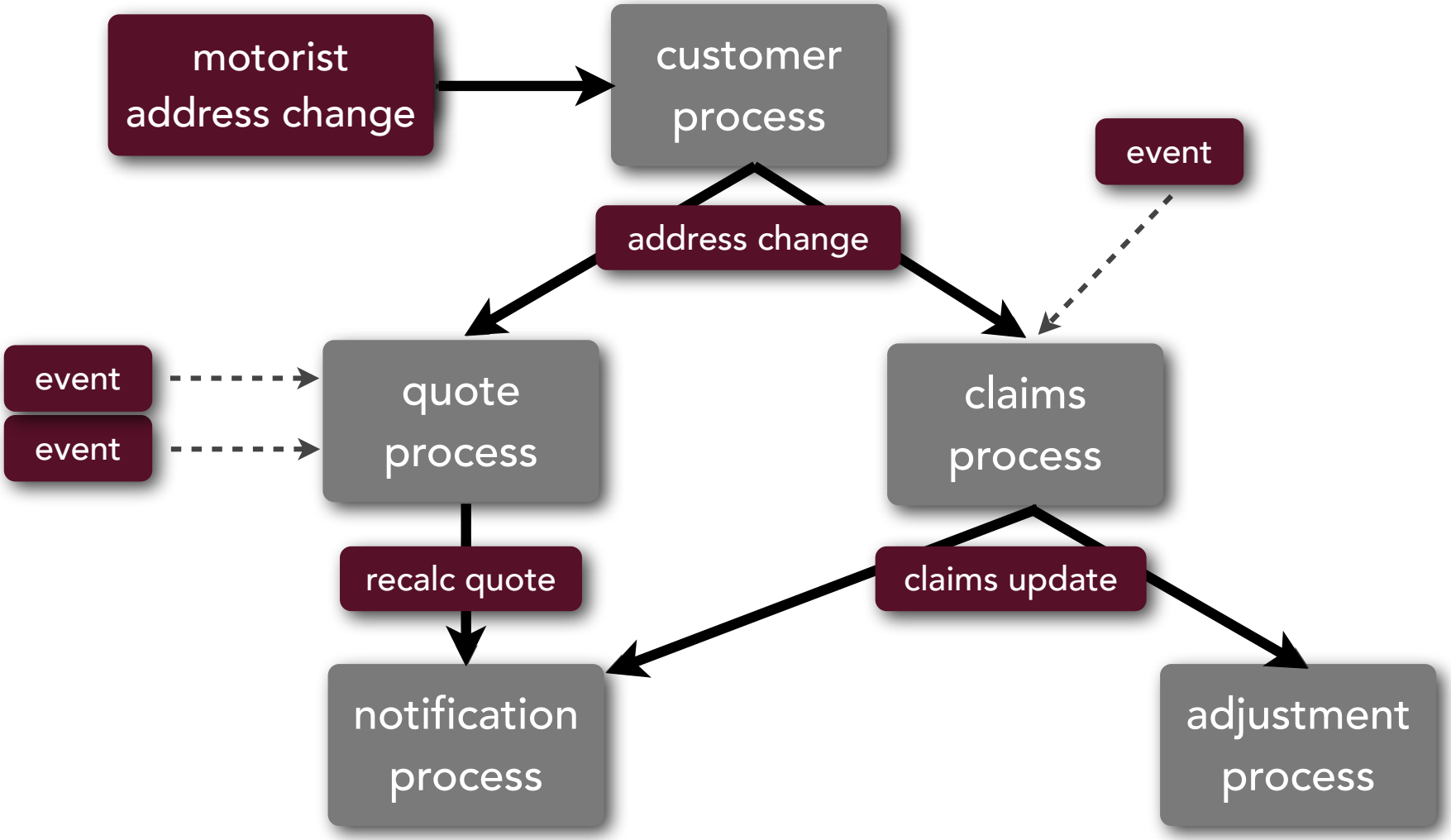


event-driven architecture

broker-less topology



event-driven architecture



event-driven architecture

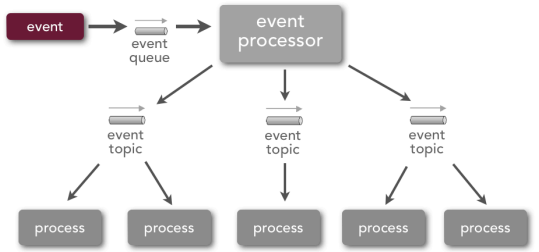
highly decoupled and distributed

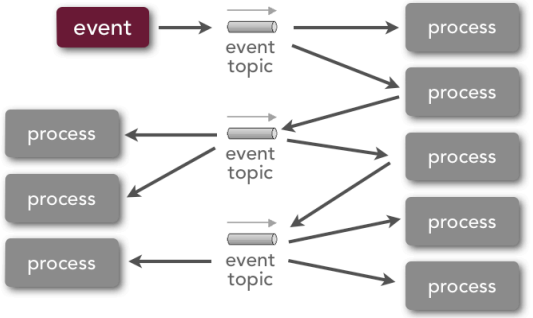
