

Hybrid clouds that just work

# Getting Started With Serverless: Key Use Cases & Design Patterns

Jennifer Gill  
Peter Fray  
Vamsi Chemitiganti

Platform9 Systems

Sept 20, 2018

# Agenda

- About Us
- Introduction to Serverless & Fission
- Fission Architecture - Based on Kubernetes
- Demo
- The Five Popular Use Cases and Design Patterns
- Fission Value Proposition
- Q&A

# About Us

# Speakers



## **Vamsi Chemitiganti, Chief Strategist**

- Over 20 years of high tech experience in various customer facing roles
- Various leadership positions at Red Hat, HortonWorks, Siemens Medical Solutions
- Check out his blog on serverless, Kubernetes, and more! <http://www.vamsitalkstech.com/>



## **Peter Fray, Sr. Solutions Architect**

- Over 25 years of high tech experience in various customer facing roles
- Experience with a wide range of technologies including software, storage, networking
- Peter - any certifications to add?



## **Jennifer Gill, Head of Global Product Marketing**

- Over 20 years of technical marketing experience
- Leadership positions at Platform9, HyperGrid, Zerto, EMC and VCE
- Recognized for excellence in marketing as part of VCE launch

# Platform9 Overview

- Founded in 2013 by a team of VMware engineers with a goal to make it easy to transform any infrastructure into a cloud
- Recognized for innovative technology



- \$36.5 million has been raised to date

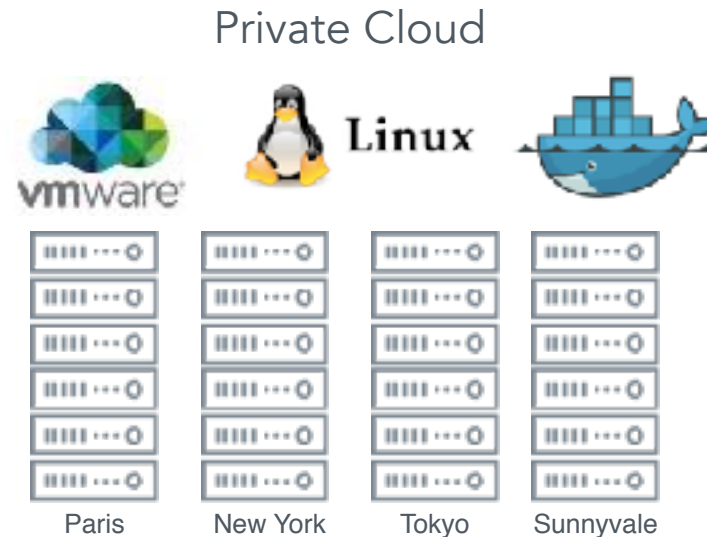


# Our Approach To Cloud Management: SOUL

1. Industry's only **SaaS** managed offering
2. **O**pen-source innovation & economics
3. **U**nified cloud experience
4. **L**everages existing IT infrastructure



Unified cloud experience with any infrastructure, for any application stack



# Introduction to Serverless & Fission

# ▶ Fission: A New Way To Build Modern Stateless Apps

Code, Deploy And Operate Instant 'Production-Ready' Apps.

Fission is a full featured **Serverless** framework for running serverless applications on Kubernetes.

Fission lets you easily code, deploy and operate applications that are production-ready from the get-go - without having to learn anything about Kubernetes.





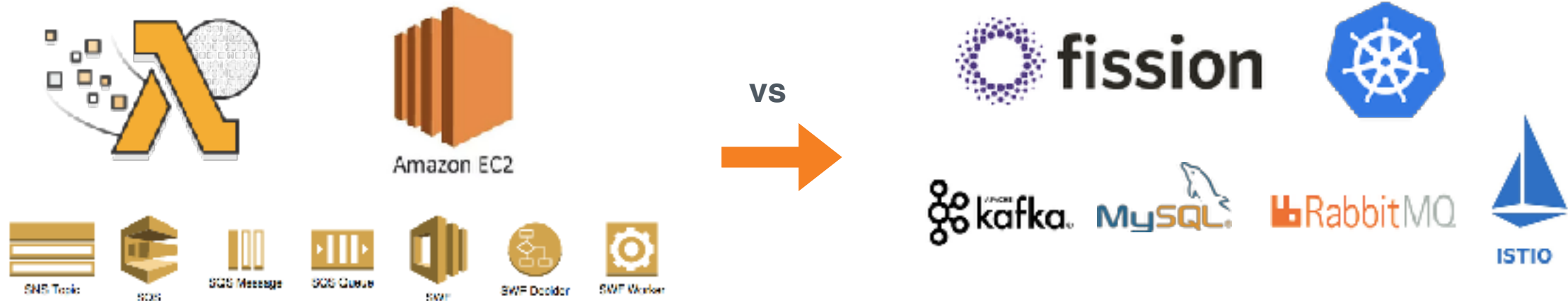
# Fission: Platform9's Serverless Framework

