goto; chicago

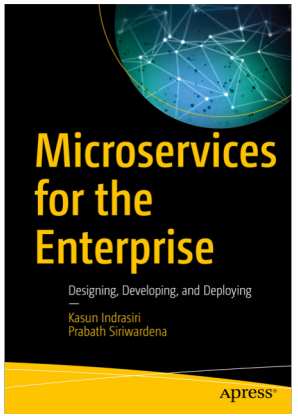
Application Integration in Cloud Native and Microservices Landscapes

Kasun Indrasiri

Author of "*Microservices for the Enterprise*", Director-Integration Architecture at *WSO2*

About Me..

- Director - Integration Architecture at WSO2
- Author “Microservices for the Enterprise”, “Beginning WSO2 ESB”
- Product Manager : WSO2 Enterprise Integrator
- Committer at Apache
- Founder : SF Bay Area “Microservices, APIs and Integration” meetup



What is Cloud native?

- Cloud native apps are composed of microservices or serverless functions
- Packaged in containers.
- Runs on a Continuous Delivery model.
- Adaptive governance.

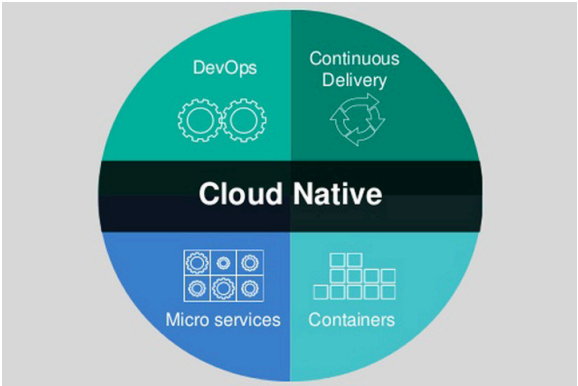


Image : https://www.eweek.com/cloud/red-hat-s-eight-steps-to-cloud-native-applications

Integrating Cloud native apps?

- Cloud native application
  - A set of interconnected microservices or serverless functions.

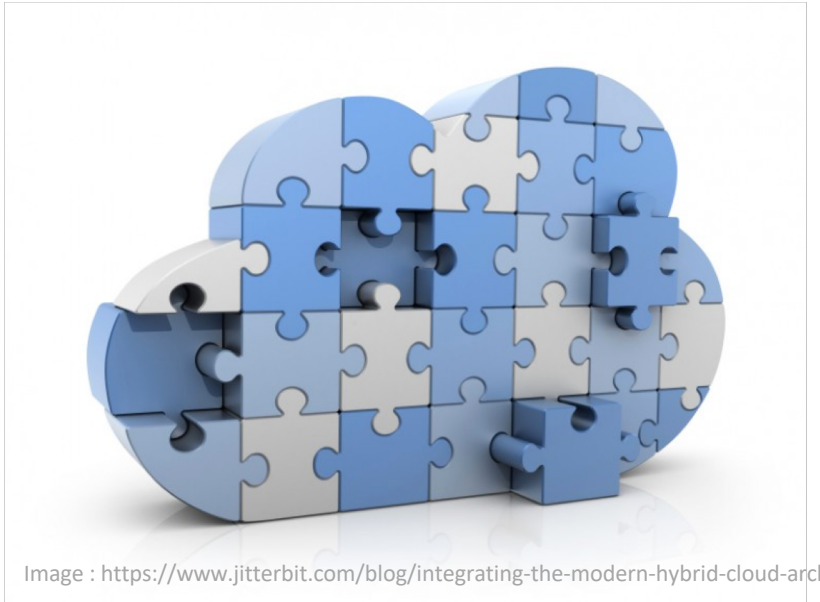
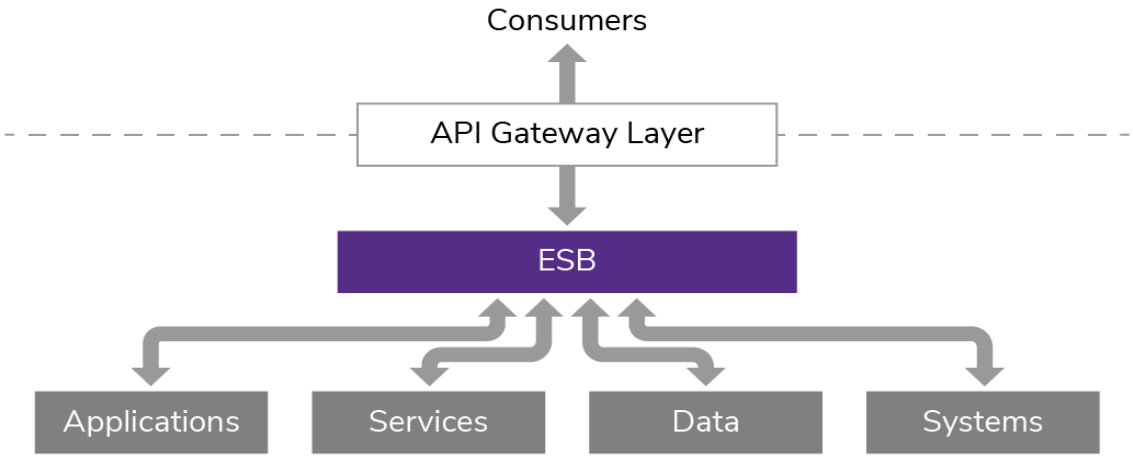


