

LIFERAY & HYSTRIX

Florian Assmus

Managing Consultant @ PRODYNA AG

ABOUT ME

MAY I INTRODUCE MYSELF

- Florian Aßmus
- Chief Architect for Development and Operations
- Joined PRODYNAG in 2007



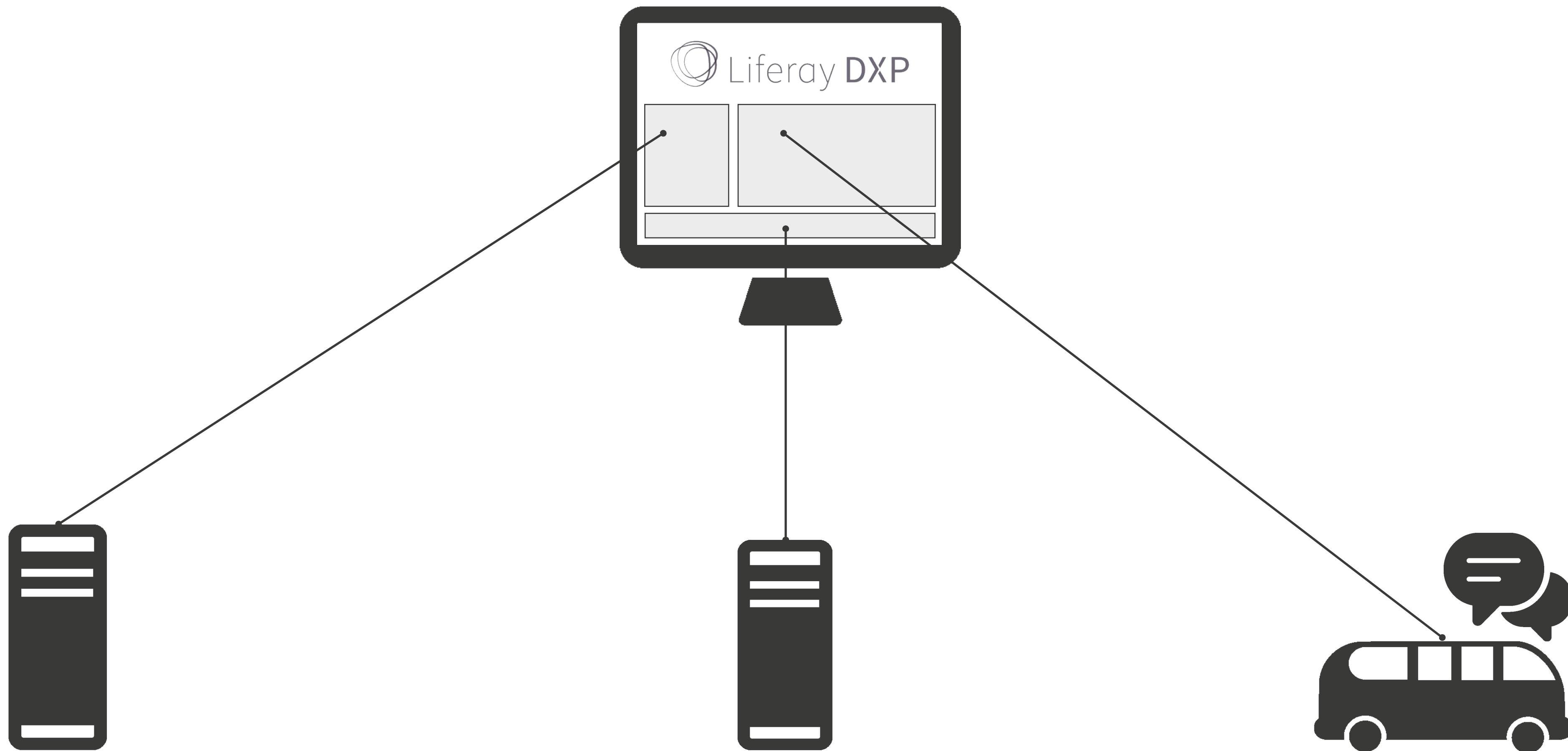


LIFERAY®



WHERE IS THE PROBLEM?