- Use Kubernetes and containers without a steep learning curve
- Write short-lived functions in any language
- Map functions to triggers (e.g., message queues, timers, HTTP requests)



- ✓ No containers to build
- ✓ No Docker registries
- ✓ Works on any Kubernetes cluster, anywhere
- ✓ Extensible to any language

# Fission: An open source alternative to AWS Lambda



- AWS started the serverless movement with Lambda, but it **locks Developers** into AWS ecosystem
  - Pricing lock-in
  - Not multi-cloud or multi-tool friendly. Developers want choice. (eg how to I consume my Kafka stream events in Lambda instead of using Kinesis?)
- Fission **opens up** these advantages to everyone on **any infrastructure**. Its open and extensible and thus enables a wider ecosystem.

# Fission

Architecture

# Fission

Developer

→ Sprint 132

→ Sprint 172

→ Sprint 119.2



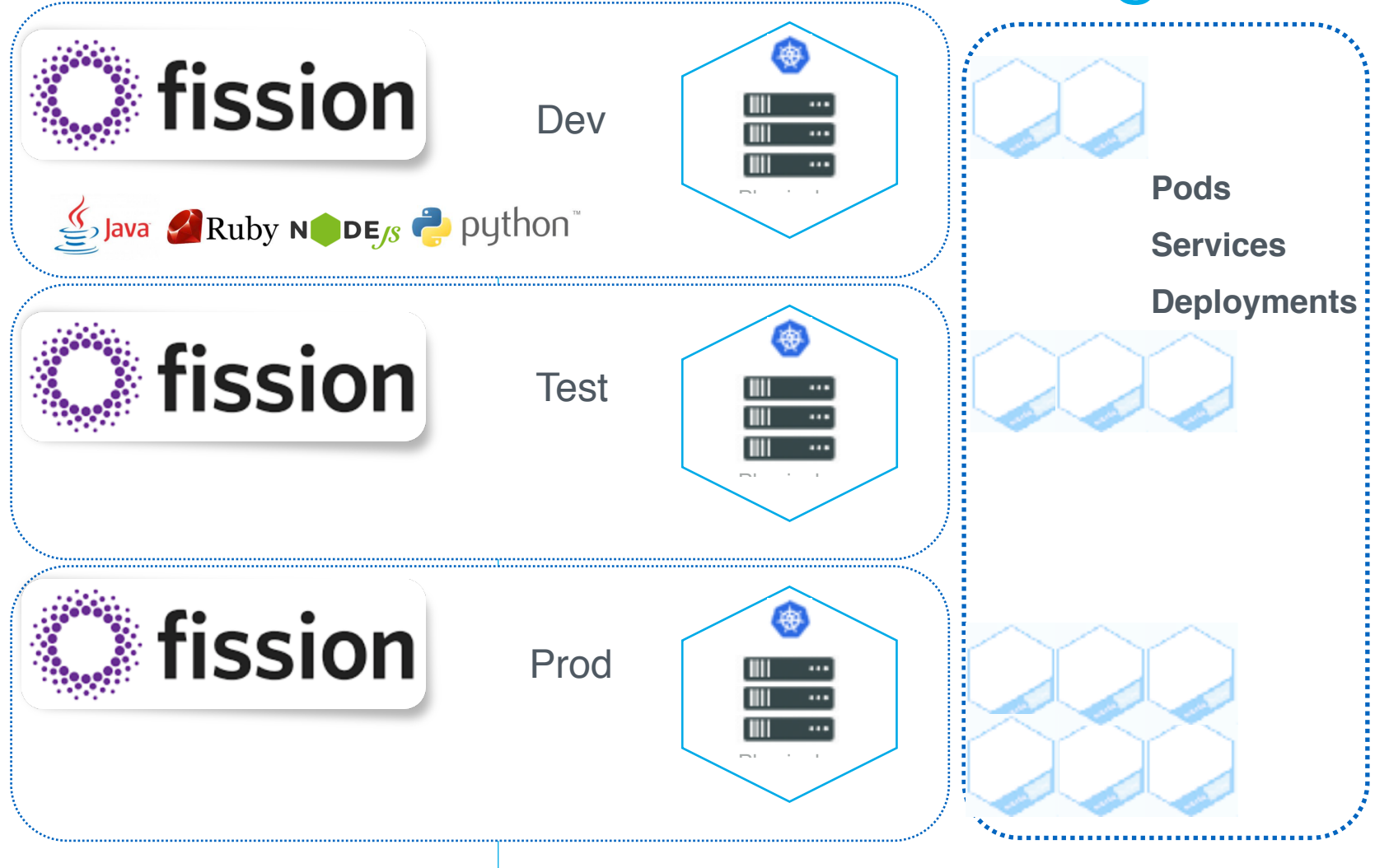
QA

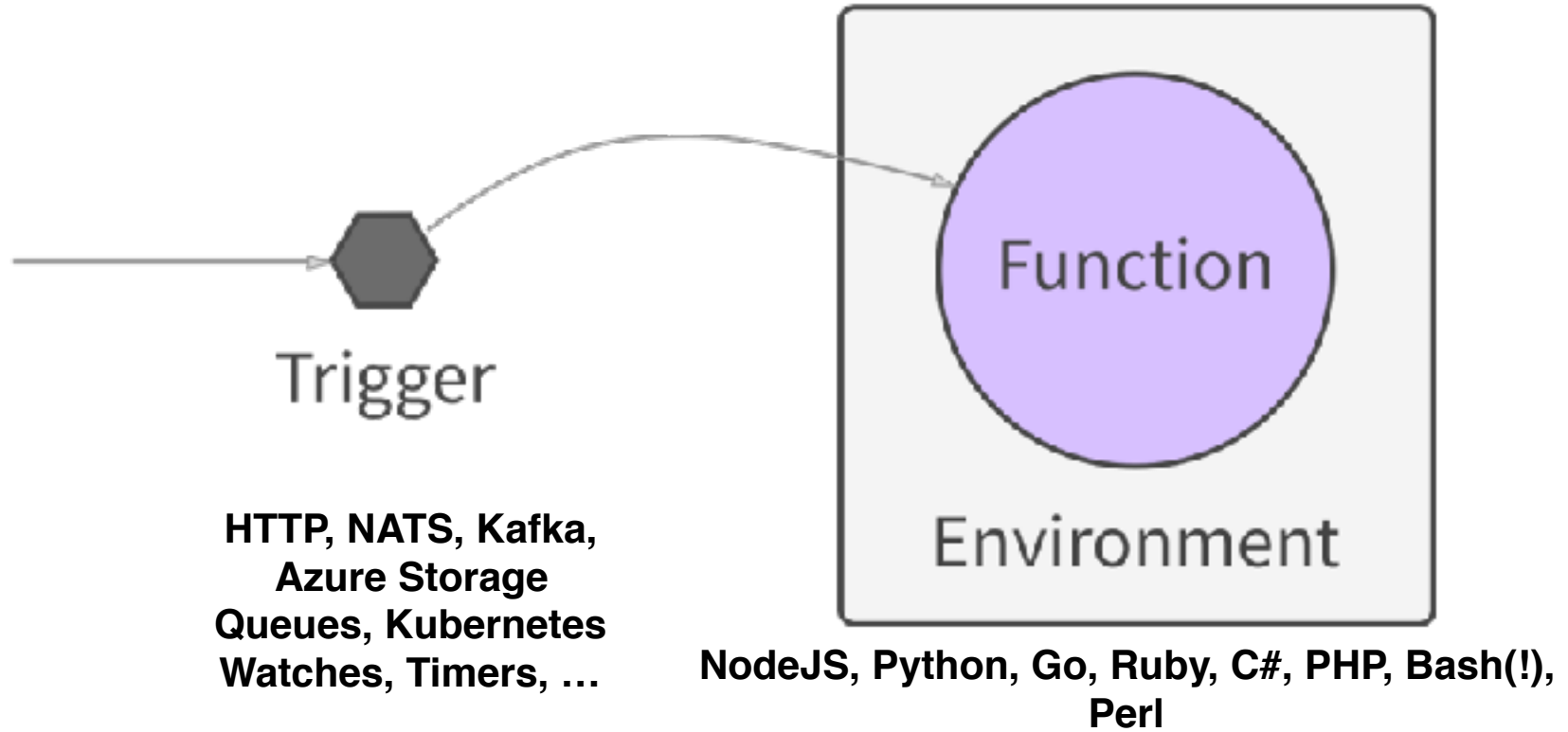
Functional Test  
Performance Testing



Operations