highly scalable

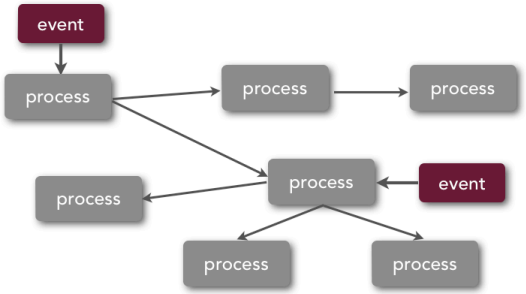
high degree of complexity

good for event-based business models and business processes

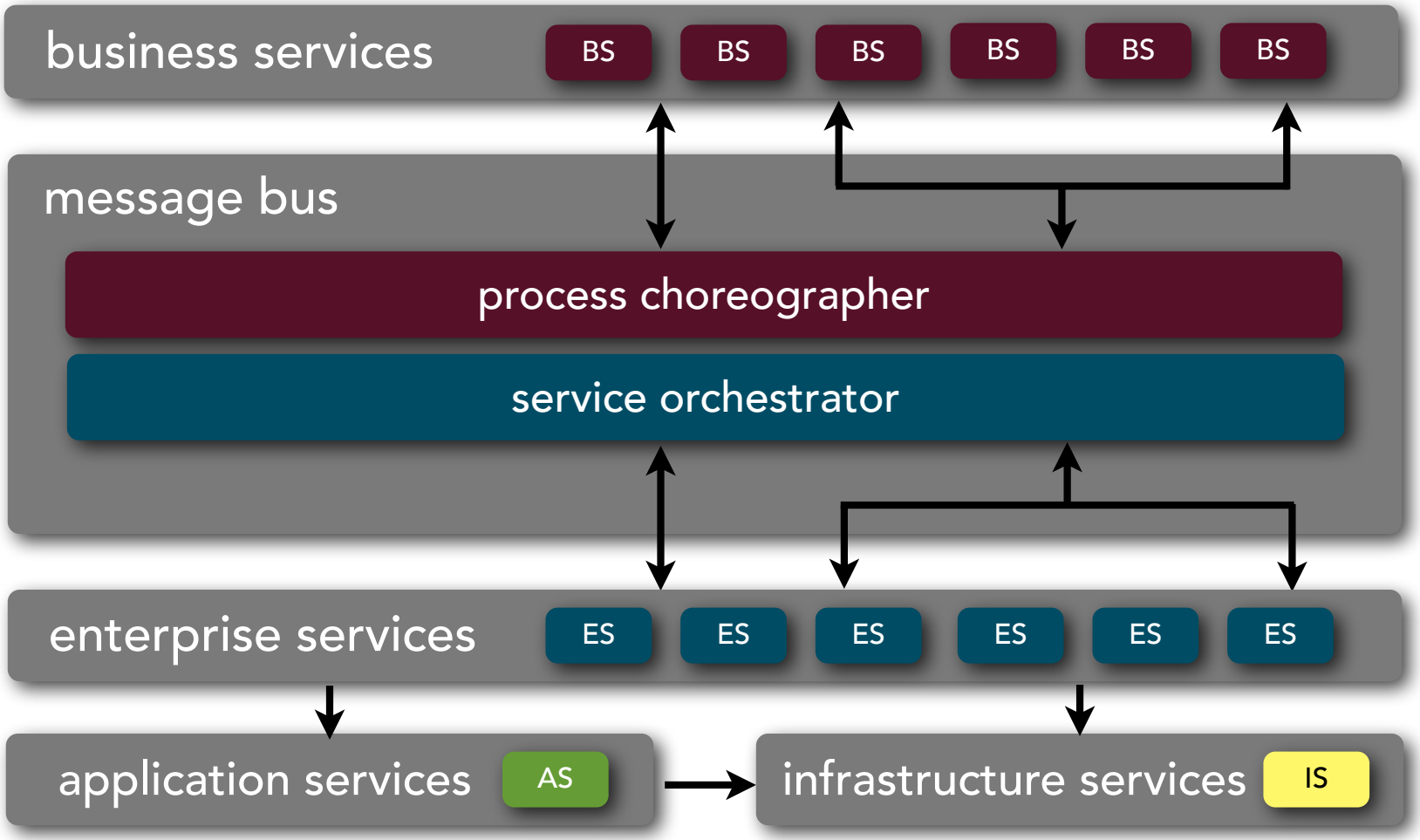
not good for processes which require a high degree of data sharing, orchestration, and reuse



