

ARCHITECTURAL STYLES IN SERVICE-BASED ARCHITECTURE



JOHAN ZETTERSTRÖM



CADEC 2019.01.24 & 2019.01.30 | CALLISTAENTERPRISE.SE

CALLISTA

— ENTERPRISE —

THE ARCHITECTS



Mark Richards



Neal Ford

AGENDA

- The three service-based architectural styles:
 - Service-oriented architecture
 - Microservices
 - Miniservices
- Background, drivers and negatives
- Miniservices & migration
- Comparing SOA to Microservices
- Summary

SERVICE-ORIENTED ARCHITECTURE

SOA - DRIVERS

Background

- Decomposition of monolithic systems
- Reuse of information and functionality
- Services, but point-to-point integration
- Heterogenous integration capabilities



DRIVERS

SOA

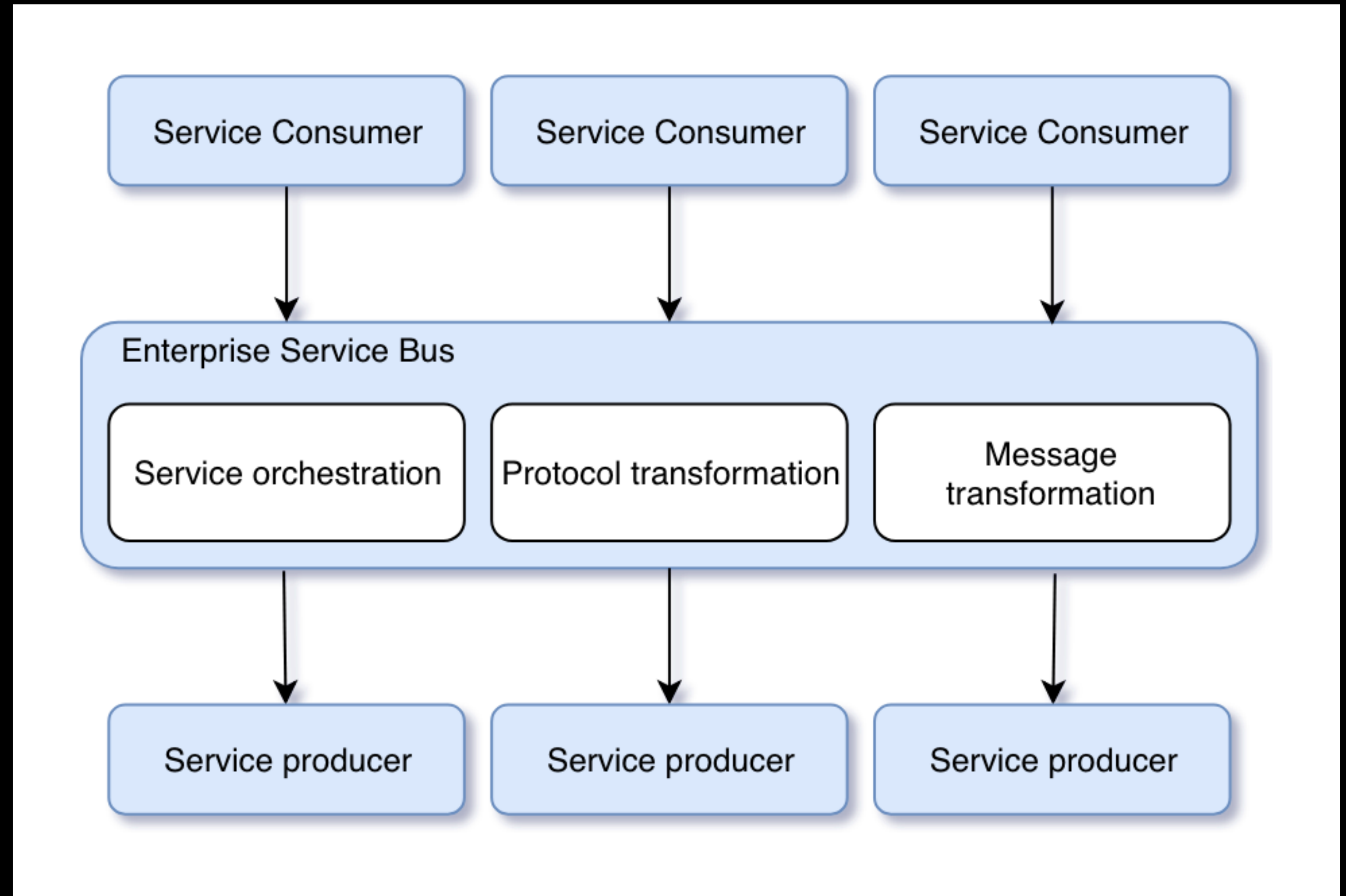
- Reuse (of services)
- Loose coupling
- Centralised integration capabilities

Microservices

Miniservices

SOA

- Integration capabilities on a central integration platform (ESB)
- Integration Competence Center (ICC)
 - Conway's law in reverse... (https://en.wikipedia.org/wiki/Conway%27s_law)



| SOA - NEGATIVES

- The integration platform becomes a bottleneck
- The organisation (ICC) becomes a bottleneck