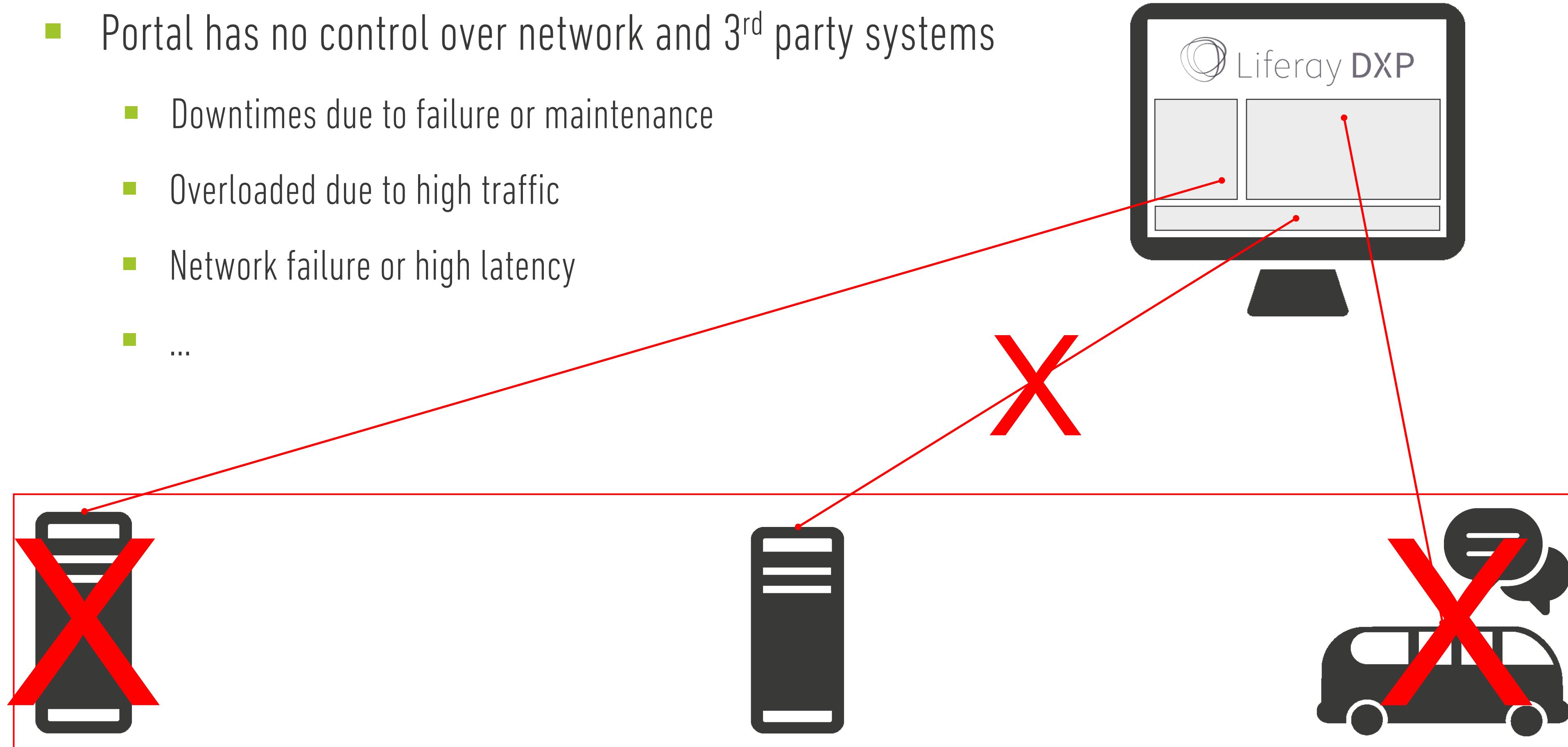
LIFERAY & HYSTRIX



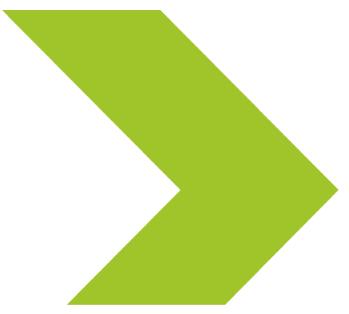
WHERE IS THE PROBLEM?