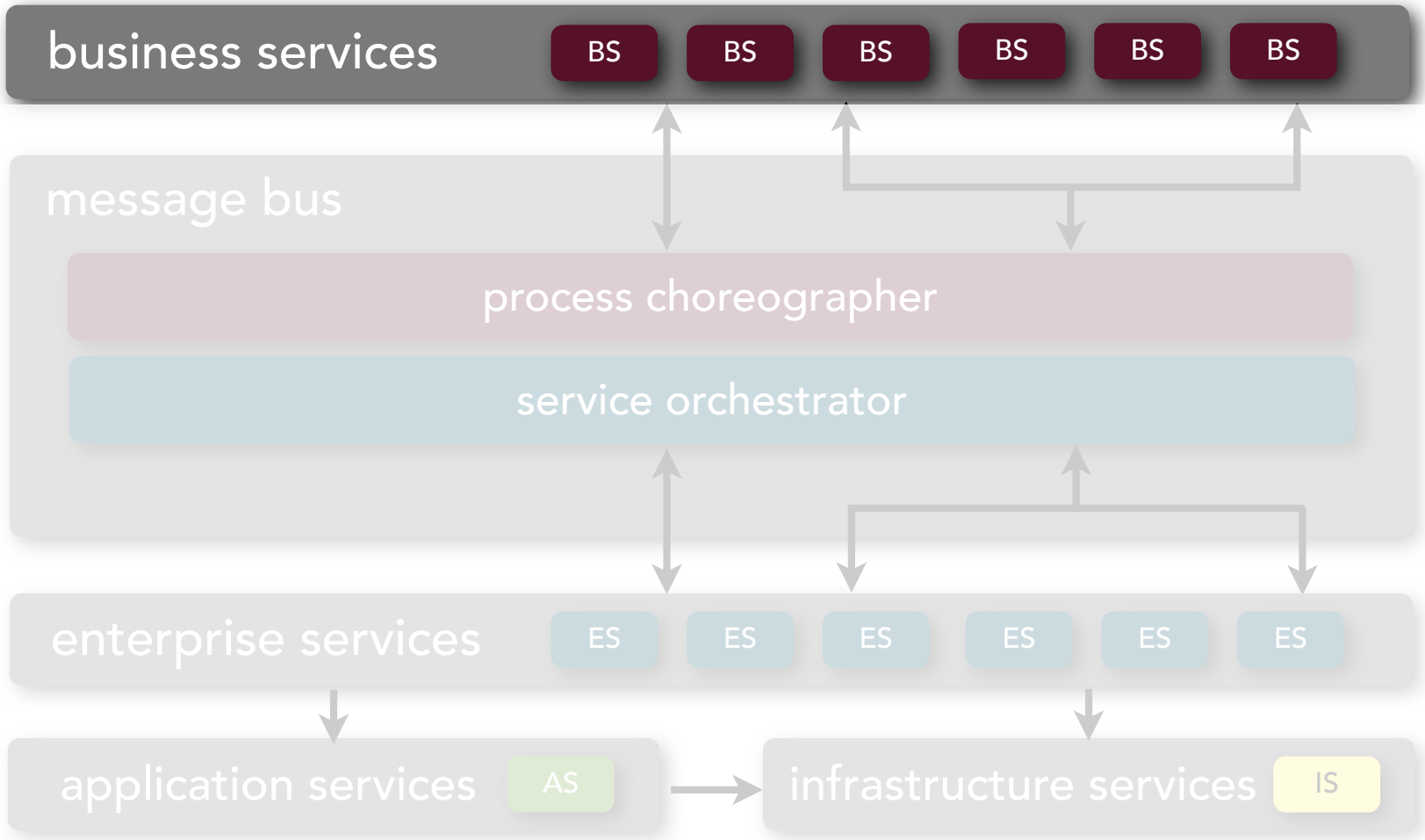




service-oriented architecture



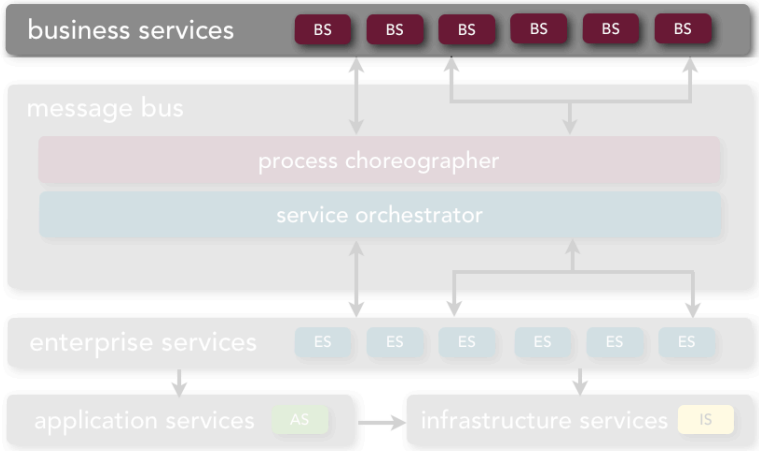
service-oriented architecture



service-oriented architecture

business services

business services template	
type	abstract
owner	business users
granularity	course-grained
scope	enterprise-level
notes	contains name, inputs, outputs, and process flow independent of any technical implementation or protocol represented in BPEL, WSDL, etc.