Image : https://www.jitterbit.com/blog/integrating-the-modern-hybrid-cloud-architecture/

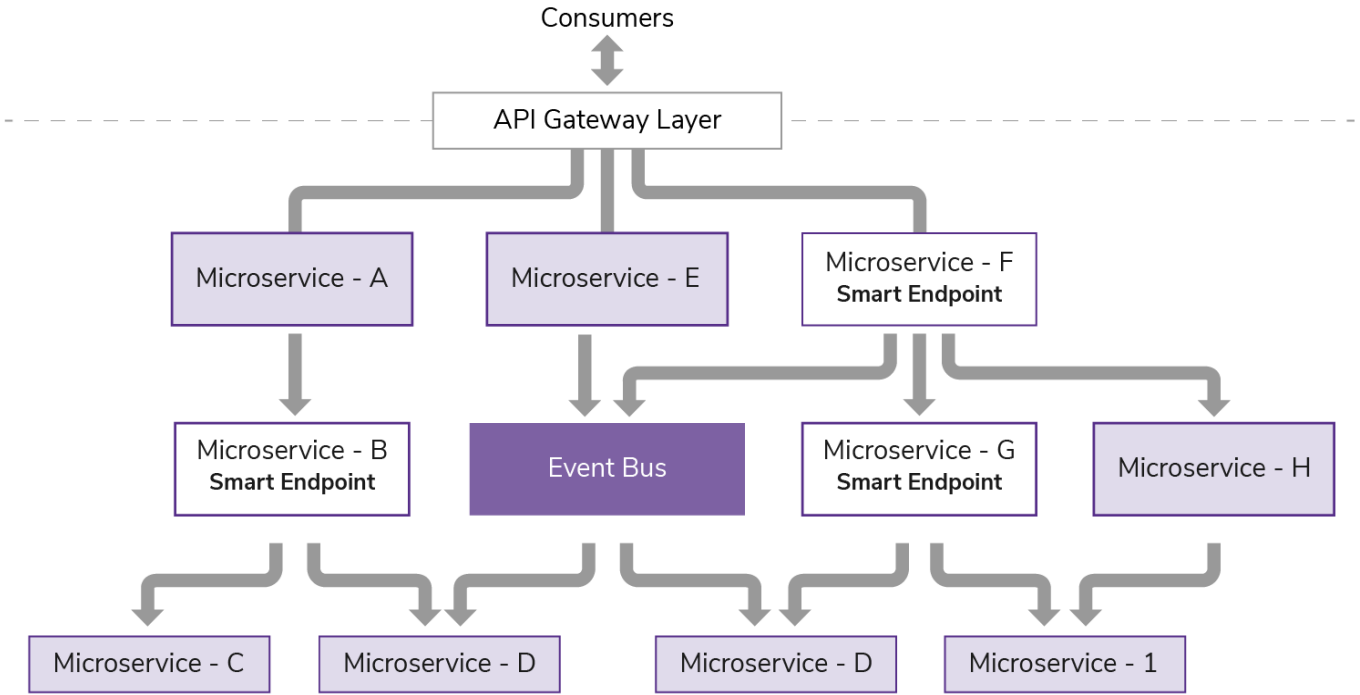
From ESB to ‘Smart-endpoints and Dumb Pipes’