MICROSERVICES

■ MICROSERVICES - DRIVERS

Background

- "Web scale" - potentially huge numbers of users
- Cloud - enables
- DevOps

DRIVERS

SOA

- Reuse (of services)
- Loose coupling
- Centralised integration capabilities

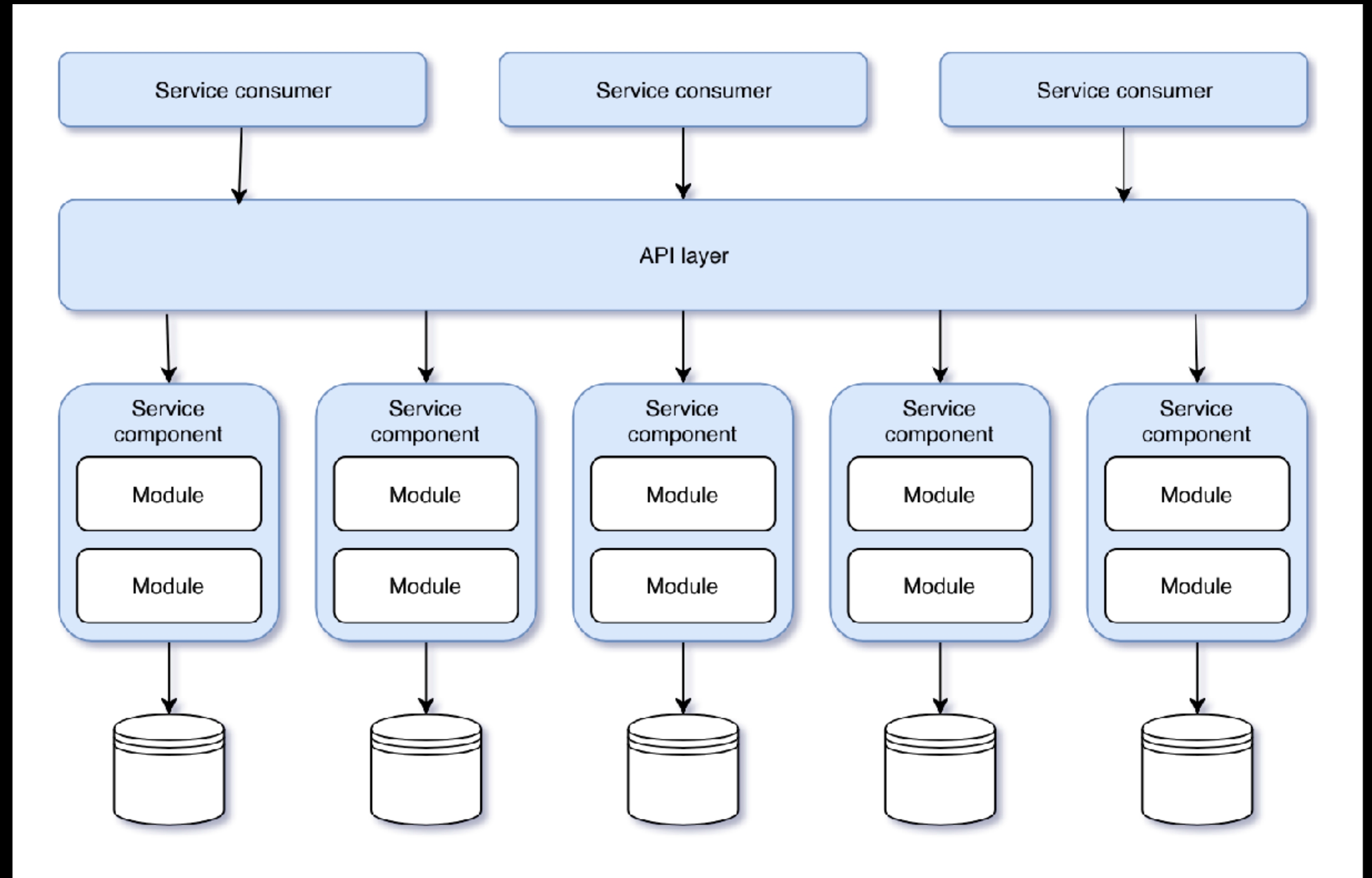
Microservices

- Scalability
- Changeability

Miniservices

MICROSERVICES

- Small independently deployable service components
- Service owns its data
- Lightweight API layer focused on non-functional aspects (service exposure, routing, security, throttling)



MICROSERVICES - NEGATIVES

It's complicated...

EDGE SERVER

HOW TO HIDE PRIVATE SERVICES?
HOW TO PROTECT PUBLIC SERVICES?

CENTRALIZED CONFIGURATION

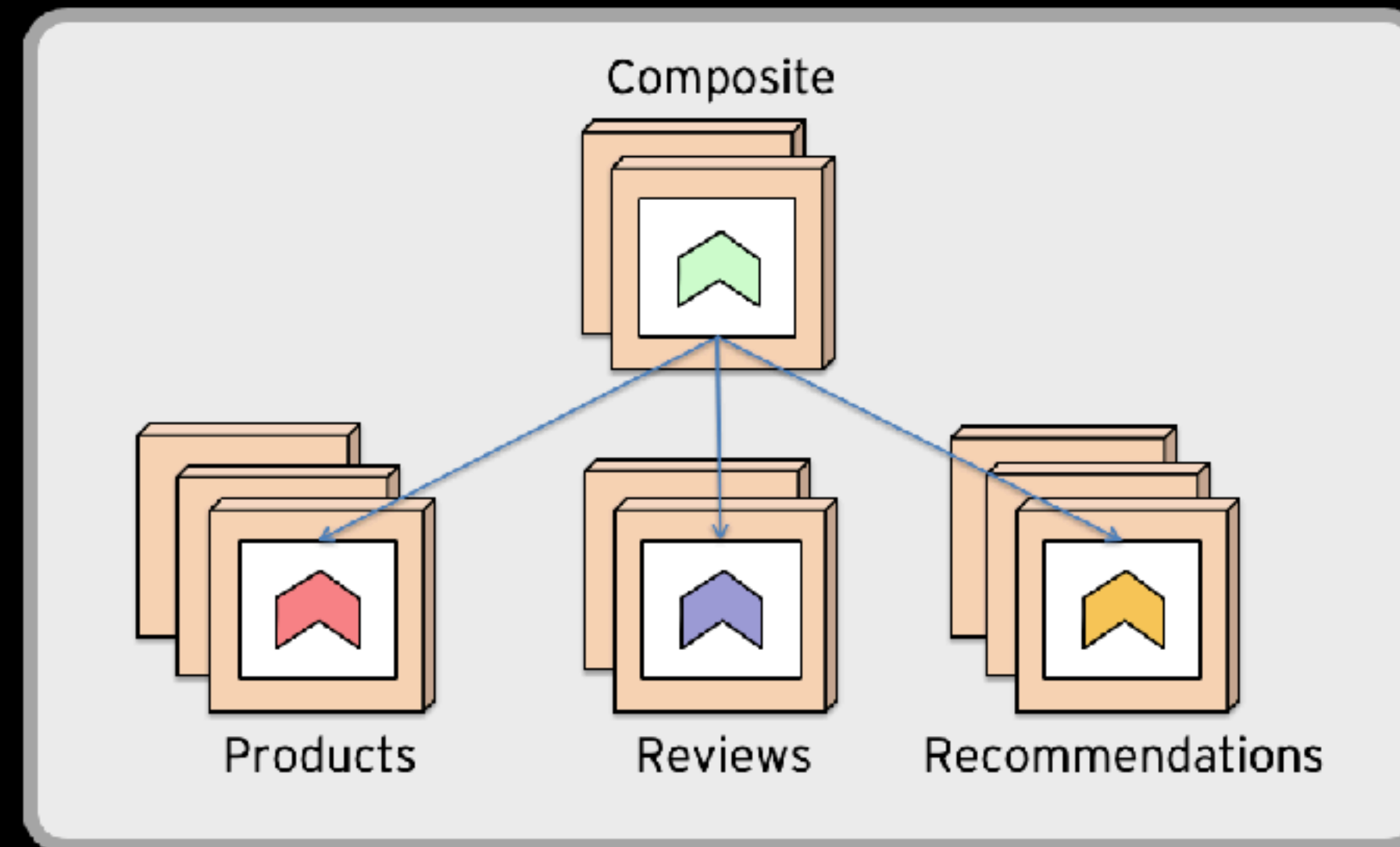
WHERE IS MY CONFIGURATION?
ARE ALL SERVICES
CONFIGURATION UP TO DATE?

