Comparing Service-based Architectures
Service-oriented Architecture
origins: hubs
origins: hubs
origins: hubs

System A

System B

System C

System D (ftp only)

System E (http only)
origins: hubs

System A

System D (ftp only)

System E (http only)

System B

System C
origins: hubs

System A
System D (ftp only)
System E (http only)
System B
System C

Integration Hub
origins: hubs

System A

Integration Hub

System B

System C

System D (ftp only)

System E (http only)
origins: hubs
origins: hubs

looks great, but what about single point of failure and performance bottleneck considerations?
orchestration

hub

intelligent hub

service oriented architecture / enterprise service bus pattern
service-oriented architecture
service-oriented architecture
service-oriented architecture

abstract
enterprise-level
coarse-grained

owned and defined by business users
data represented as WSDL, BPEL, XML, etc.

no implementation - only name, input, and output

Are we in the business of...?

ExecuteTrade  PlaceOrder  ProcessClaim
service-oriented architecture

owned by shared services teams

custom or vendor implementations that are one-to-one or one-to-many relationship with business services
service-oriented architecture

owned by application teams

concrete application-level fine-grained

bound to a specific application context

AddDriver  UpdateAddress  CalcSalesTax

application services
service-oriented architecture

owned by infrastructure or shared services teams

concrete enterprise-level fine-grained

implements non-business functionality to support both enterprise and business services

WriteAudit CheckUserAccess LogError infrastructure services
service-oriented architecture
mediation and routing

business services

message bus

enterprise services
service-oriented architecture

process choreography
service-oriented architecture

service orchestration
service-oriented architecture

service registry

business service

CreateQuote

service registry
calcQuote()
enterprise service
saveQuote()
exterprise service

location, name, and implementation transparency

CreateQuote
calcQuote
https://grid1/CalcQuote
saveQuote
rmi://atlas:1299/saveQuote
service-oriented architecture
message enhancement

cusip, mm/dd/yy

client application

enhance message

enterprise service

sedol, yyyy.mm.dd, symbol

037833100
04/23/15

037833100
04/23/15

2046251
2015.04.23
AAPL

2046251
2015.04.23
AAPL

contract decoupling
service-oriented architecture
message transformation

contract decoupling
service-oriented architecture
protocol transformation

access decoupling
implementation transparency
service-oriented architecture
service-oriented architecture
service-oriented architecture

- Business services (BS)
- Message bus
  - Process choreographer
  - Service orchestrator
- Enterprise services (ES)
- Application services (AS)
- Infrastructure services (IS)

- Maximize reuse
- Maximize canonicality
service-oriented architecture

business services

message bus

process choreographer

service orchestrator

enterprise services

application services

infrastructure services

❌ incremental change

❌ operationally coupled
domain driven design
bounded context

Maintaining Model Integrity

CONTINUOUS INTEGRATION

UBQUITOUS LANGUAGE

BOUNDED CONTEXT

CONTEXT MAP

ASSess/overview relationships with

KEEP model unified by

OVERlap allied contexts through
relate allied contexts as

OVERlap unilaterally as

SUPPORT multiple clients through
freeteams to go

SEPARATE WAYS

TRANslate and insulate unilaterally with

BIG BALL OF MUD

ANTICORRUPTION LAYER
microservices architecture
microservices architecture

distributed architecture

client requests

api layer

client requests

client requests
microservices architecture
separately deployed components
microservices architecture

service component

client requests

api layer

client requests

client requests
microservices architecture
bounded context
bounded context ≠ entity
prefer duplication over coupling
prefer duplication over coupling

microservices
smart endpoints, dumb pipes

<- HTTP ->
<- messaging ->
standardize on integration, not platform

embrace polyglot solutions where sensible

Standardize in the gaps between services - be flexible about what happens inside the boxes.

Pick some sensible conventions, and stick with them.

Have one, two or maybe three ways of integrating, not 20.

too few languages/platforms

too many languages/platforms
technical consistency

Consider Service Templates to make it easy to do the right thing!
decentralized governance
decentralized data management

ACID versus BASE

monolith - single database
microservices - application databases
decentralized data management

