

Creating Service Fabric Services



Ivan Gavryliuk

SOFTWARE ARCHITECT

@aloneguid <http://isoline ltd.com>



Overview



Reliable Services intro

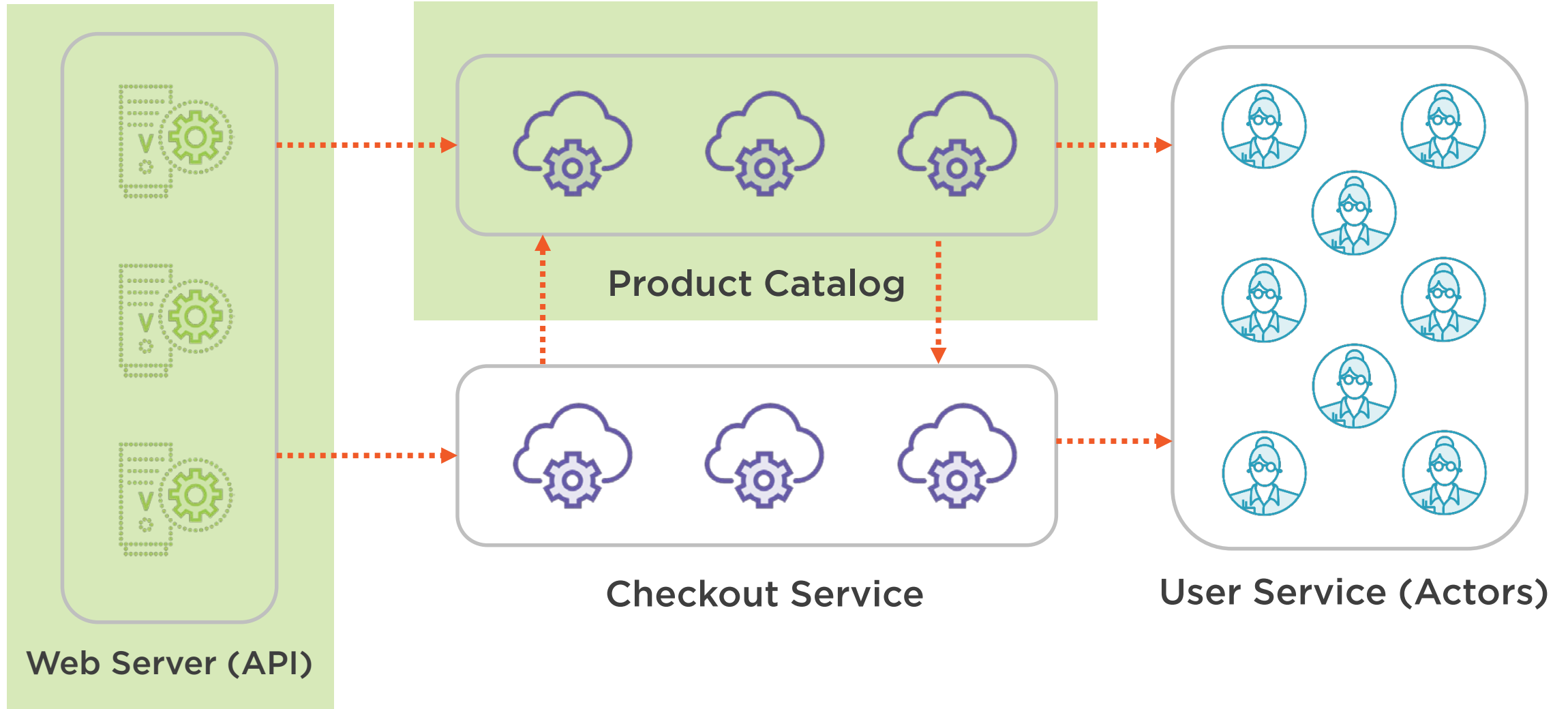
Application recap

Build two of four microservices

Service communication



Application



Two Services



Web API
(Stateless Service)



Product Catalog
(Stateful Service)

Reliable Services



Normal App vs. Reliable Service

An Application

Easy to write (established frameworks)

Great choice of libraries

No learning curve

A Reliable Service

Easy to write (established frameworks)

Great choice of libraries (x64 only)

