

Enterprise Swift Services: A Recipe for Deployment

Tripta Gupta

@triptagupta



Colin Campbell

@Colin_Campbell



iOS architecture, where MVC stands for
Massive View Controller

Reply

Retweet

Favorite

More

205

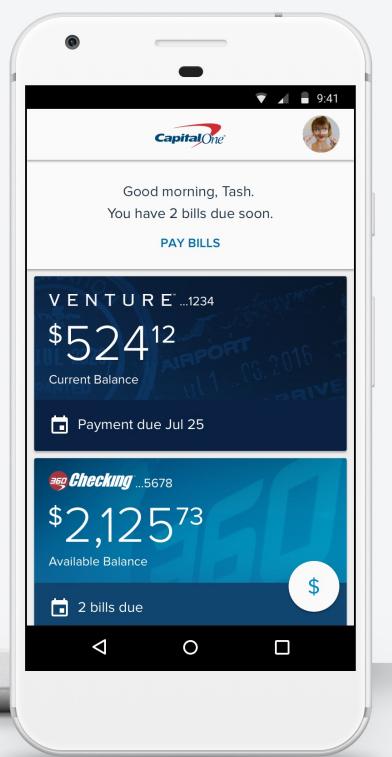
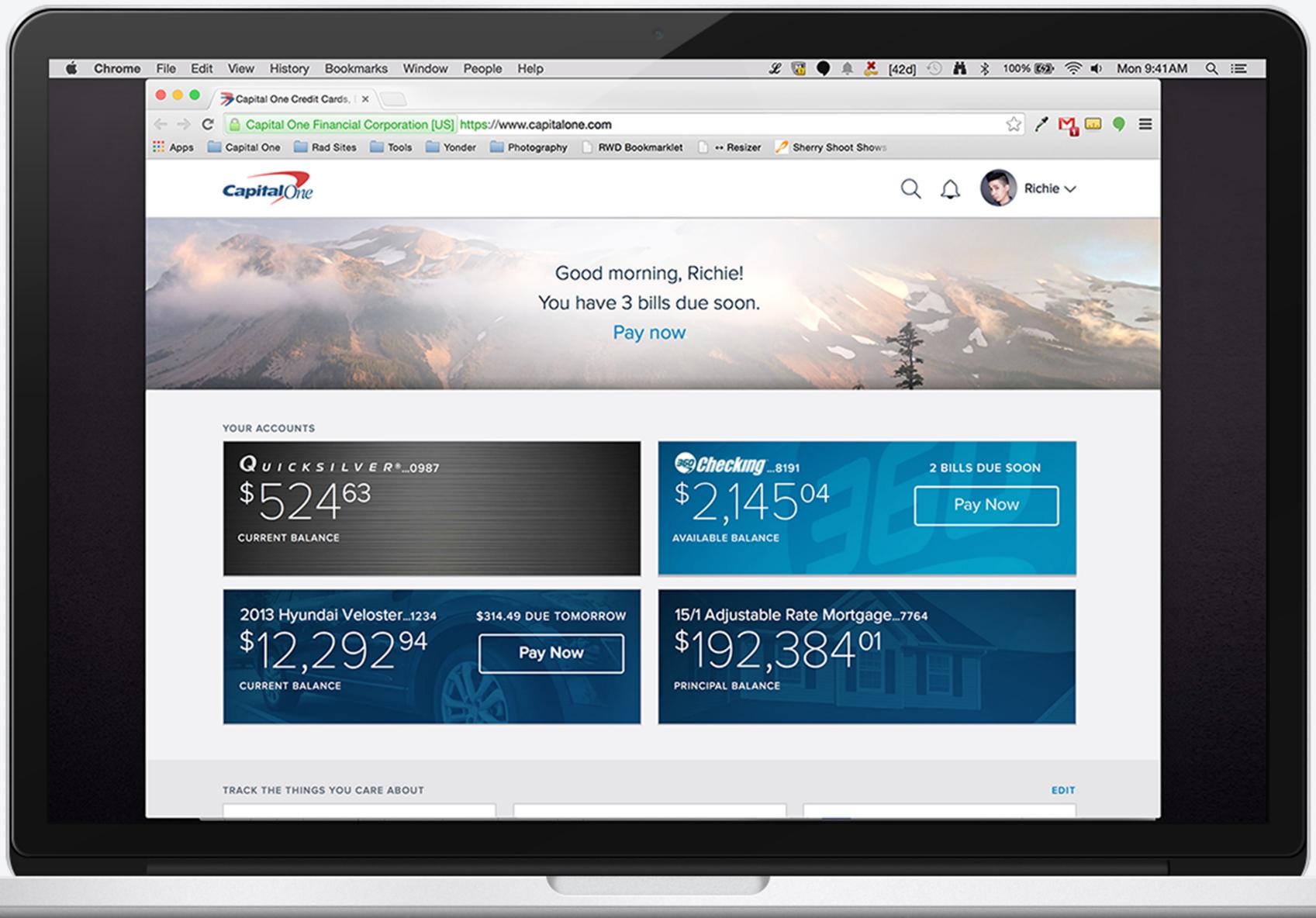
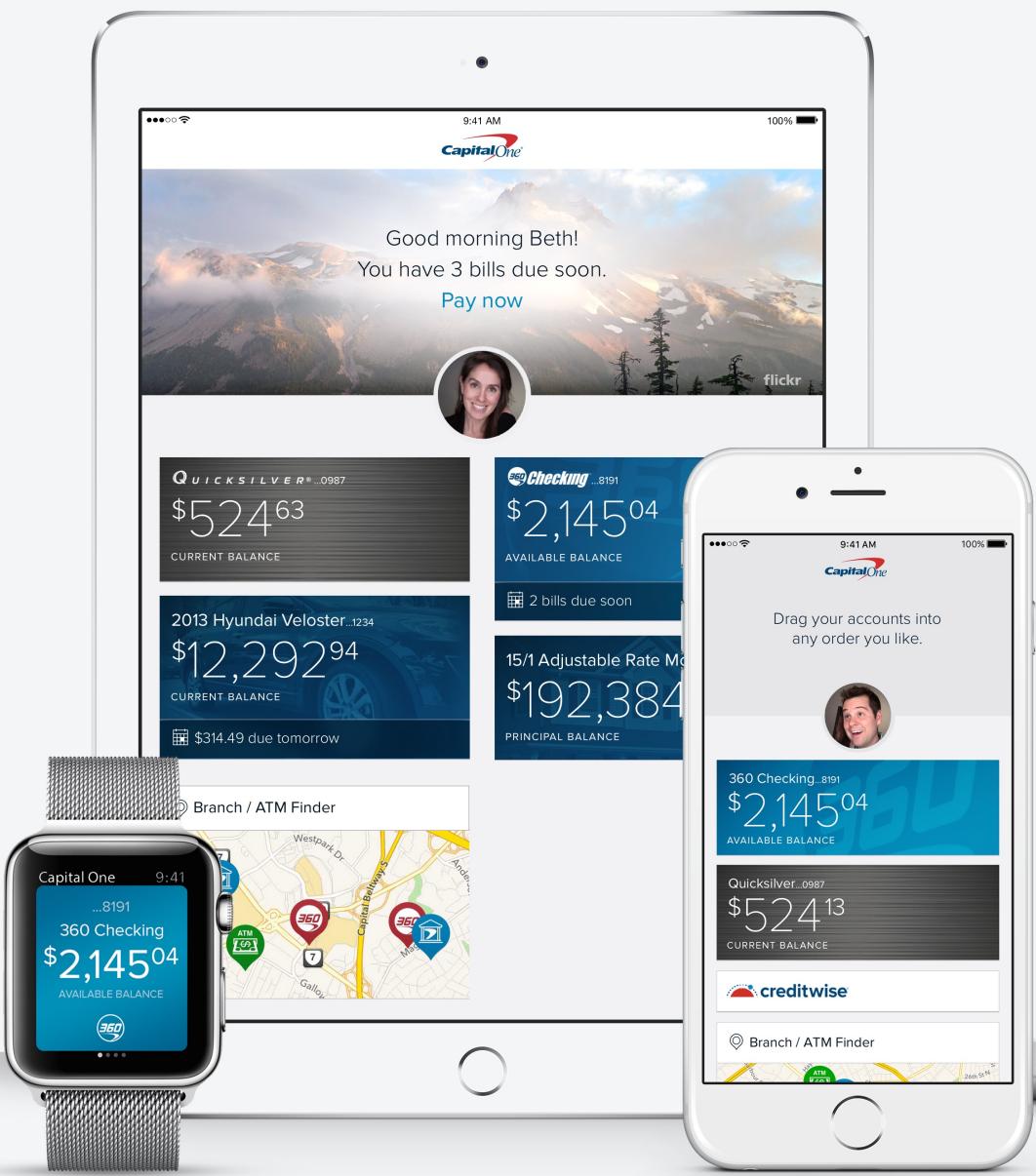
RETWEETS

80

FAVORITES



Server Side Swift Services





Capital One Mobile [4+]

Capital One >

★★★★★ (148)

+ OPEN

Offers Apple Watch App

Details

Reviews

Related

Ratings and Reviews

Current Version

All Versions

★★★★★ 148 Ratings



Great For Banking On The Go

Thurs

★★★★★ by Shinobree

I can handle ALL of my banking needs from the app. Transferring money is a breeze. Paying multiple bills is simple. Being able to check my credit score (for free) is greatly appreciated.

The home screen show you the balances of all of your acc... [More](#)

Amazing app

Sat

★★★★★ by Lock Card Girl

Love this app, makes keep track of my accounts very easy. I absolutely love the Lock Card feature option, I am a very cautious person so it's normally on lock when I am not using my card. Not sure but with this new update seems that the Lock Card feature was removed and I am heartbroken by it. Can you please b... [More](#)

Capital One is the best!