LIFERAY & HYSTRIX

- Portal has no control over network and 3rd party systems
 - Downtimes due to failure or maintenance
 - Overloaded due to high traffic
 - Network failure or high latency
 - ...



THE NETFLIX APPROACH

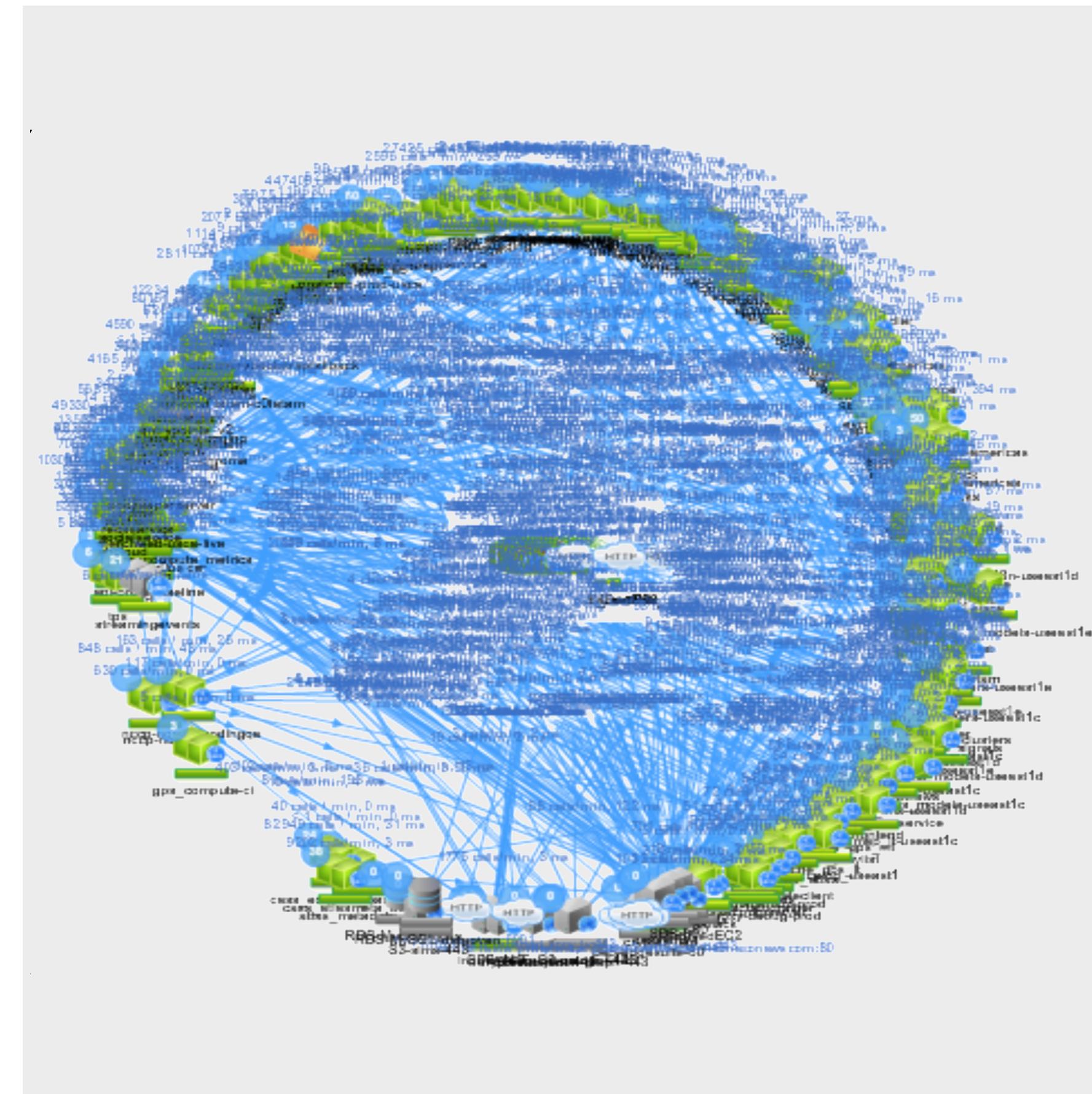


LIFERAY & HYSTRIX

- Globally distributed service-oriented architecture
 - Thousands of servers and services
 - Billion requests a day

- Fault tolerance is a requirement, not a feature!