- Using an ESB for integration
- API Gateway to expose your APIs



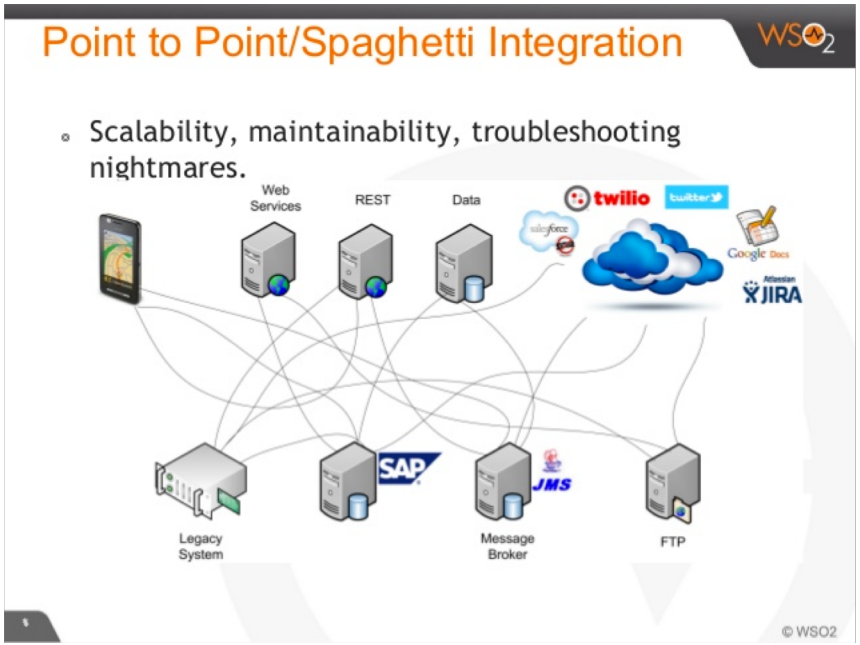
From ESB to ‘Smart-endpoints and Dumb Pipes’

- Smart endpoints and dumb pipes



From ESB to ‘Smart-endpoints and Dumb Pipes’

- Bringing back ’point to point’ integration… again?