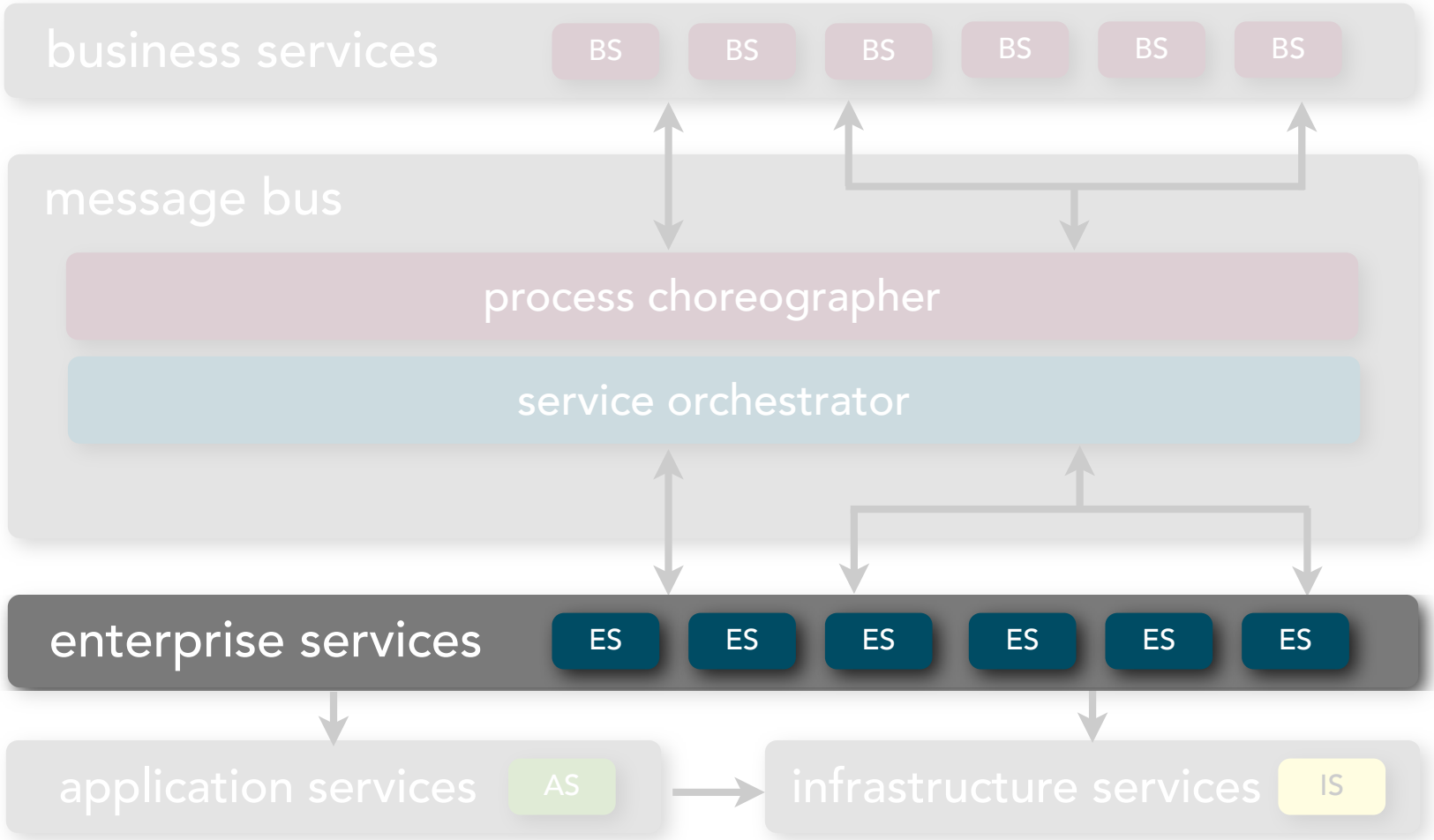


ProcessClaim

ExecuteTrade

PlaceOrder

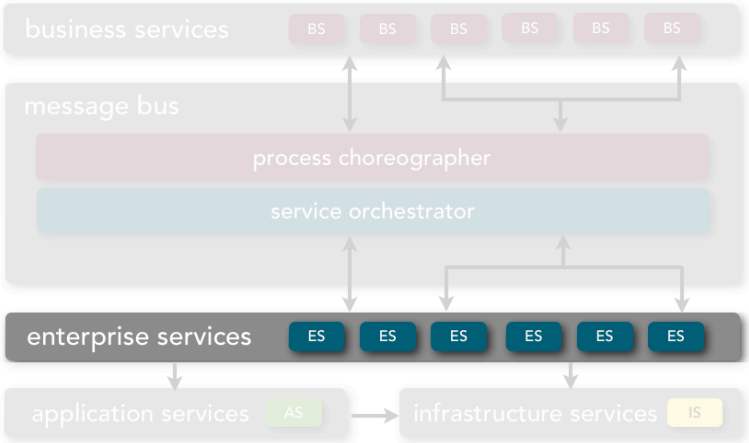
service-oriented architecture



service-oriented architecture

enterprise services

enterprise services template	
type	concrete
owner	architect / shared services team
granularity	course-grained
scope	enterprise-level
notes	custom or vendor implementation one-to-one or many-to-one relationship with business services