Tseitlin, A.: „Resiliency through failure – Netflix’s Approach to Extreme Availability in the Cloud“, QCon NY, 2013





THE NETFLIX APPROACH

LIFERAY & HYSTRIX

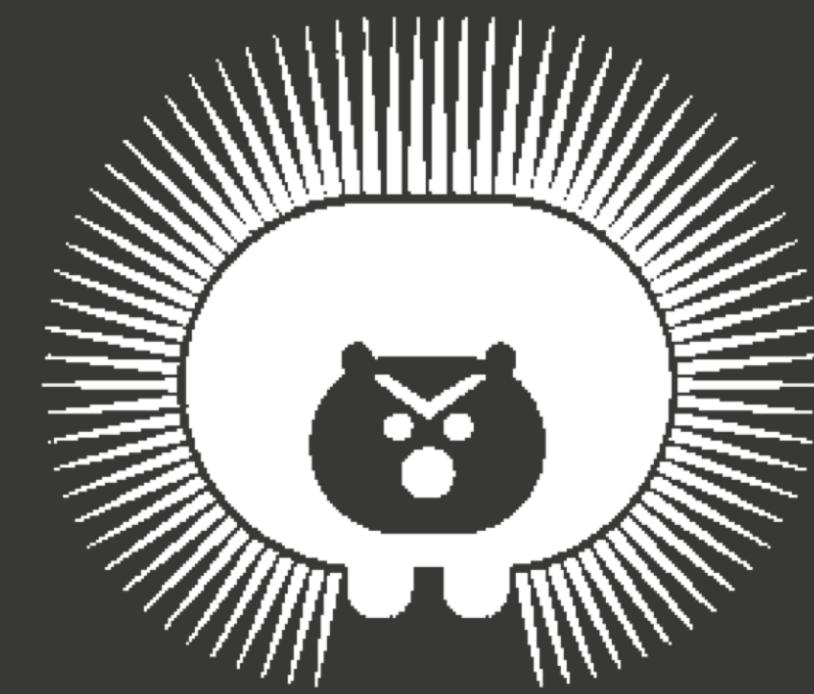
- Isolate 3rd party service interaction
 - Always use timeouts when waiting for a response
 - Limit the number of outstanding requests
 - Make use of isolation patterns (e.g. circuit breaker)
- Fail fast and rapidly recover
- Fallback and degrade gracefully

REALIZATION OF THIS APPROACH INVOLVES...

LIFERAY & HYSTRIX

- Highly complex threading model
 - Multiple Thread-Pools
 - Semaphores
 - ...
- Network timeouts and retries
- Isolation techniques
 - Bulkhead
 - Swimlane
 - Circuit breaker





HYSTRIX
DEFEND YOUR APP



HYSTRIX OVERVIEW

LIFERAY & HYSTRIX

- Simple Java library with minimal dependencies
- Build on RxJava an implementation of Reactive Extensions for the JVM
- Wraps any call to external systems in a separate command
- Commands are typically executed in a separate thread

