

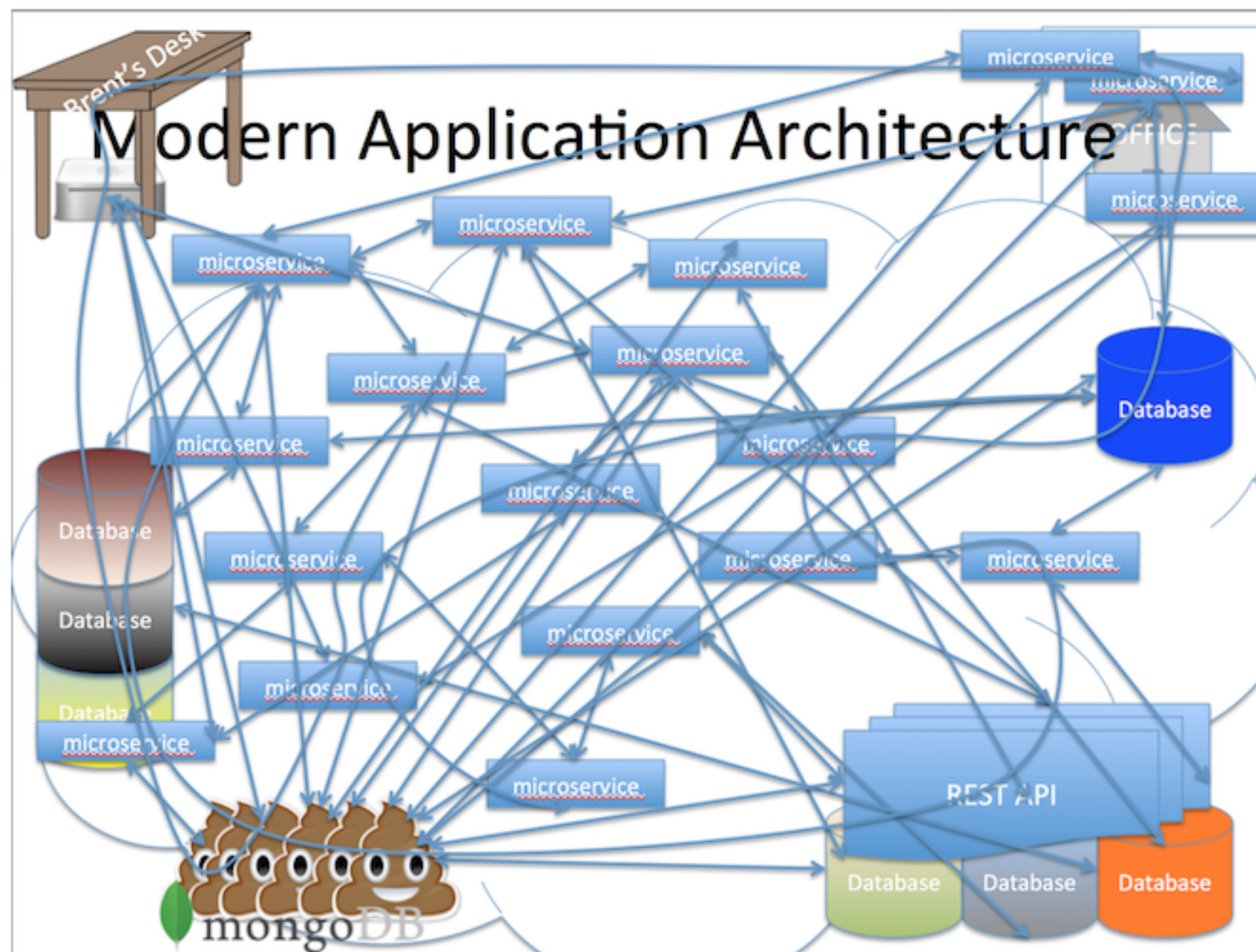
REAL TIME TRAFFIC VISUALIZATION IN A MICROSERVICES WORLD

WHO AM I?