Thurs

★★★★★ by james238

When I need to get things done and done right and on time, My capital one card is what I turn to... Time and time again they've

Edge Orchestration

(gateway between backend & clients)

Edge

1. Extend functionality of existing APIs
at Capital One

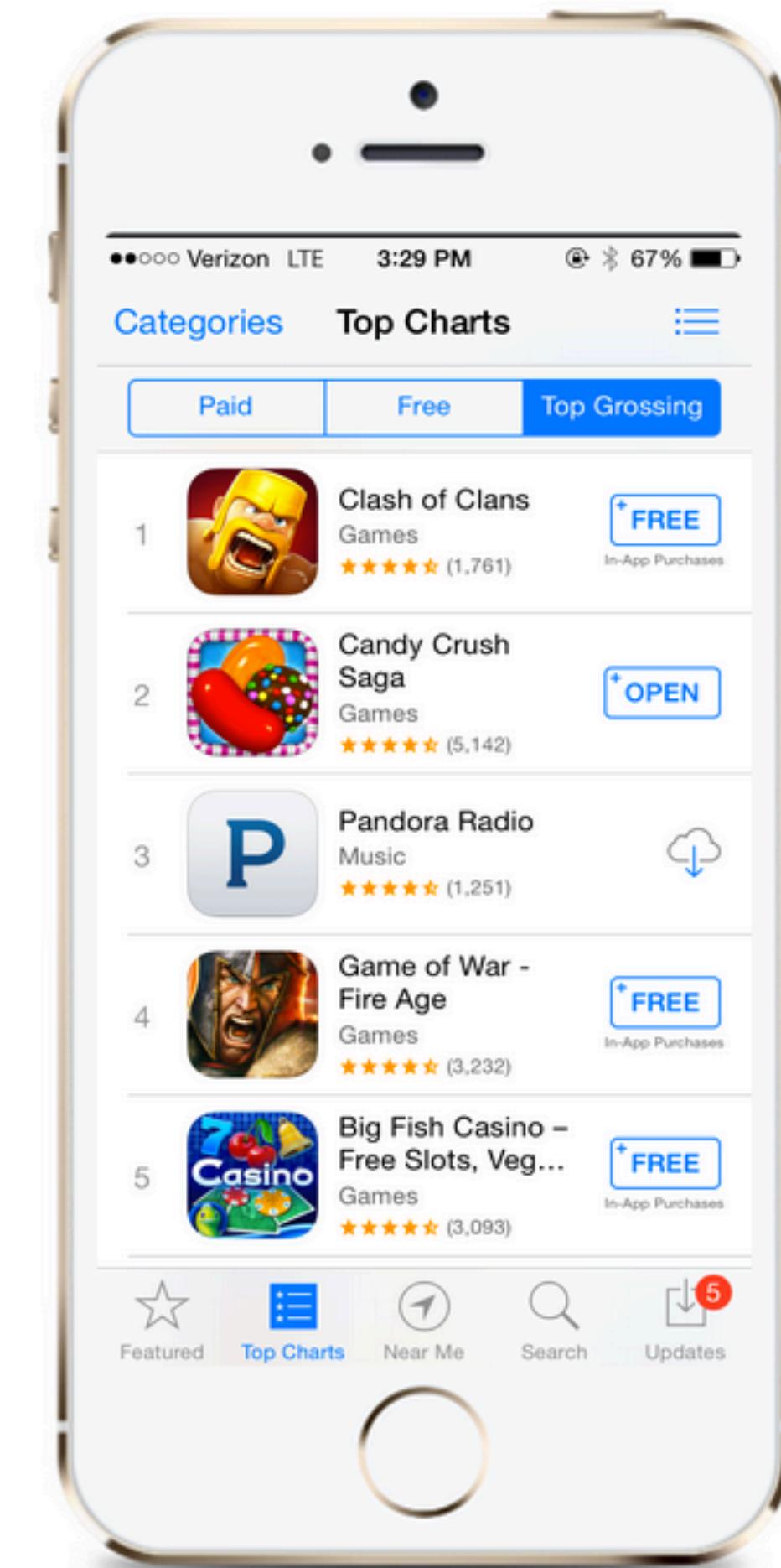
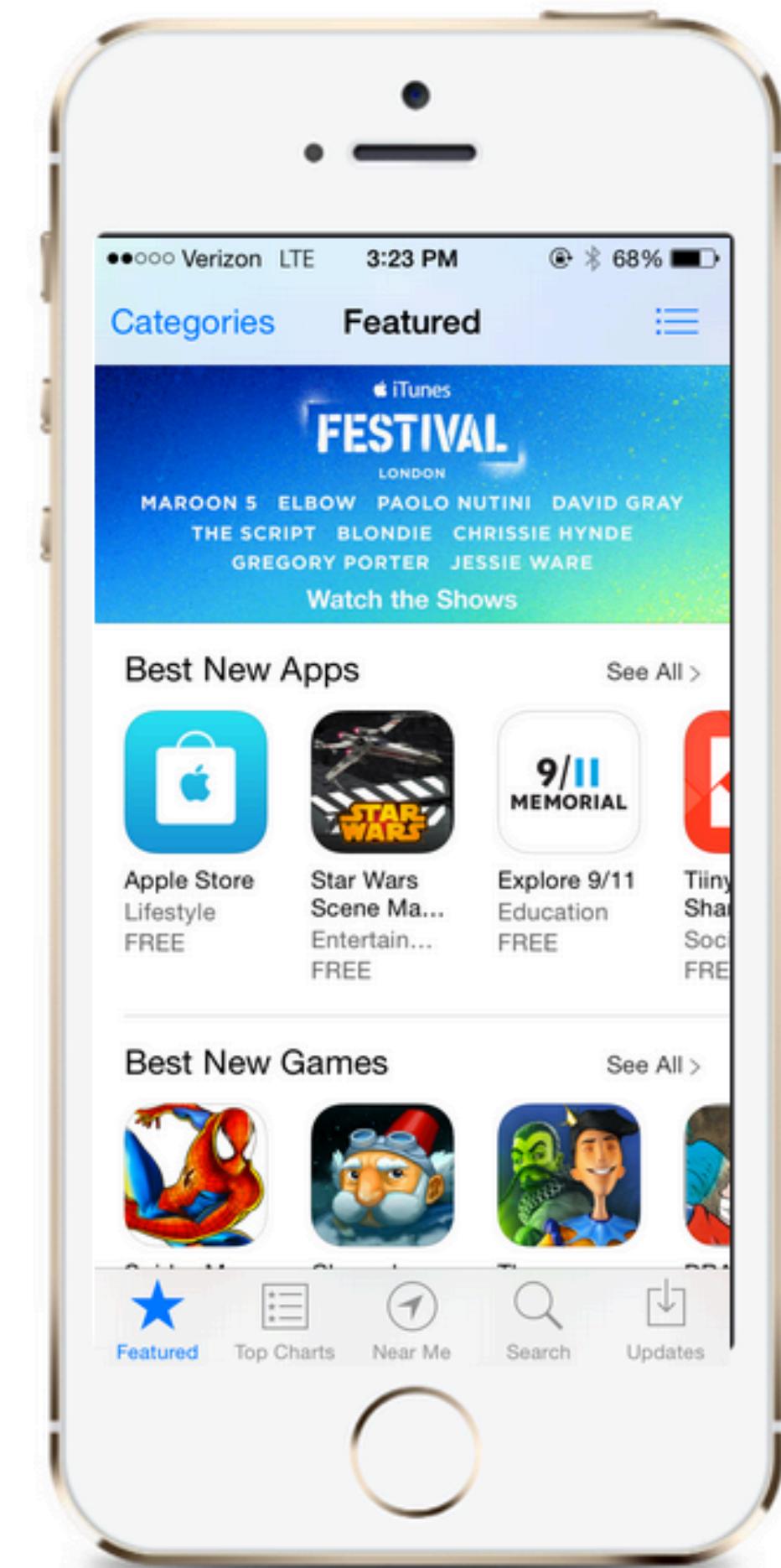
Edge

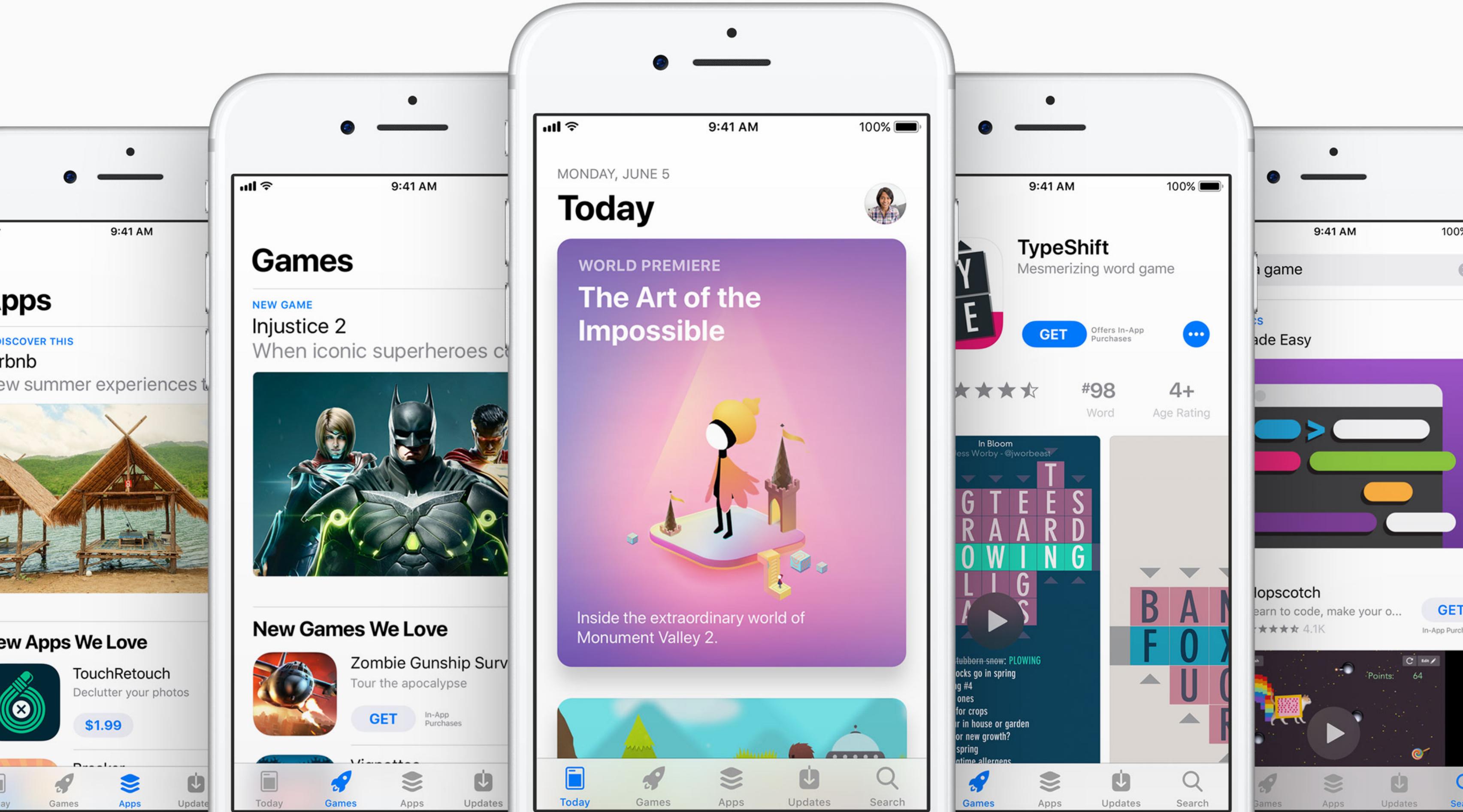
2. Support engineering teams in
feature delivery