<https://github.com/Netflix/Hystrix>

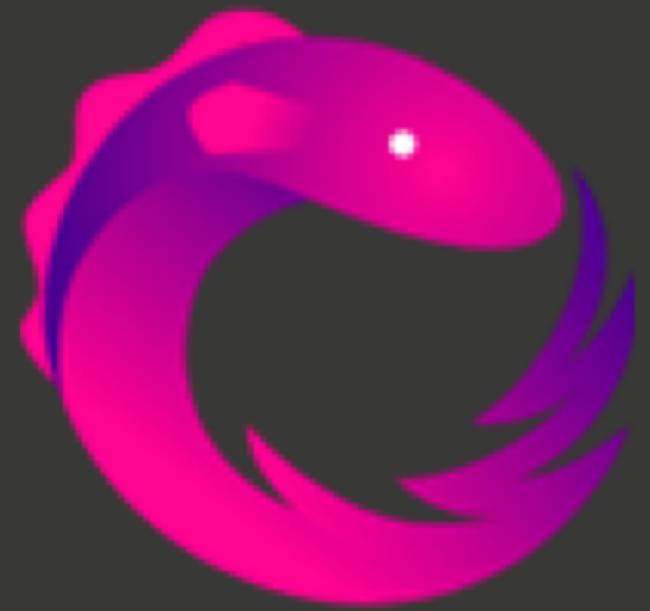


HYSTRIX OVERVIEW

LIFERAY & HYSTRIX

- Each command can be individually configured
 - Timeout
 - Queue size
 - Circuit breaker threshold
 - ...
- Fallback logic when a request fails, is rejected, times-out, or short-circuits
- Monitoring metrics and configuration changes in near real-time