Roberto Perez Alcolea

- » Mexican
- » Works @ Target
- » Groovy Enthusiast
- » roberto@perezalcolea.info
- » @rpalcolea

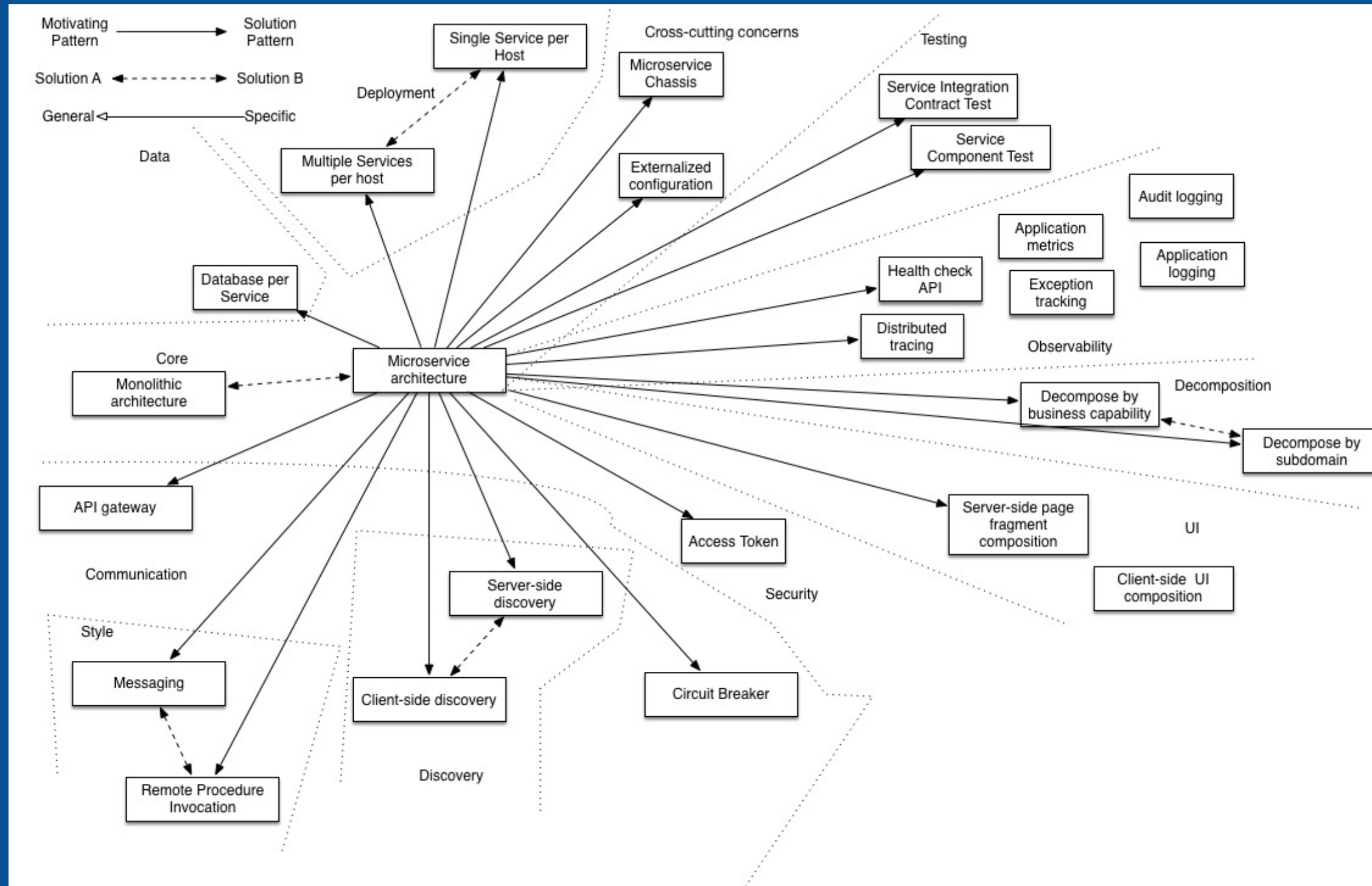
MICROSERVICES



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