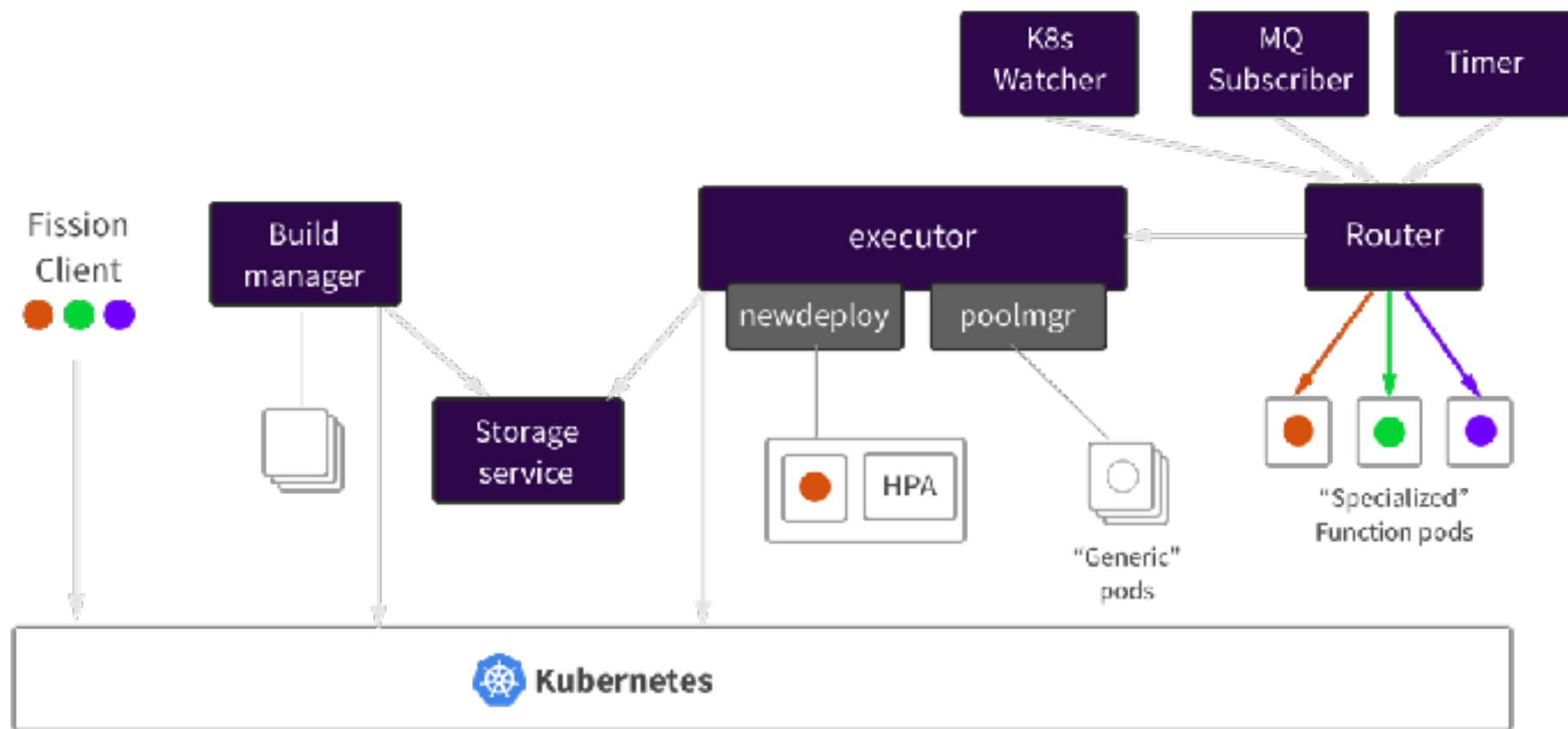
Deployments  
Upgrades  
Hot fixes



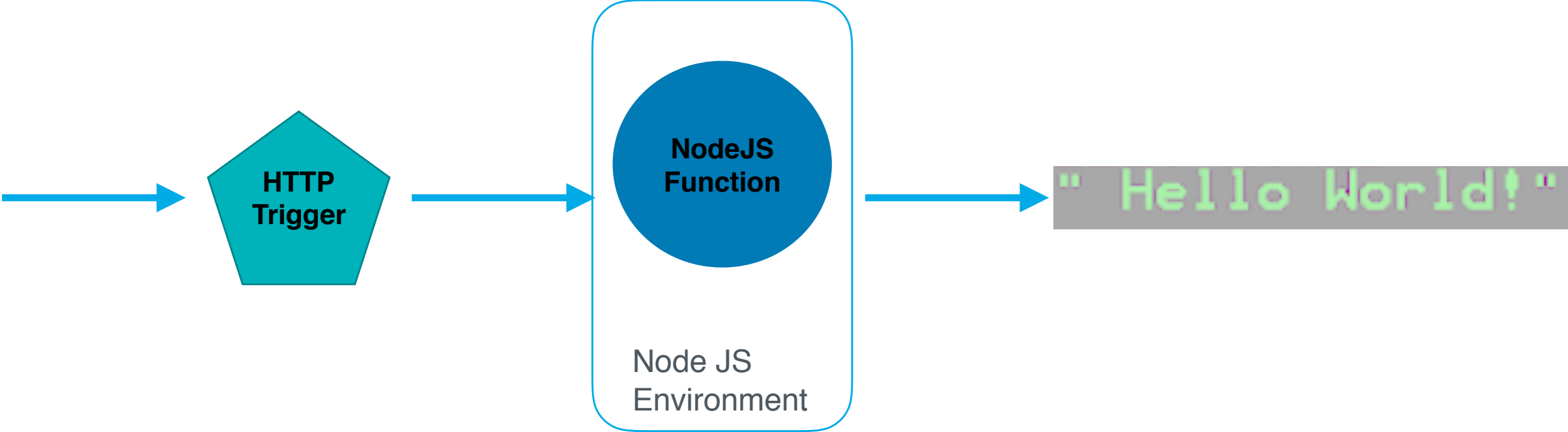


# Function Executors

- Pool-based executor: low-latency, small idle overhead
  - “Pre-warmed” pool of environment containers
  - Functions loaded into containers on-demand
- New-deployment executor: high latency, no idle overhead