<https://github.com/Netflix/Hystrix>

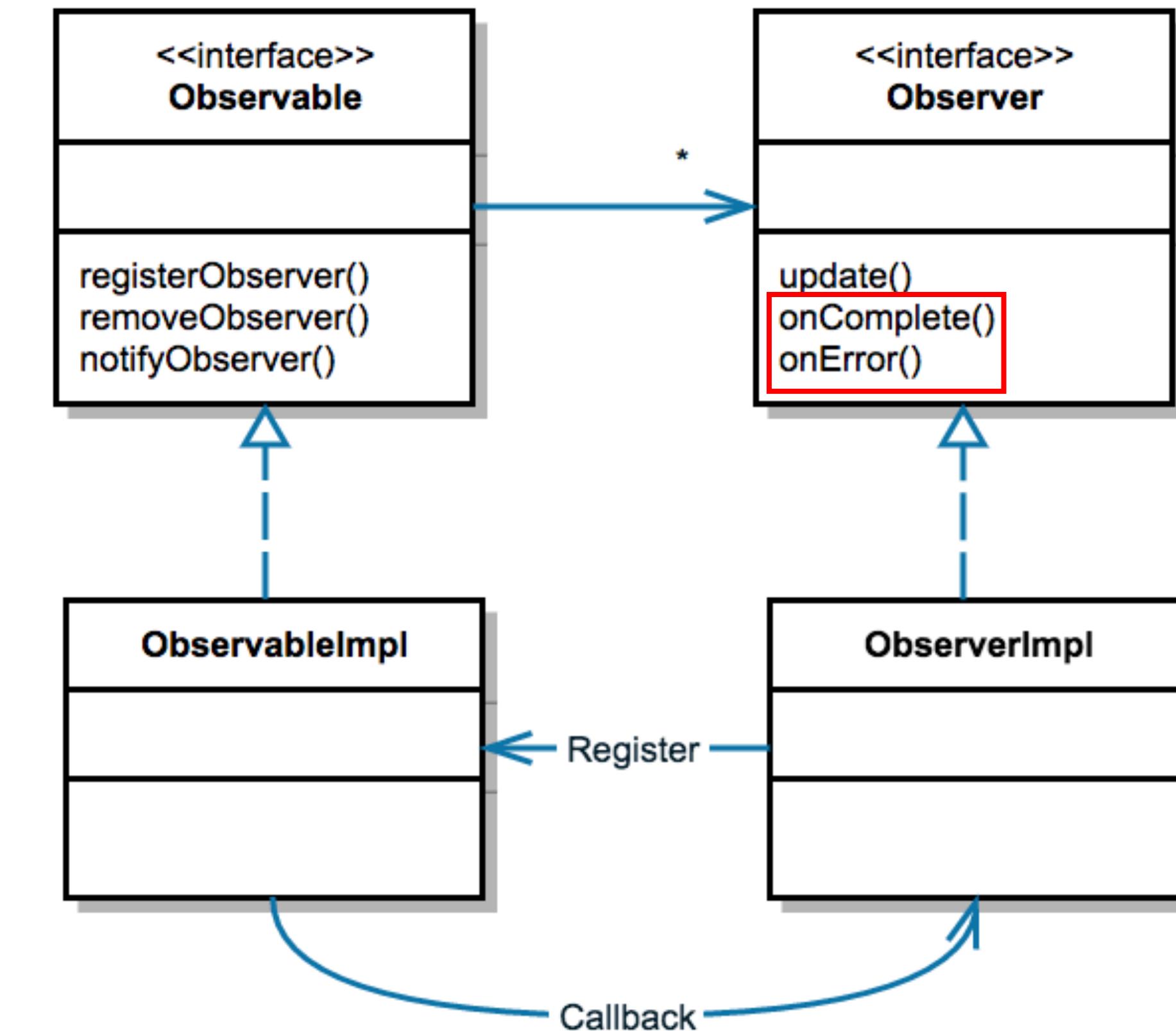


ReactiveX

RXJAVA

LIFERAY & HYSTRIX

- Library for composing asynchronous and event-based programs
- Extends the observer pattern to support sequences of data/events
- Provides operators that allow you to compose sequences
 - Filter, select, transform, combine, and compose
- Observable becomes “push” equivalent to Iterable