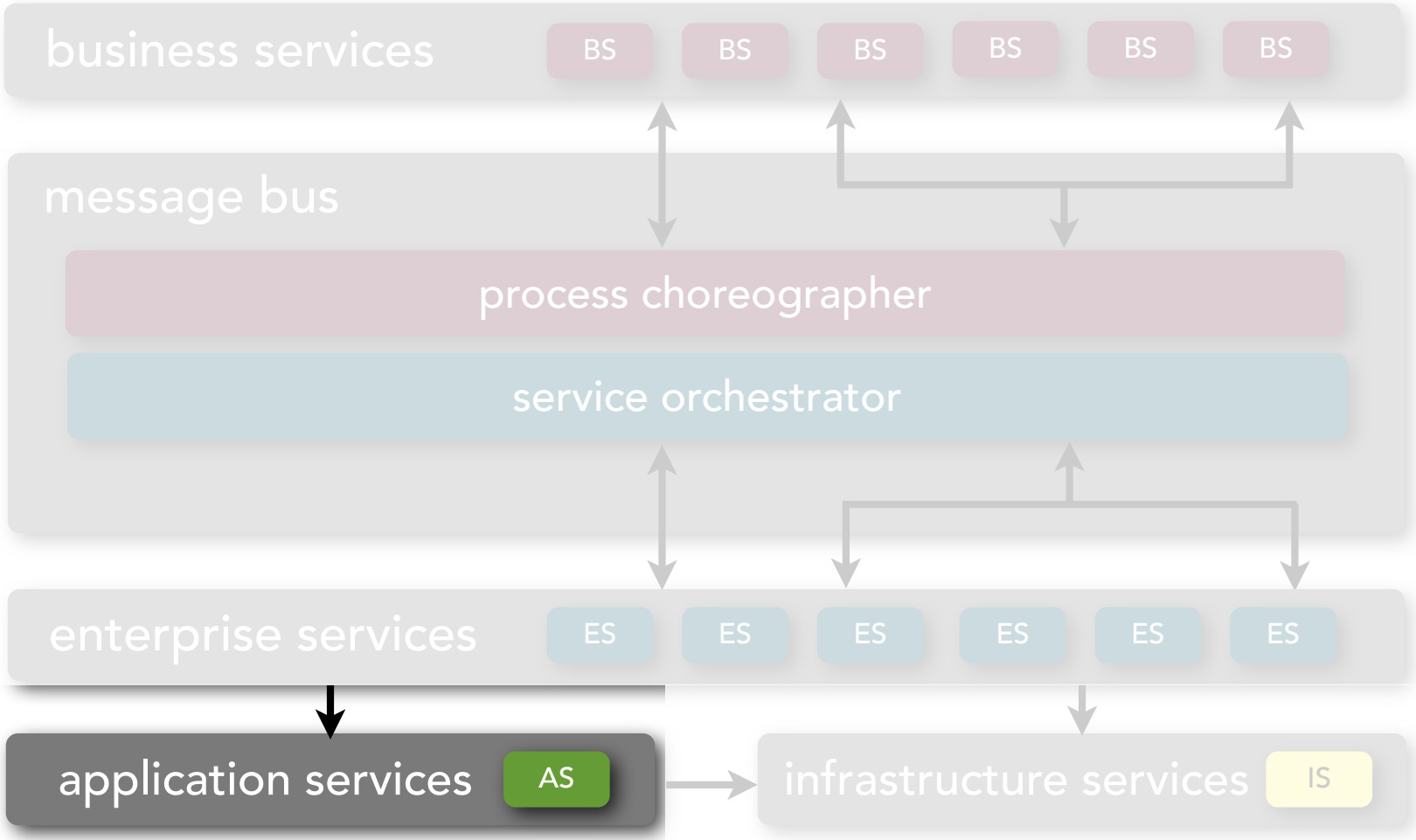


CreateCustomer

CalculateQuote

CheckCompliance

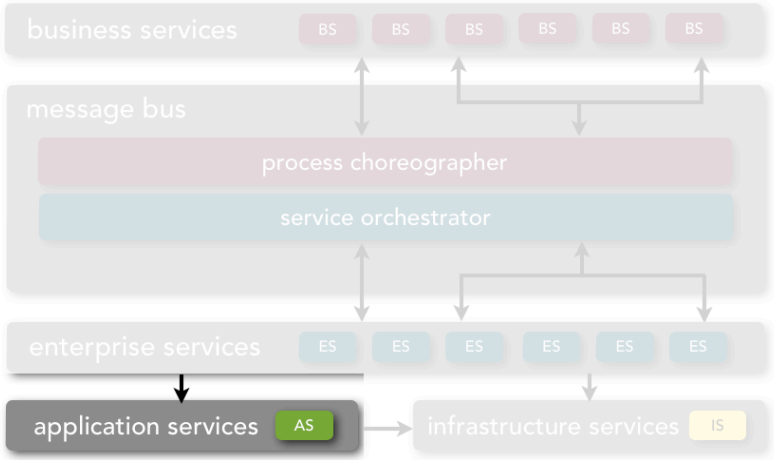
service-oriented architecture



service-oriented architecture

application services

application services template	
type	concrete
owner	application development team
granularity	fine-grained
scope	application-level
notes	bound to a specific application context; generally not shared used for validation, database query and updates

