REAL TIME TRAFFIC VISUALIZATION IN A MICROSERVICES WORLD

ROBERTO PEREZ ALCOLEA - CODEMASH 2018

WHO AM I?

Roberto Perez Alcolea

» Mexican

» Works @ Target

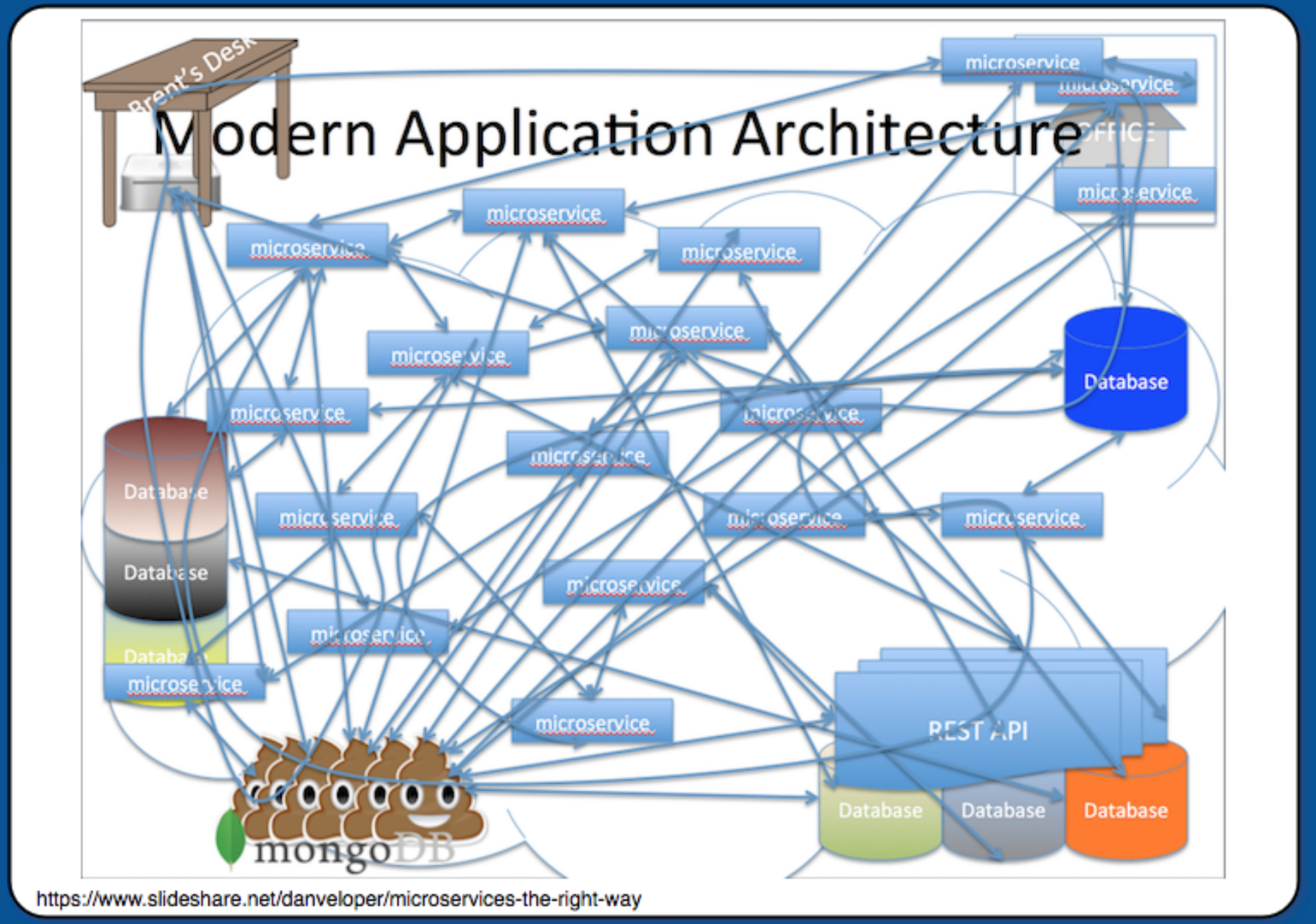
» Groovy Enthusiast