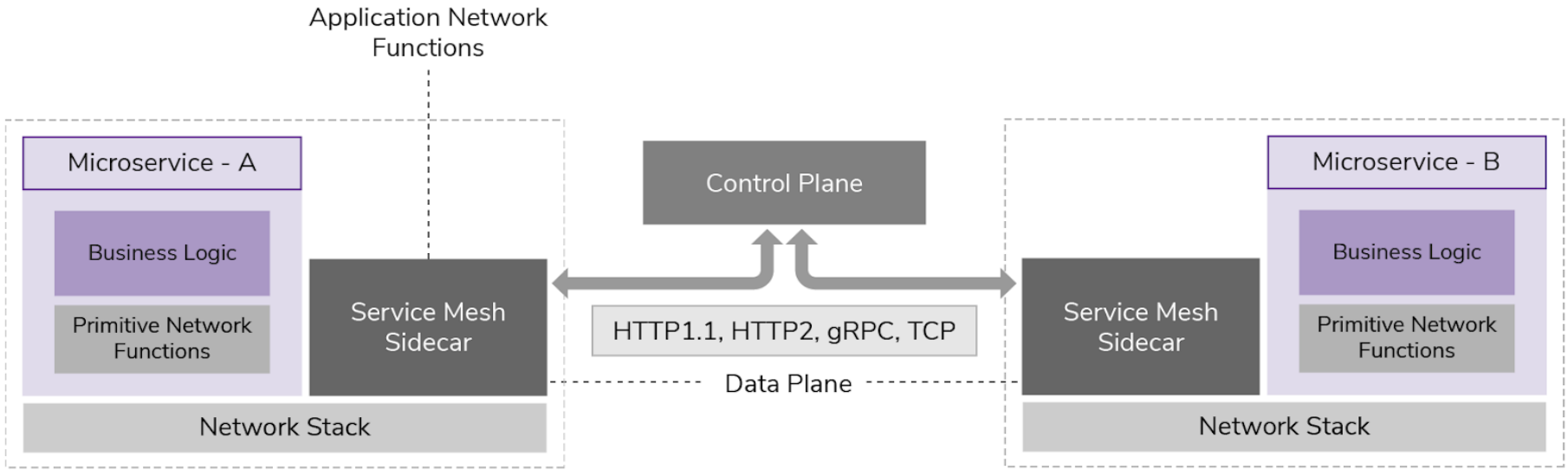


Building Smart-endpoints

- Why it is hard?
  - Capabilities that ESB provided has to be implemented at each service level.
  - Integration complexity is not reduced but dispersed across services.
  - Proliferation of services makes it even more harder.

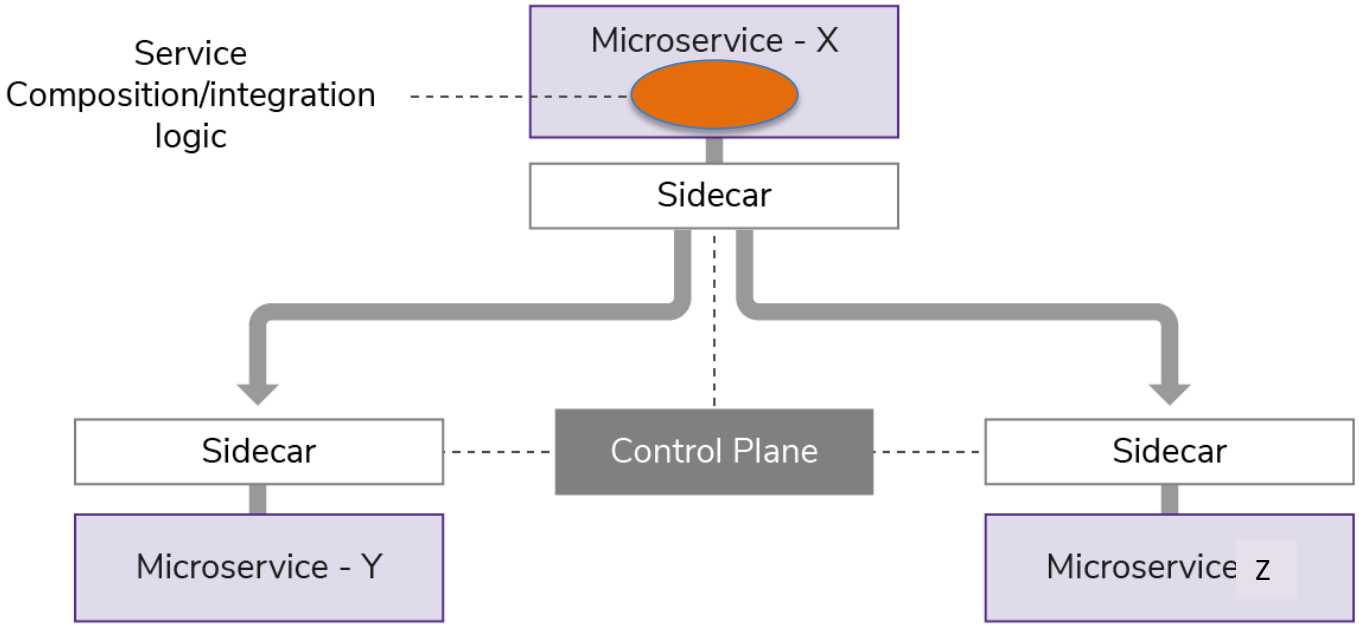
Service Mesh

- A application network infrastructure to build cloud native apps.



Service Mesh vs ESB?

- Service Mesh is ***NOT*** a distributed ESB.