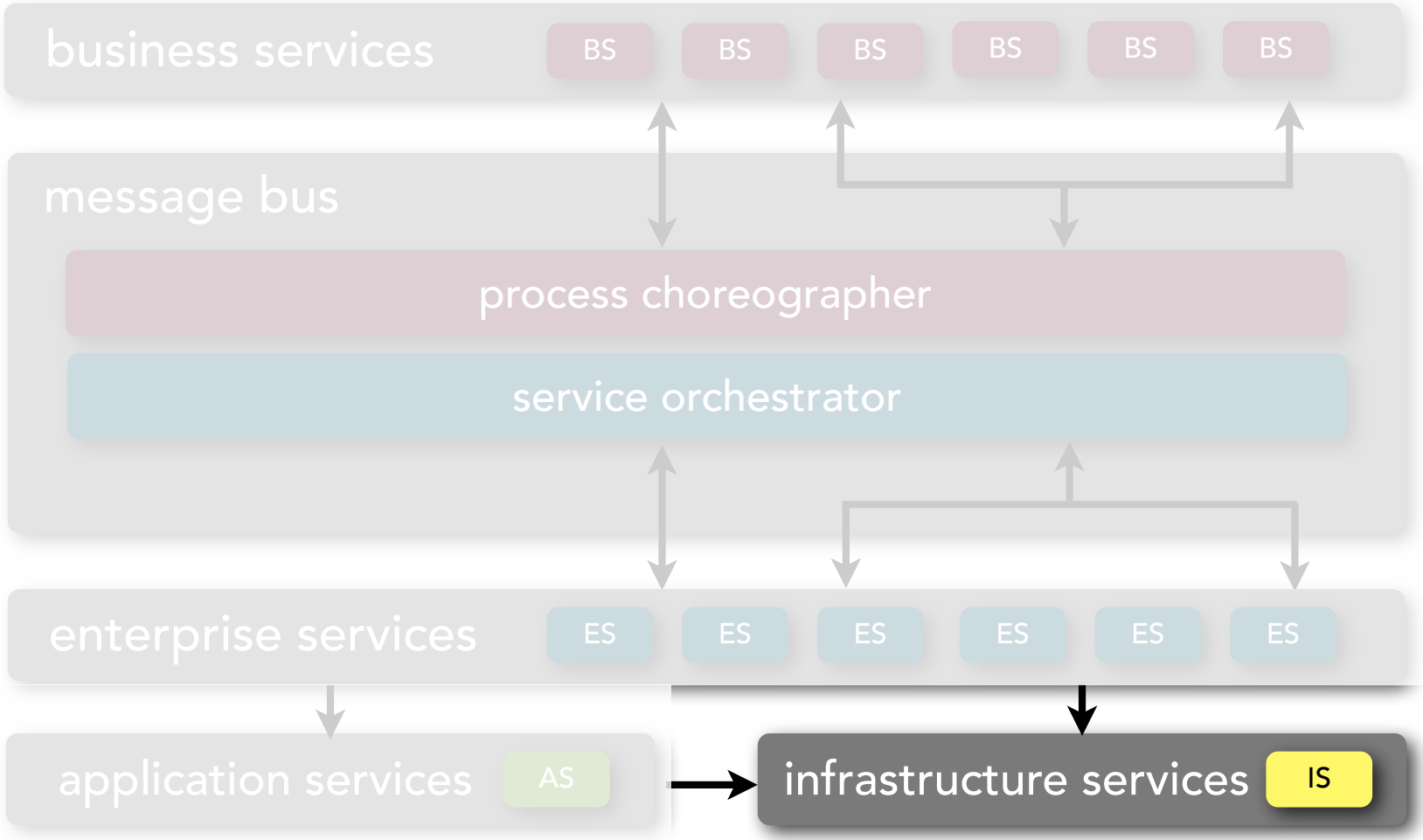


addDriver

addVehicle

getInventoryCount

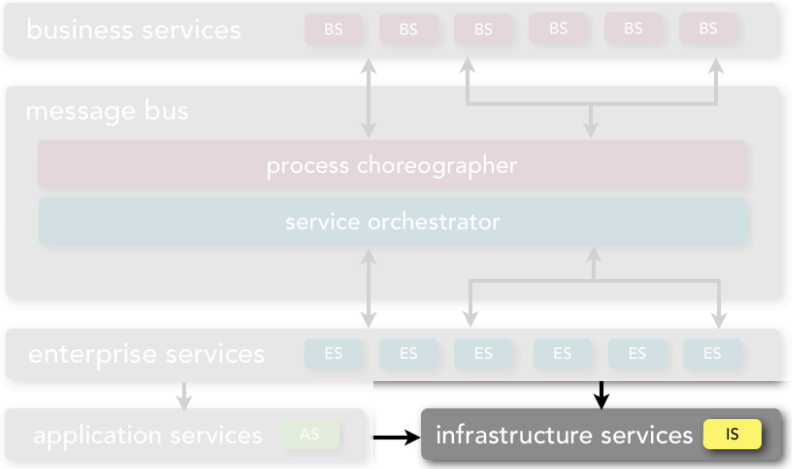
service-oriented architecture



service-oriented architecture

infrastructure services

infrastructure services template	
type	concrete
owner	application development team
granularity	fine-grained
scope	enterprise-level
notes	supports application and enterprise services implements non-business functionality