» roberto@perezalcolea.info

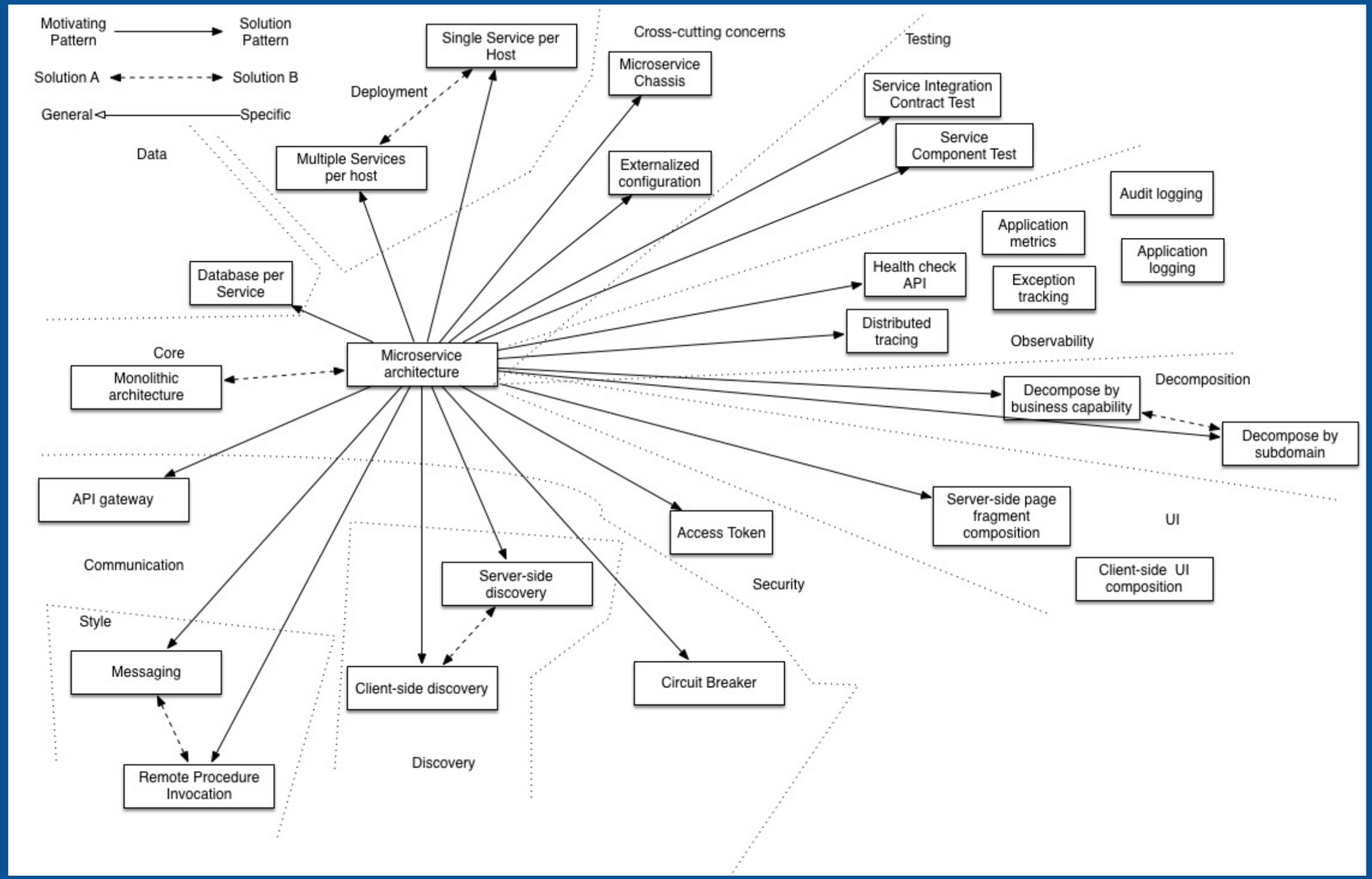
» @rpalcolea

MICROSERVICES

ROBERTO PEREZ ALCOLEA - CODEMASH 2018



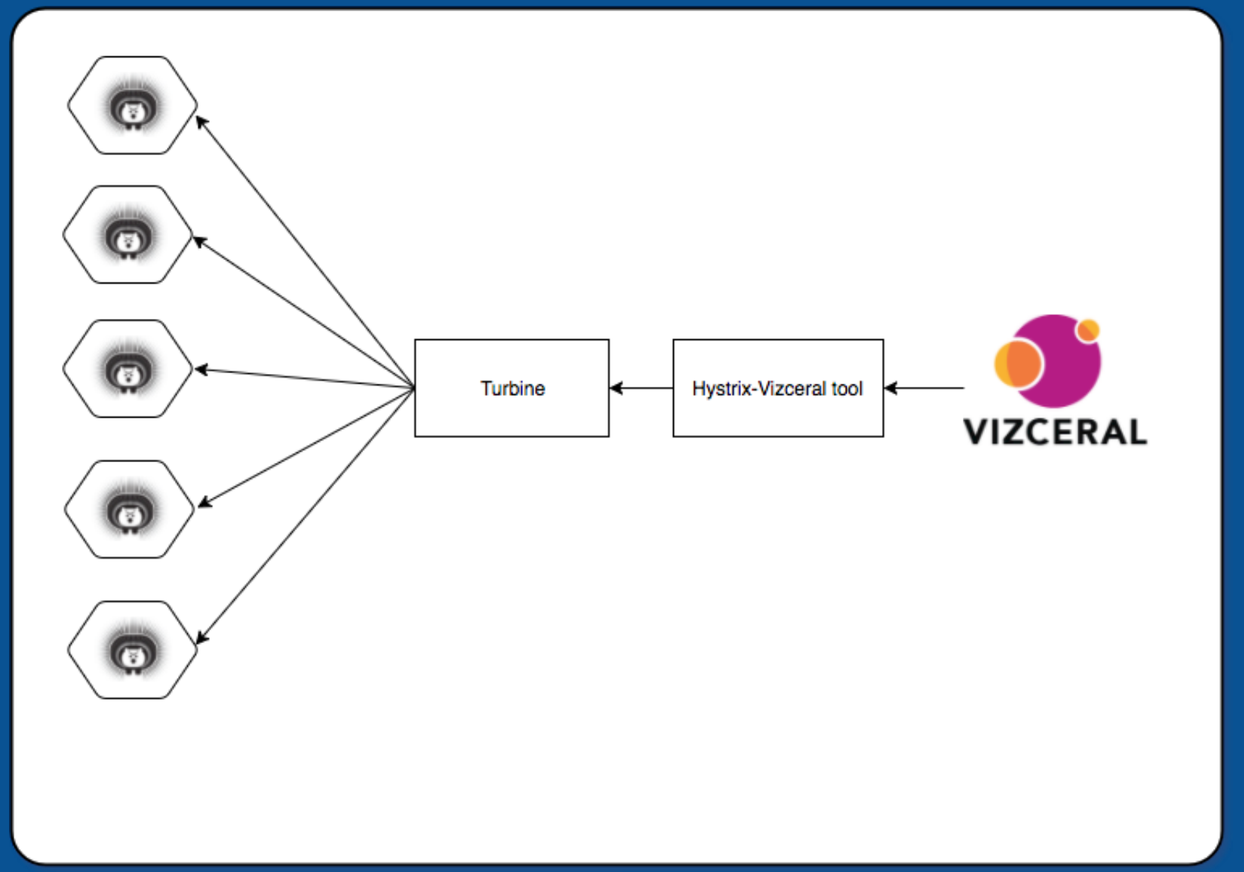
https://www.slideshare.net/danveloper/microservices-the-right-way