Edge



3. Deliver new & user-specific micro experiences

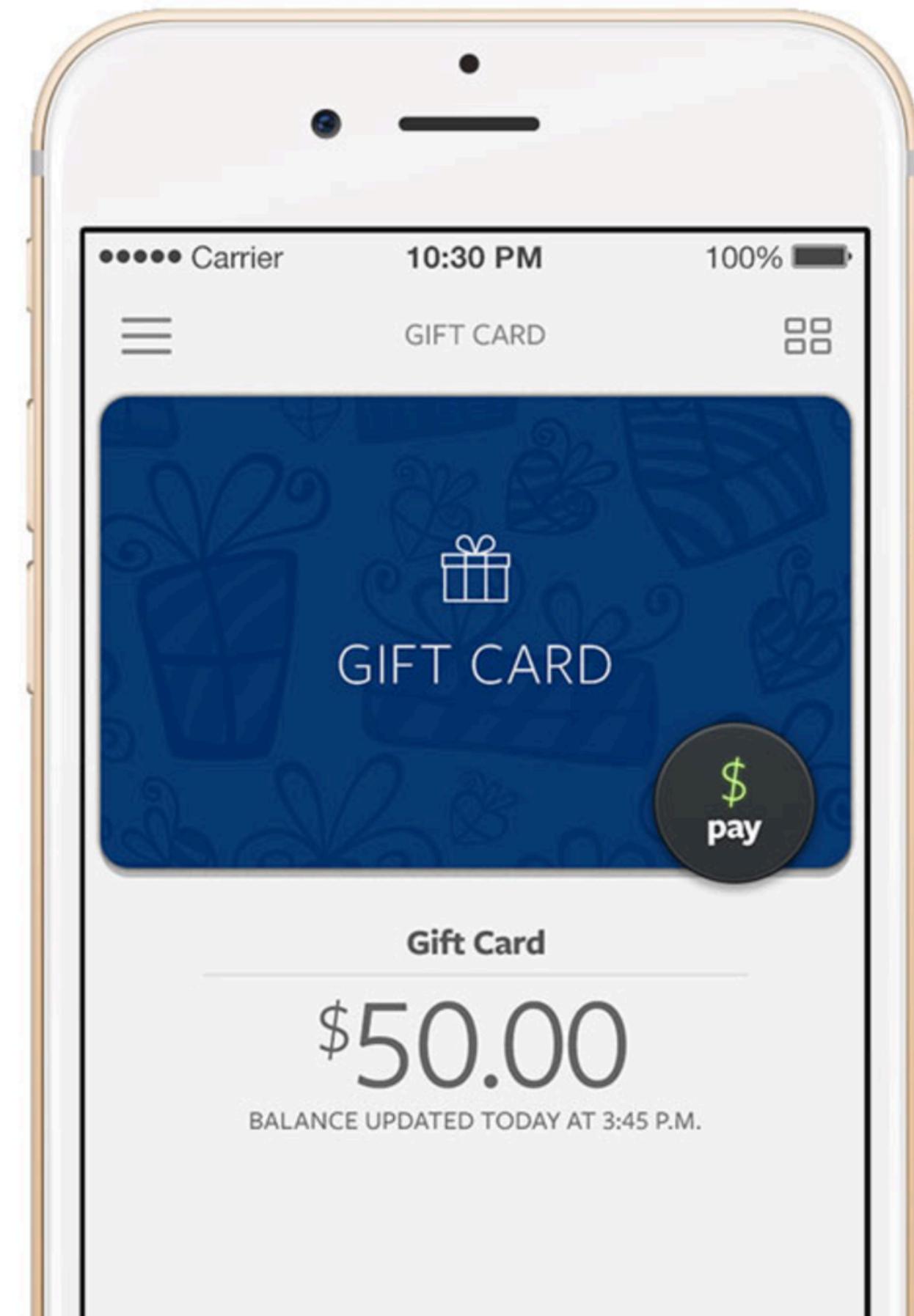




Micro experiences (xAPIs)

Redeem Rewards xAPI

1. **GET** rewards balance
2. **POST** redemption
3. **GET** eligible transactions





Feedback Loop

Mission

Develop a standardized process that allows client engineers to deploy production-ready Swift services into an enterprise environment.

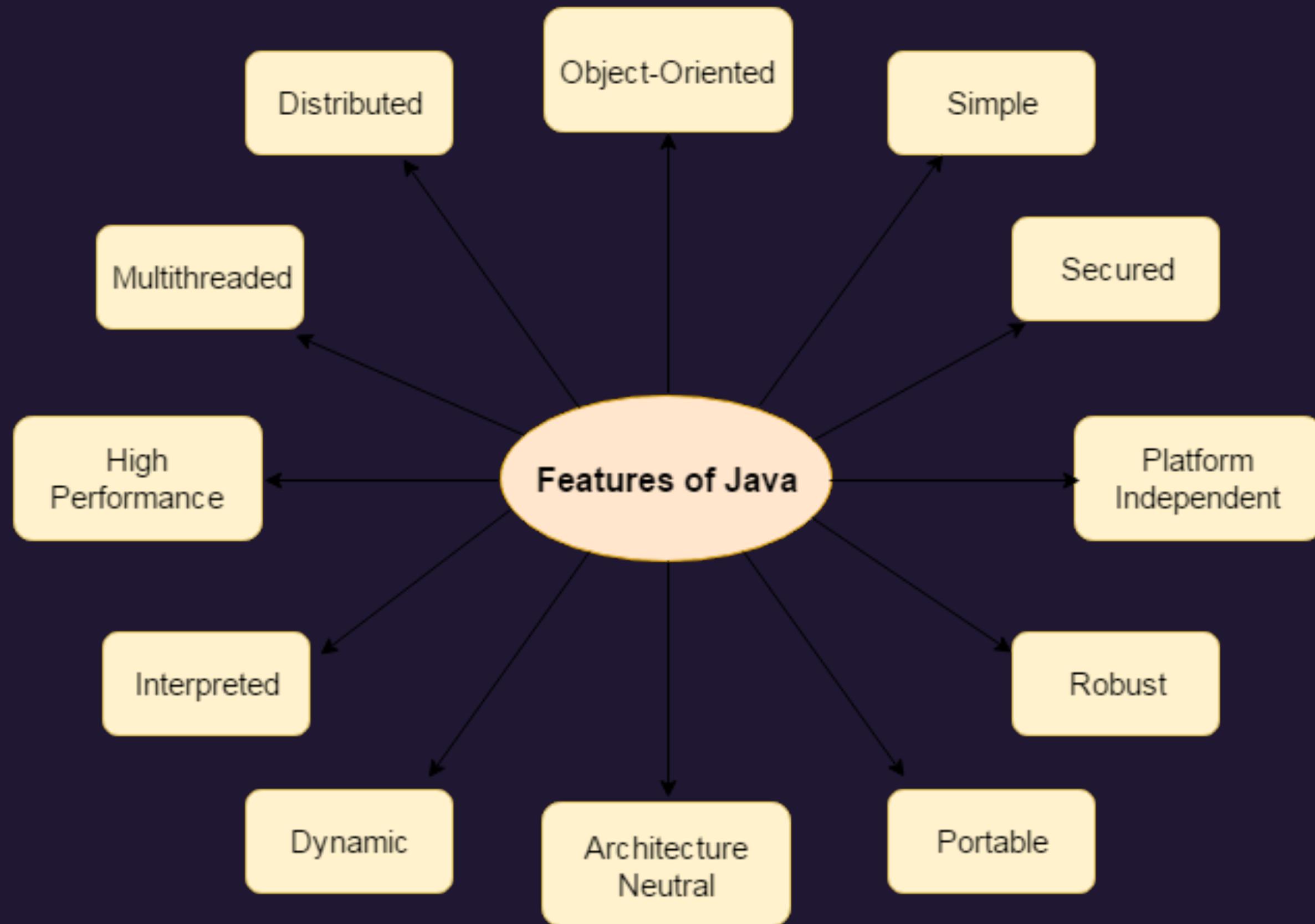


Why Swift?