LOG ANALYSIS

WHERE ARE THE LOGS?
HOW TO CORRELATE LOGS
FROM DIFFERENT SERVICES?

DISCOVERY SERVER

WHERE ARE THE SERVICES?
WHICH SERVICE TO CALL?



SERVICE MANAGEMENT

HOW TO

- DEPLOY SERVICES?
- SCALE SERVICES?
- UPGRADE SERVICES?
- RESTART FAILING SERVICES?

RESILIENCE

HOW TO HANDLE FAULTS?

- SLOW OR NO RESPONSE
- TEMPORARY FAULTS
- OVERLOAD

DISTRIBUTED TRACING

WHO IS CALLING WHO?

TRAFFIC MANAGEMENT

HOW TO CONTROL ROUTING?

- RATE LIMITING
- CANARY & BLUE/GREEN UPGRADES

OBSERVABILITY

HOW ARE MY SERVICES PERFORMING?

MONITORING

WHAT HARDWARE RESOURCES ARE USED?

MINISERVICES

MINISERVICES - DRIVERS

Background

- Microservices...

DRIVERS

SOA

- Reuse (of services)
- Loose coupling
- Centralised integration capabilities

Microservices

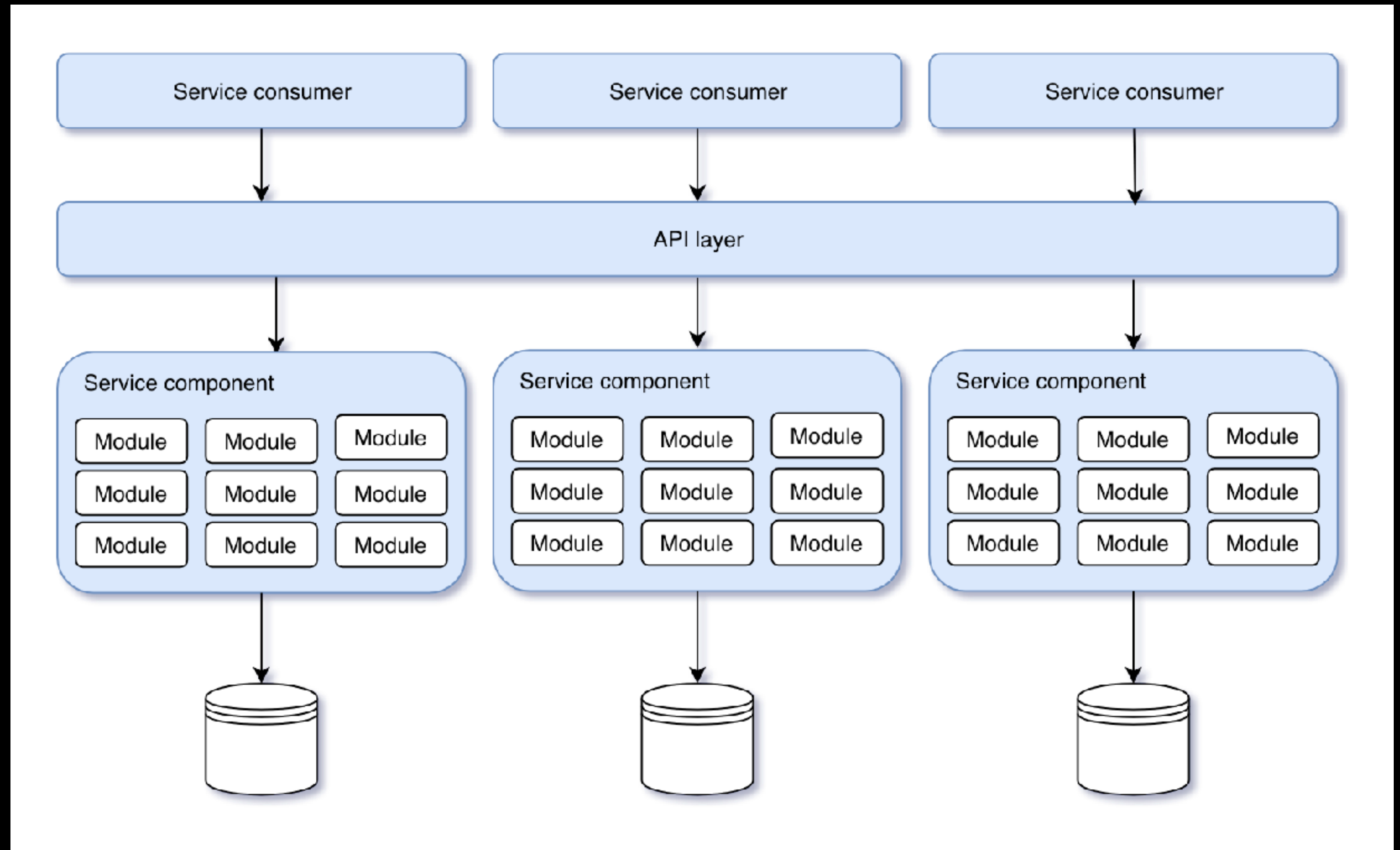
- Scalability
- Changeability

Miniservices

- Complex business logic
- Less decomposition of data

MINISERVICES

- A bit larger independently deployable service components
- To the consumer, indistinguishable from a microservice
- Slower to change, less stress on DevOps



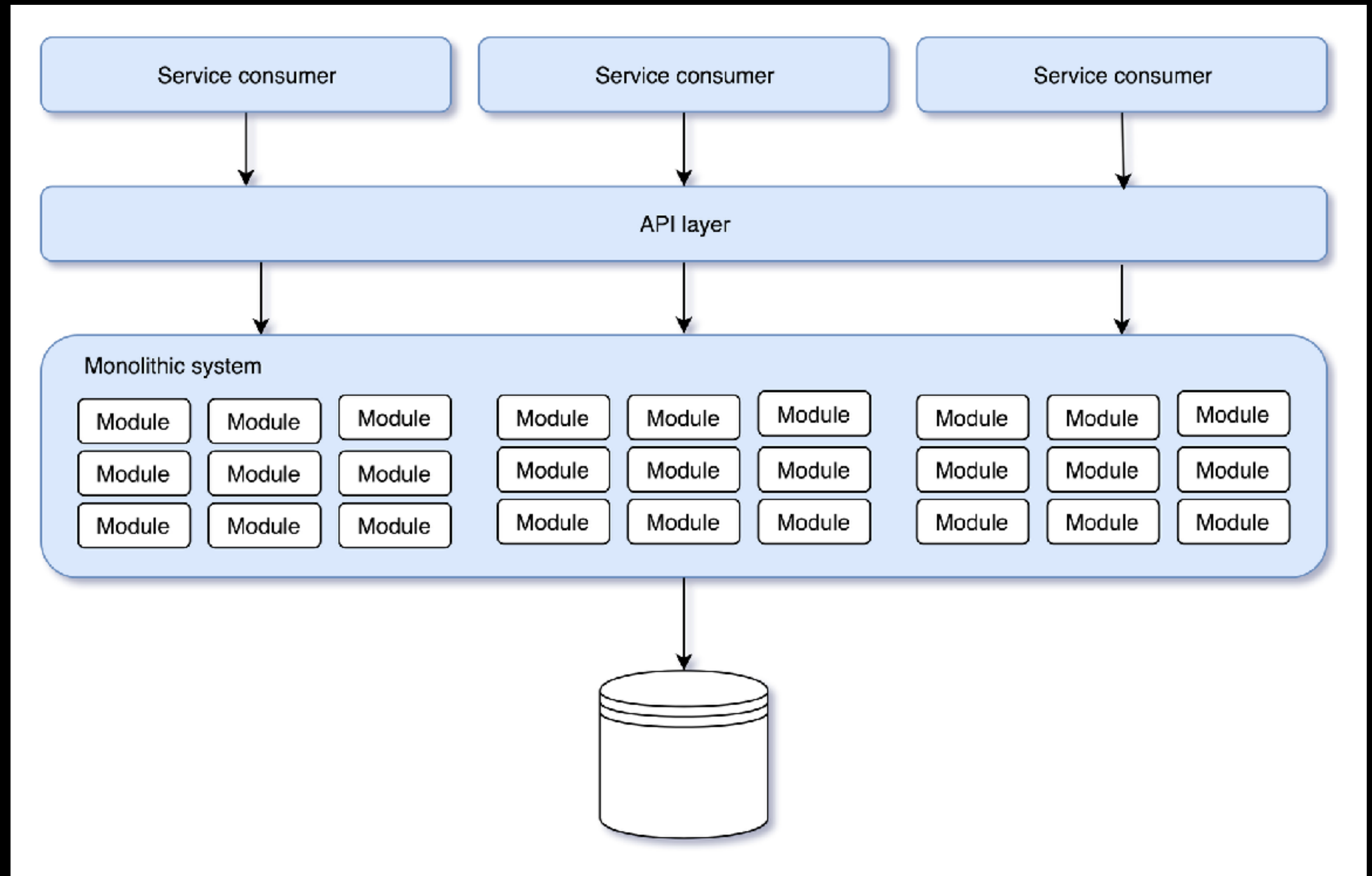
MINISERVICES - NEGATIVES

- Changeability
- Scalability
- Modules within service become entangled over time