Motivating Pattern
Solution Pattern
Solution A
Solution B
General
Specific
Data
Deployment
Single Service per Host
Multiple Services per host
Database per Service
Cross-cutting concerns
Microservice Chassis
Externalized configuration
Testing
Service Integration Contract Test
Service Component Test
Audit logging
Application metrics
Application logging
Health check API
Exception tracking
Distributed tracing
Observability
Core
Monolithic architecture
Microservice architecture
Decomposition
Decompose by business capability
Decompose by subdomain
API gateway
Communication
Server-side discovery
Client-side discovery
Discovery
Access Token
Security
Circuit Breaker
Server-side page fragment composition
UI
Client-side UI composition
Style
Messaging
Remote Procedure Invocation

TRAFFIC VISUALIZATION

ROBERTO PEREZ ALCOLEA - CODEMASH 2018

Hystrix + Turbine + Vizceral



HYSTRIX

ROBERTO PEREZ ALCOLEA - CODEMASH 2018

HYSTRIX

Library to help control interactions between distributed services

» Times out slow commands

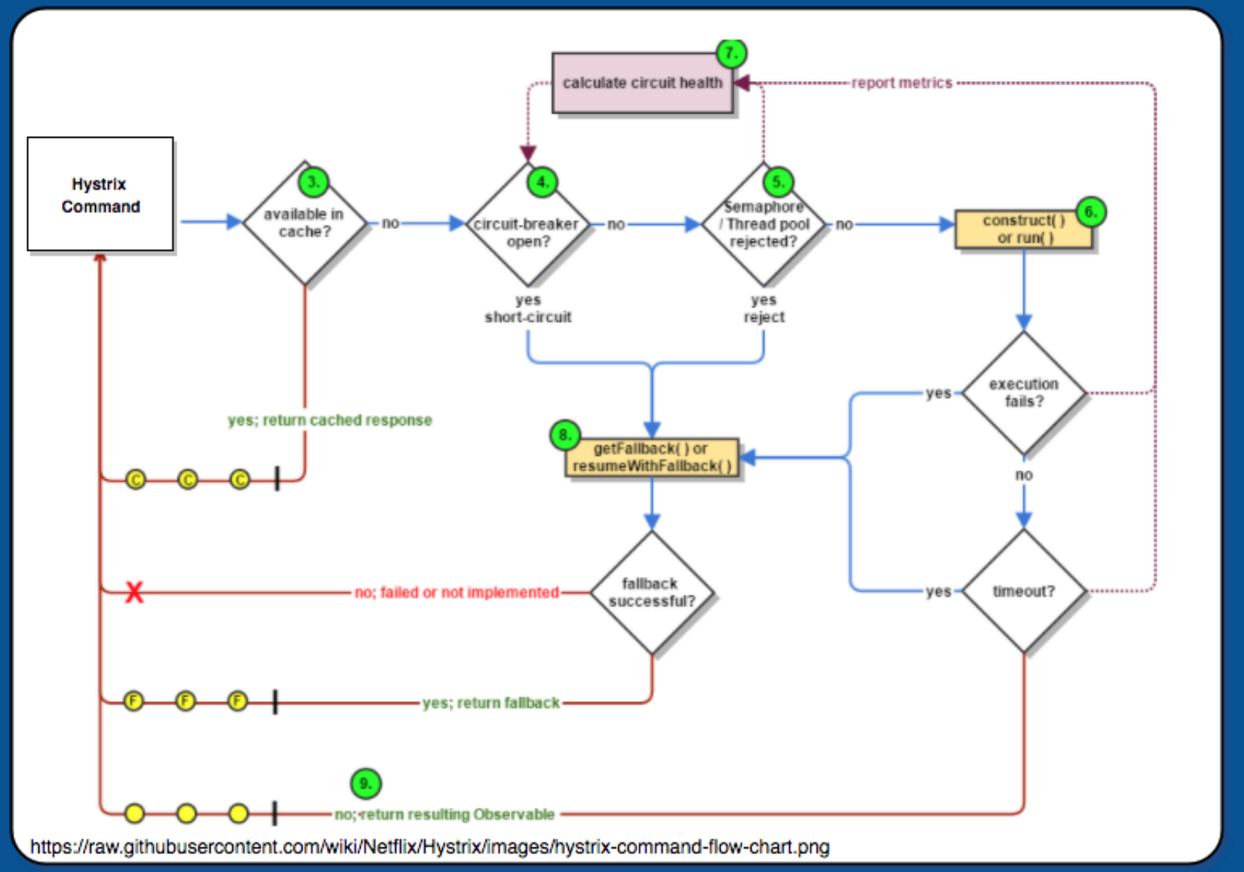
» Rate limit number of concurrent commands

