Software Architecture Patterns
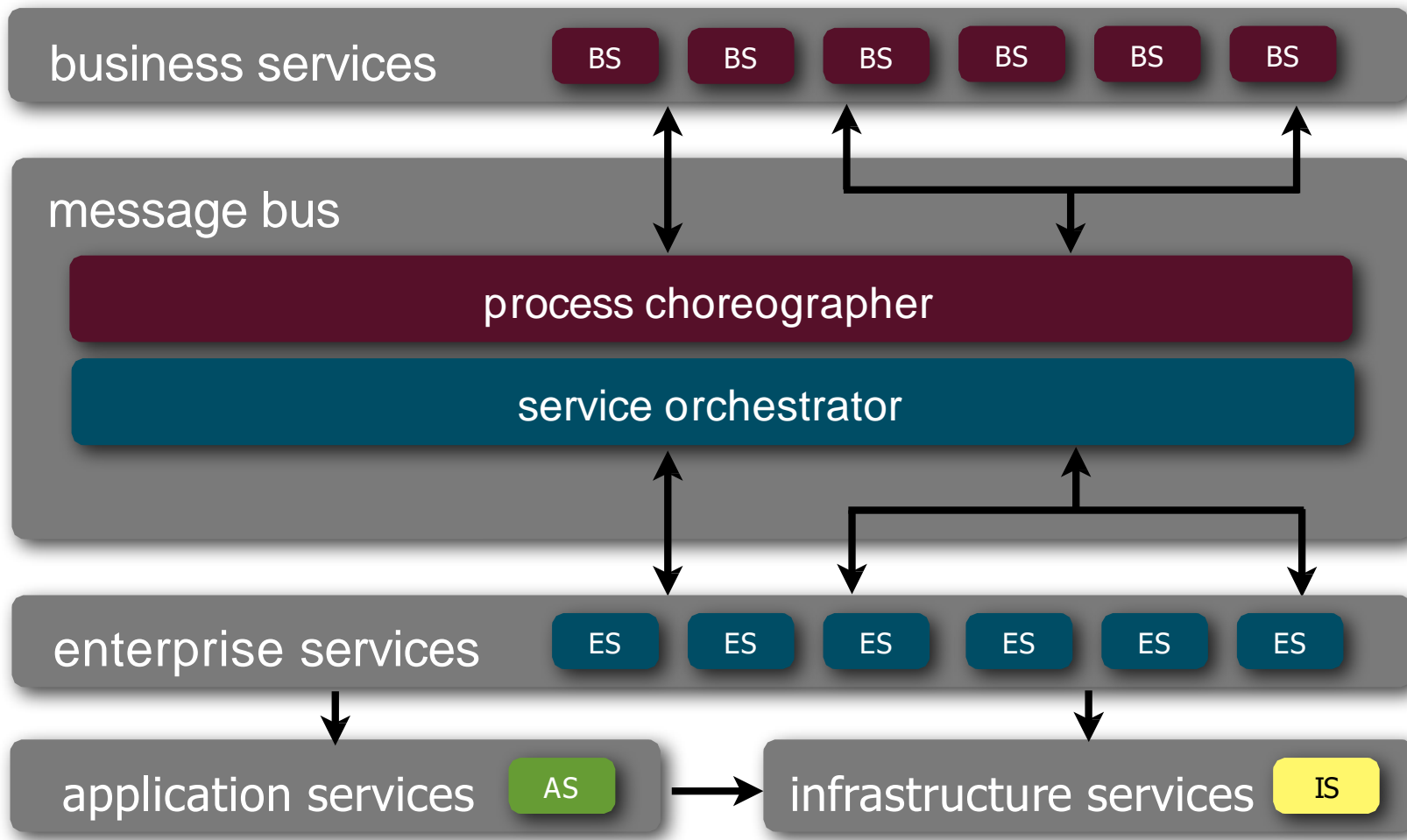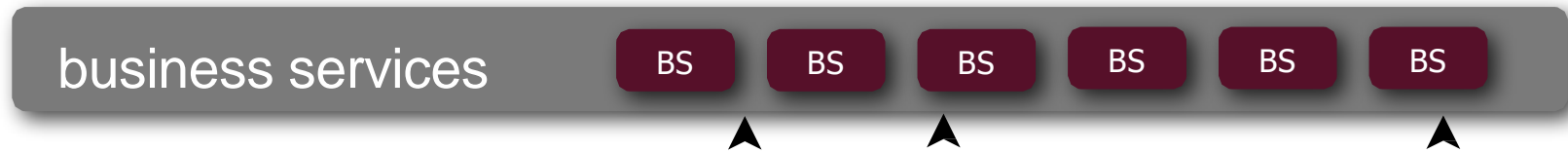
Service-oriented architecture