Raspberry Pi
AWS Lambda





Java vs. Swift

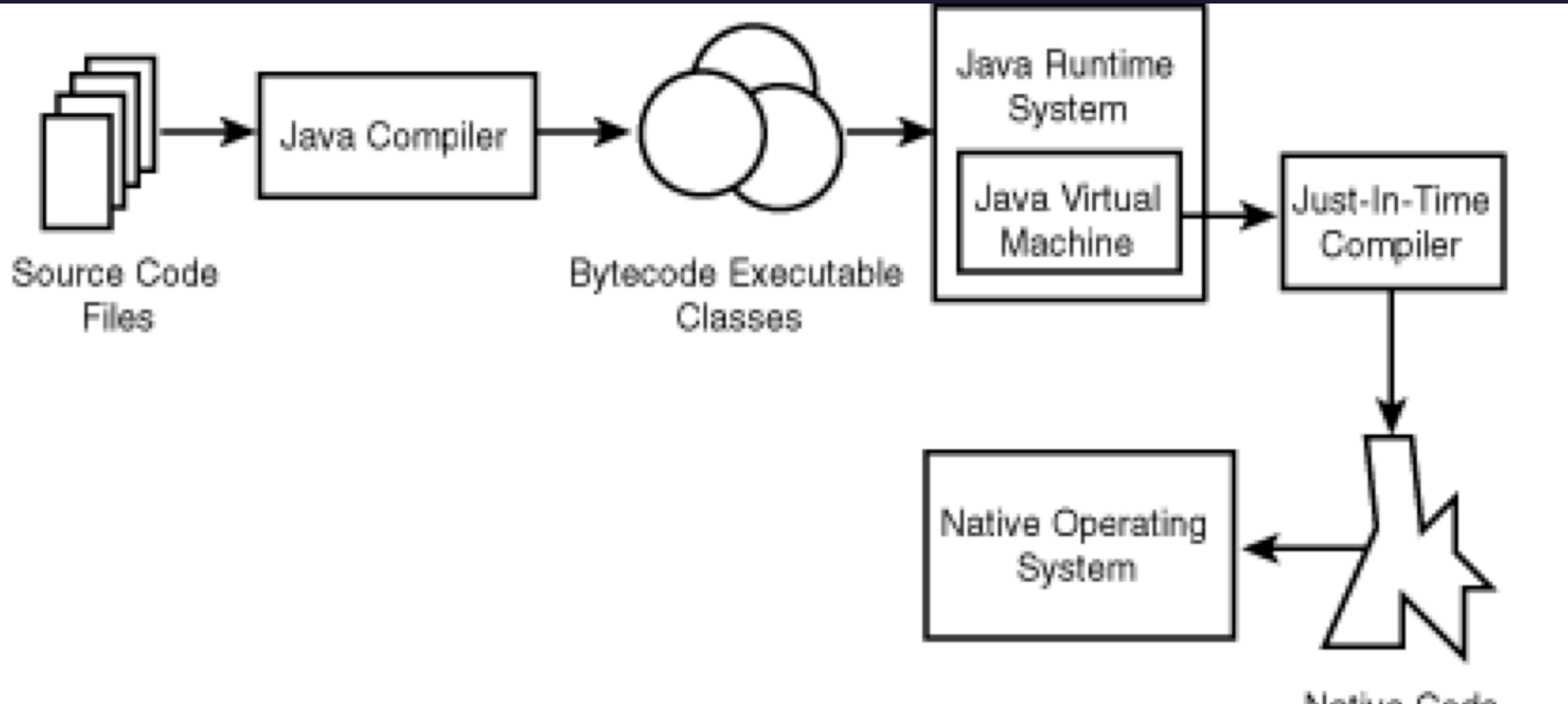
Java™

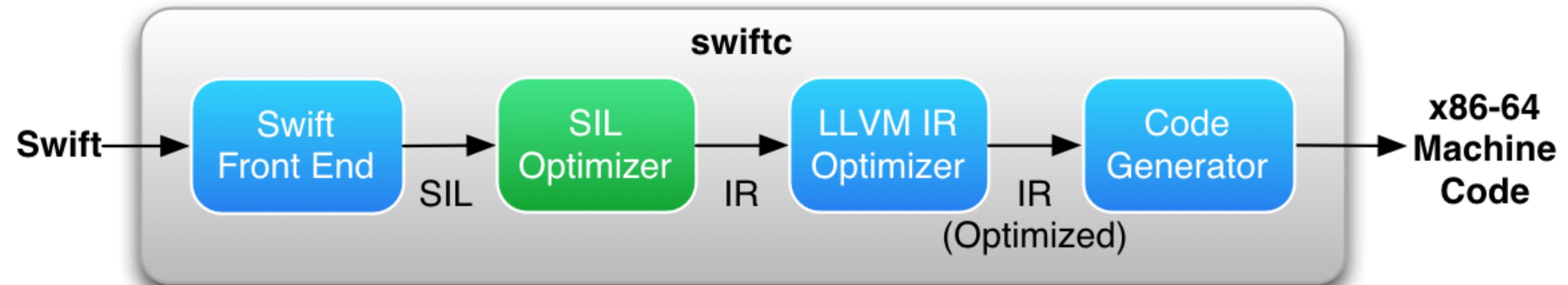
Developer Productivity

Safety

Memory Footprint

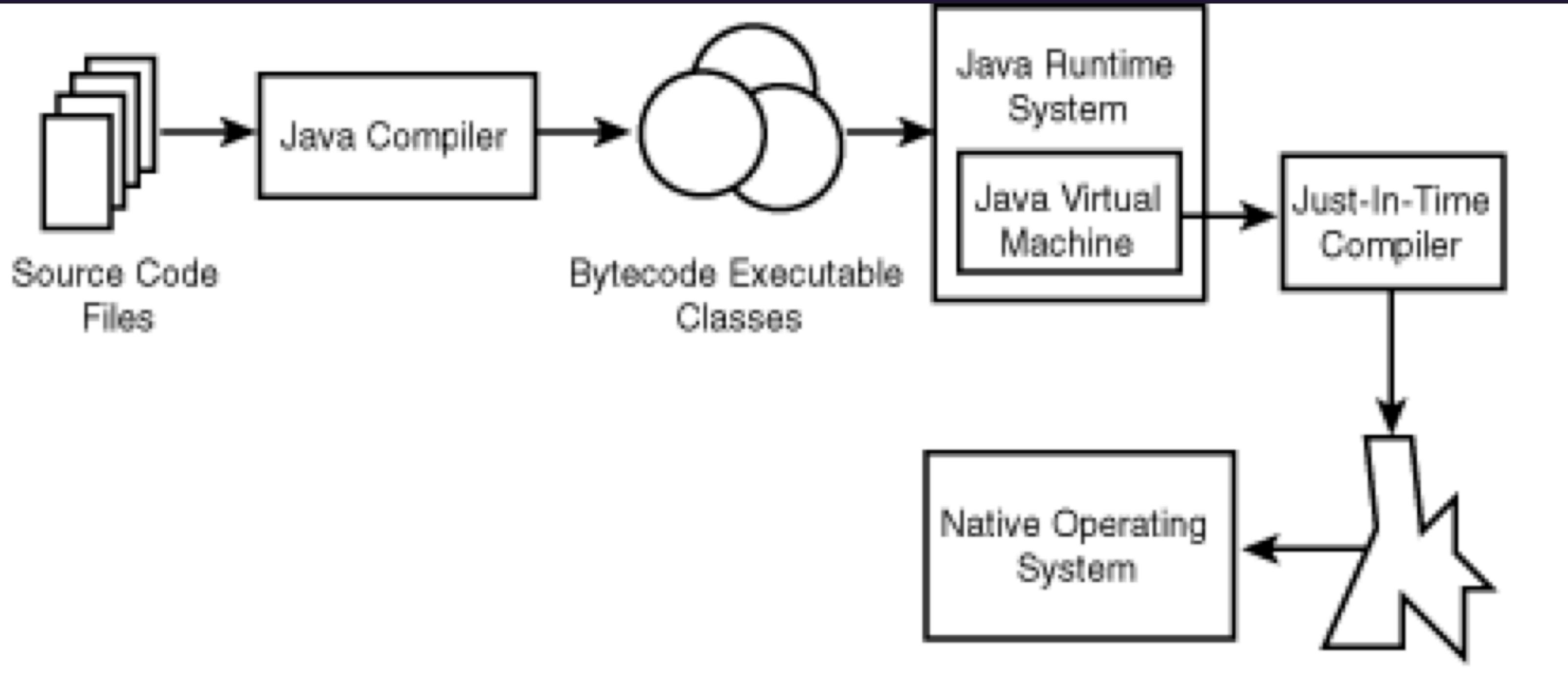
Performance

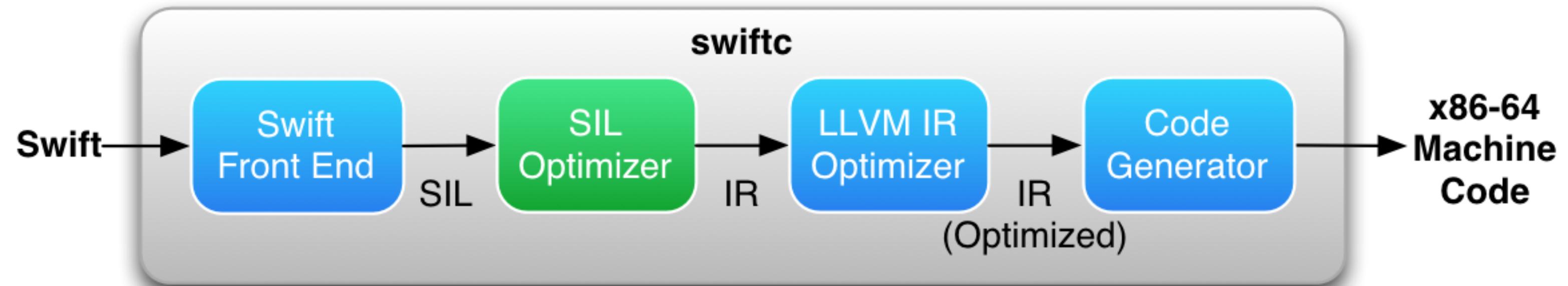






	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
General Purpose - Current Generation					
t2.nano	1	Variable	0.5	EBS Only	\$0.0059 per Hour
t2.micro	1	Variable	1	EBS Only	\$0.012 per Hour
t2.small	1	Variable	2	EBS Only	\$0.023 per Hour
t2.medium	2	Variable	4	EBS Only	\$0.047 per Hour
t2.large	2	Variable	8	EBS Only	\$0.094 per Hour
t2.xlarge	4	Variable	16	EBS Only	\$0.188 per Hour
t2.2xlarge	8	Variable	32	EBS Only	\$0.376 per Hour
m4.large	2	6.5	8	EBS Only	\$0.1 per Hour
m4.xlarge	4	13	16	EBS Only	\$0.2 per Hour
m4.2xlarge	8	26	32	EBS Only	\$0.4 per Hour
m4.4xlarge	16	53.5	64	EBS Only	\$0.8 per Hour
m4.10xlarge	40	124.5	160	EBS Only	\$2 per Hour
m4.16xlarge	64	188	256	EBS Only	\$3.2 per Hour
m3.medium	1	3	3.75	1 x 4 SSD	\$0.067 per Hour
m3.large	2	6.5	7.5	1 x 32 SSD	\$0.133 per Hour
m3.xlarge	4	13	15	2 x 40 SSD	\$0.266 per Hour
m3.2xlarge	8	26	30	2 x 80 SSD	\$0.532 per Hour







Memory Footprint



Safety



java™

The background features a large, abstract graphic element composed of overlapping orange and red curved shapes, resembling a stylized flame or a rising sun, positioned in the lower right quadrant.

Developer Productivity

Backing



ORACLE

Java vs. Swift

The Java logo, featuring the word "Java" in its signature font with a trademark symbol, and a stylized sunburst graphic to its left.