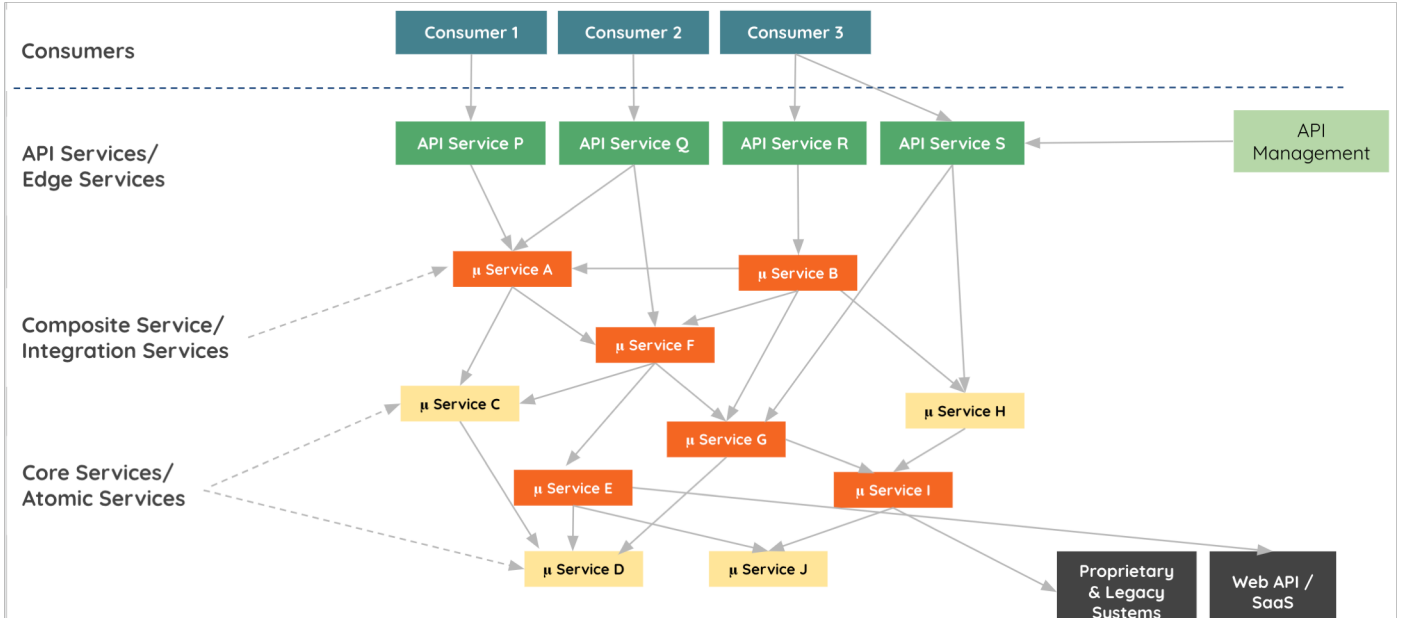
Cloud native application integration requirements

Requires a cloud native runtime

- Kubernetes native: Runs smoothly within Docker/K8s.
- Startup time and resource consumption.
- Seamless integration with the cloud native ecosystem.

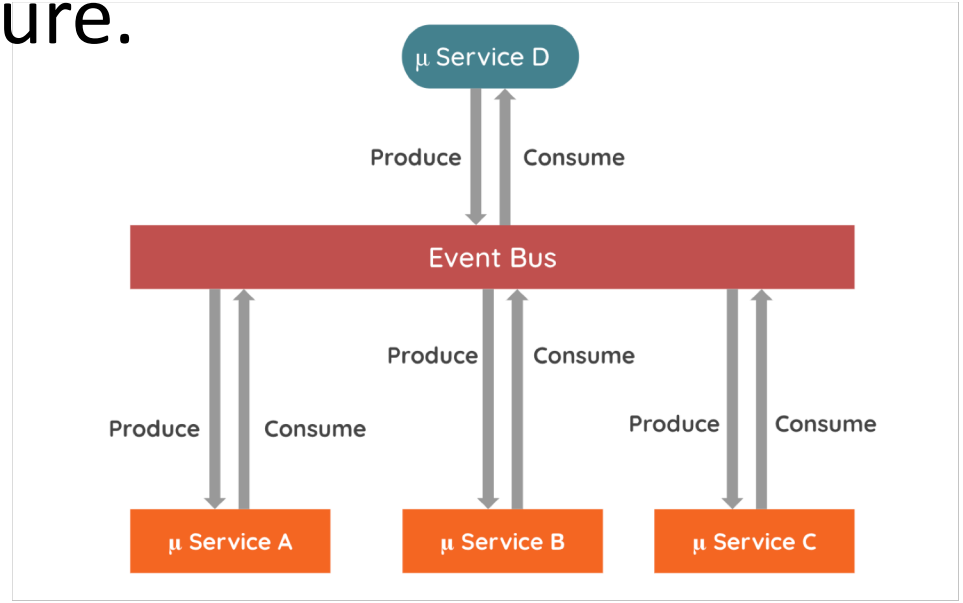
Service Orchestrations (Active compositions)