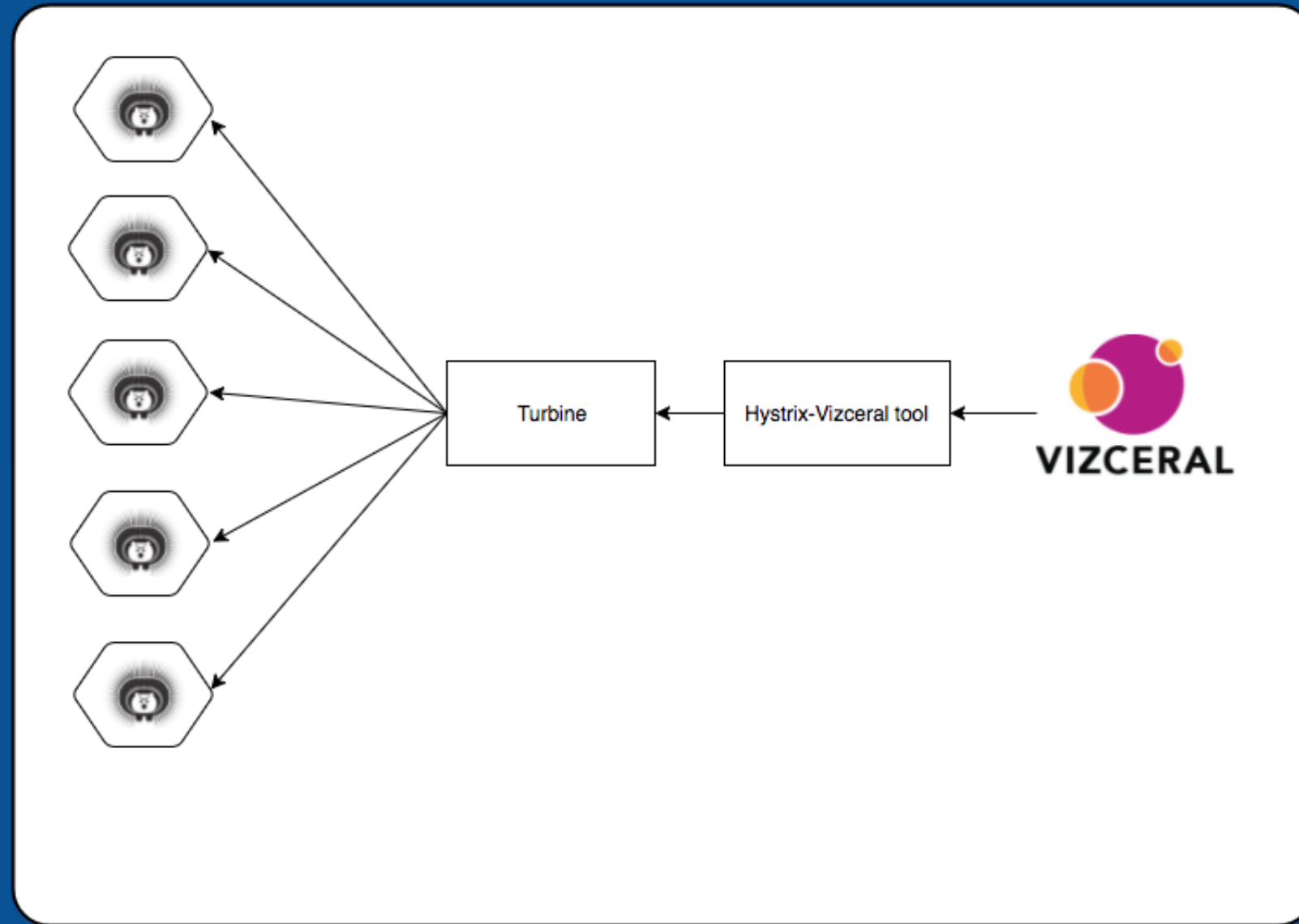
PRINCIPLES OF MICROSERVICES

- » Modeled around business domain
- » Culture of automation
- » Hide implementation details
- » Highly observable
- » Isolate failure
- » Deploy independently