None to little learning curve

Access ASF API

Pluggable communication model

Access to Reliable Storage

Simple programming model



Creating Product Catalog



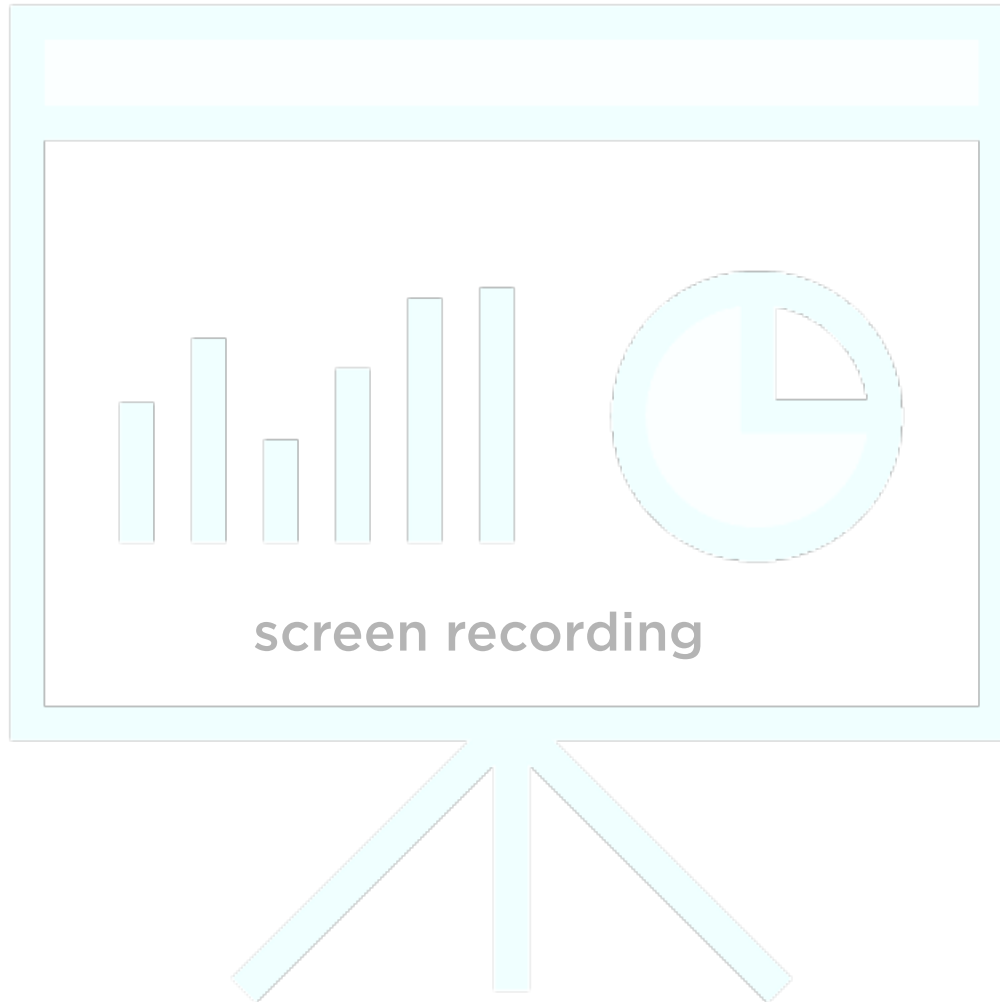
Solution Structure



Azure Service Fabric project (one)



Service Project (many)