Developer Productivity

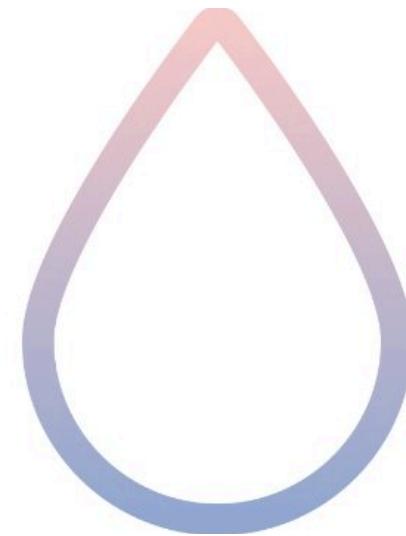
Safety

Memory Footprint

Performance

Swift 🤔

Server Side Swift Frameworks



Kitura

Independent Modules

Express.js-based

Educational resources

Bluemix



Vapor



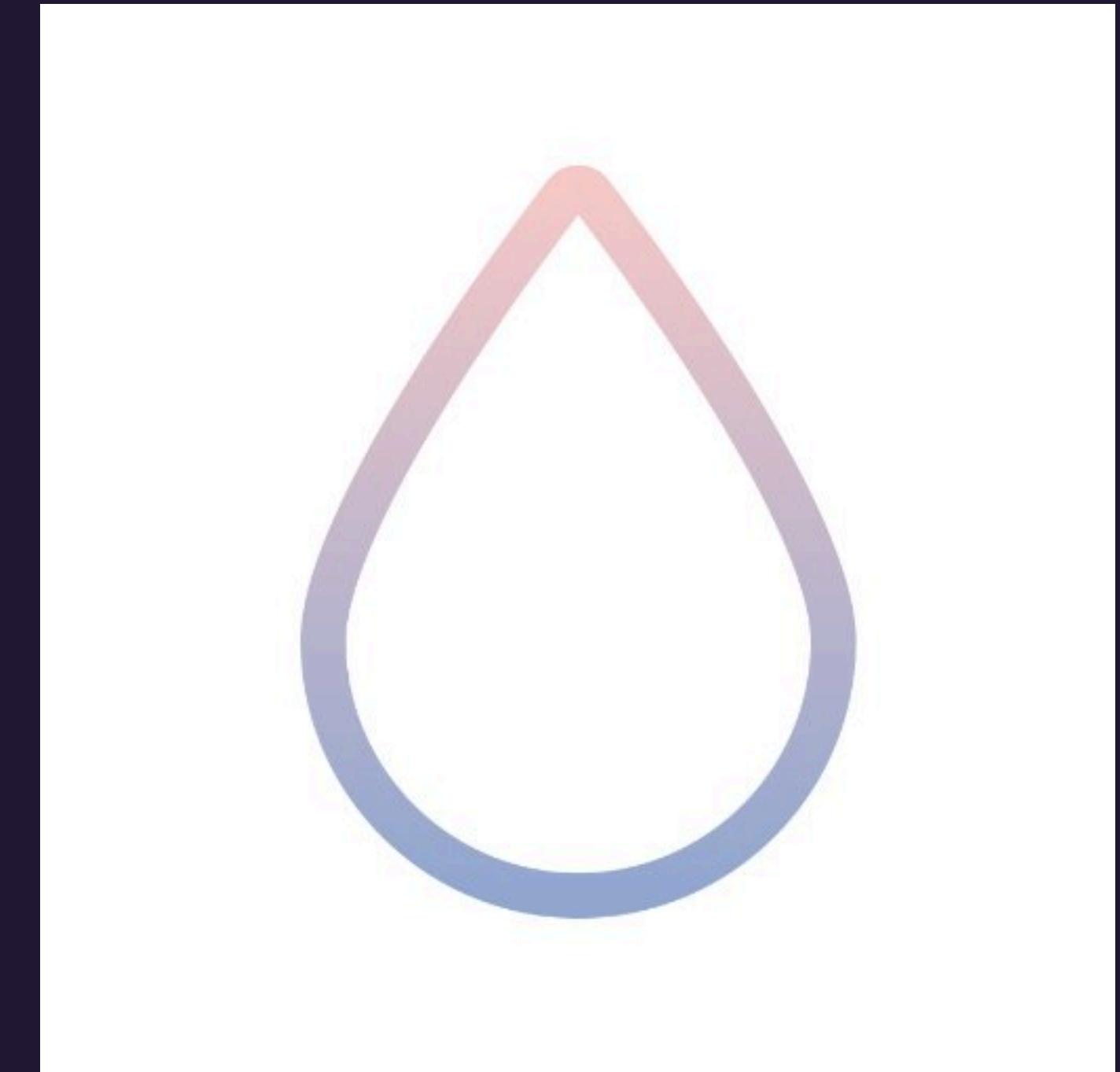
Pure Swift



Documentation



Collaborative



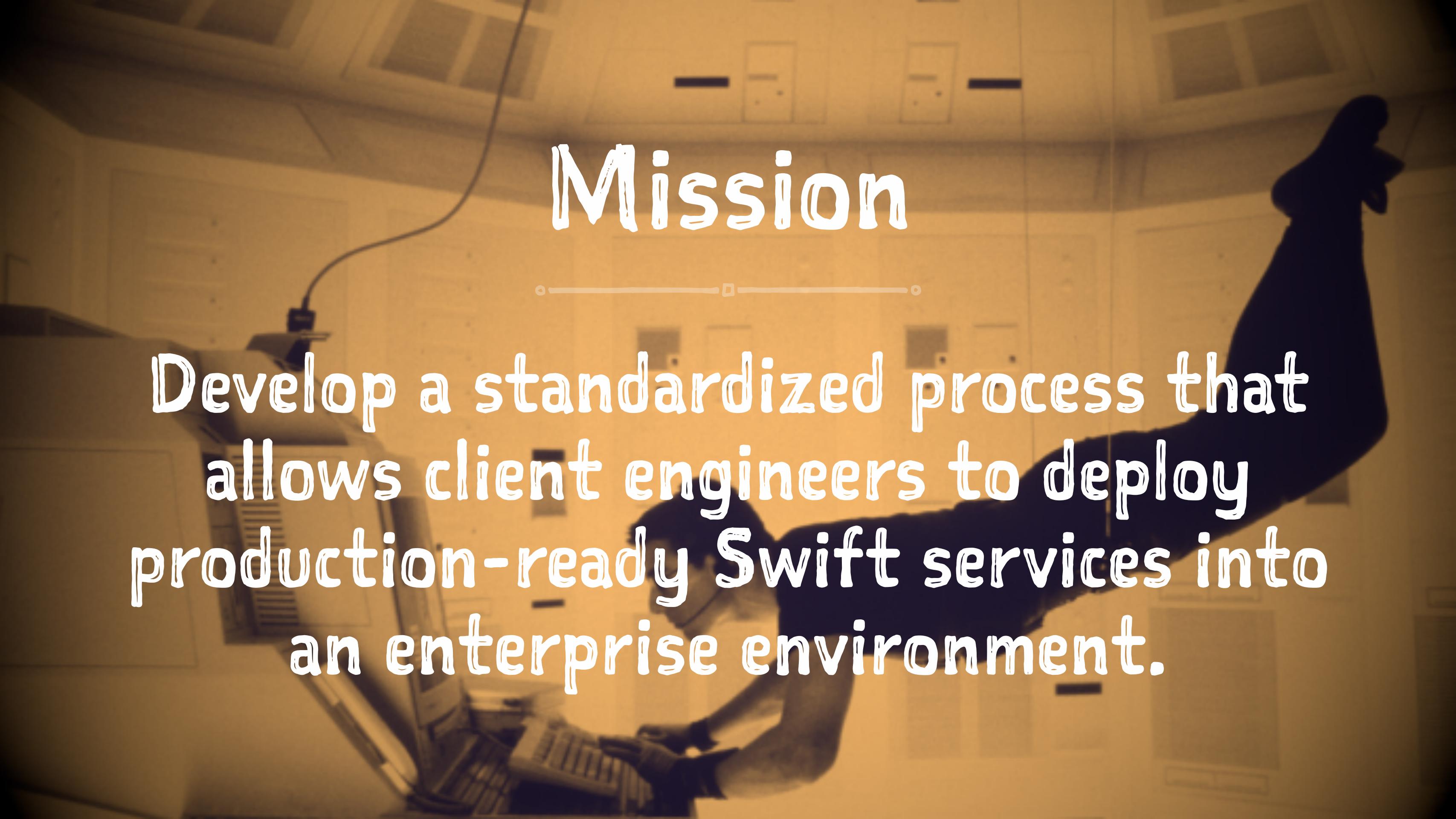
Server APIs Project

Focus Areas

The project will initially focus on producing API proposals for base networking, security and encryption, and HTTP parsing:

- **Base Networking:** Provide a portable interface for low level socket-based network I/O, including TCP/IP and UDP protocols, IPv4 and IPv6 support and domain name resolution. Support should be provided to create and use both synchronous and asynchronous non-blocking connections.
- **Security and Encryption:** Provide common cryptographic constants and cyphers along with keychain and certificate management, and SSL/TLS based secure transport. This must integrate with the base networking support to provide secure sockets, and with the HTTP parsing library to provide HTTPS support.
- **HTTP and WebSockets:** Provide low level HTTP parsing, including HTTP, HTTP/2 and WebSocket support making it possible, in conjunction with the security and networking APIs, to create secure HTTP and WebSocket servers.