MINISERVICES & MIGRATION

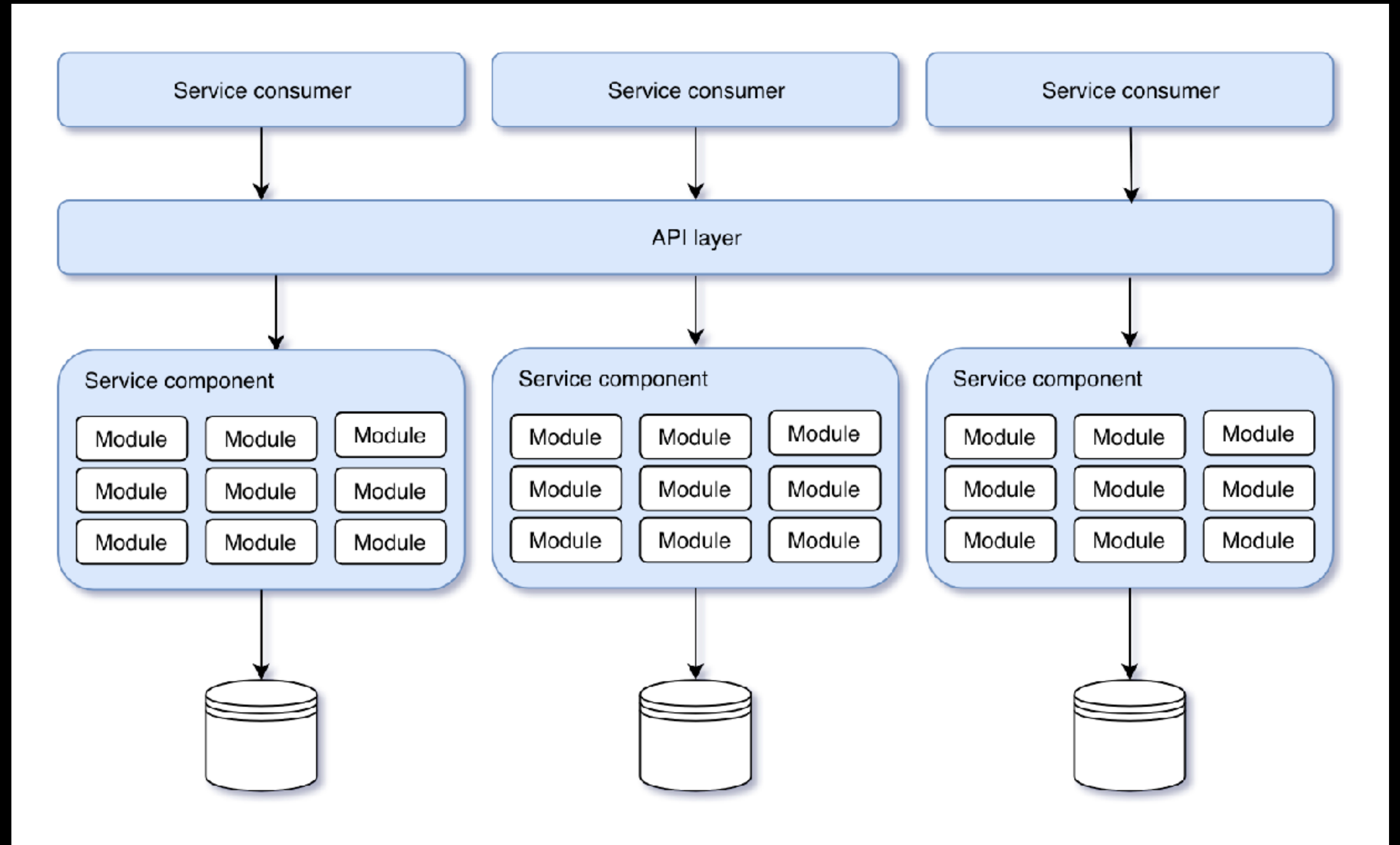
MINISERVICES & MIGRATION

- Monolith



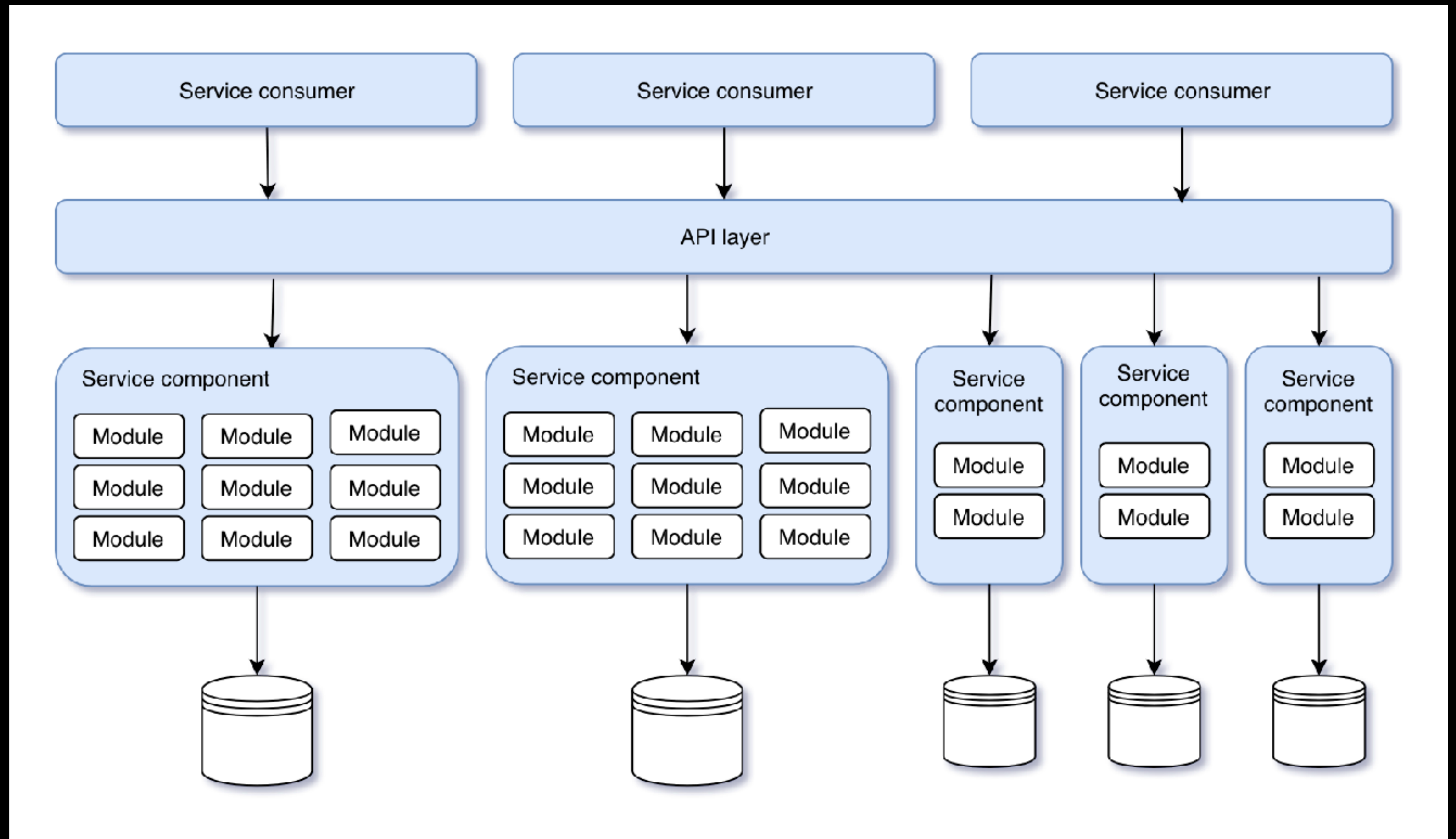
MINISERVICES & MIGRATION

- Monolith -> Miniservices



MINISERVICES & MIGRATION

- Monolith -> Miniservices -> Microservices



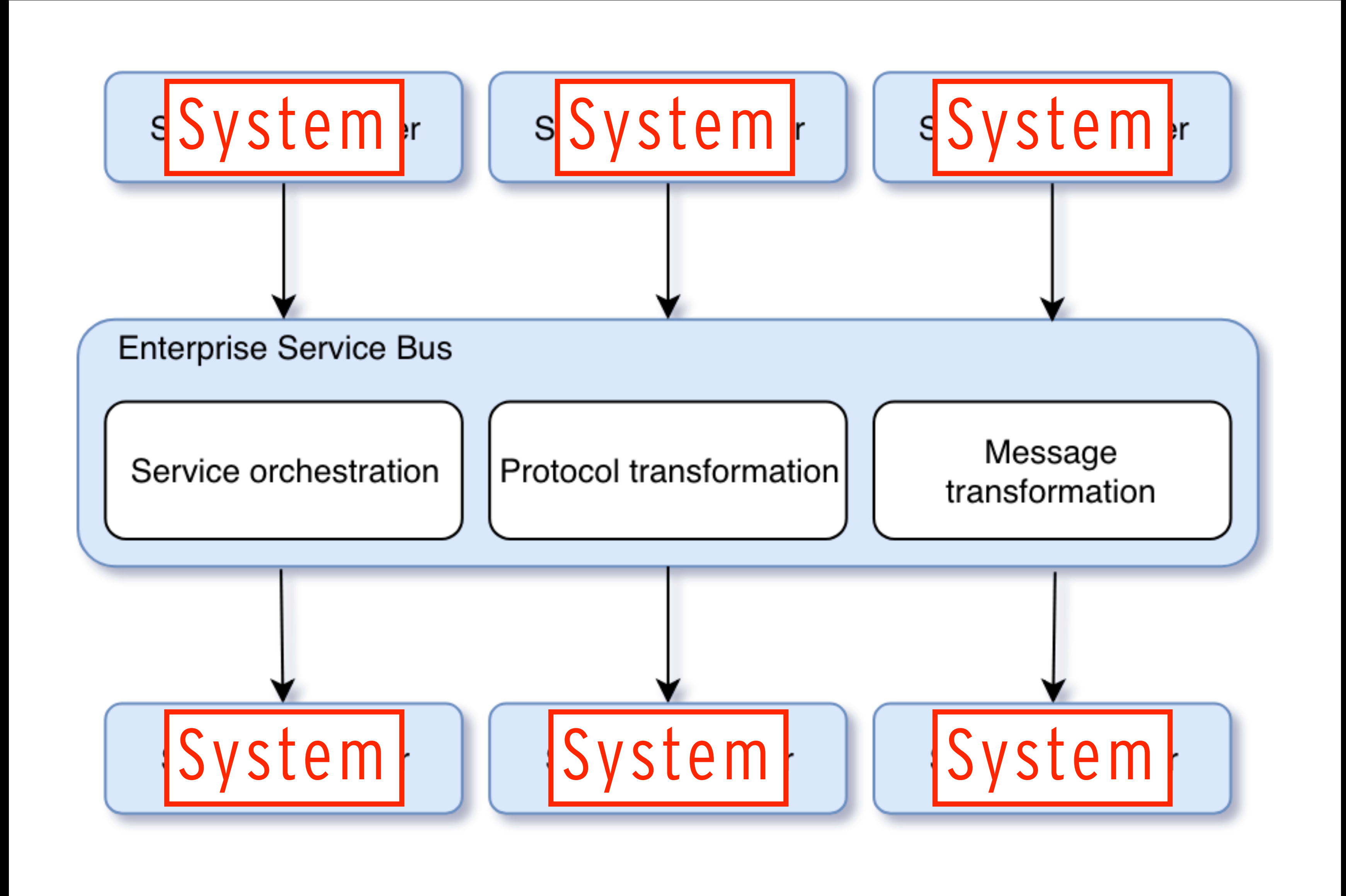
MINISERVICES & MIGRATION

- A caveat: Providing a stable API requires analysis of how the miniservice will be decomposed into microservices.
- Implementing a hybrid architecture
 - Minimize the differences...
 - » Use common infrastructure
 - » Use common routines (when possible)
 - ...but don't ignore them!
 - » Create a service taxonomy
 - » Different testing routines?
 - » Build & deploy