» Circuit breaker

» Fallback on failures

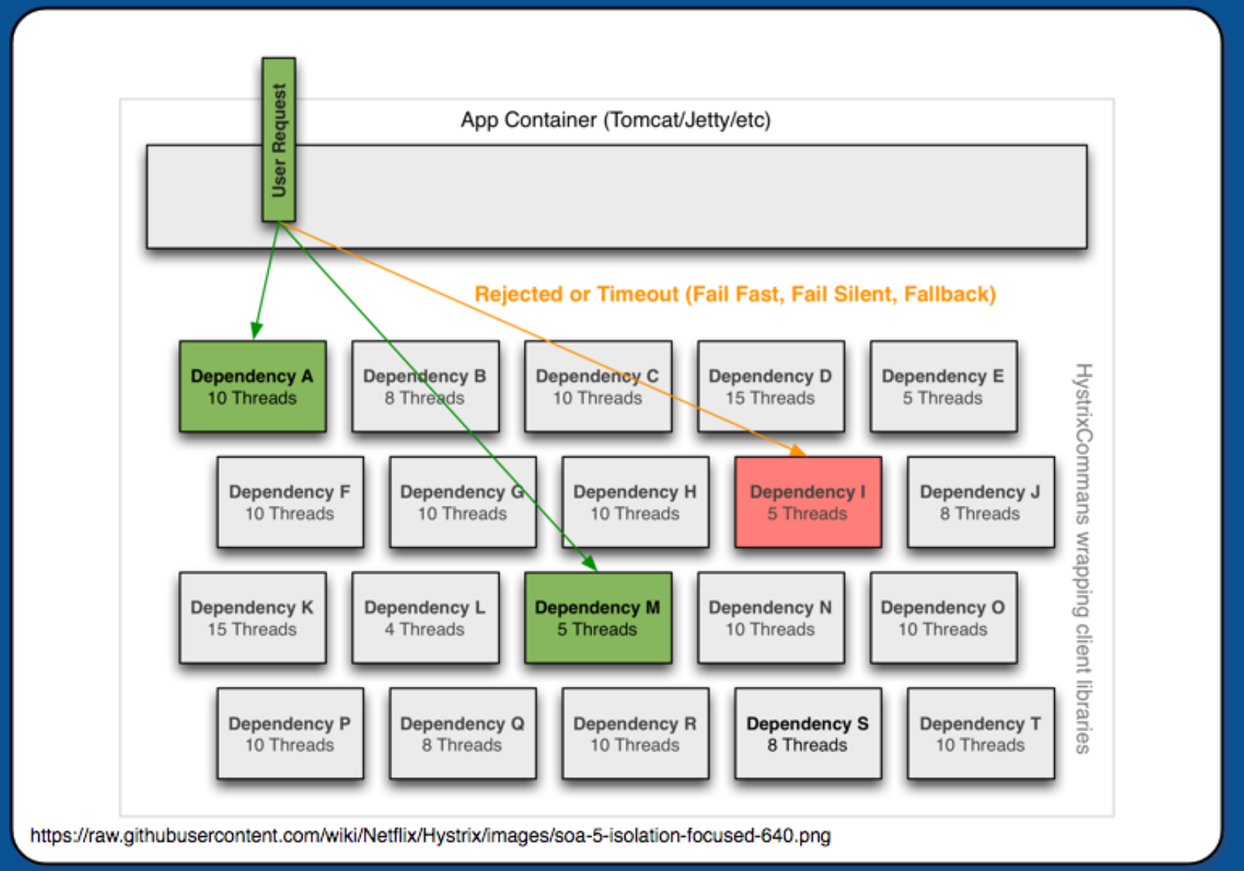
» Real-Time dashboard for visibility

HYSTRIX FLOW



https://raw.githubusercontent.com/wiki/Netflix/Hystrix/images/hystrix-command-flow-chart.png