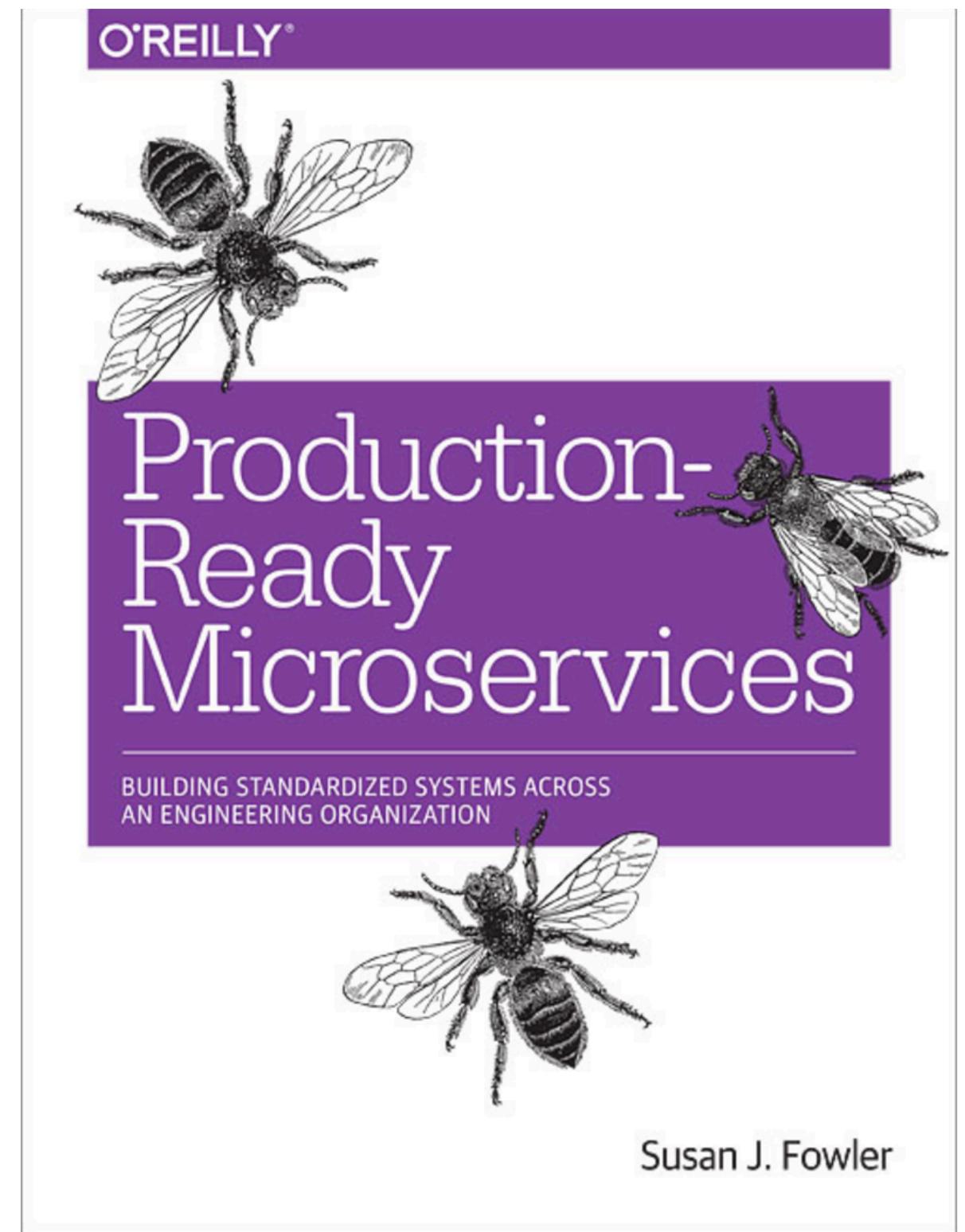
Mission



Develop a standardized process that allows client engineers to deploy production-ready Swift services into an enterprise environment.

Production-Ready

1. Stability
2. Reliability
3. Scalability
4. Performance
5. Fault Tolerance
6. Monitoring



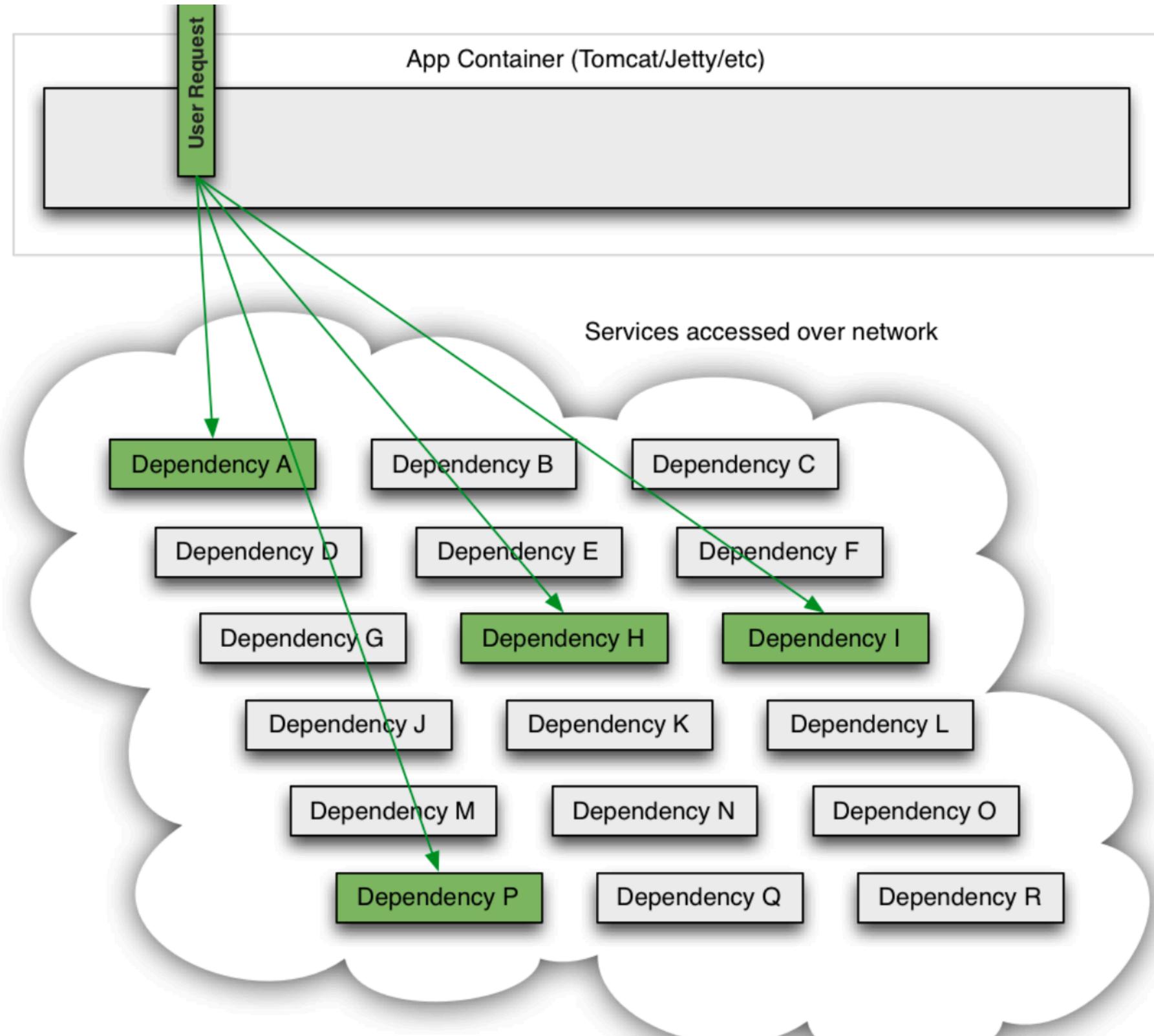
Fault Tolerance

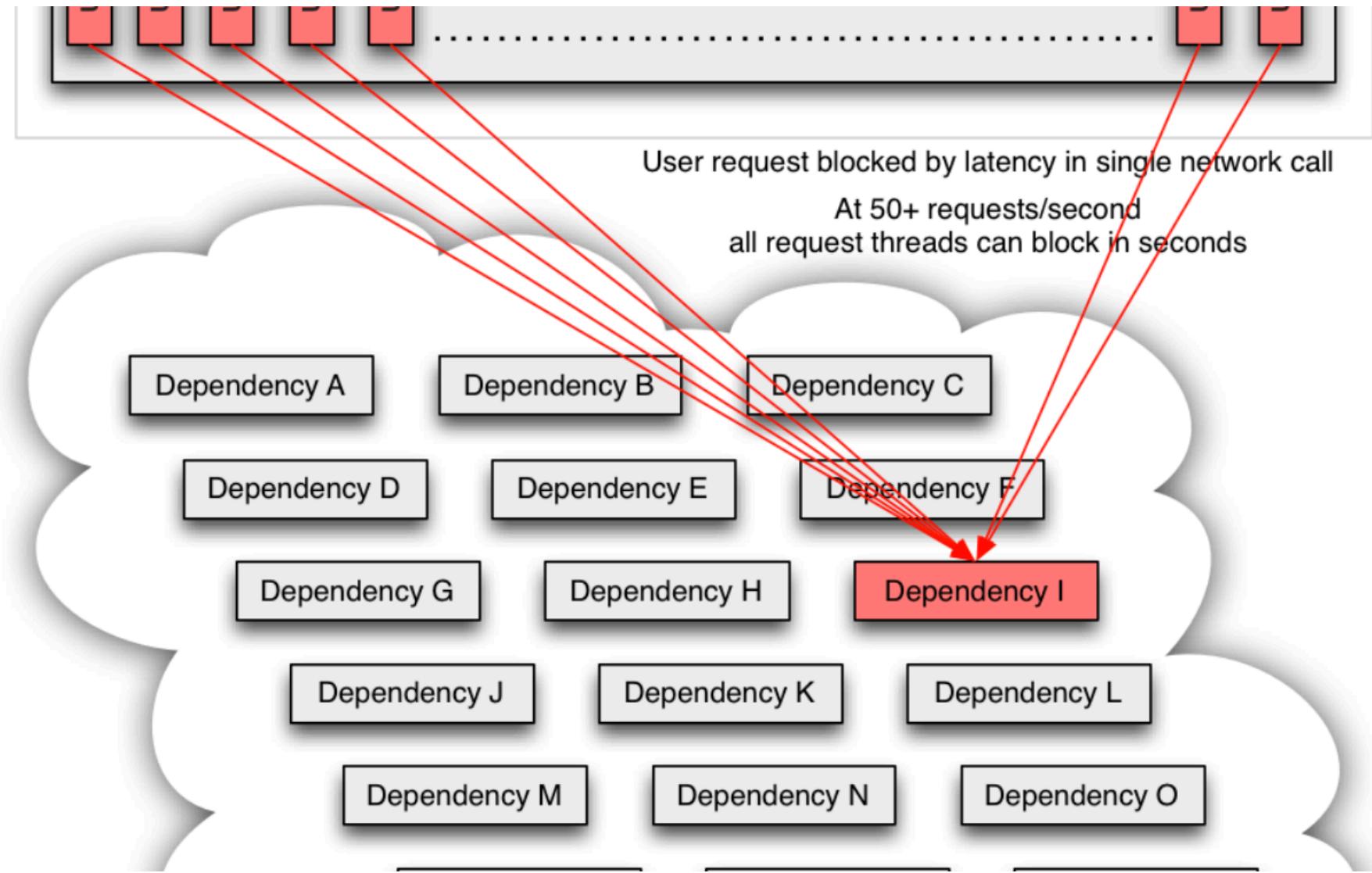
"... a capability of a computer system or network to deliver uninterrupted service and resolve potential service interruptions, despite one or more of its components failing.

"The purpose is to prevent catastrophic failure that could result from a single point of failure."

Points of Failure

1. Hardware
2. Communication
3. Application
4. Microservice





- when service suffers, system suffers
- end-to-end availability of entire system compromised

Production-Ready

1. Stability
2. Reliability
3. Scalability
4. Performance
5. Fault Tolerance
6. Monitoring

"The Netflix API receives more than 1 billion incoming calls per day which results in several billion outgoing calls to dozens of underlying subsystems with peaks of over 100k dependency requests per second."

