MESSAGE BASED APPLICATIONS AS FIRST-CLASS CITIZENS IN CLOUD FOUNDRY


Geoff Arnold, CTO, Sensity Systems
garnold@sensity.com
The tl;dr

• Today, Cloud Foundry is optimized for applications which are driven by web (HTTP) requests
  – Inputs are dispatched via a router; scaling is handled by a load balancer
  – Application lifecycle management involves orchestrating these services and injecting endpoint bindings into the application

• Many new applications – IoT, Smart City – are [also] driven by pub/sub message flows
  – Inputs are dispatched by a message broker; scaling involved adding new subscribers
  – Application lifecycle management involves orchestrating broker configuration and subscription - but Cloud Foundry doesn’t know how

• I think we should fix this
Motivation

• Provide the same level of ease-of-development and operational automation for message driven applications as for web services
  – Deployment via application manifests
  – Blue/green deployment

• Facilitate the development of "reactive microservices" application frameworks (a.k.a. “serverless architectures”, e.g. AWS Lambda)
Example: Smart City IoT Application Architecture

Device Command Center

Data Dealer

Interface Service

Real Time Event Feed

Historic Data Query

Asset State Read/Update

Administrative Functions

Business Application, e.g. Parking, Lighting Control

Administrative Application, e.g. Node Provisioning, Add Users, Enable VMS Feed

Customer Identity Mgmt Service

TSDB Service

Asset Service

Identity Service

Third-Party Video Mgmt Service
Message-based and transactional flows

Device Command Center

Data Dealer

Interface Service

Real Time Event Feed

Historic Data Query

Asset State Read/Update

Administrative Functions

Business Application, e.g. Parking, Lighting Control

Administrative Application, e.g. Node Provisioning, Add Users, Enable VMS Feed

Customer Identity Mgmt Service

TSDB Service

Asset Service

Identity Service

Third-Party Video Mgmt Service

Copyright © 2016 Sensity Systems Inc. All Rights Reserved.
Topics to discuss

• One standard message broker (compare gorouter) or a standard proxy for multiple messaging systems?

• Health Manager?

• Broker-based vs. brokerless designs?

• Broker HA?

• Placement?

• Application manifest semantics?

• Mixed applications – some components message-driven, others with web service interfaces?

• Integrating Services (Marketplace and User Provided) into message flows?
THANK YOU