ISOLATION



https://raw.githubusercontent.com/wiki/Netflix/Hystrix/images/soa-5-isolation-focused-640.png

HYSTRIX

```
...
private static final HystrixObservableCommand.Setter hystrixSetter = HystrixObservableCommand.Setter
        .withGroupKey(HystrixCommandGroupKey.Factory.asKey("github-service"))
        .andCommandKey(HystrixCommandKey.Factory.asKey("GithubService.getRepositories"))
        .andCommandPropertiesDefaults(HystrixCommandProperties.defaultSetter())

...
Observable<Map> call(String language) {
        return new HystrixObservableCommand<Map>(hystrixSetter) {
            @Override
            protected Observable<Map> construct() {
                return getRepositories(language)
            }

            @Override
            protected Observable<Map> resumeWithFallback() {
                return Observable.just([:])
            }

        }.toObservable()
    }
```

HYSTRIX

```
...
private static final HystrixObservableCommand.Setter hystrixSetter = HystrixObservableCommand.Setter
        .withGroupKey(HystrixCommandGroupKey.Factory.asKey("github-service"))
        .andCommandKey(HystrixCommandKey.Factory.asKey("GithubService.getRepositories"))
        .andCommandPropertiesDefaults(HystrixCommandProperties.defaultSetter())

...
Observable<Map> call(String language) {
        return new HystrixObservableCommand<Map>(hystrixSetter) {
            @Override
            protected Observable<Map> construct() {
                return getRepositories(language)
            }

            @Override
            protected Observable<Map> resumeWithFallback() {
                return Observable.just([:])
            }

        }.toObservable()
    }
```

HYSTRIX COMMAND METRICS


```
{
 "type": "HystrixCommand",
 "name": "GithubService.findRepositories",
 "group": "github-api",
 "currentTime": 1498355684319,
 "isCircuitBreakerOpen": false,
 "errorPercentage": 0,
 "errorCount": 0,
 "requestCount": 1,
 "rollingCountBadRequests": 0, "rollingCountCollapsedRequests": 0,  "rollingCountEmit": 1, "rollingCountExceptionsThrown": 0,
 "rollingCountFailure": 0, "rollingCountFallbackFailure": 0,  "rollingCountFallbackRejection": 0, "rollingCountFallbackSuccess": 0,
 "rollingCountResponsesFromCache": 0, "rollingCountSemaphoreRejected": 0, "rollingCountShortCircuited": 0, "rollingCountSuccess": 1,
 "rollingCountThreadPoolRejected": 0,  "rollingCountTimeout": 0,
 "currentConcurrentExecutionCount": 0,
 "rollingMaxConcurrentExecutionCount": 1,
 "latencyExecute_mean": 0,
 "latencyExecute": {
   "99": 0,
 },
 "latencyTotal_mean": 0,
 "latencyTotal": {
   "99": 0,
 },
}
```

HYSTRIX COMMAND METRICS


```
{
 "type": "HystrixCommand",
 "name": "GithubService.findRepositories",
 "group": "github-api",
 "currentTime": 1498355684319,
 "isCircuitBreakerOpen": false,
 "errorPercentage": 0,
 "errorCount": 0,
 "requestCount": 1,
 "rollingCountBadRequests": 0, "rollingCountCollapsedRequests": 0,  "rollingCountEmit": 1, "rollingCountExceptionsThrown": 0,
 "rollingCountFailure": 0, "rollingCountFallbackFailure": 0,  "rollingCountFallbackRejection": 0, "rollingCountFallbackSuccess": 0,
 "rollingCountResponsesFromCache": 0, "rollingCountSemaphoreRejected": 0, "rollingCountShortCircuited": 0, "rollingCountSuccess": 1,
 "rollingCountThreadPoolRejected": 0,  "rollingCountTimeout": 0,
 "currentConcurrentExecutionCount": 0,
 "rollingMaxConcurrentExecutionCount": 1,
 "latencyExecute_mean": 0,
 "latencyExecute": {
   "99": 0,
 },
 "latencyTotal_mean": 0,
 "latencyTotal": {
   "99": 0,
 },
}
```

HYSTRIX COMMAND METRICS


```
{
 "type": "HystrixCommand",
 "name": "GithubService.findRepositories",
 "group": "github-api",
 "currentTime": 1498355684319,
 "isCircuitBreakerOpen": false,
 "errorPercentage": 0,
 "errorCount": 0,
 "requestCount": 1,
 "rollingCountBadRequests": 0, "rollingCountCollapsedRequests": 0,  "rollingCountEmit": 1, "rollingCountExceptionsThrown": 0,
 "rollingCountFailure": 0, "rollingCountFallbackFailure": 0,  "rollingCountFallbackRejection": 0, "rollingCountFallbackSuccess": 0,
 "rollingCountResponsesFromCache": 0, "rollingCountSemaphoreRejected": 0, "rollingCountShortCircuited": 0, "rollingCountSuccess": 1,
 "rollingCountThreadPoolRejected": 0,  "rollingCountTimeout": 0,
 "currentConcurrentExecutionCount": 0,
 "rollingMaxConcurrentExecutionCount": 1,
 "latencyExecute_mean": 0,
 "latencyExecute": {
   "99": 0,
 },
 "latencyTotal_mean": 0,
 "latencyTotal": {
   "99": 0,
 },
}
```