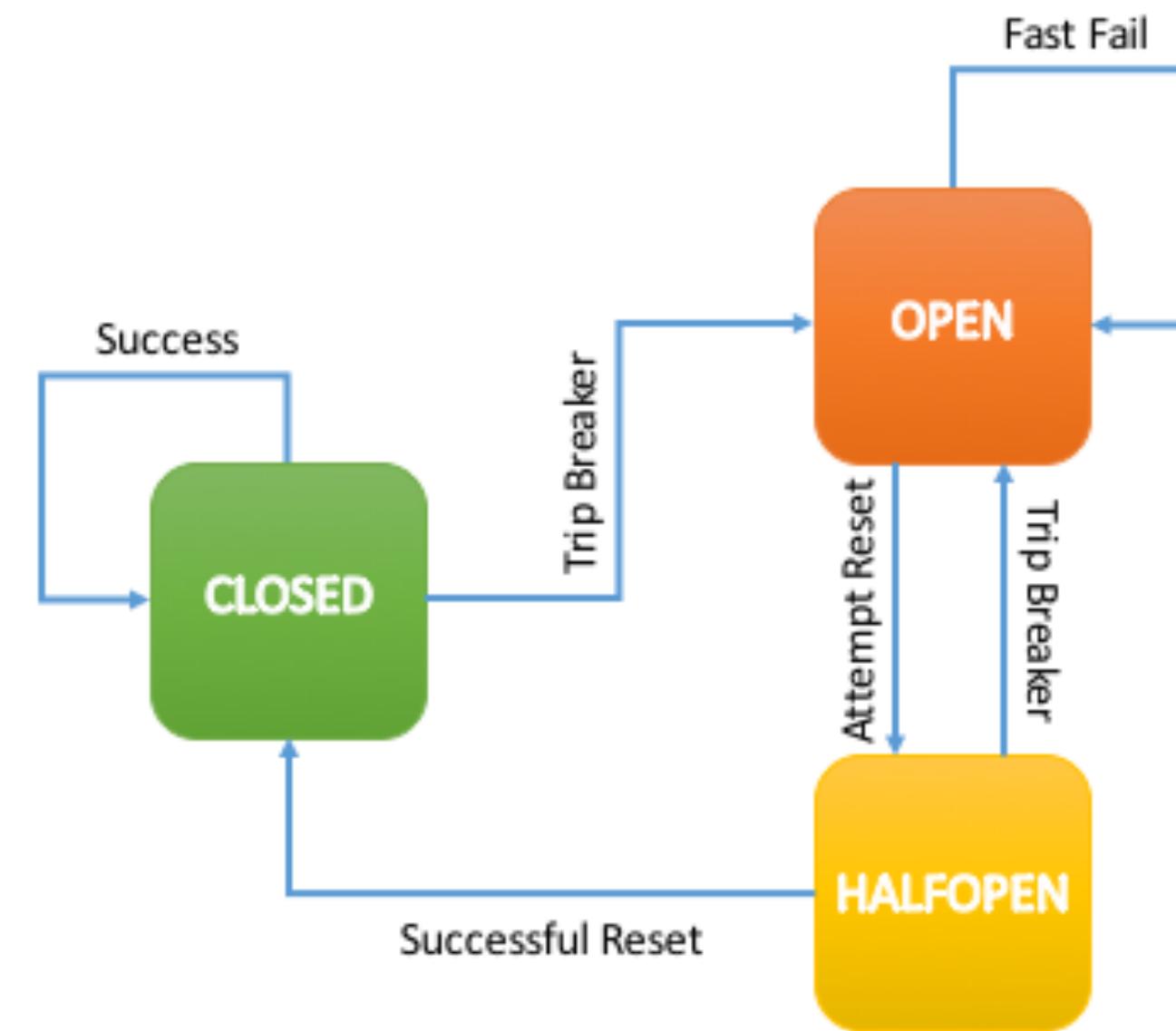


- Netflix Tech Blog

Microservices Concern	Spring Cloud & Netflix OSS
Configuration Management	Config Server, Consul, Netflix Archaius
Service Discovery	Netflix Eureka, Hashicorp Consul
Load Balancing	Netflix Ribbon
API Gateway	Netflix Zuul
Service Security	Spring Cloud Security
Centralized Logging	ELK Stack (Logstash)
Centralized Metrics	Netflix Spectator & Atlas
Distributed Tracing	Spring Cloud Sleuth, Zipkin
Resilience & Fault Tolerance	Netflix Hystrix, Turbine & Ribbon
Auto Scaling & Self Healing	-
Packaging, Deployment & Scheduling	Spring Boot
Job Management	Spring Batch
Singleton Application	Spring Cloud Cluster



Hystrix - Circuit Breaker Pattern



Swiftrix



Java <--> C <--> Swift



Java to C calls are handled by Java Native Interface (JNI)

... The Bad, And The Ugly



Compilation



Debugging



Error Handling



XCode --> Terminal

IBM's Circuit Breaker Library

 [IBM-Swift / CircuitBreaker](#)

 [Code](#)  [Issues 0](#)  [Pull requests 0](#)  [Projects 0](#) [Insights ▾](#)

 [Watch](#) 12  [Star](#) 6  [Fork](#) 0

A Swift Circuit Breaker library – Improves application stability and reliability.

[swift](#) [circuit-breaker](#) [bulkhead](#)

 **Swift 100.0%**



Circuit Breaker Constructor

```
private init(timeout: Int, resetTimeout: Int, maxFailures: Int, rollingWindow: Int, bulkhead: Int, command: (AnyFunction<A, B>)?, contextCommand: AnyFunction<A, B>?) {
    self.timeout = timeout
    self.resetTimeout = resetTimeout
    self.maxFailures = maxFailures
    self.rollingWindow = rollingWindow
    self.fallback = fallback
    self.command = command
    self.contextCommand = contextCommand
    self.failures = FailureQueue(size: maxFailures)
    self.bulkhead = (bulkhead > 0) ? Bulkhead.init(limit: bulkhead) : nil
}
```

1.) Define the Fallback Function

```
func myFallback(err: BreakerError, msg: String) {
    // The fallback will be called if the request does not return before the specified timeout
    // or if the CircuitBreaker is currently in Open state and set to fail fast.
    // Client code can use the fallback function to do alternate processing, such as show an error page.
    Log.verbose("Error: \(error)")
    Log.verbose("Message: \(msg)")
}
```

2.) Create a Function to Circuit Break

```
func myFunction(a: Int, b: Int) -> Int {  
    // do stuff  
    let value: Int = ...  
    return value  
}
```

3.) Create a `CircuitBreaker` instance
for each endpoint you want to circuit
break

```
let breaker = CircuitBreaker(command: myFunction, fallback: myFallback)
```

Optional configurations: `timeout`, `resetTimeout`, `maxFailures`,
`etc`

4.) Invoke the call to the function by calling CircuitBreaker run()

```
breaker.run(commandArgs: (a: 10, b: 20), fallbackArgs: (msg: "Something went wrong."))
```

Tracked Stats

- Total Requests
- Concurrent Requests
- Rejected Requests
- Successful Responses
- Average Response Time
- Failed Responses
- Total Timeouts
- Total Latency

Advantages:

1. Pure Swift
2. Ability to set many parameters for the Circuit Breaker constructor (timeout, resetTimeout, maxFailures, rollingWindow, etc)
3. Open source + IBM

Mission

Develop a standardized process that allows client engineers to deploy production-ready Swift services into an enterprise environment.



ubuntu®





docker



Sample Dockerfile

```
# Set work dir to /vapor
WORKDIR /vapor

# Copy files
COPY files/$HarmonyTar .
COPY files/$HystrixTar .
COPY files/$SwiftTar .
COPY files/SwiftJNI SwiftJNI/

# Install Swift xAPI Source Code
# Note: These are not GZIPped ... so just use tar
RUN tar xfz $HystrixTar 2>\&1 >/dev/null
RUN tar xfz $SwiftTar
RUN make ./swiftrix/CLibrary
RUN make ./swiftrix/JavaSource
RUN tar xfz $HarmonyTar 2>\&1 >/dev/null

RUN rm \&1
```