TRAFFIC VISUALIZATION

Hystrix + Turbine + Vizceral



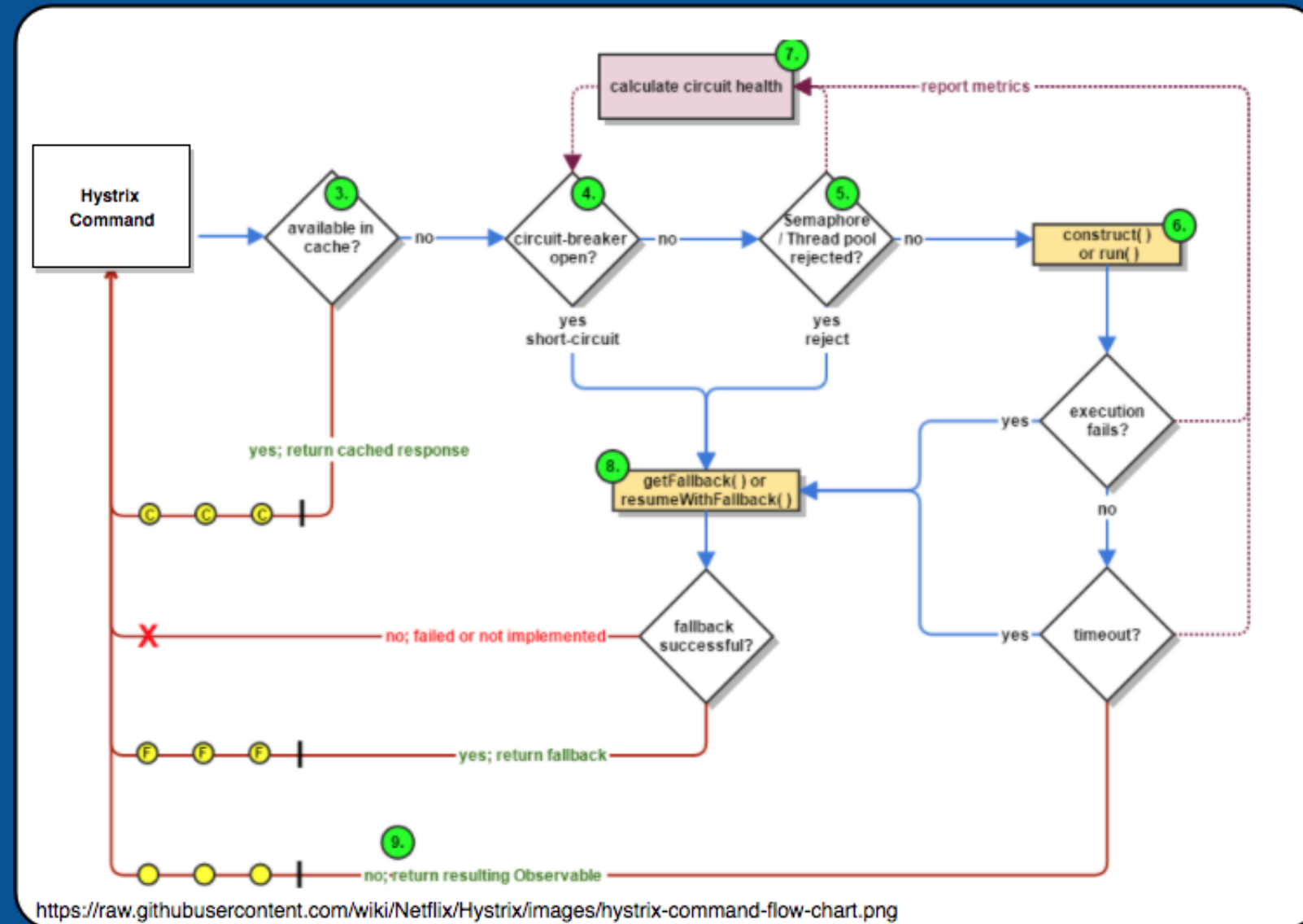
HYSTRIX

HYSTRIX

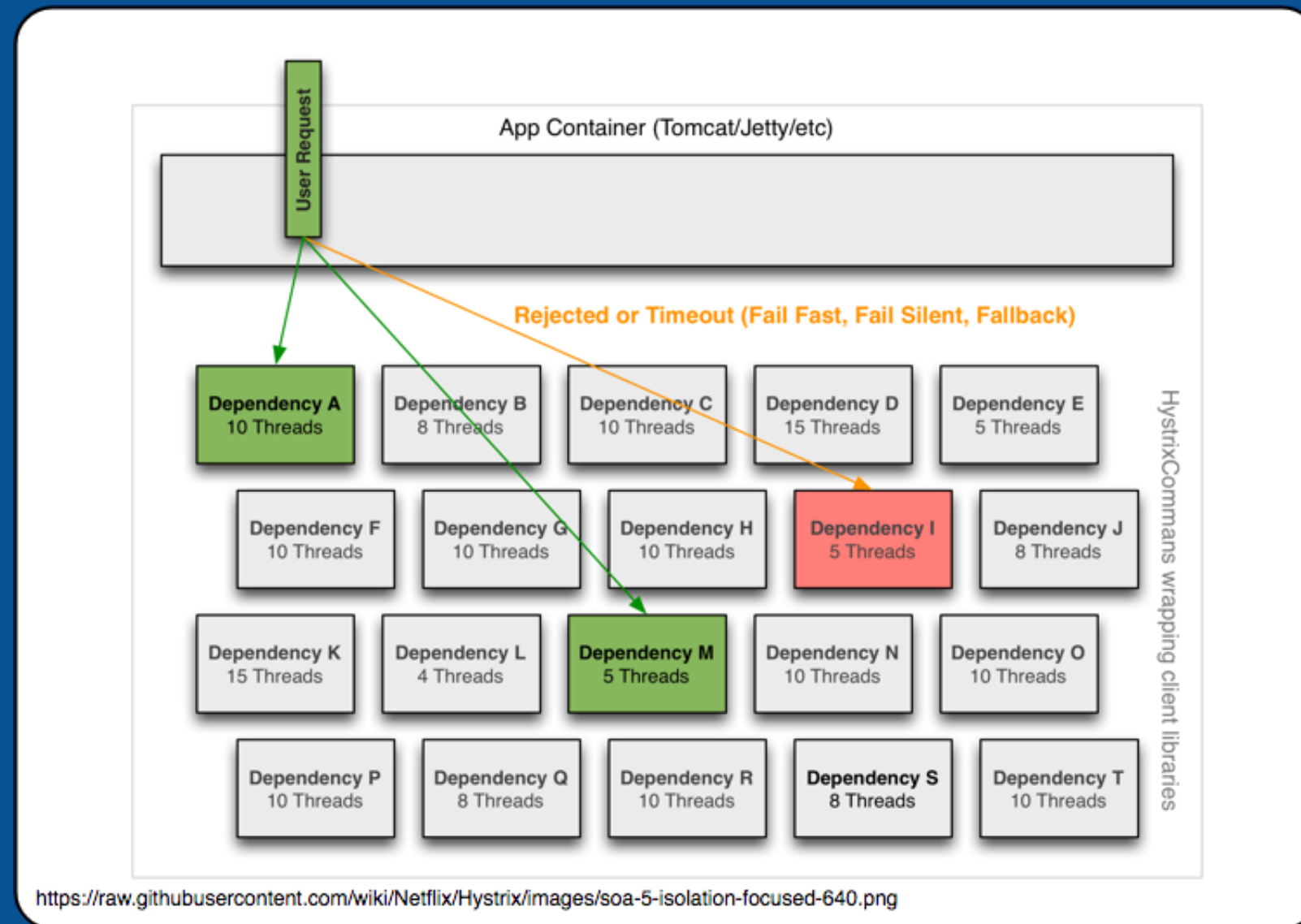
Library to help control interactions between distributed services