Service-oriented architecture
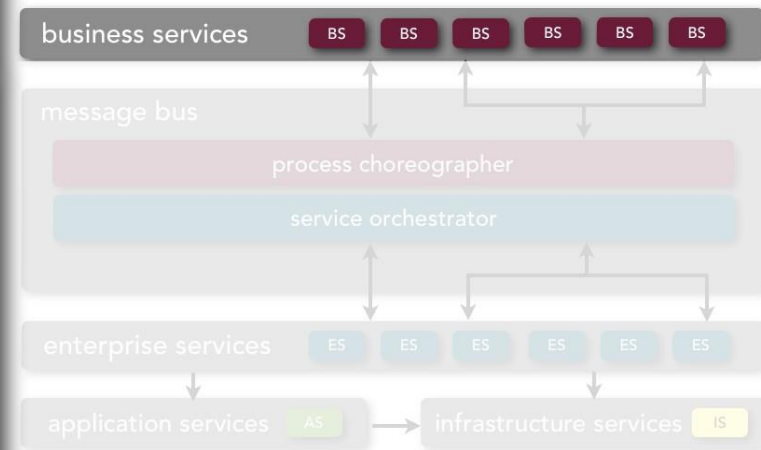
Service-oriented architecture (Cont.)

business services — BS BS BS BS BS BS

Service-oriented architecture (Cont.)
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

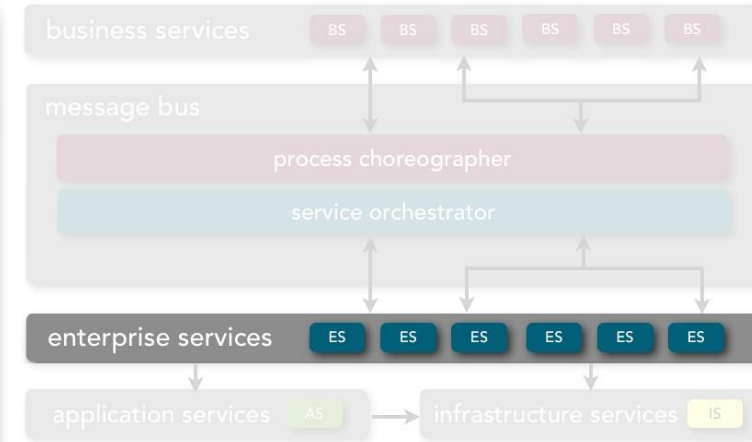# Service-oriented architecture (Cont.)

enterprise services | ES | ES | ES | ES | ES | ES
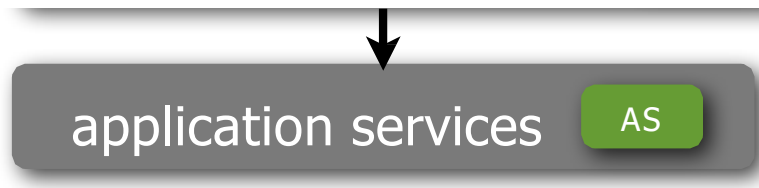
Service-oriented architecture (Cont.)
enterprise services

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
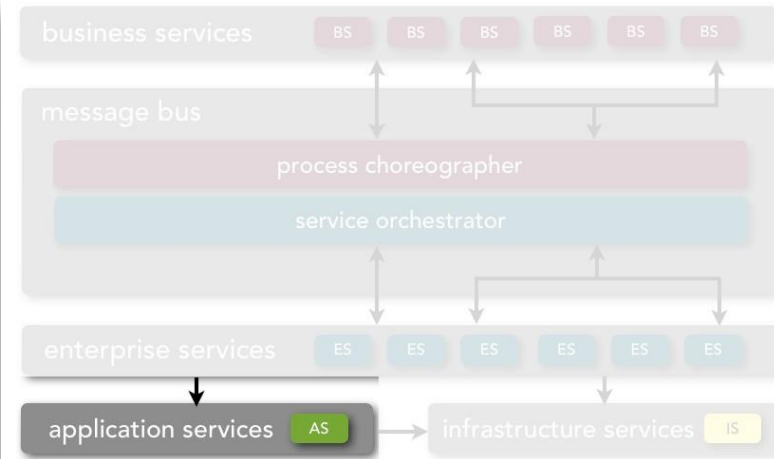
CalculateQuote