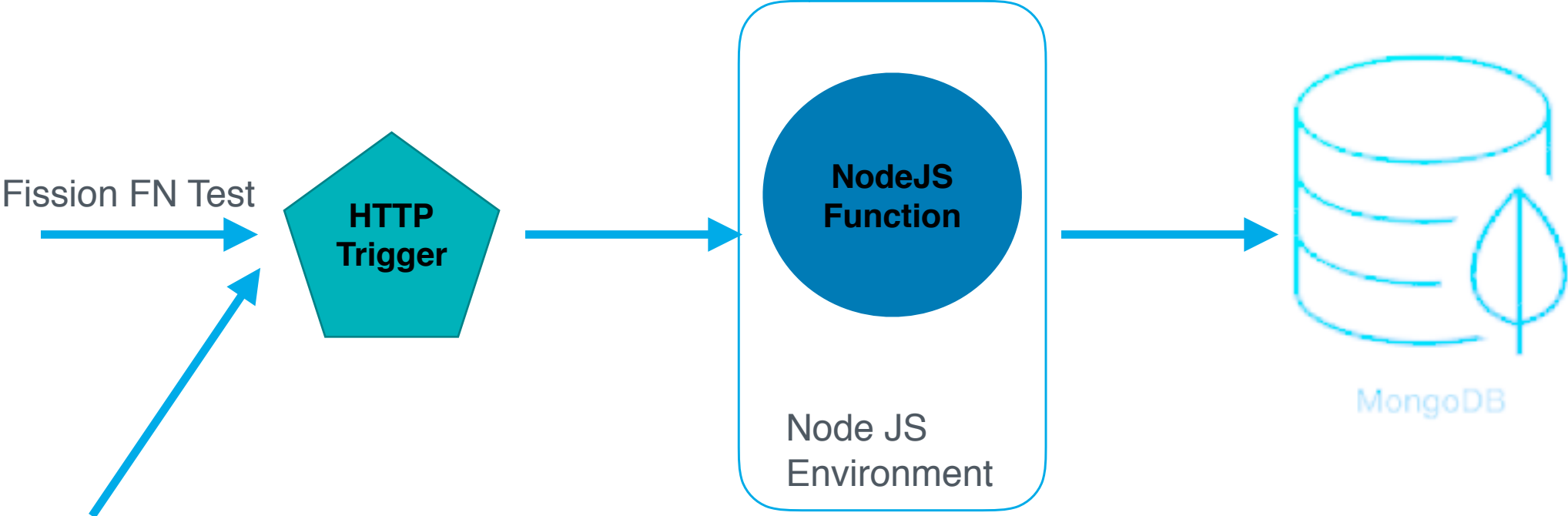


# Fission: DEMO Hello World





# Fission: DEMO Mongo DB Insert



Load Generated 10,000 Executions

# DEMO

# Five Key Serverless Design Patterns

# Use Case Pattern One - Edge Data Processing

## IoT Architecture using Fission

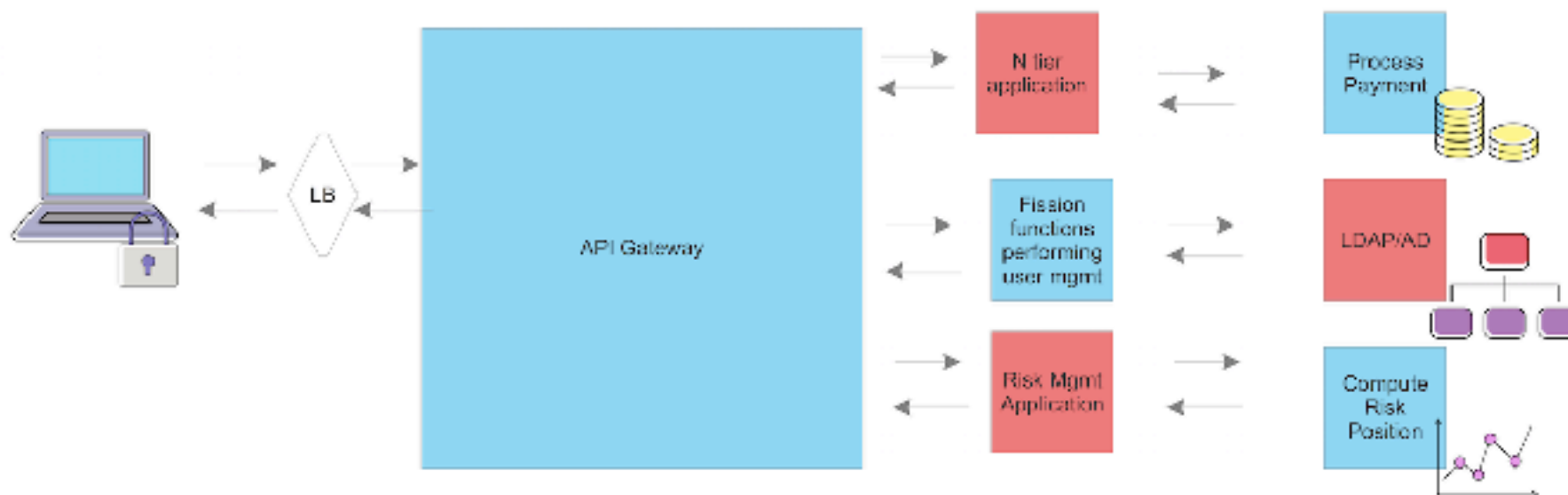
IoT devices send data streams to a Gateway which can combine the data and send them into an API Gateway or Message Queue (leveraging Kafka). Functions perform data transformation, normalization & work in conjunction with traditional microservices to update enterprise applications, BI tools and other services.