Avoid distributed transactions if at all possible
infrastructure automation

- compile, unit and functional test
- acceptance test
- integration test
- user acceptance test
- performance test

run on build machine
- deployed on build machine
- deployed to integration environment
- deployed to UAT environment
- deployed to performance environment

deploy to production

BUILD

TEST & RELEASE
microservices architecture

service orchestration

front orchestrator

process claims
  module
  module

process quotes
  module
  module

process adjustments
  module
  module

update customer
  module
  module

notification
  module
  module
microservices architecture

service orchestration

process claims
module
module

process quotes
module
module

process adjustments
module
module

update customer
module
module

notification
module
module
microservices architecture

service orchestration
microservices architecture

service orchestration

- process claims
  - module
  - module

- process quotes
  - module
  - module

- process adjustments
  - module
  - module

- update customer
  - module
  - module

- notification
  - module
  - module
microservices architecture

service orchestration
consumer driven contracts
http://martinfowler.com/articles/consumerDrivenContracts.html
microservices architecture

maximize easy evolution
support

Microservice is the first architectural style developed post-Continuous Delivery.
microservice implementation

http://www.infoq.com/presentations/Micro-Services
Service-based Architecture

user interface layer

integration hub middleware

client requests

service component
module module module module

service component
module module

service component
module module module
service-based architecture

is there a middle ground?
service-based architecture

service granularity

database scope

integration hub
service-based architecture

service granularity
service-based architecture

service granularity

client requests

user interface layer
service-based architecture

service granularity

single-purpose micro-service to "portion of the application" macro-service

👍 macro-services resolves orchestration and transactional issues

👍 allows for complex business processing within a service context
service-based architecture

service granularity

single-purpose micro-service to "portion of the application"
macro-service

- services become harder to develop and test
- deployment pipeline requires more planning
- change control becomes more difficult
service-based architecture

database scope
service-based architecture

database scope
service-based architecture

database scope

single-purpose service-based database to globally shared application database

Storyboard:
- reduces service orchestration and contract dependencies
- improves performance due to fewer remote calls
- refactoring entire database may not be feasible or possible
service-based architecture

database scope

single-purpose service-based database to globally shared application database

- looser bounded context of services
- tighter service coupling based on schema
- schema changes become expensive and difficult
service-based architecture

integration hub
service-based architecture

integration hub
service-based architecture
integration hub

lightweight message broker to heavier integration hub

- allows for transformation of contract differences
- allows for non-transactional orchestration of services
- allows for protocol-agnostic heterogeneous interoperability
- allows for common processing logic across all services
service-based architecture

integration hub

lightweight message broker to heavier integration hub

- decrease in overall performance
- added complexity and cost
- increased need for governance
- deployment pipeline requires much more planning
- services become harder to develop and test
what people really mean when they say “microservice”:

A domain-centric service based architecture with modern DevOps practices.
migrating architectures
electronics recycling application

- public requests
- receiving department
- recycling and accounting
electronics recycling application
service-based architecture

adding microservices

client requests

user interface layer

client requests

client requests

service component

service component

service component

module module module module
module module module module
module module module module
module module module module

module module module module
module module module module
module module module module
module module module module

module module module module
module module module module
module module module module
module module module module

module module module module
module module module module
module module module module
module module module module

module module module module
module module module module
module module module module
module module module module
service-based architecture

adding microservices
service-based architecture

adding microservices
comparing:

Micro

Service-oriented

Service-based
characteristics differences

overall agility

ability to respond quickly to constant change in both business and technology

monolithic architecture

service-oriented architecture

service-based architecture

microservices architecture
characteristics differences

ease of deployment

promotes an effective and fast deployment pipeline; features are quick and easy to deploy

- monolithic architecture
- service-oriented architecture
- microservices architecture
- service-based architecture
characteristics differences

ease of testing

ease at which features can be tested and verified; confidence level in completeness of testing

monolithic architecture

service-oriented architecture

service-based architecture

microservices architecture
characteristics differences

overall performance

which patterns relatively promote better performing applications?

- monolithic architecture
- service-oriented architecture
- microservices architecture
characteristics differences

overall scalability

how well does the architecture pattern lend itself to highly scalable applications?

- monolithic architecture
- service-based architecture
- service-oriented architecture
- microservices architecture

level of scalability
characteristics differences

overall simplicity

level of complexity in applications implemented using the architecture pattern

monolithic architecture

service-oriented architecture

service-based architecture

microservices architecture
service differences

service-oriented architecture

microservices architecture

service-based architecture

service granularity

service numbers
service differences

service granularity

what is the typical granularity of services within this pattern?
service numbers

what is the typical upper limit of the number of services found?
Which ?!?
Which ?!? 

Monolith

default

easy to understand/build

doesn’t scale well in any dimension
Which ?!? 

service taxonomy

high degree of (potential) reuse

highly compatible in integration-heavy environments

operationally complex
Which ?!?&

incremental change

highly evolvable

complex interactions
Which ?!? 

best target for migration 

good compromise 

domain-centric without going $\mu$service
Which ?!? 

it depends
domain/architecture
isomorphism

mechanical sympathy
nealford.com

@neal4d

nealford.com/videos

www.oreilly.com/software-architecture-video-training-series.html