- » Times out slow commands
- » Rate limit number of concurrent commands
- » Circuit breaker
- » Fallback on failures
- » Real-Time dashboard for visibility

HYSTRIX FLOW



ISOLATION



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,
  },
}
```

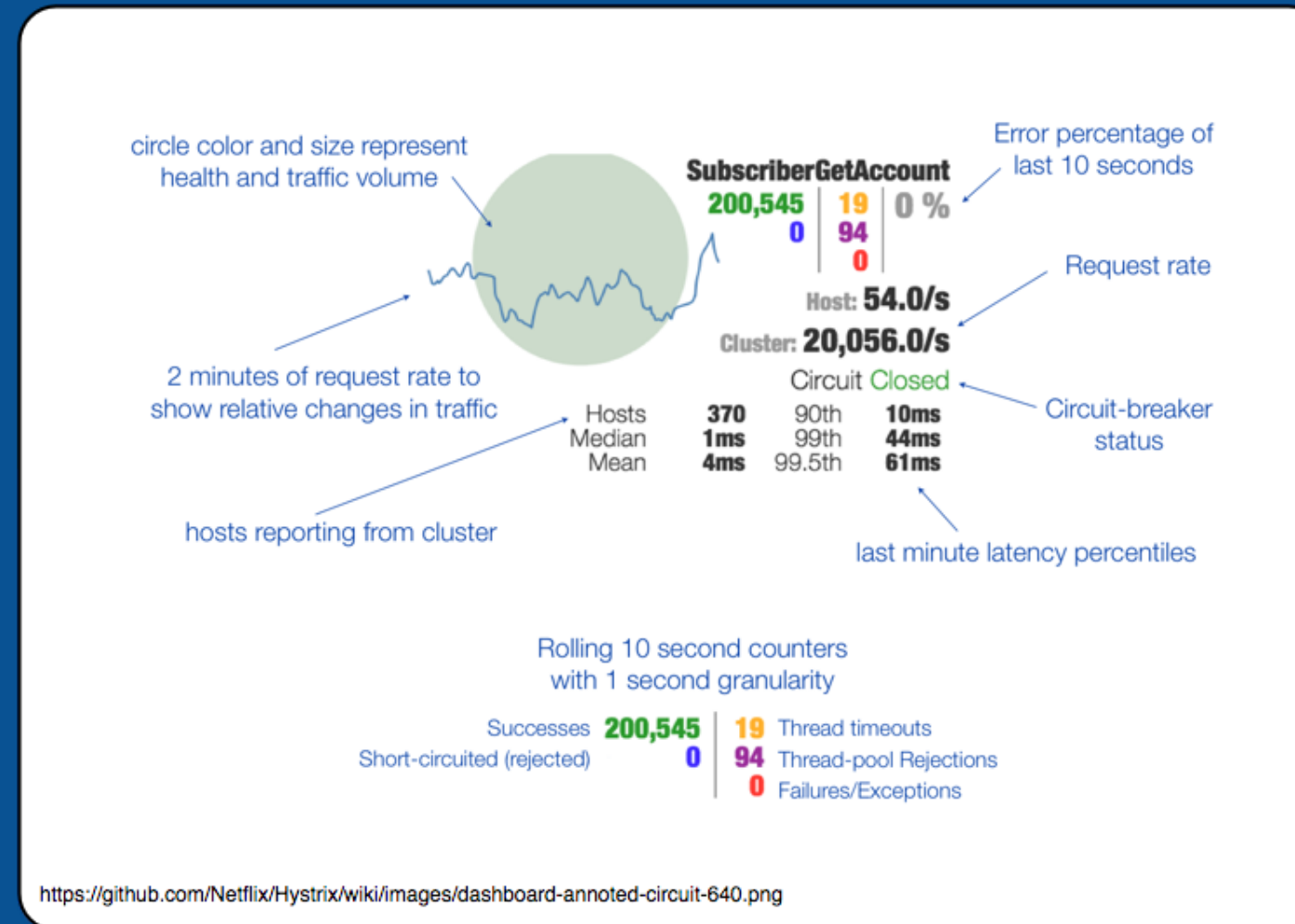
HYSTRIX COMMAND METRICS