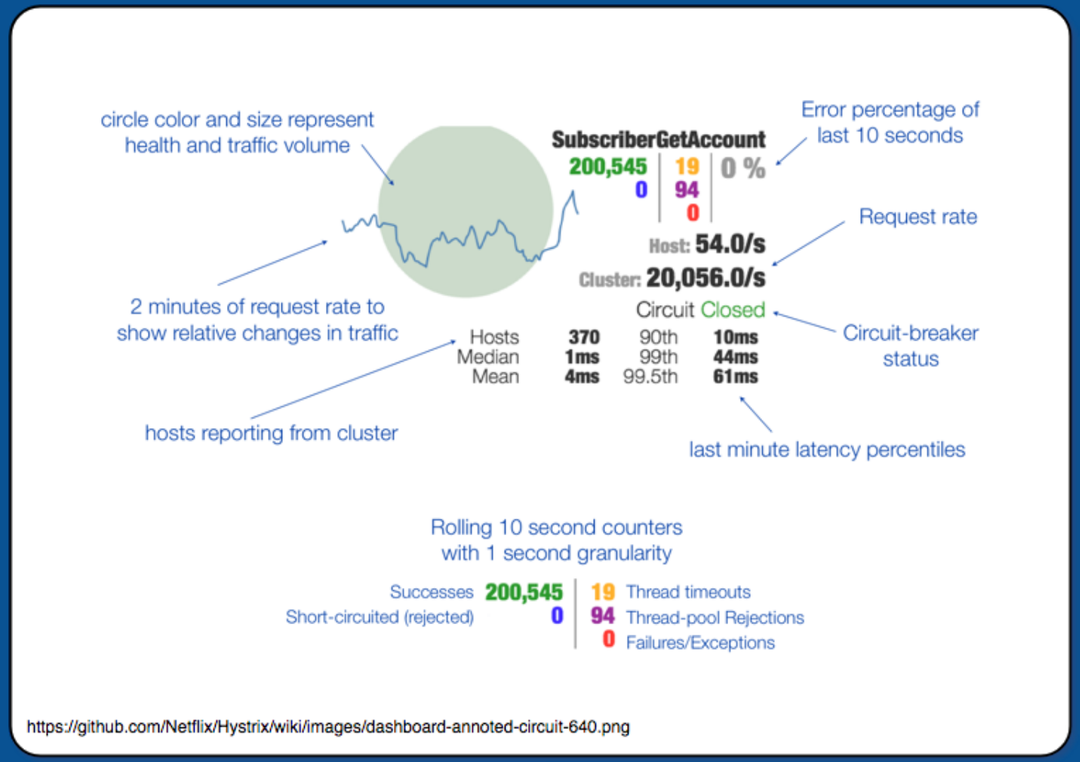
HYSTRIX COMMAND METRICS


```
{
 "type": "HystrixCommand",
 "name": "GithubService.findRepositories",
 "group": "github-api",
 "currentTime": 1498355684319,
 "isCircuitBreakerOpen": false,
 "errorPercentage": 0,
 "errorCount": 0,
 "requestCount": 1,
 "rollingCountBadRequests": 0, "rollingCountCollapsedRequests": 0,  "rollingCountEmit": 1, "rollingCountExceptionsThrown": 0,
 "rollingCountFailure": 0, "rollingCountFallbackFailure": 0,  "rollingCountFallbackRejection": 0, "rollingCountFallbackSuccess": 0,
 "rollingCountResponsesFromCache": 0, "rollingCountSemaphoreRejected": 0, "rollingCountShortCircuited": 0, "rollingCountSuccess": 1,
 "rollingCountThreadPoolRejected": 0,  "rollingCountTimeout": 0,
 "currentConcurrentExecutionCount": 0,
 "rollingMaxConcurrentExecutionCount": 1,
 "latencyExecute_mean": 0,
 "latencyExecute": {
   "99": 0,
 },
 "latencyTotal_mean": 0,
 "latencyTotal": {
   "99": 0,
 },
}
```