<https://github.com/ReactiveX>

RXJAVA

LIFERAY & HYSTRIX

```
getDataFromLocalMemory()  
    .skip(10)  
    .take(5)  
    .map({ s -> return s + " transformed" })  
    .forEach({ println "next => " + it })
```

Iterable

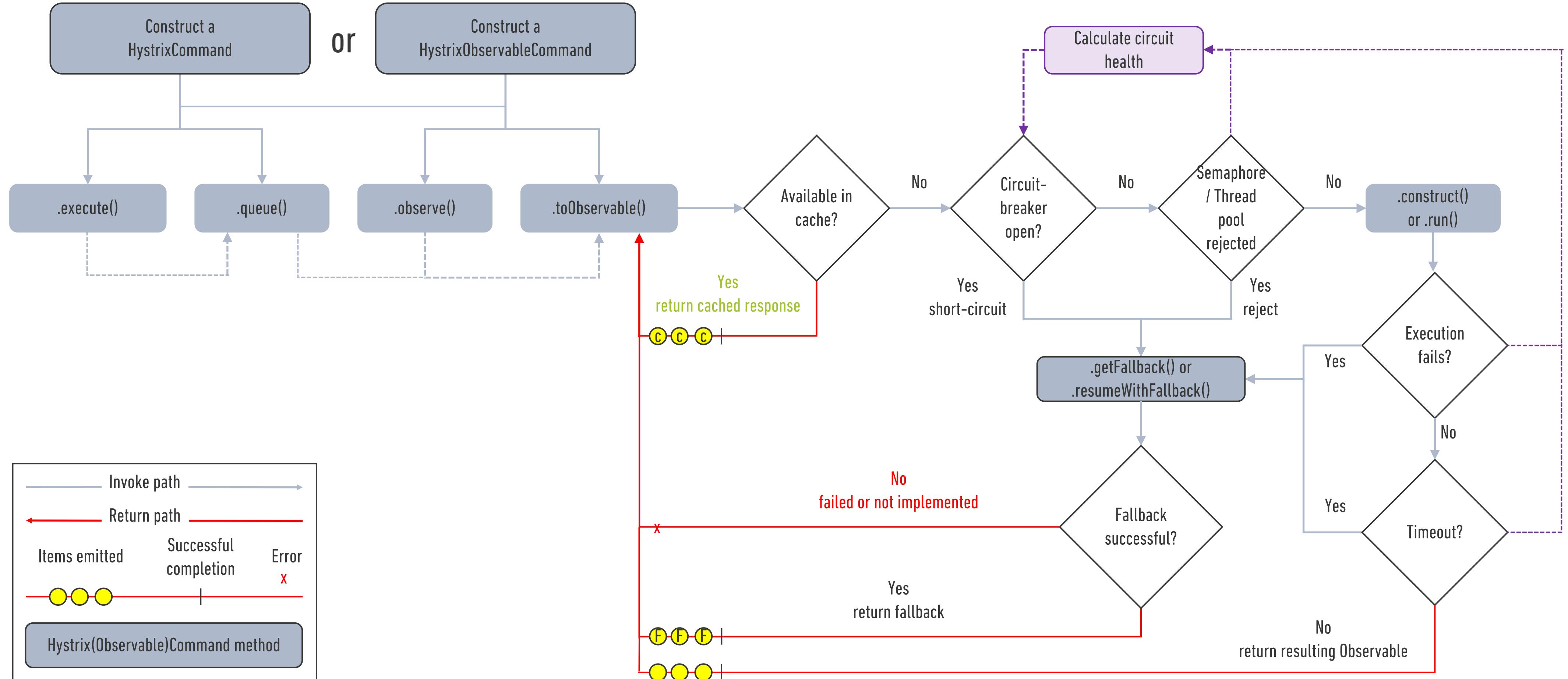
```
getDataFromNetwork()  
    .skip(10)  
    .take(5)  
    .map({ s -> return s + " transformed" })  
    .subscribe({ println "onNext => " + it  
})
```