```
{
  "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

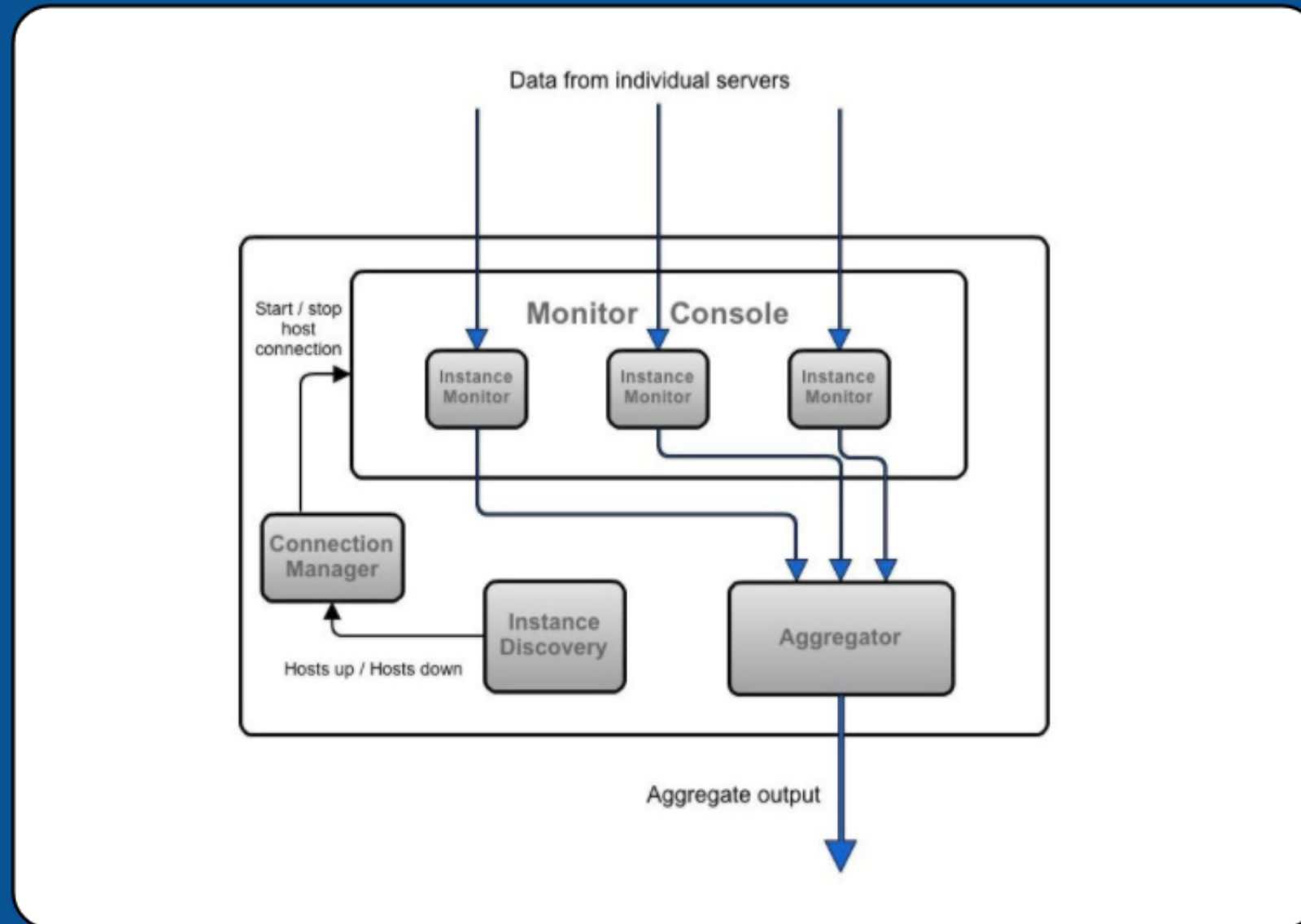
HYSTRIX DASHBOARD



TURBINE