- A given service invokes one or more other services to fulfill a business use case.



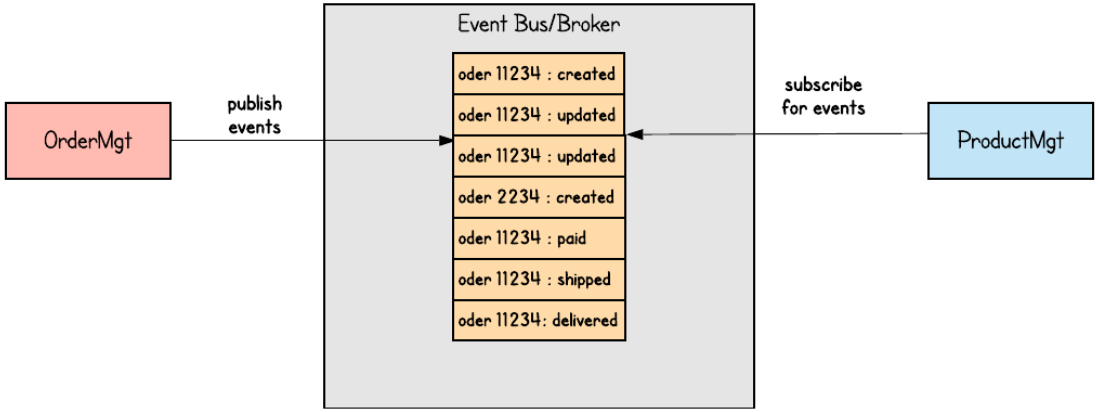
Service Choreography (Reactive compositions)

- Services communicates using event driven architecture.



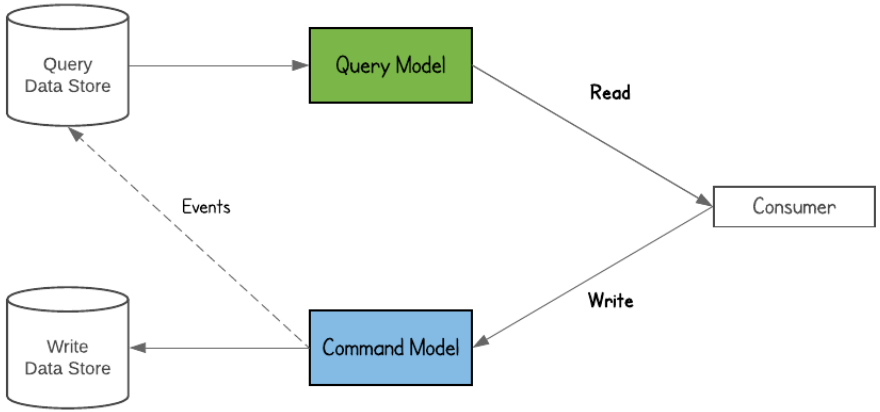
Service Choreography/Reactive compositions

- Event Sourcing
  - Persist each state changing events of an entity as a sequence of events.
  - All such events are stored in an event bus and consumers can derive the state by processing a sequence of events.



Service Choreography/Reactive compositions

- CQRS
  - **C**ommand **Q**uery **R**esponsibility **S**egregation.
  - CRUD model is not always feasible.
  - Split the common data model into query and command models.