Service Initialization



Register the Reliable Service

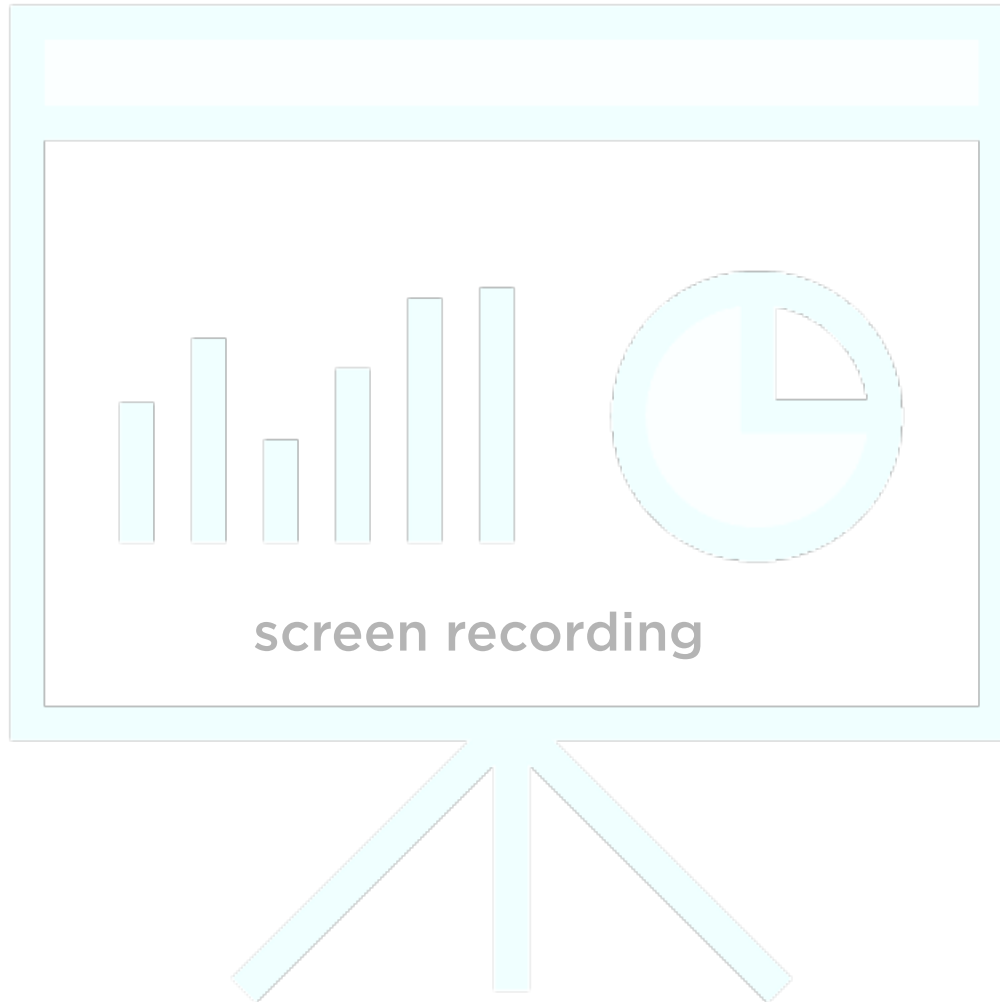


Log Reliable Service has started



Sleep forever

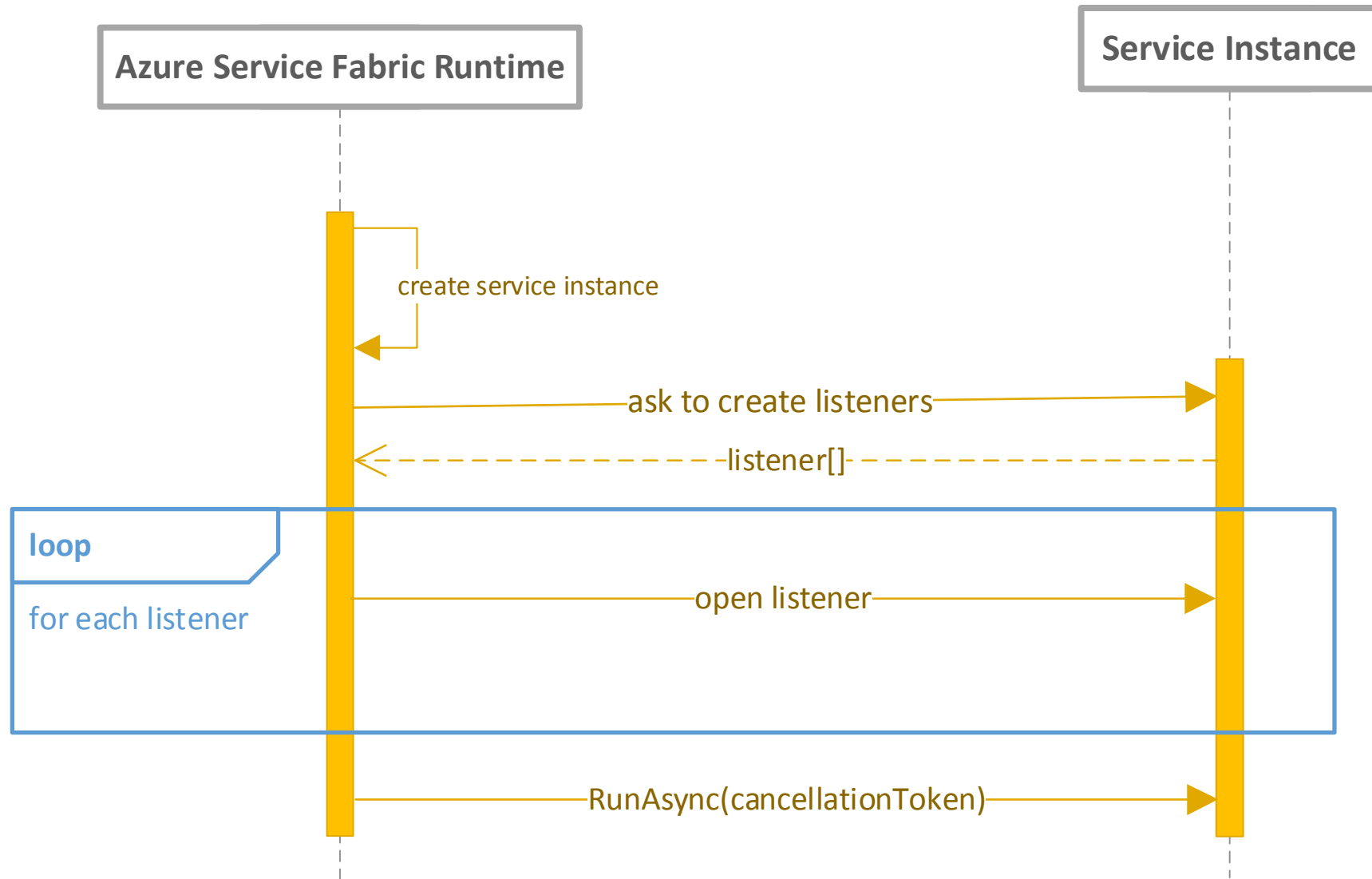




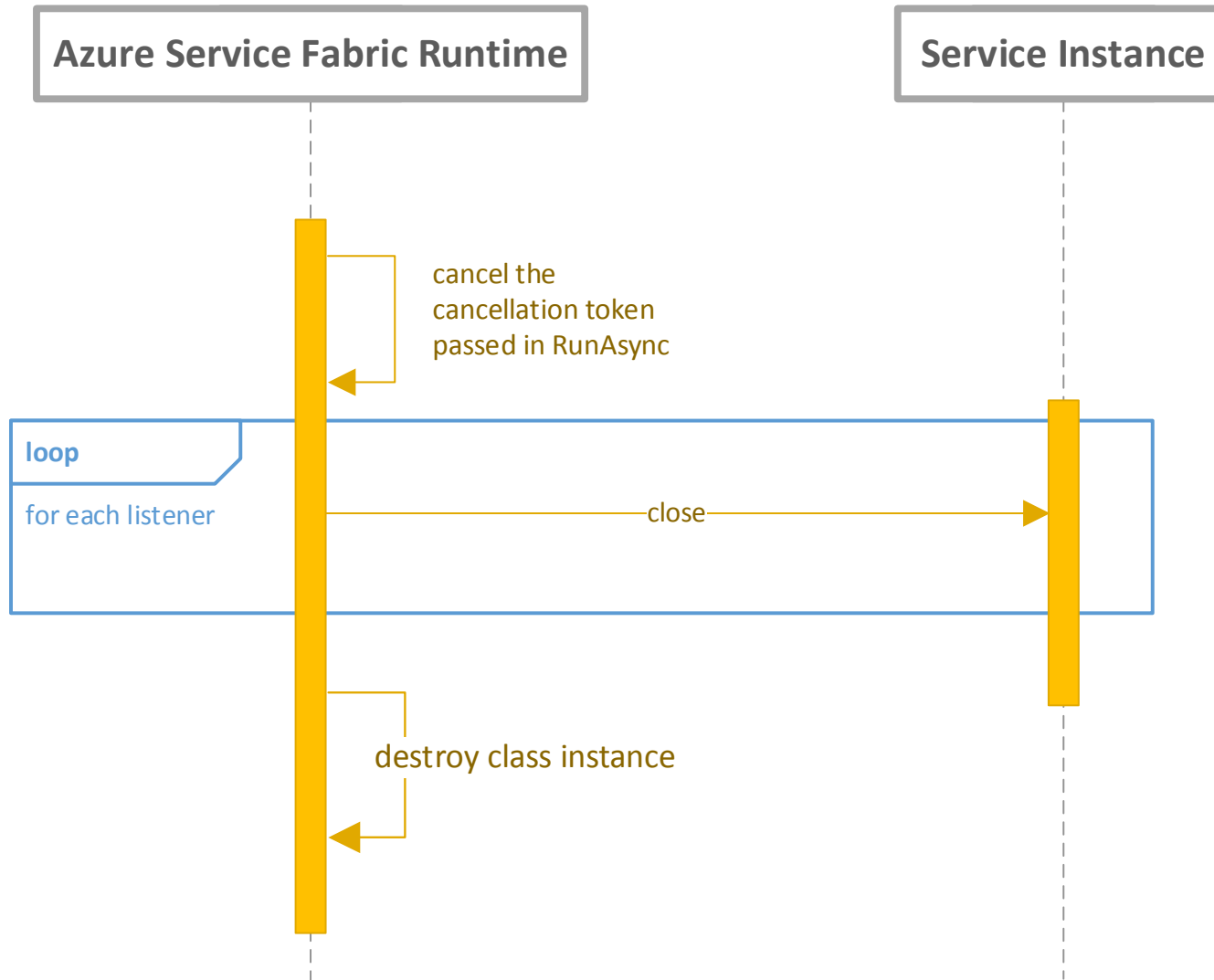
Service Lifecycle



Startup



Shutdown



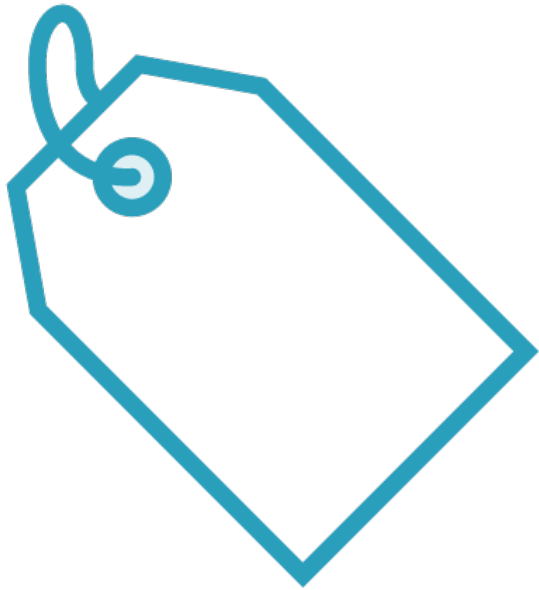
Always respond to
cancellationToken event as
soon as possible