CheckCompliance

Service-oriented architecture (Cont.)

application services AS

Service-oriented architecture (Cont.)
application services

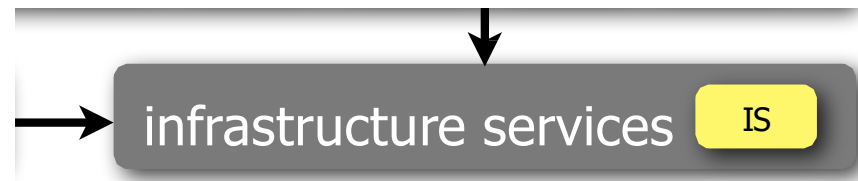| application services template	
type	concrete
owner	application development team
granularity	fine-grained
scope	application-level
notes	bound to a specific application context; generally not shared used for validation, database query and updates

addDriver

addVehicle

getInventoryCount

Service-oriented architecture (Cont.)

infrastructure services IS

Service-oriented architecture (Cont.)
infrastructure services

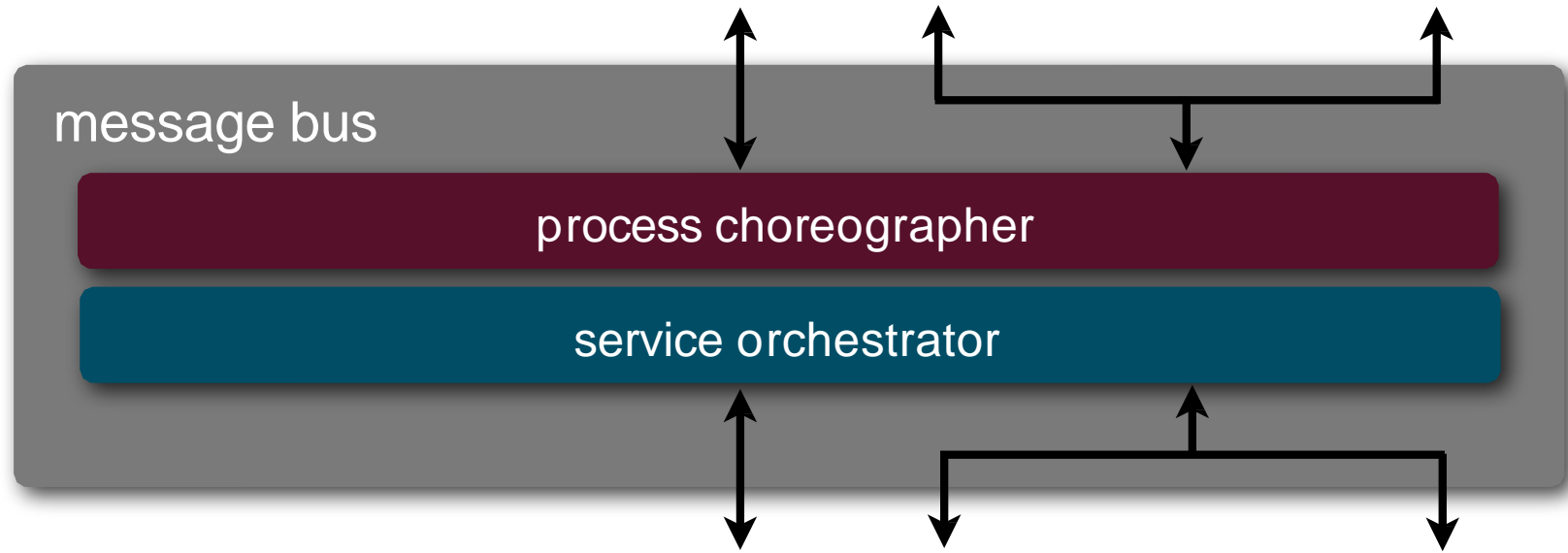| infrastructure services template	
type	concrete