HYSTRIX DEMO

ROBERTO PEREZ ALCOLEA - CODEMASH 2018

HYSTRIX DASHBOARD



https://github.com/Netflix/Hystrix/wiki/images/dashboard-annoted-circuit-640.png

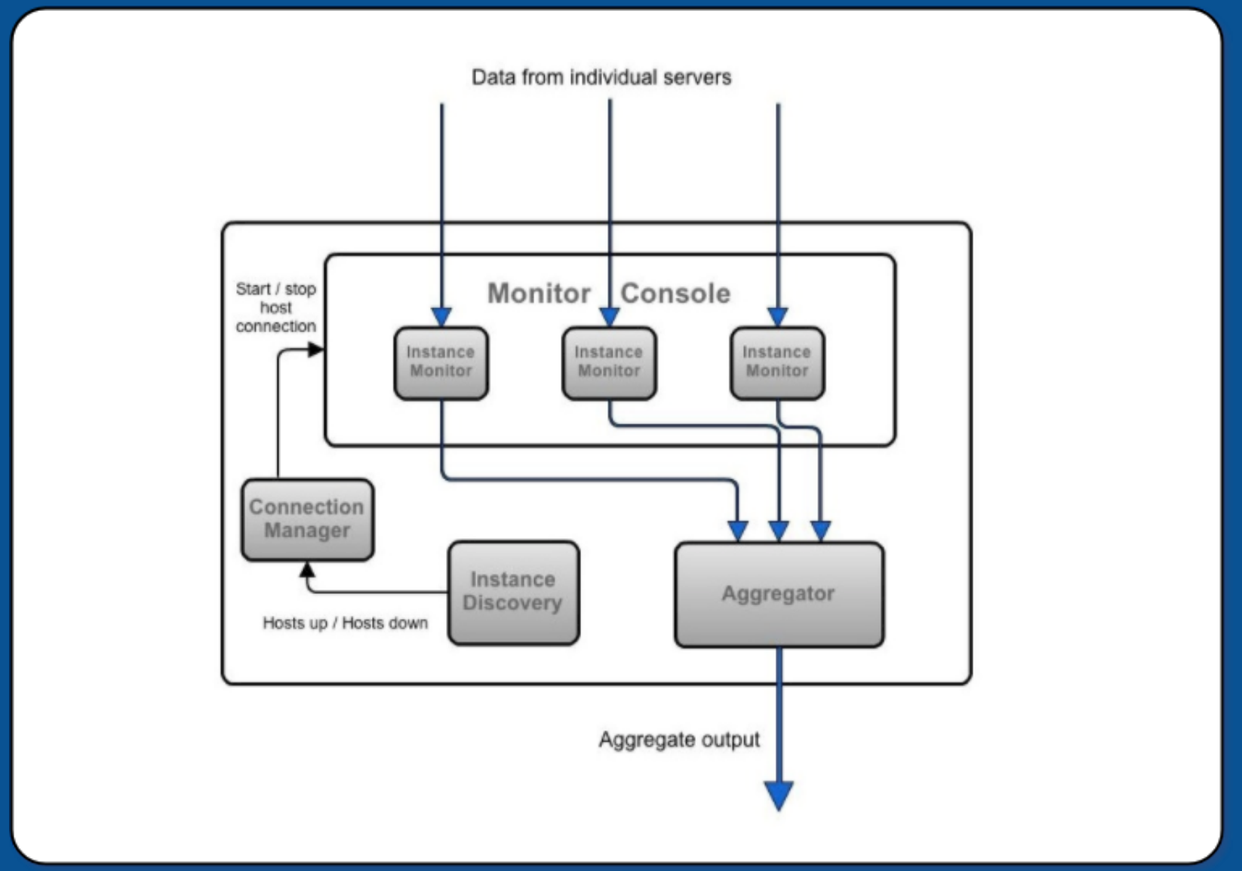
TURBINE

» Aggregates streams of Server-Sent Event (SSE) JSON data into a single stream

» Clusters

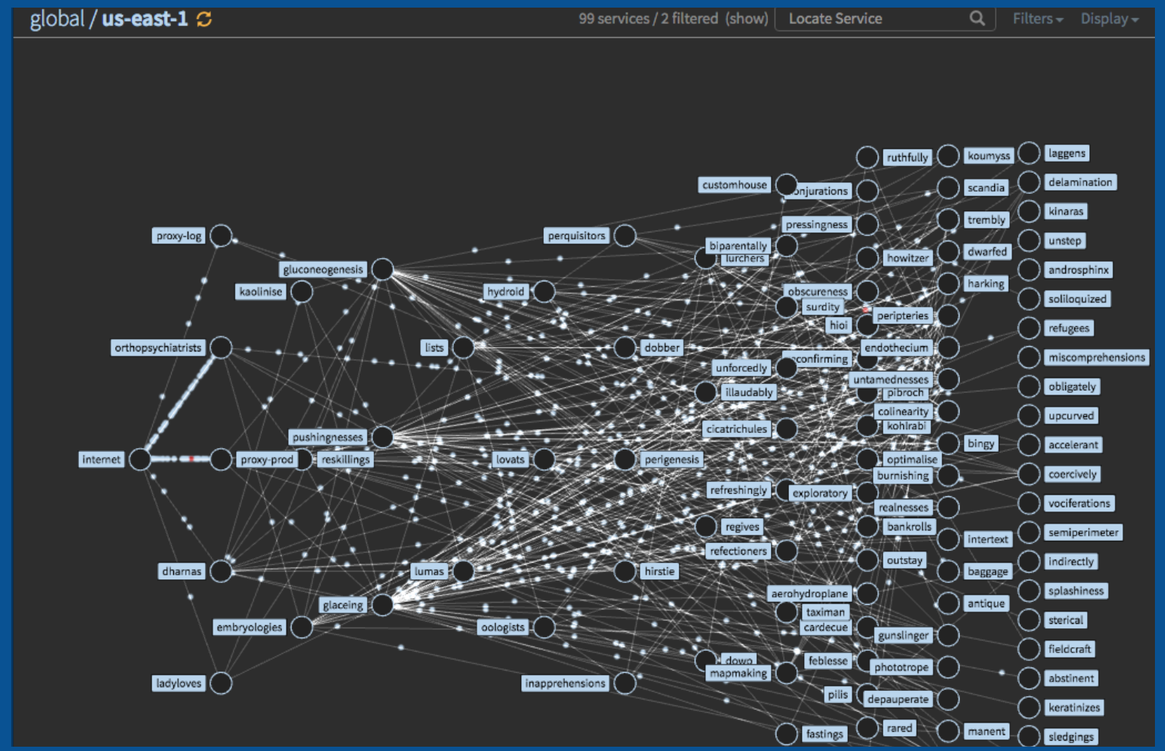
» Instance discovery