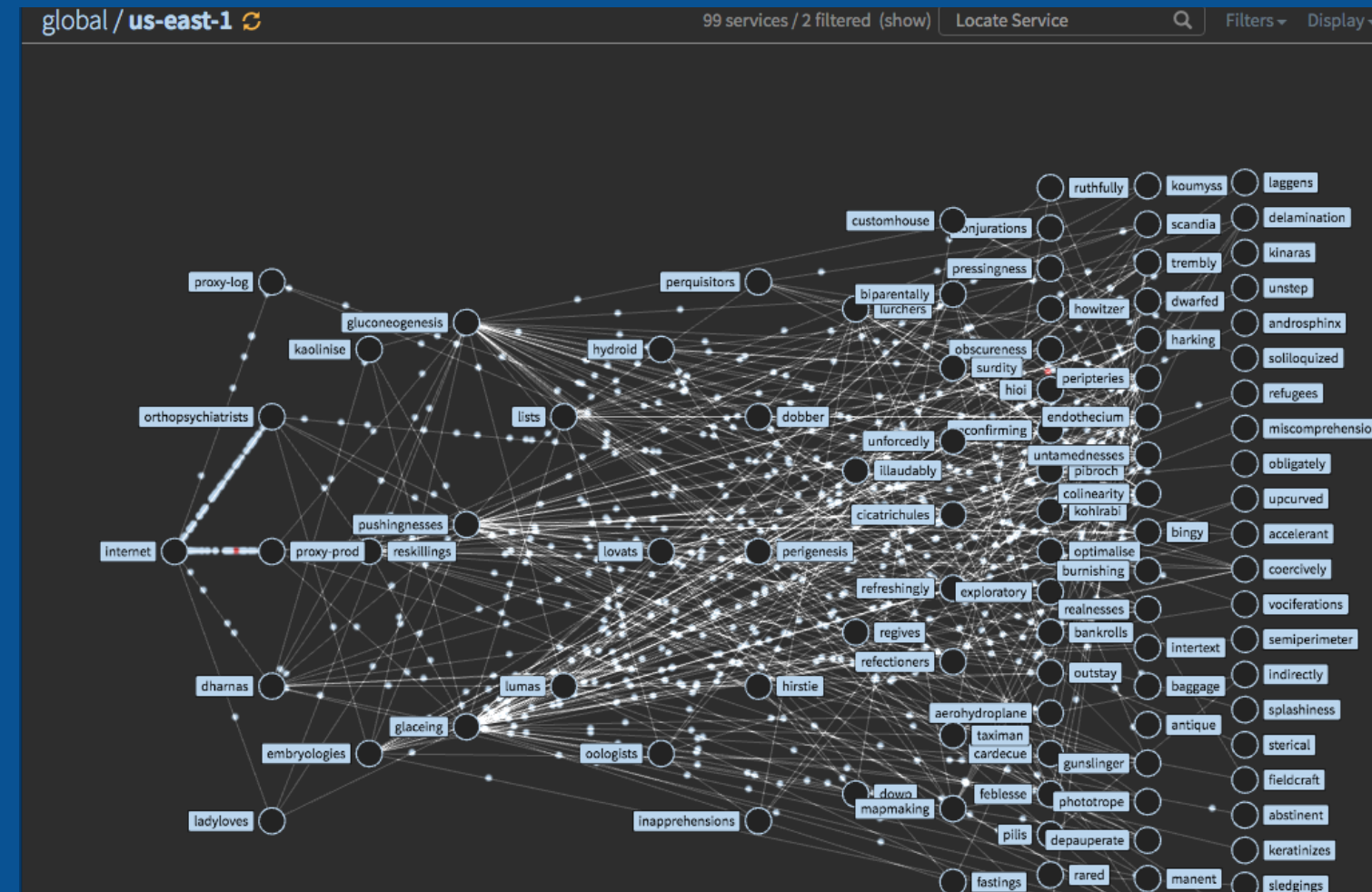
- » Aggregates streams of Server-Sent Event (SSE) JSON data into a single stream
- » Clusters
- » Instance discovery