# Use Case Pattern Two - Vertical Specific

## Financial Services Use Cases using Fission

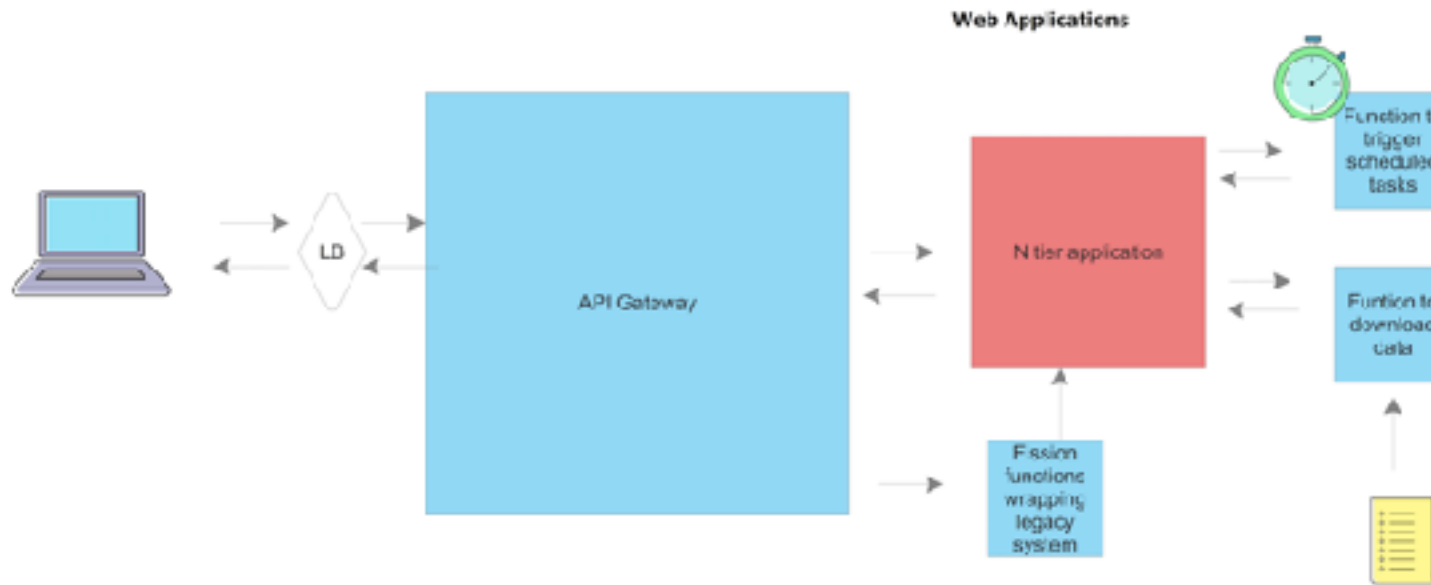
Fission can perform a variety of tasks to help customers in specific industry verticals such as Banking and Retail. A key example is to perform targeted, stateless functions in financial services platforms. These can include payments processing, computing Risk positions, running Compliance checks and for performing User Management.



# Use Case Pattern Three - Augment Web Applications

## Augment Web Applications using Fission

Fission can act as a helper in augmenting traditional web applications by performing a range of tasks. These can include data processing usecases such as transformation/normalization, running scheduled tasks on time, sending alerts etc.