Product Catalog Database



Product Entity



Product ID (GUID)

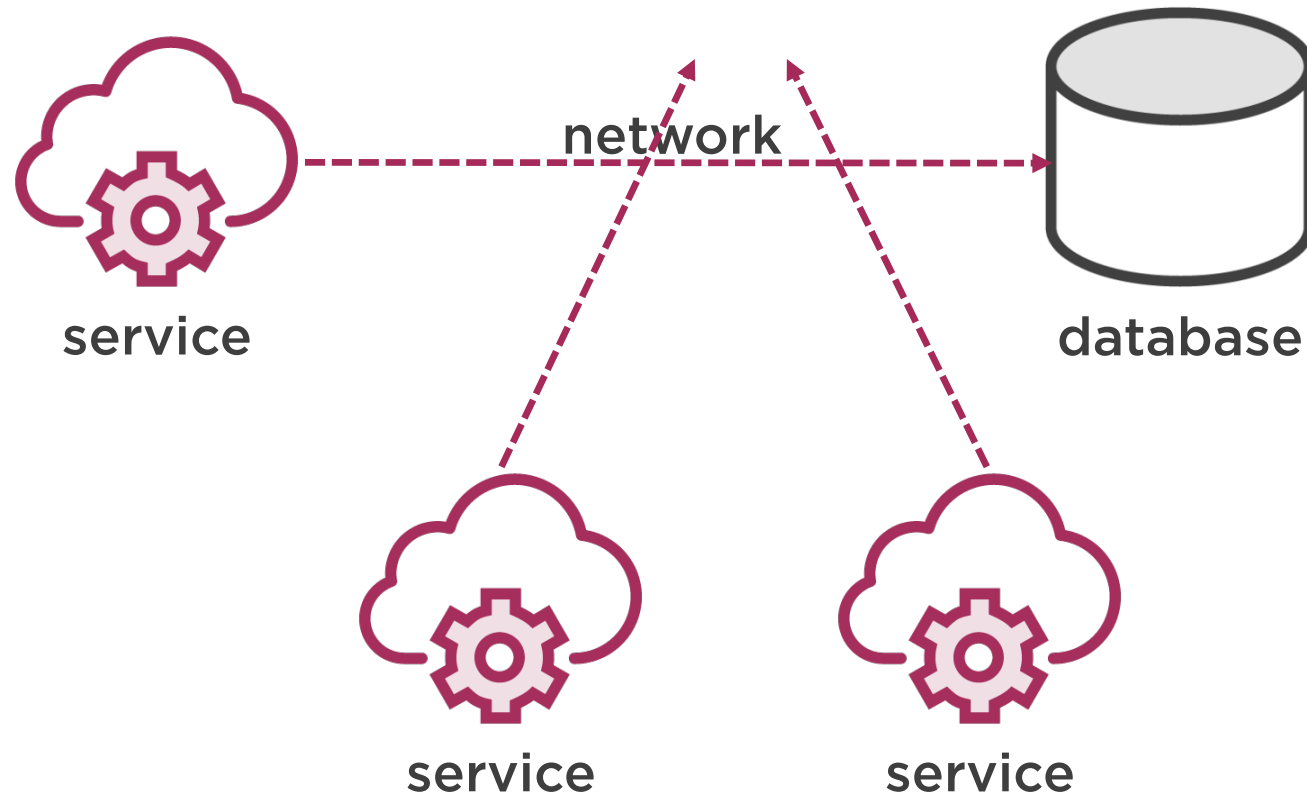
Name (String)

Description (String)

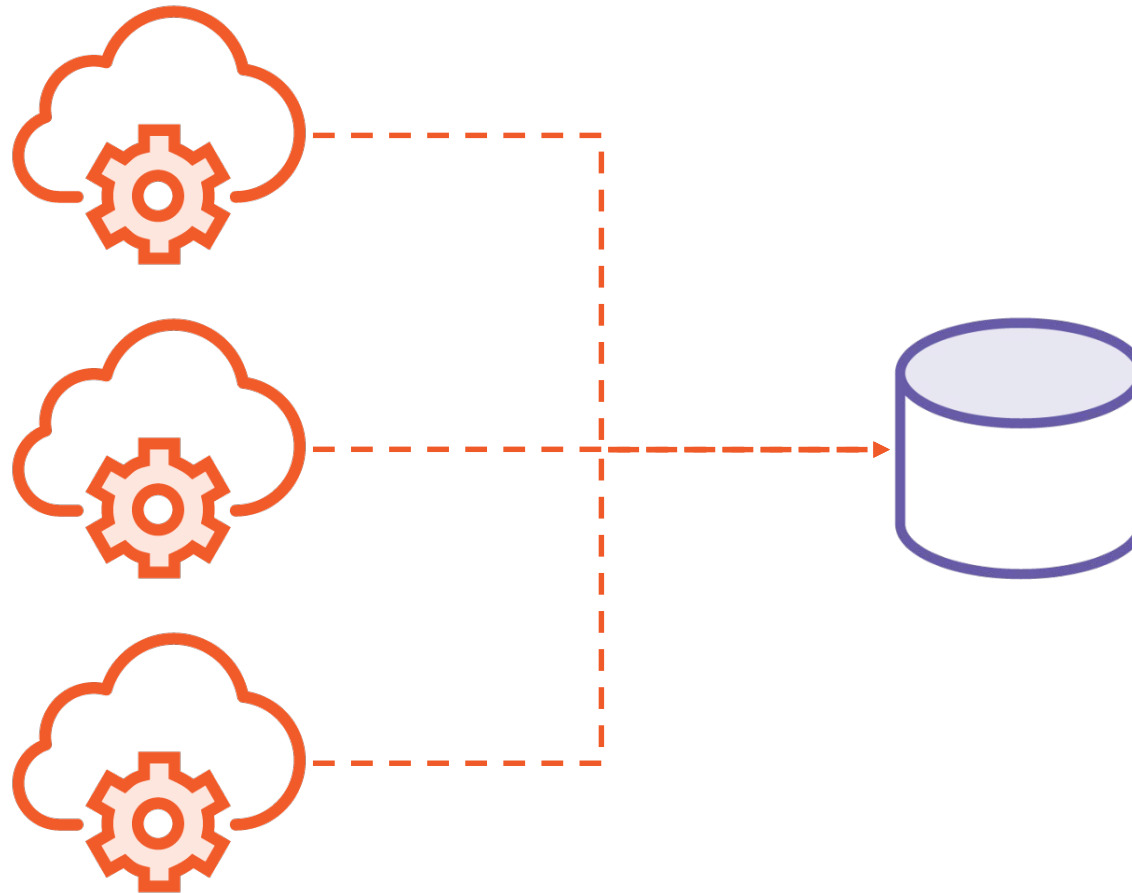
Price (double)