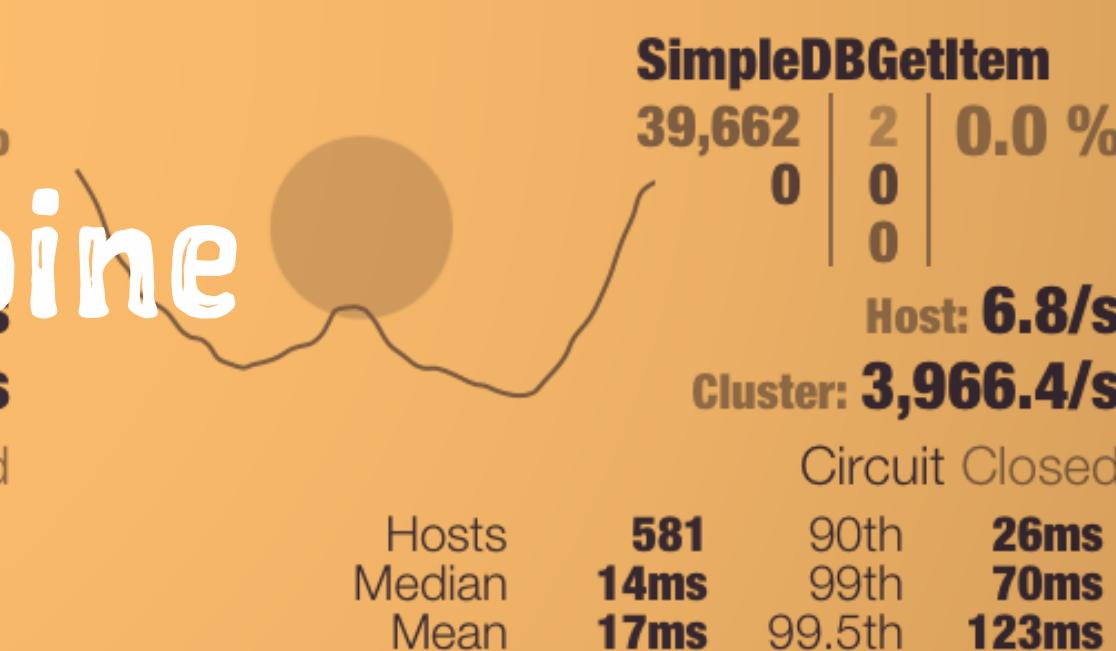
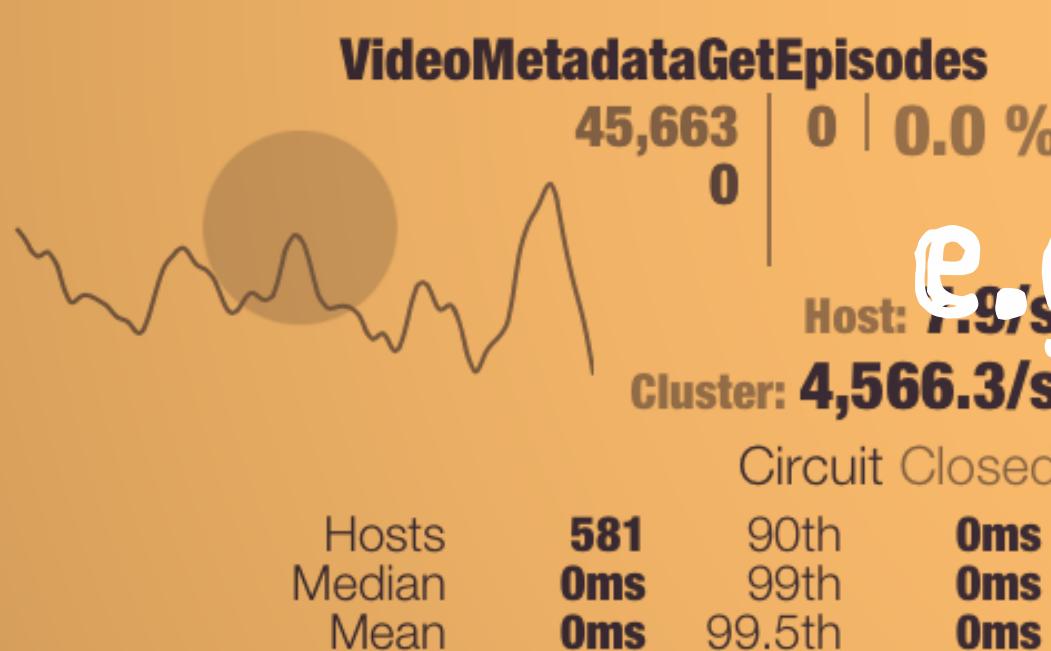
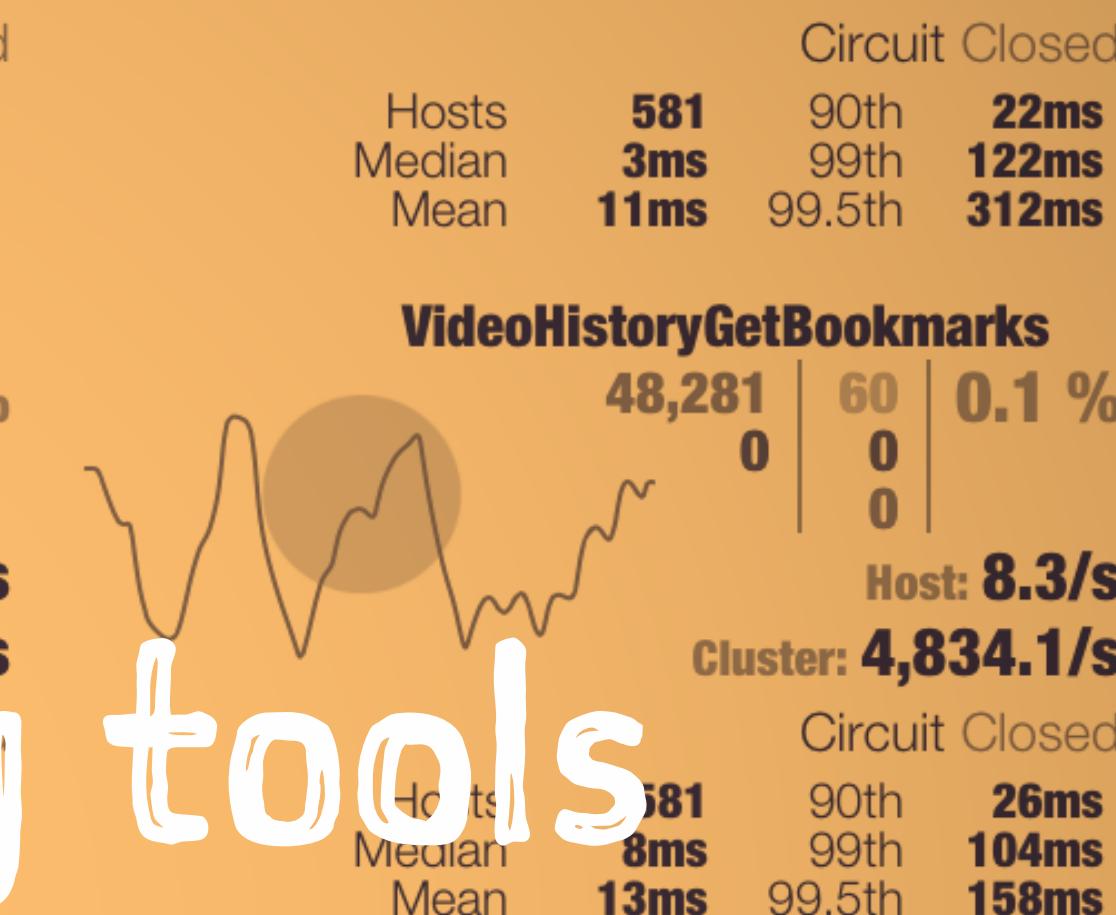
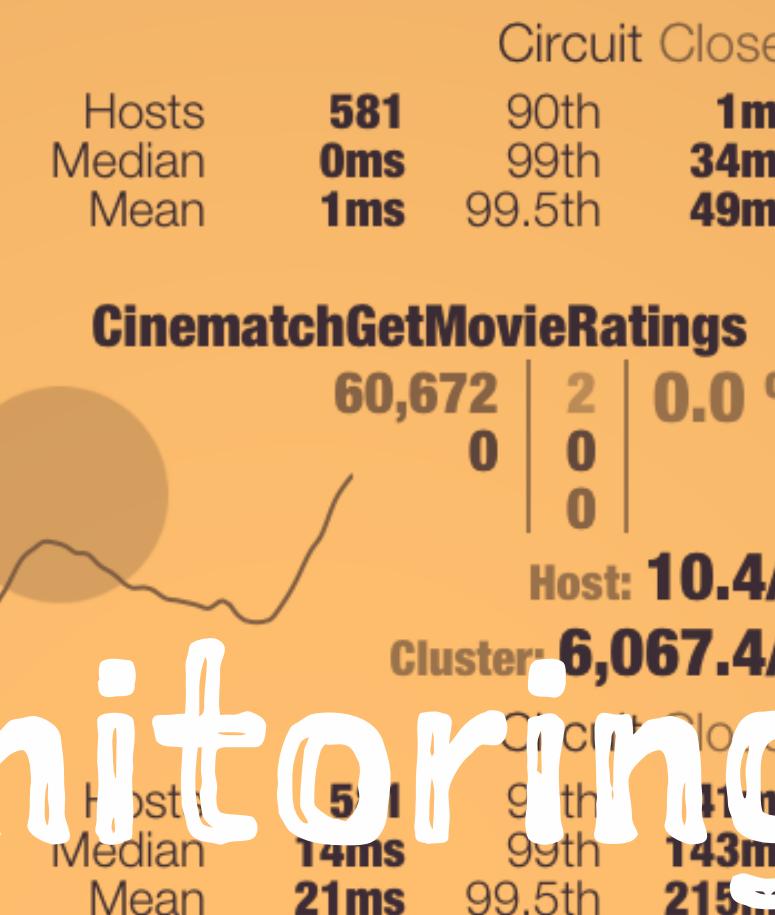
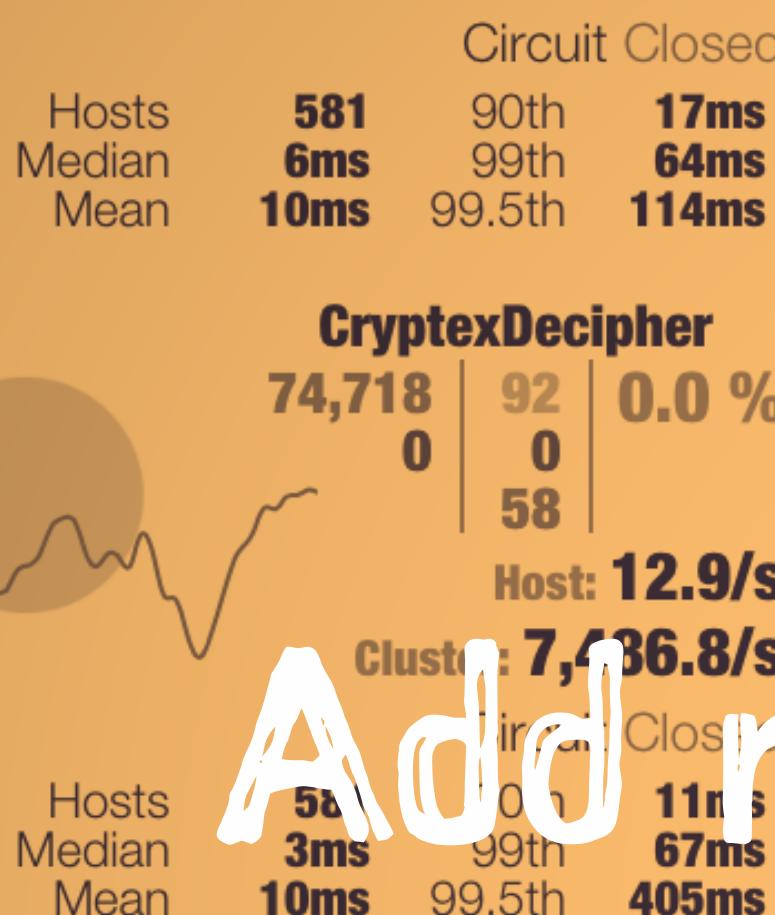
Next Steps

A young boy with blonde hair, wearing a black mask and a dark blue cape over a white t-shirt with a large black star on the chest, stands on a concrete ledge against a plain, light-colored wall. He is looking directly at the camera with a neutral expression.

Create a framework for our
client developers

Add monitoring tools

e.g. **Hystrix Turbine**





Contribute back to Server
Side Swift community

Questions?



tripta.gupta@capitalone.com



@triptagupta