Protocols abstractions

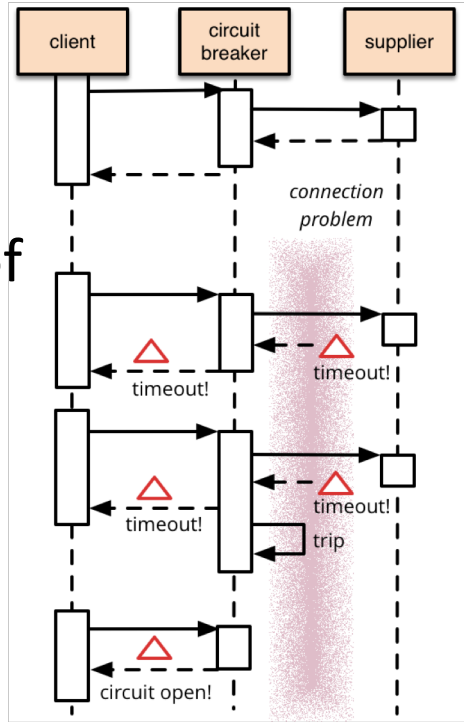
- Building applications using disparate protocols.
  - HTTP, gRPC, GraphQL, Web Sockets, Web Hooks
  - Kafka, NATS, AMQP
  - FTP, SFTP
  - TCP, Rsocket

Integration Patterns

- Forking, Joining, Splitting, Looping, and Aggregation of messages or service calls.
- Some of the Enterprise Integration Patterns (EIPs) are now implemented at each smart endpoint.

Resiliency

- Network is unreliable.. Now and forever!
- Not all apps runs on top of a service mesh
- Resiliency may be coupled to the business logic of the service
- Patterns
  - Timeout, Retry, Circuit Breaker, Fail-fast, Bulk-head

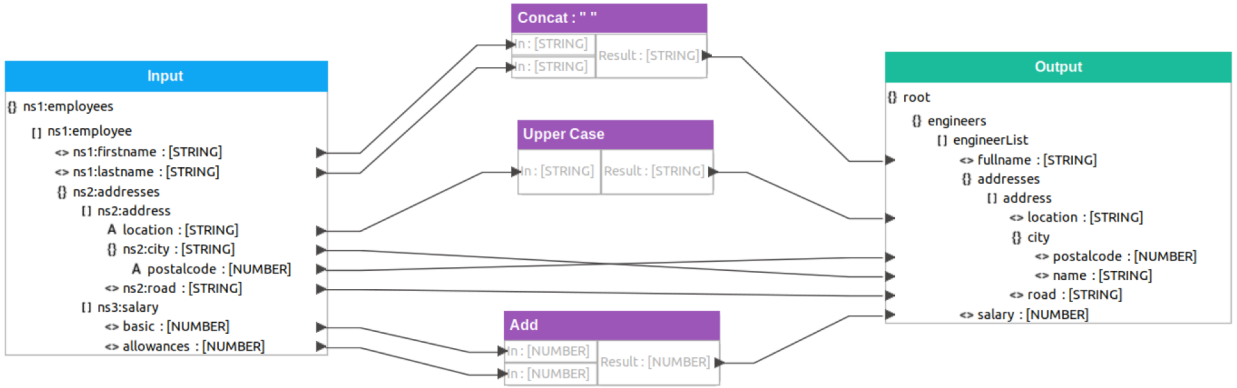


Message Delivery Semantics

- Store and forward.
- Persistent delivery.
- Idempotent messaging.

Type Conversions

- Message type mapping and transformations.
  - JSON, XML, ProtoBuf, Avro
  - Implemented graphically or programmatically.
    - E.g. [WSO2 Enterprise Integrator](#) – Data Mapper



APIs, Events and Streams

- APIs
- Events
- Streams

Integrating with SaaS and legacy on-prem

- Integration with SaaS (e.g. Salesforce)
- Proprietary and legacy systems (e.g. SAP)