Availability (int)

Approach



Single Point of Failure





Single point of failure

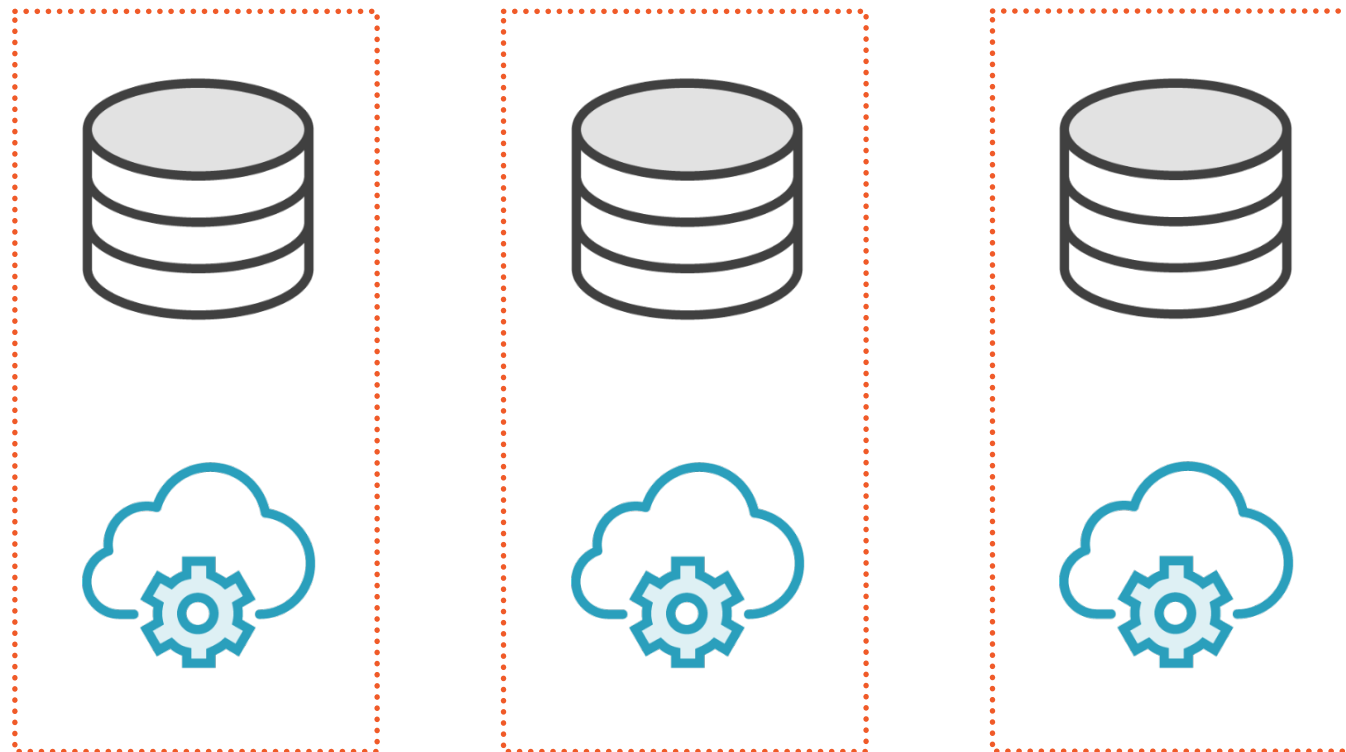
Doesn't scale

No ownership control

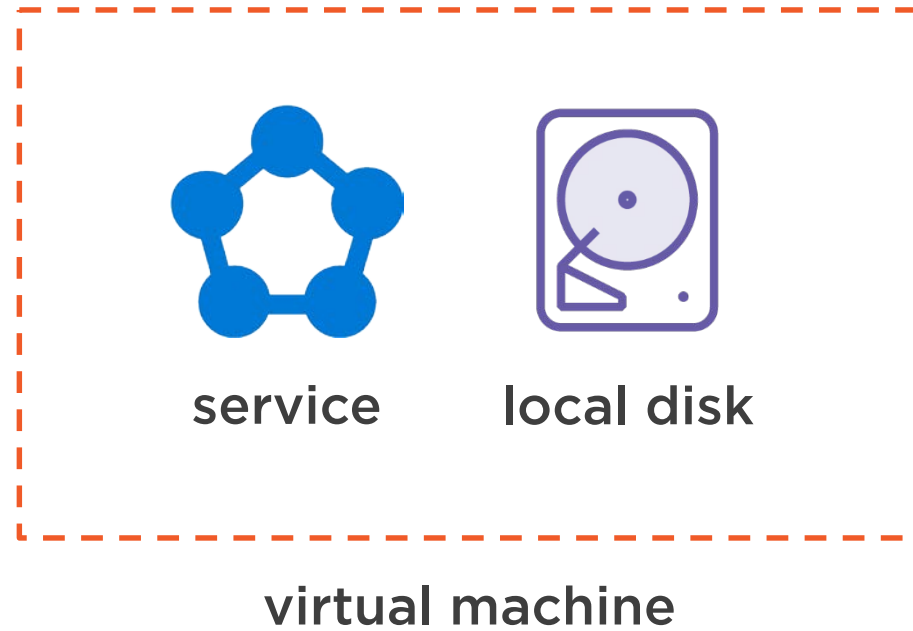
Hard to add/change features

Provider lock-in

Database per Service



Service Fabric State



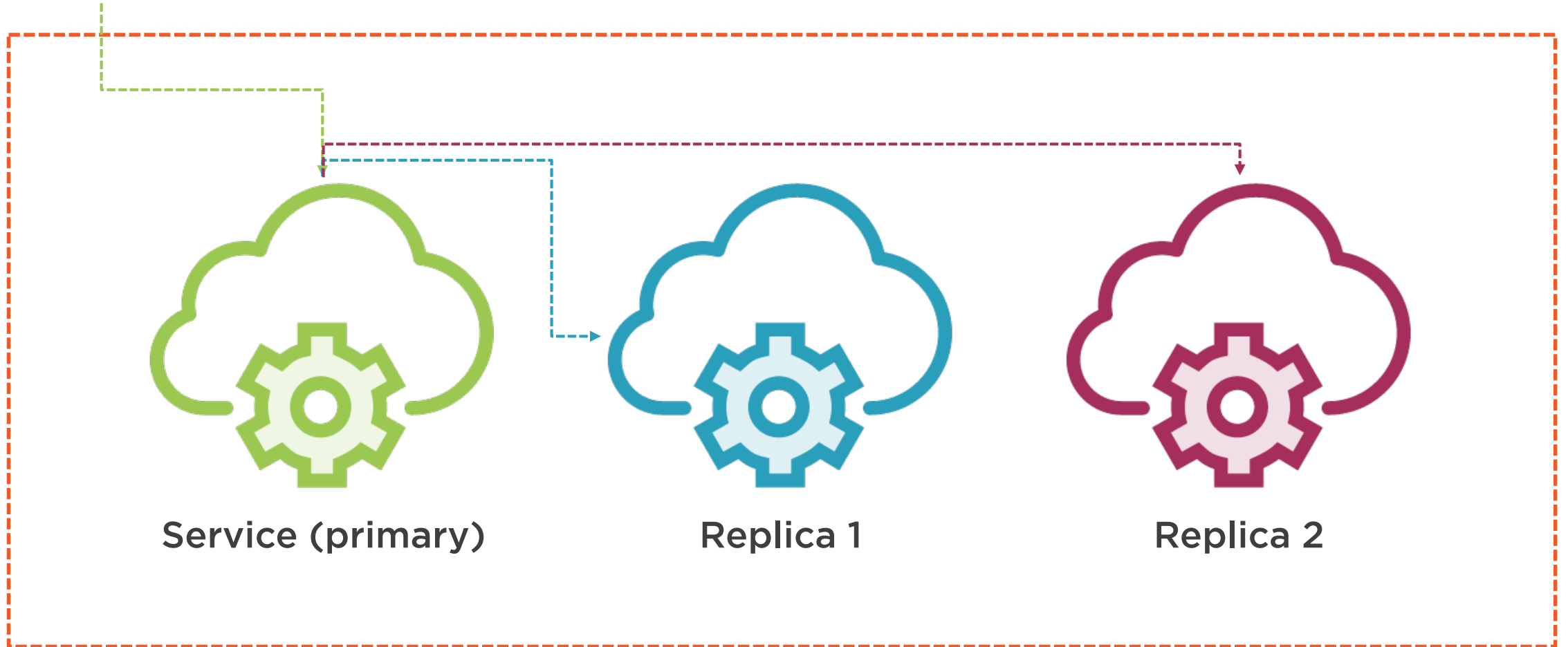