TURBINE (HOW IT WORKS)



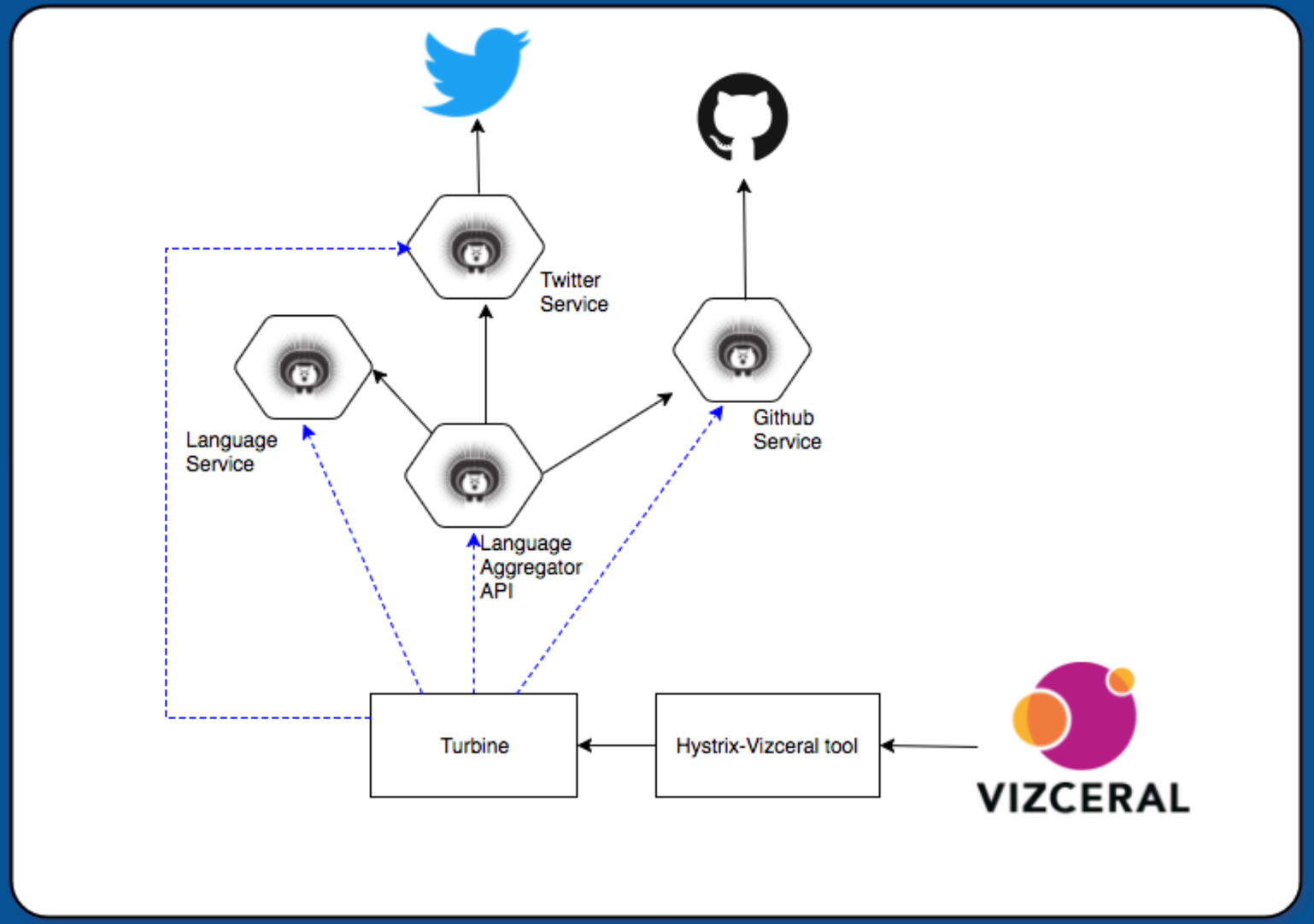
VIZCERAL

Vizceral is a tool from Netflix to visualize traffic between components



PUTTING ALL TOGETHER

DEMO


Twitter
Service
Language
Service
Github
Service
Language
Aggregator
API
Turbine
Hystrix-Vizceral tool
VIZCERAL

THANK YOU

ROBERTO PEREZ ALCOLEA - CODEMASH 2018