owner	application development team
granularity	fine-grained
scope	enterprise-level
notes	supports application and enterprise services implements non-business functionality
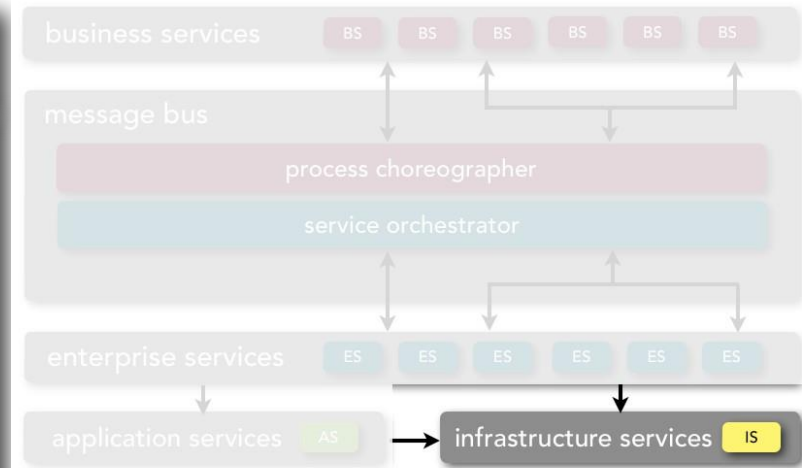
writeAuditLog
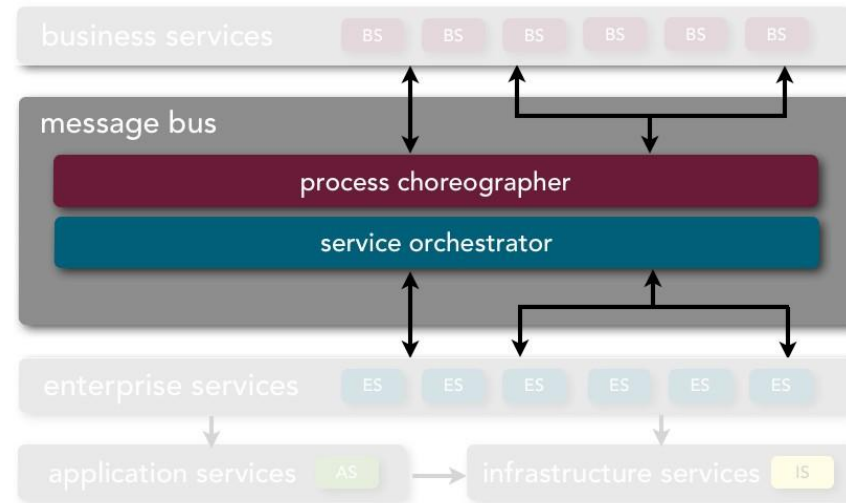
checkUserAccess

writeErrorLog

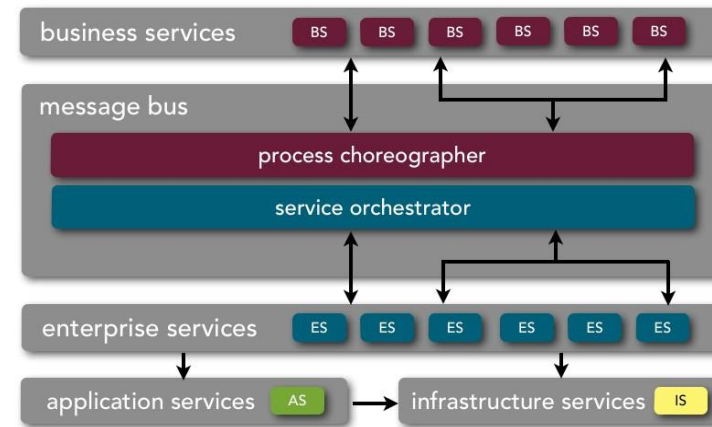# Service-oriented architecture (Cont.)

Service-oriented architecture (Cont.)

- Message bus
  - Process choreography
  - Service orchestration
  - Service registry protocol
  - Transformation message
  - Enhancement message
  - Transformation

Service-oriented architecture (Cont.)

- 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).

Thanks!