Service Fabric State



Service Fabric State

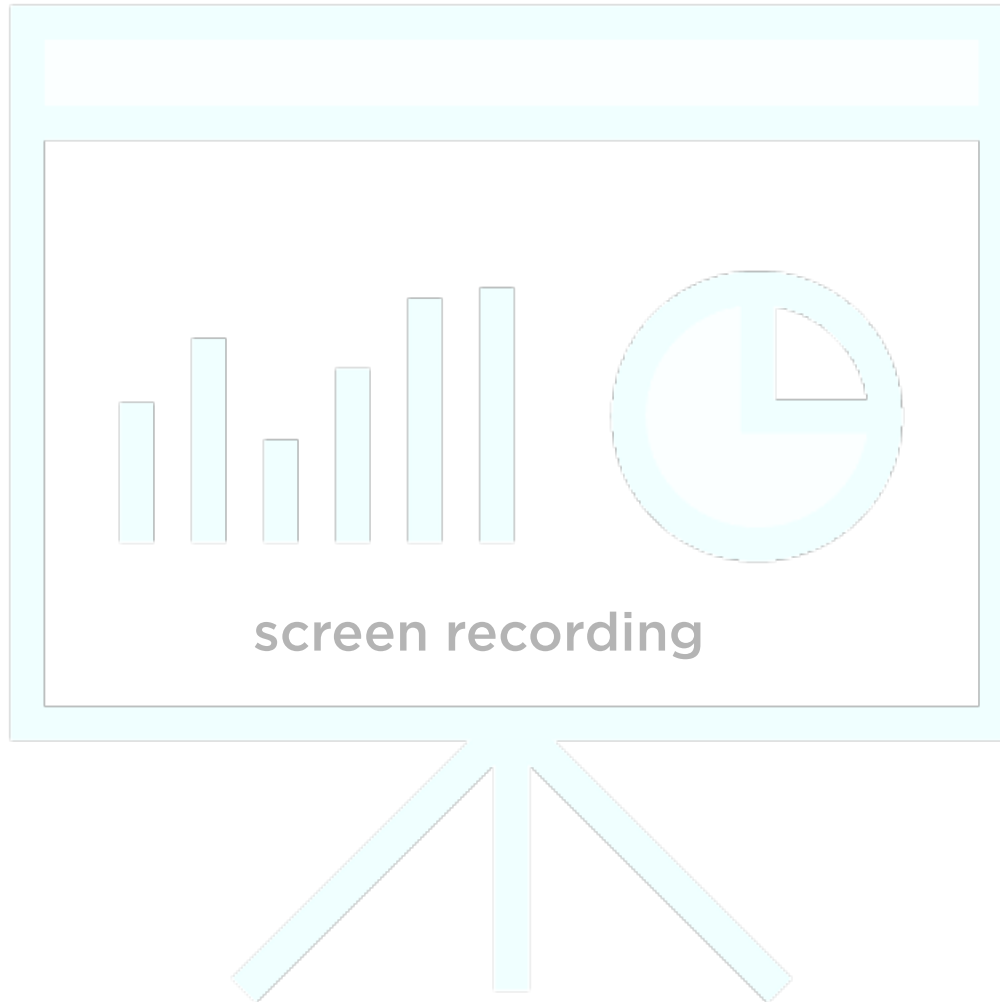


write



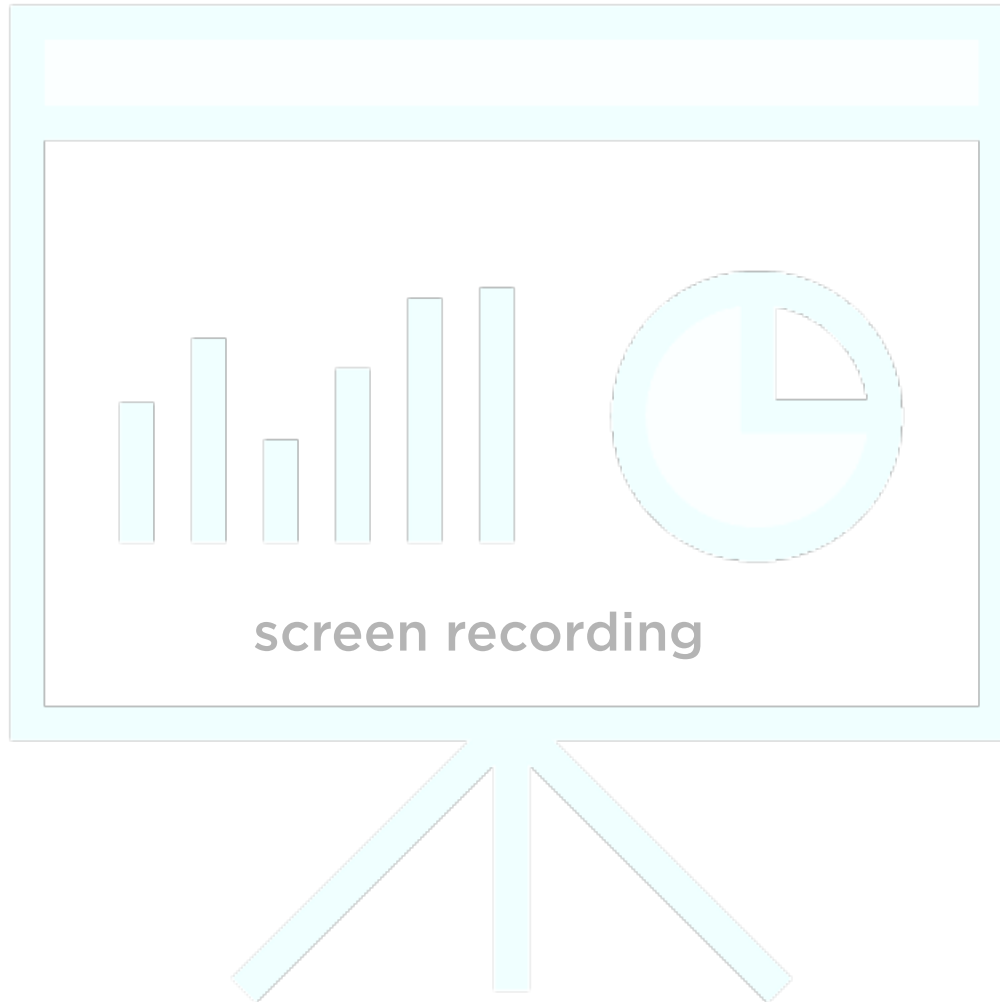
quorum





Creating Web API





Connecting Two Services Together



3 Node Cluster



active



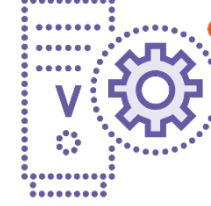
passive



active



active



active



passive