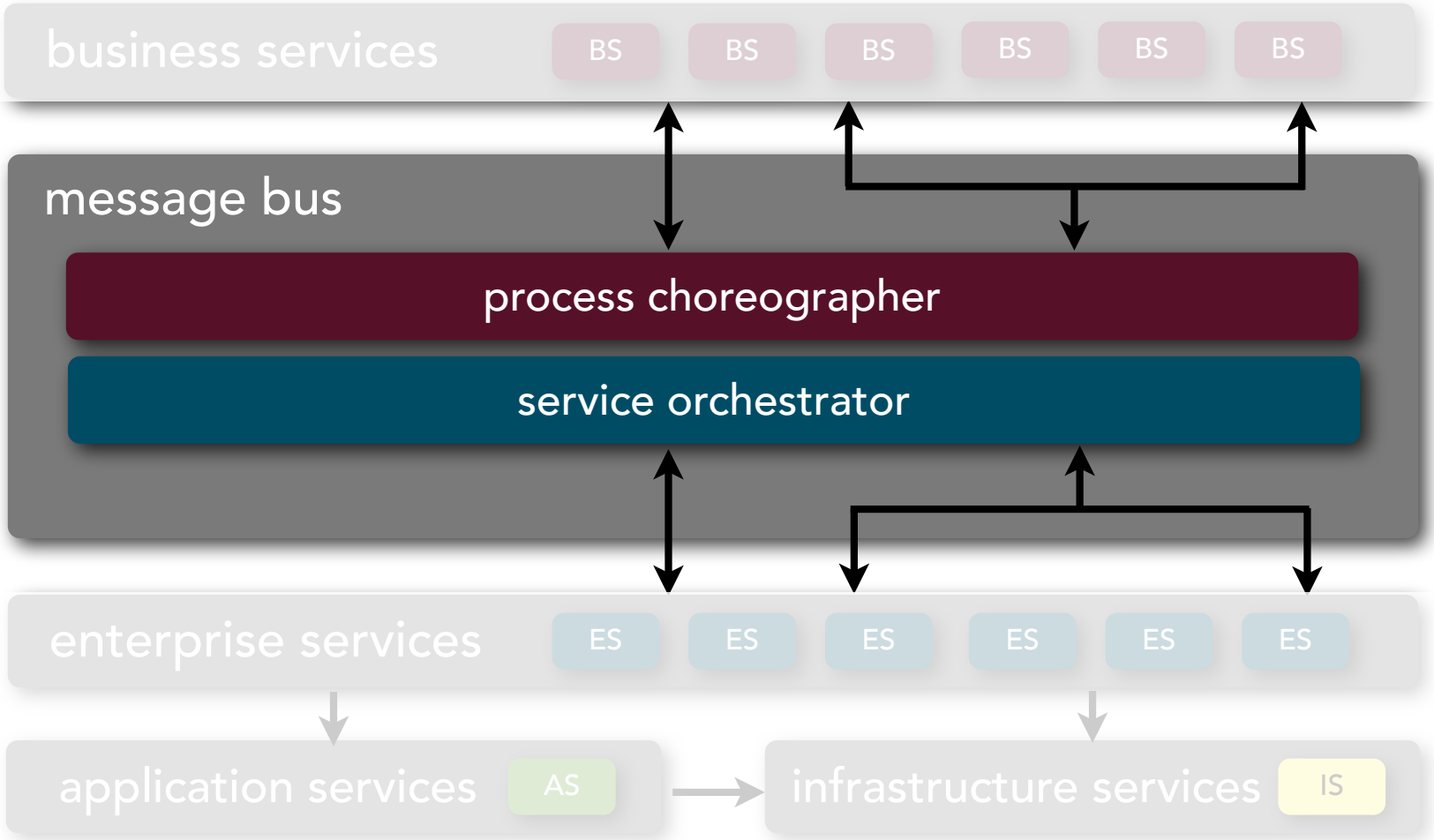


writeAuditLog

checkUserAccess

writeErrorLog

service-oriented architecture



service-oriented architecture

message bus

process choreography

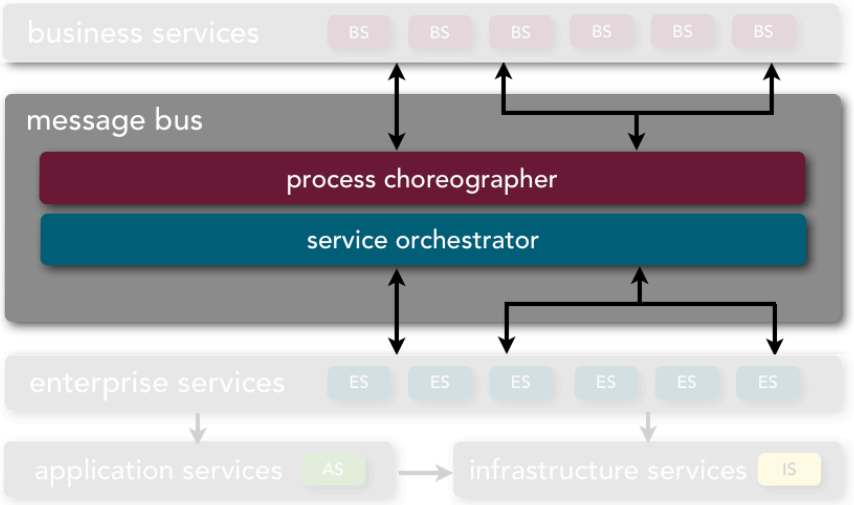
service orchestration

service registry

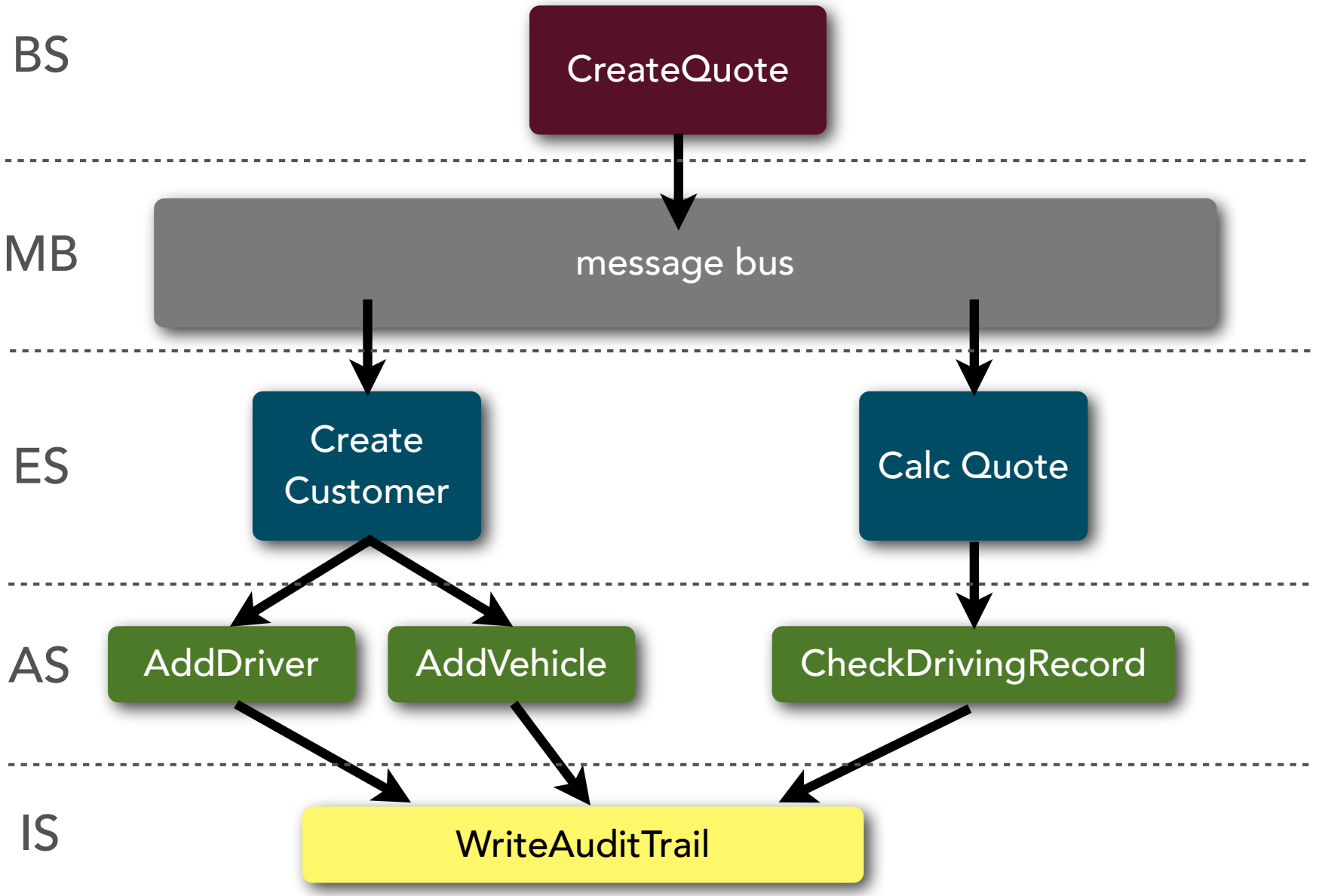
protocol transformation

message enhancement

message transformation

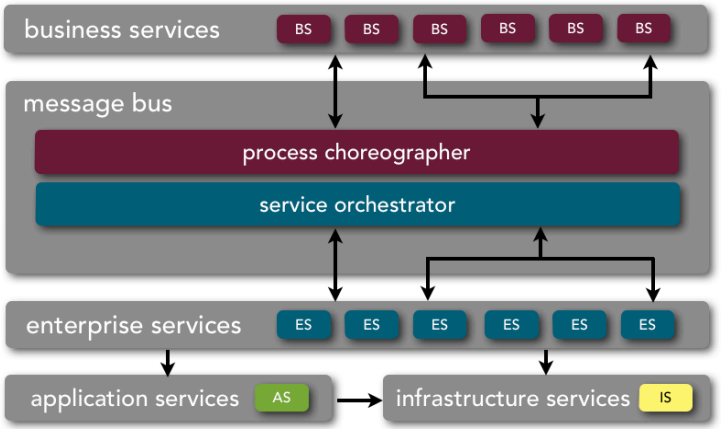


service-oriented architecture



service-oriented architecture

good pattern for understanding and implementing business processes and services



very high level of complexity

difficult to implement due to complex tools, hype, misconceptions, and heavy business user involvement

good pattern for large, complex, heterogeneous businesses that have a large number of common services (e.g., insurance)

for more information

Wikipedia (Event Driven Architecture)
http://en.wikipedia.org/wiki/Event-driven_architecture

Creating an Effective SOA Service Taxonomy
Mark Richards, SOA World, 2008
http://soa.sys-con.com/node/738704

?’S

Mark Richards

Independent Consultant

Hands-on Enterprise / Integration Architect
Published Author / Conference Speaker

http://www.wmrichards.com
http://www.linkedin.com/pub/mark-richards/0/121/5b9

Published Books:

Java Message Service, 2nd Edition
97 Things Every Software Architect Should Know
Java Transaction Design Strategies

Neal Ford

Director / Software Architect /
Meme Wrangler

ThoughtWorks®

2002 Summit Blvd, Level 3, Atlanta, GA 30319, USA
T: +1 40 4242 9929 Twitter: @neal4d
E: nford@thoughtworks.com W: thoughtworks.com