# Use Case Pattern Four - Front Legacy Applications

## Front Legacy Systems using Fission

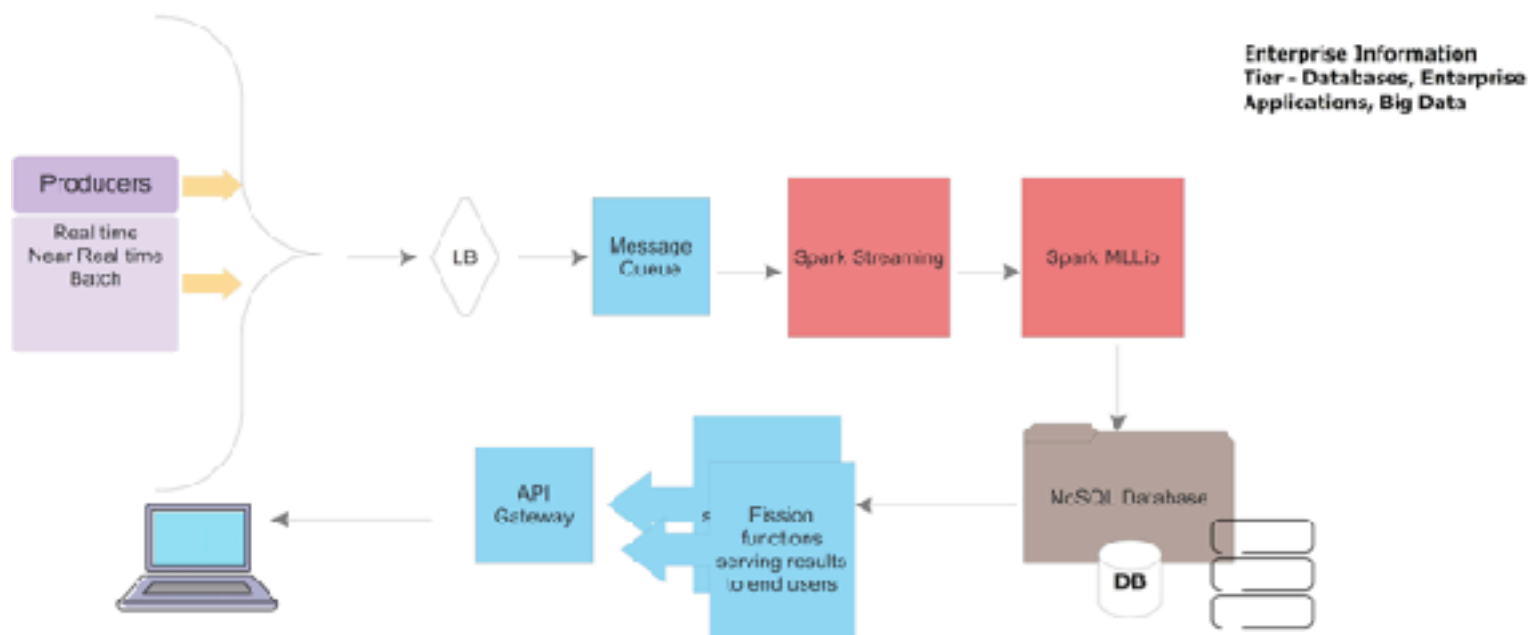
Fission acts as a proxy in fronting legacy applications with a modern RESTful API. Functions are created which can intake data in any format, convert them to a target legacy system and to process the response. Functions can also provide translation to messages sent back to consuming applications via the API Gateway.



# Use Case Pattern Five - Data Processing

## Big Data Processing using Fission & Apache Spark

Data producers send data in both Real time and Batch mode into a Hadoop cluster running Apache Spark. As the Data is ingested, Spark Streaming pulls data into microbatches from a Kafka queue. Data Science models hosted in MLLib are updated in real time and the results are persisted into a database. A cluster Speed Layer in the architecture and they read from the database and communicate with end users.





# Fission Value Proposition

# Fission Unique Value Proposition

- **Simple on-ramp** for both devs and ops: up and running in your first hour
- **Fast:** cold start optimizations, autoscaling
- **Out of the box** support for **10+** languages and **5+** triggers
- **Batteries included:** builders, "alt-tab" dev/test cycle, metrics, tracing, canaries (and lots more)
- **Composable:** Workflow engine for composing functions

# QUESTIONS