Never assume a service is
running in a fixed location



Choice of Protocols

WCF

HTTP

Service
Remoting



Service Remoting

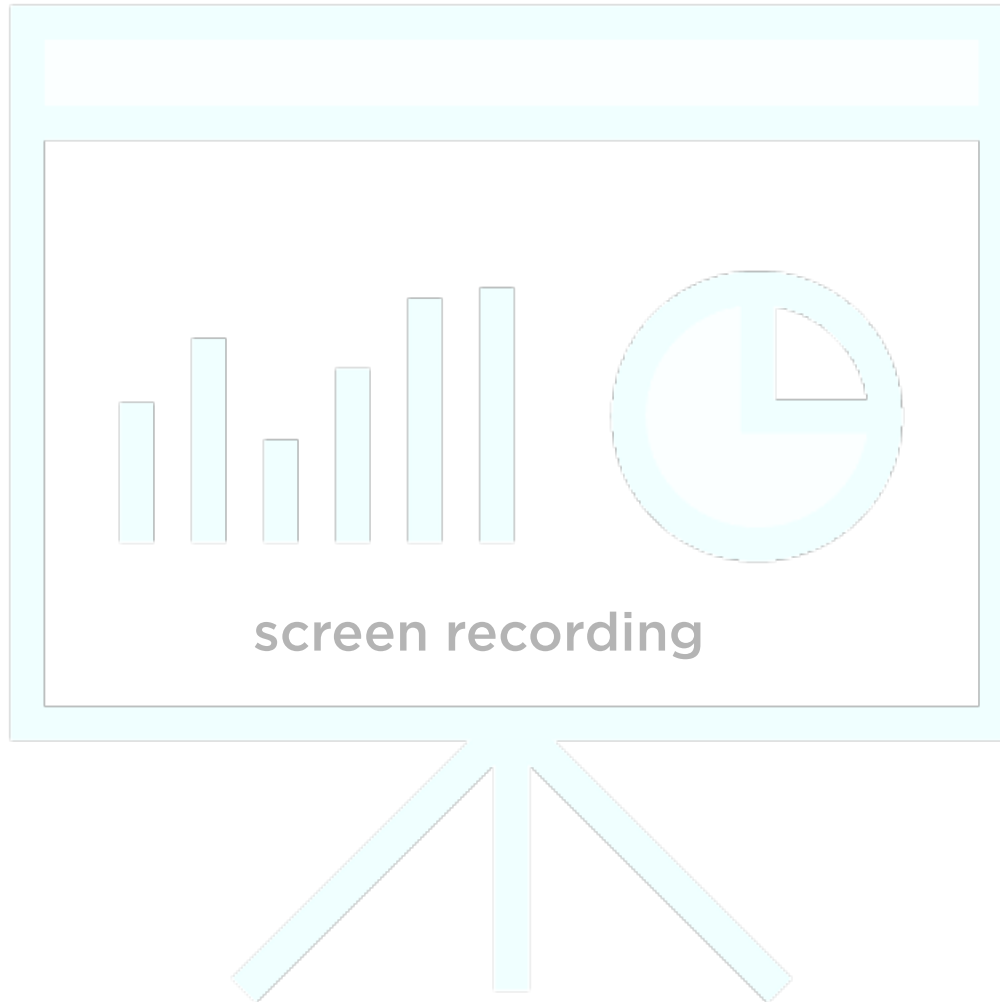
Pros

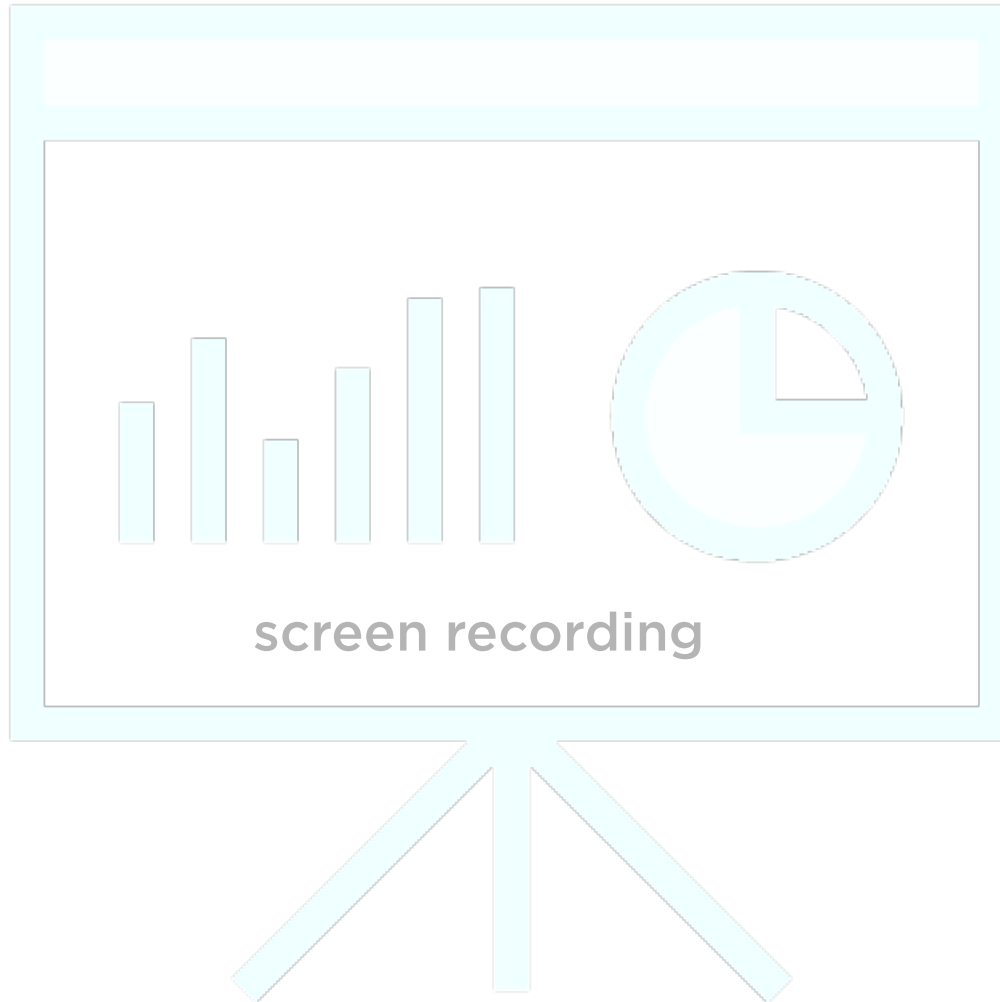
Automatic service address resolution
Establishing connection
Retries
Error handling
Strong typed
Fast
C# and Java compatible