Observable

- Pulls values from producer
- Blocks until values arrive
- Producer pushes values
- Values arrive whenever available
- Can be synchronously or asynchronously

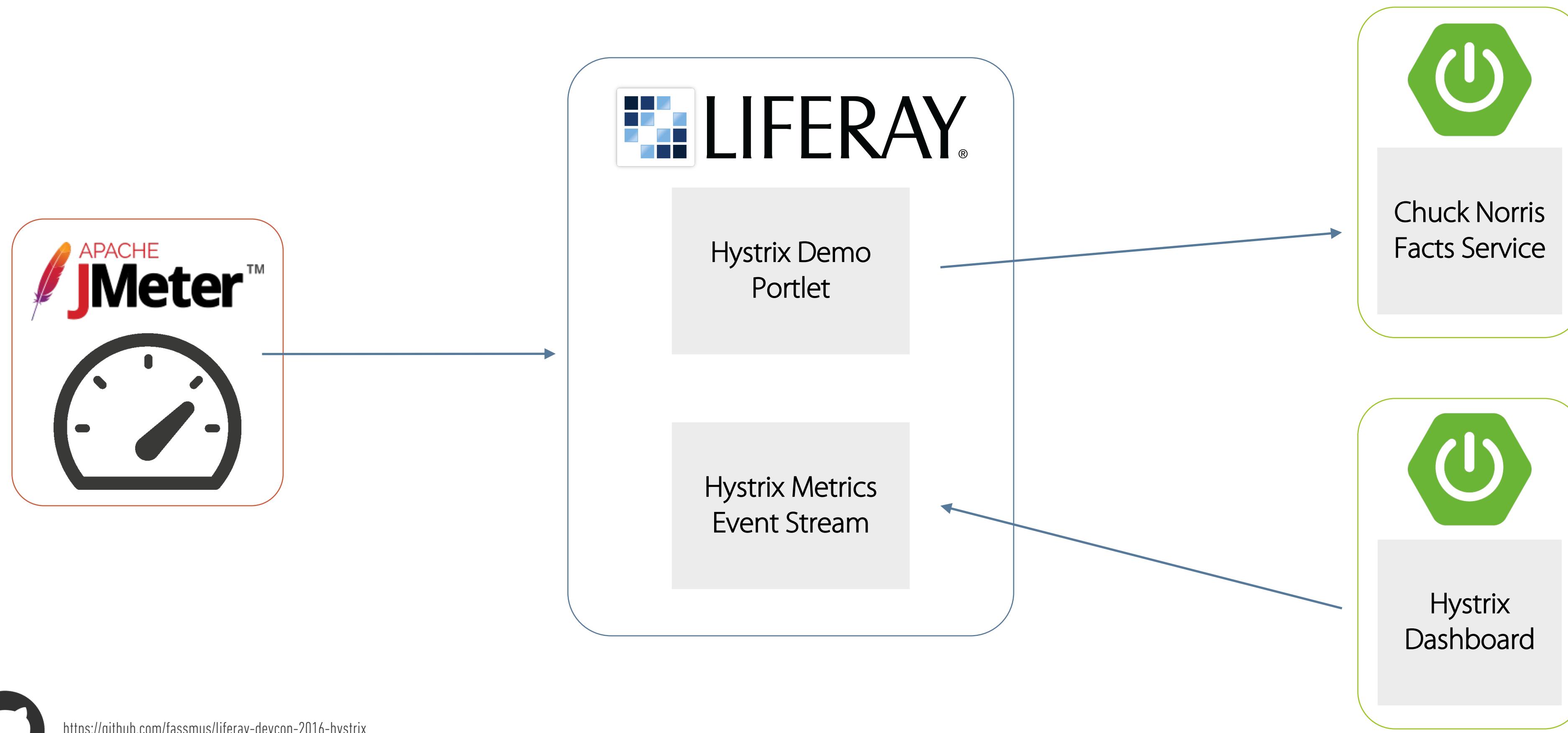
HYSTRIX COMMAND

LIFERAY & HYSTRIX



DEMO SETUP

LIFERAY & HYSTRIX

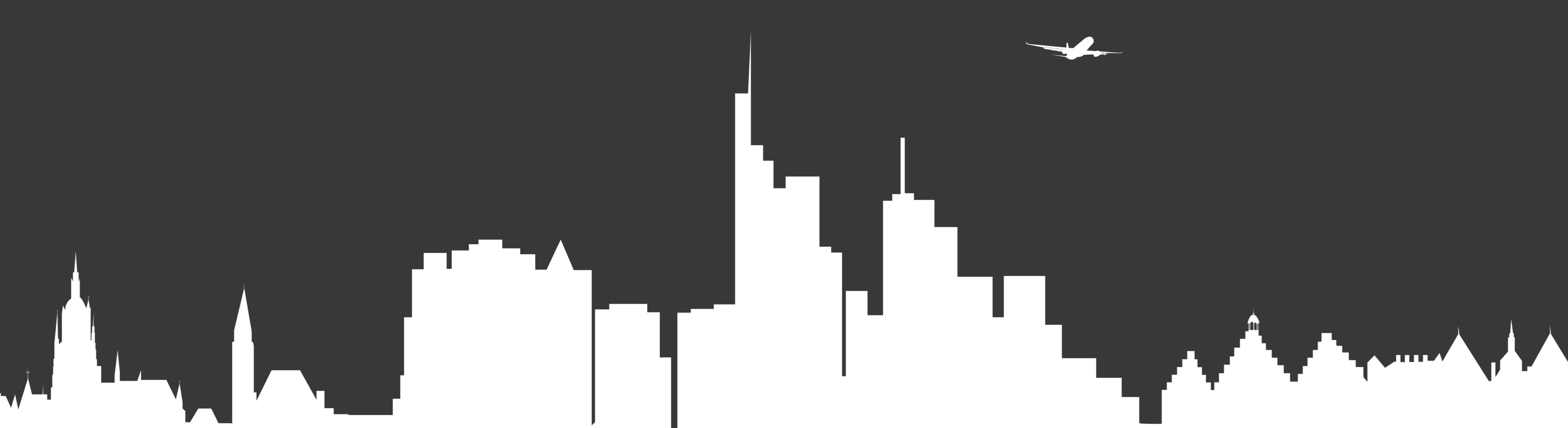


<https://github.com/fassmus/liferay-devcon-2016-hystrix>



ANY QUESTIONS?

VISIT US AT OUR PRODYN^A BOOTH



PRODYNA IN YOUR REGION

FLORIAN AßMUS – PRODYNA ESCHBORN

- PRODYNA AG
- Ludwig-Erhard-Straße 12-14 65760 Eschborn
- T +49 69 597 724 - 0 F +49 69 597 724 - 700 M +49 176 178 70 155
- florian.assmus@prodyna.com prodyna.com





CONTACT US!



+49 69 597 724 - 0



info@prodyna.com



Facebook/prodynaag



Twitter/prodynaag