TURBINE (HOW IT WORKS)



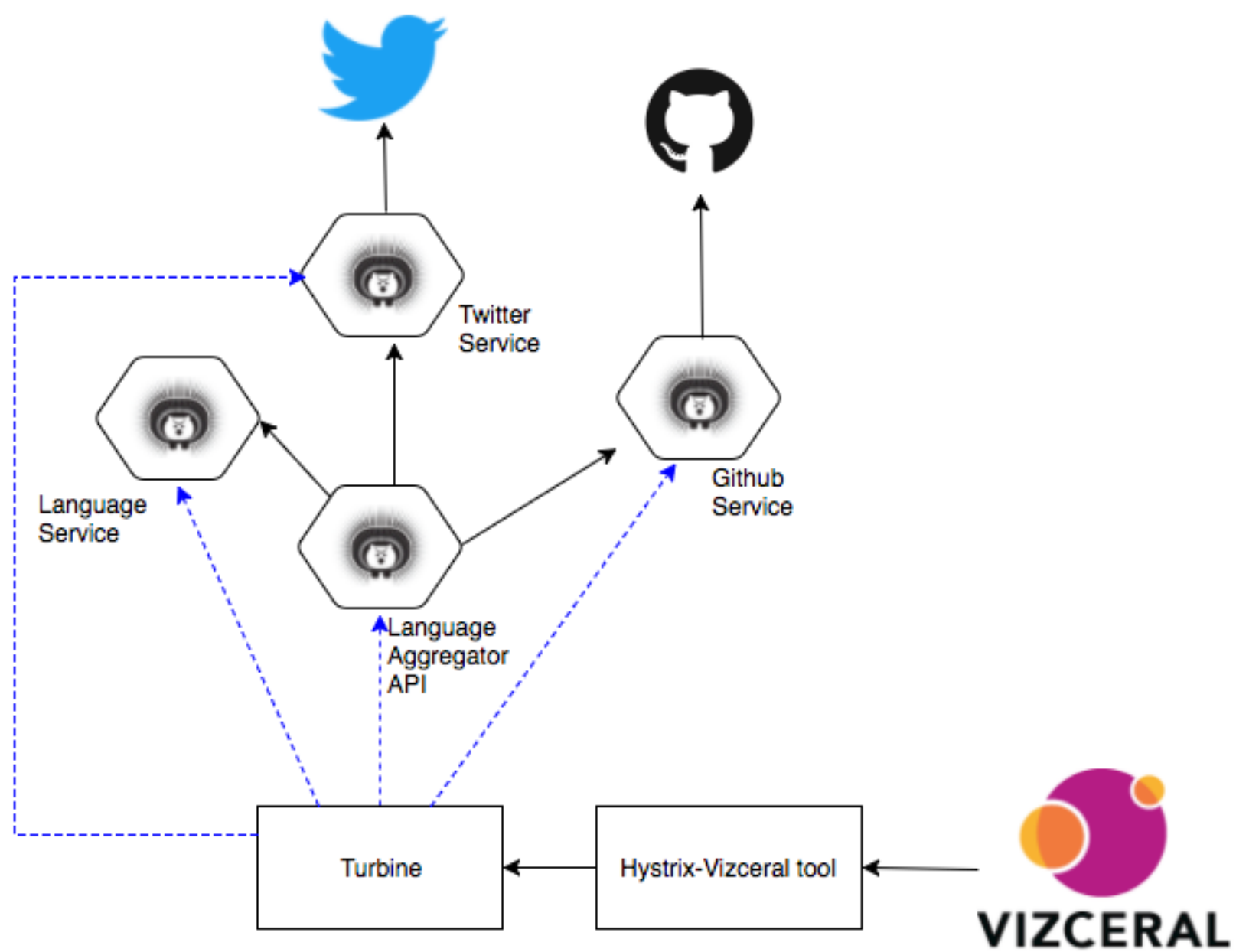
VIZCERAL

Vizceral is a tool from Netflix to visualize traffic between components



PUTTING ALL TOGETHER

DEMO



THANK

YOU