Don't use

Custom binary protocol
Unknown external consumers
Protocol not supported by ASF

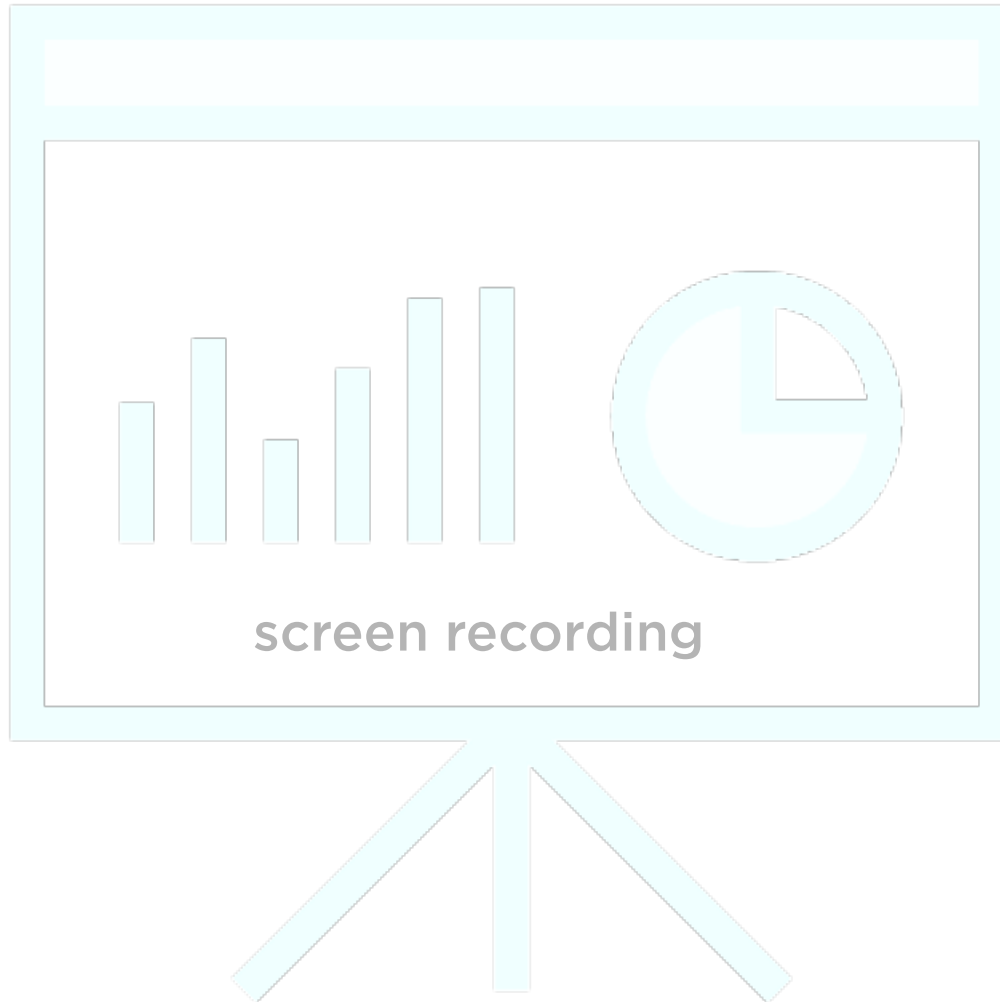






URI Format





Summary



Differences between an app and a
Reliable Service

Stateless vs Stateful services

Application lifecycle

Reliable state

Service Remoting

Connected two microservices