COMPARING SOA TO MICROSERVICES

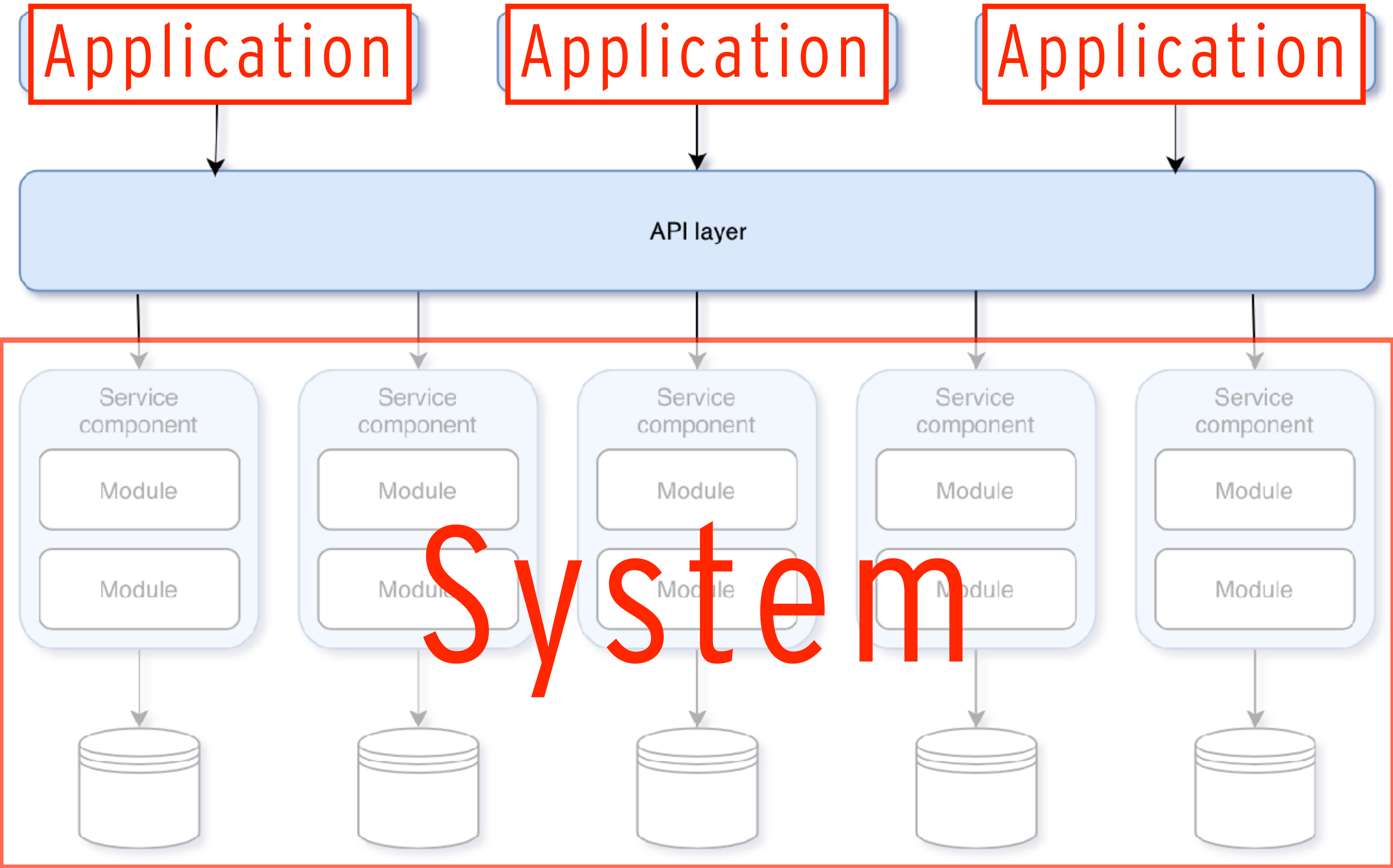
COMPARASION - GRANULARITY

SOA

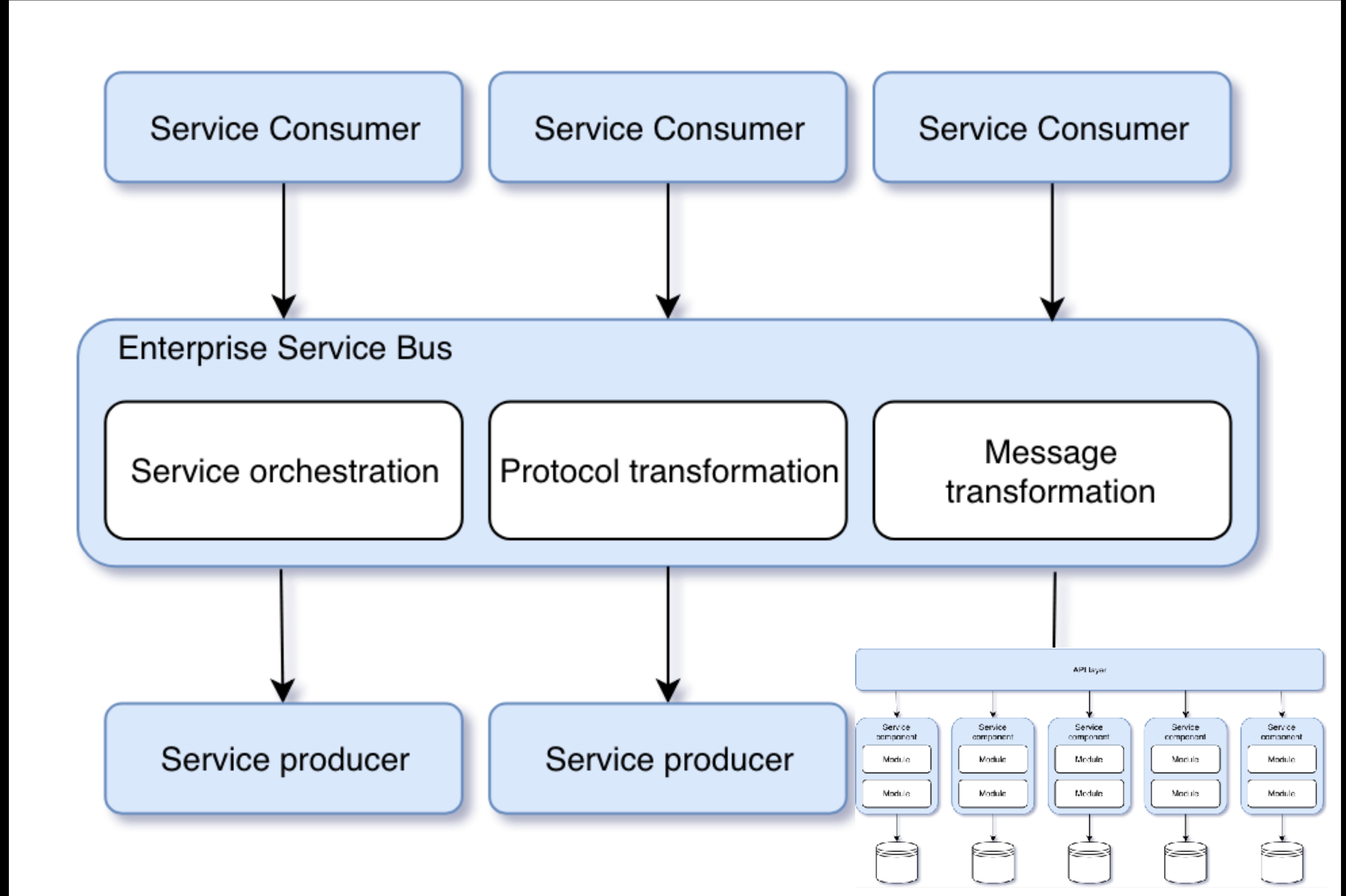


COMPARASION - GRANULARITY

MICROSERVICES



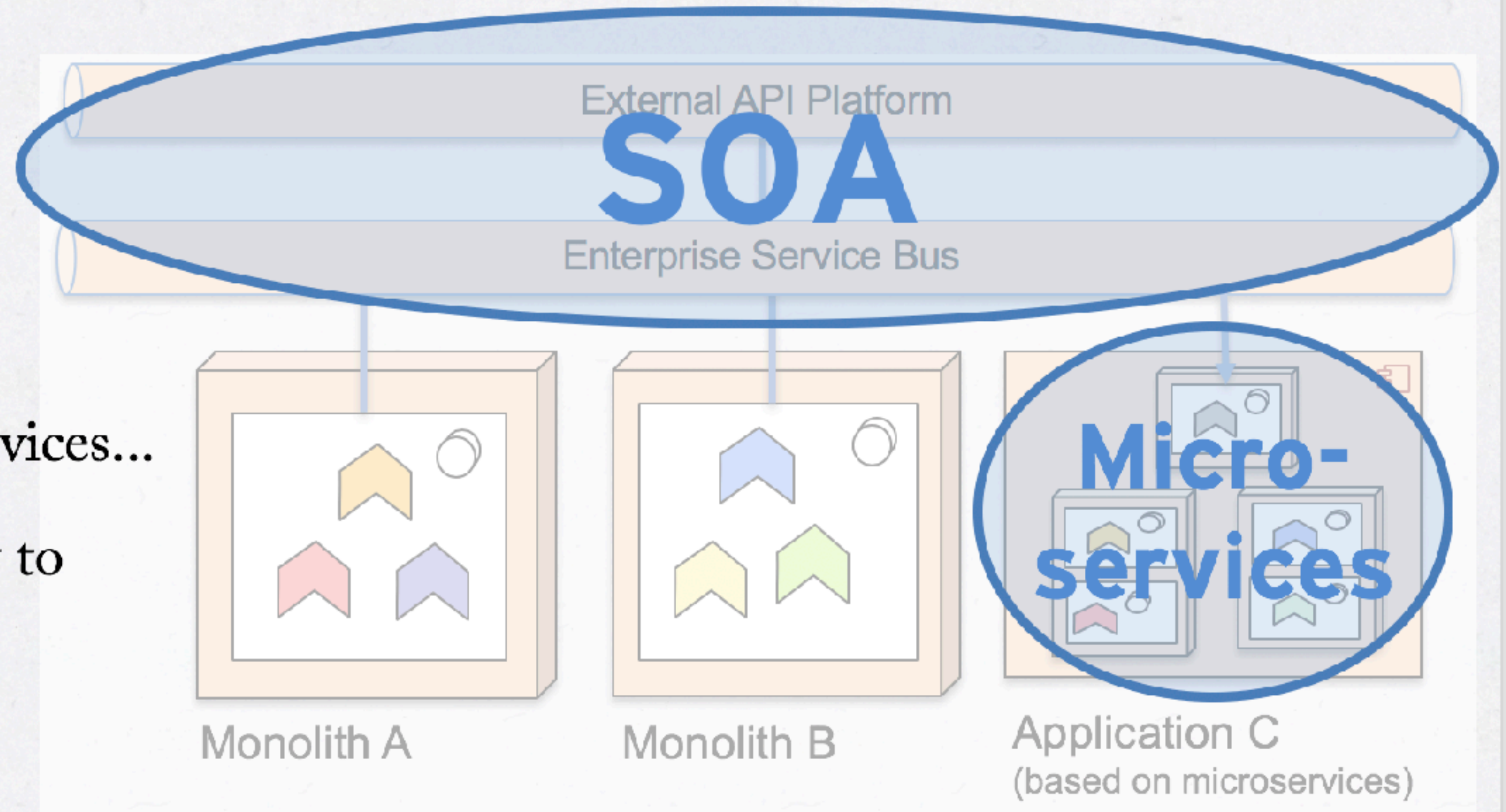
COMPARASION - GRANULARITY



WHAT'S A MICROSERVICE?

- SOA vs. Microservices

- SOA and microservices don't conflict, they complement each other!
- SOA is about how to reuse existing functionality as services...
- Microservices is about how to make functionality to scale better with high resilience and short release cycles...



| DIRECT COMPARISONS BECOME MISLEADING

”Microservices are SOA done right”

”Miniservices are SOA done right”

- Different problem areas
 - Integration in a heterogenous world
 - Scalable, changeable systems

| SUMMARY

- Recognize your miniservices!
- Neither microservices nor miniservices are SOA done right