Workflows and SAGAs

- Long running processes
- Compensation and distributed transactions
  - SAGA

Technologies for cloud native application integration

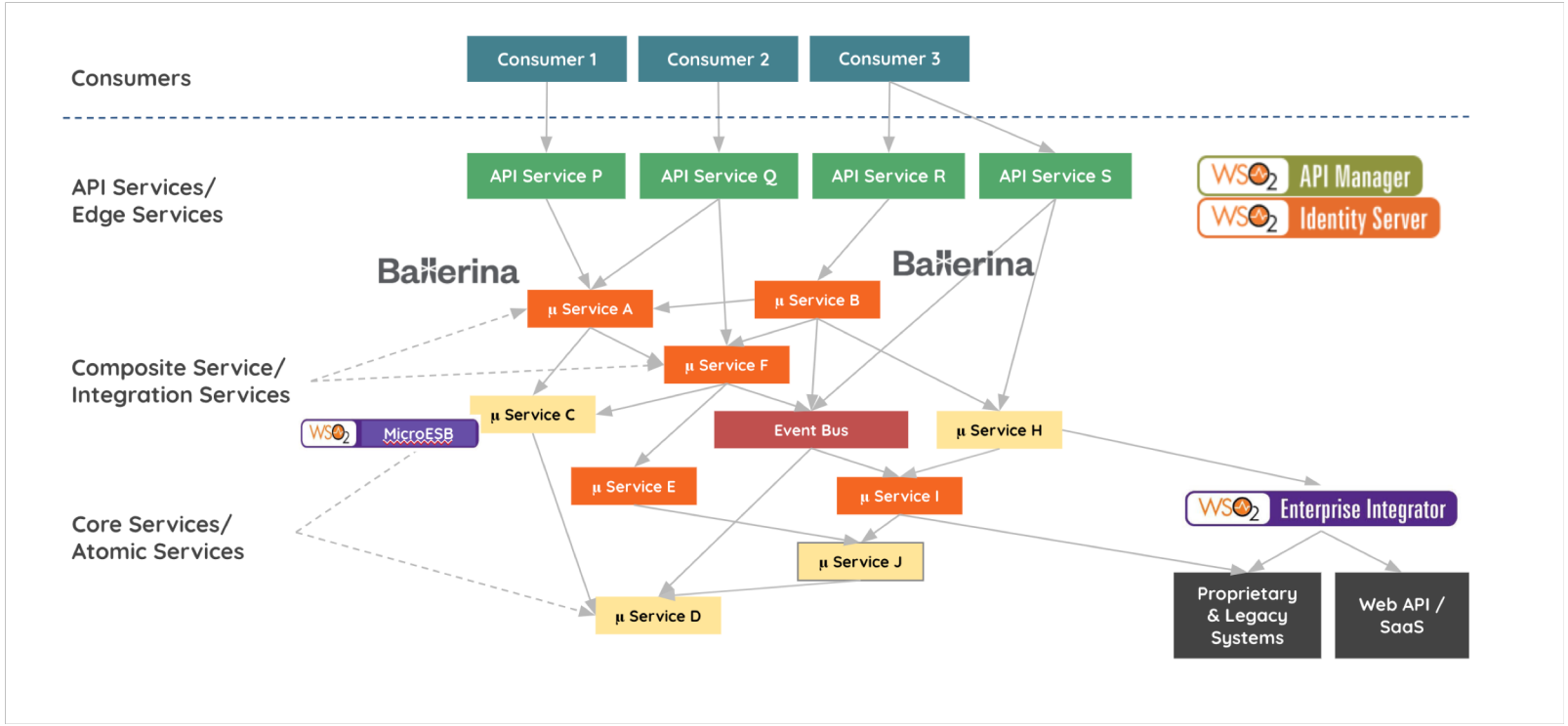
Integration Frameworks

- Camel-K
- WSO2 Micro integrator
- Spring Integration

Application Development Frameworks and languages

- Spring Boot, Vert.x
- Ballerina.io, Go lang, Node
- Micronaut, Quarkus.io

How WSO2 helps?



Summary

- Application integration is everywhere.
- Service Mesh is not for application integration.
- Selecting the best of breed technologies for application integration.

goto; chicago

Click ‘Rate Session’
to rate session
and ask questions.

follow us @gotochgo

goto; chicago

Please

Remember to rate this session

Thank you!

follow us @gotochgo

goto; chicago

Click ‘Rate Session’

Rate **5** sessions to get the supercool GOTO reward

follow us @gotochgo