MICROSERVICES ON NODE: FROM ZERO TO HERO

ERICH DE SOUZA OLIVEIRA
CTO - WINNIN.COM
TWITTER: @OLIVEIRA_ERICH
GITHUB:ERICHOLIVEIRA
WWW.ERICHOLIVEIRA.COM

ABOUT ME

✦FROM BRAZIL

✦CTO WINNIN (VIDEO PLATFORM)

✦9 YEARS DEVELOPMENT EXPERIENCE

MICROSERVICES ON NODE: FROM ZERO TO HERO

- ✦MONOLITHIC ARCHITECTURE PROBLEMS
- ✦MICROSERVICES DEFINITION
- ✦MICROSERVICES ADVANTAGES, BENEFITS AND PROBLEMS
- ✦NODE
- ✦LIVE CODING

MONOLITHIC APPLICATIONS

- ALL SERVICES RUNNING IN THE SAME PROCESS
- SHARED RESOURCES

SHARED RESOURCES

✦INFINITE LOOP = 💀

✦DATABASE LOCK = 💀

✦MEMORY LEAK = 💀

✦IN NODE A SINGLE NON-TREATED EXCEPTION = 💀

NO SHARED RESOURCES

MONOLITHIC APPLICATIONS

- ✦HARD TO CHANGE
- ✦HARD TO DEPLOY
- ✦HARD TO MONITOR

REACTIVE MANIFESTO

4 PATTERNS TO ACHIEVE SCALABILITY AND “MODERN” REQUIREMENTS (LOW RESPONSE TIME, 100% UPTIME, LOTS OF DATA)

WWW.REACTIVEMANIFETO.ORG

RESPONSIVE

RESPONSIVE SYSTEMS FOCUS ON PROVIDING RAPID AND CONSISTENT RESPONSE TIMES, ESTABLISHING RELIABLE UPPER BOUNDS SO THEY DELIVER A CONSISTENT QUALITY OF SERVICE. THIS WILL IMPROVE YOUR SYSTEM *USABILITY* AND *UTILITY*.

RESILIENT

THE SYSTEM STAYS RESPONSIVE IN THE FACE OF FAILURE.

RESILIENCE IS ACHIEVED BY REPLICATION, CONTAINMENT, ISOLATION AND DELEGATION.

FAILURES ARE CONTAINED WITHIN EACH COMPONENT, ISOLATING COMPONENTS FROM EACH OTHER AND THEREBY ENSURING THAT PARTS OF THE SYSTEM CAN FAIL AND RECOVER WITHOUT COMPROMISING THE SYSTEM AS A WHOLE.

THIS WILL IMPROVE YOUR SYSTEM *AVAILABILITY*.

BIG CONCERN FOR MONOLITHIC APPLICATIONS (SHARED RESOURCES)

ELASTIC

THE SYSTEM STAYS RESPONSIVE UNDER VARYING WORKLOAD.

THIS IMPLIES DESIGNS THAT HAVE NO CONTENTION POINTS OR CENTRAL BOTTLENECKS, RESULTING IN THE ABILITY TO SHARD OR REPLICATE COMPONENTS AND DISTRIBUTE INPUTS AMONG THEM.

REACTIVE SYSTEMS PROVIDES RELEVANT LIVE PERFORMANCE MEASURES.

MESSAGE PASSING

REACTIVE SYSTEMS RELY ON ASYNCHRONOUS MESSAGE-PASSING TO ESTABLISH A BOUNDARY BETWEEN COMPONENTS THAT ENSURES LOOSE COUPLING, ISOLATION AND LOCATION TRANSPARENCY.

THIS BOUNDARY ALSO PROVIDES THE MEANS TO DELEGATE FAILURES AS MESSAGES.

A REACTIVE SYSTEM MUST USE SAME CONSTRUCTS AND SEMANTICS ACROSS A CLUSTER OR WITHIN A SINGLE HOST.

MICROSERVICES DEFINITION

"THE MICROSERVICE ARCHITECTURAL STYLE IS AN APPROACH TO DEVELOPING A SINGLE APPLICATION AS A SUITE OF SMALL SERVICES, EACH RUNNING IN ITS OWN PROCESS AND COMMUNICATING WITH LIGHTWEIGHT MECHANISMS, OFTEN AN HTTP RESOURCE API. THESE SERVICES ARE BUILT AROUND BUSINESS CAPABILITIES AND INDEPENDENTLY DEPLOYABLE BY FULLY AUTOMATED DEPLOYMENT MACHINERY. THERE IS A BARE MINIMUM OF CENTRALIZED MANAGEMENT OF THESE SERVICES, WHICH MAY BE WRITTEN IN DIFFERENT PROGRAMMING LANGUAGES AND USE DIFFERENT DATA STORAGE TECHNOLOGIES." - MARTIN FOWLER

KEY CONCEPTS

- ✦ SMALL SERVICES
- ✦ LIGHTWEIGHT COMMUNICATION
- ✦ INDEPENDENTLY DEPLOYABLE
- ✦ MINIMUM MANAGEMENT CENTRALIZATION

MICROSERVICES: BENEFITS

- ✦NO SHARED RESOURCES (MORE LOVE ❤️)
- ✦INDEPENDENCY
- ✦EASY TO CHANGE AND TEST
- ✦EASY TO DEPLOY
- ✦EASY TO MONITOR

THE PROBLEMS

- COMMUNICATION THROUGH NETWORK
- SERVICE DISCOVERY (HOW A SERVICE CAN FIND ANOTHER SERVICE?)
- MIGHT BE HARD TO DECIDE HOW/WHEN TO DECOUPLE SERVICES

MICRO SERVICES ON NODE

- ROUTER (MESSAGE PASSAGE - RECEIVE/DELIVER ASYNC MESSAGES)
- ALWAYS ASYNC
- IMMUTABILITY
- MODULES/NAMESPACES
- PLUGINS/ADD-ONS (PERFORMANCE MEASUREMENT)

STUDIO JS

✦INSPIRED BY AKKA / ERLANG / ACTORS MODEL

✦STARTED IN 2014

✦HOW CAN I CREATE A SYSTEM AND DISTRIBUTE AS IT GROW?

✦NEW API IN 2015 (FOCUS IN DEVELOPER HAPPINESS)

✦A POWERFUL AND EXTENSIBLE CORE

STUDIO JS: PLUGINS

- ✦CLUSTERING
- ✦USAGE METRICS (STATSD)
- ✦TIMEOUTS,RETRIES,TESTS, RICH ERRORS...
- ✦YOU CAN CREATE YOUR OWN WITH FEW LINES OF CODE

STUDIO JS: CLUSTER

- PLUGIN FOR CLUSTERIZATION.
- STUDIO FOCUS ON MAKE YOUR SERVICES FOLLOW A DISTRIBUTABLE ARCHITECTURE. STUDIO-CLUSTER EFFECTIVELY DISTRIBUTE YOUR CODE.
- 2 PILLARS:
  - SERVICE DISCOVERY (NETWORK BROADCAST/REDIS/ PUB-SUB)
  - REMOTE PROCEDURE CALL (PEER-TO-PEER WEBSOCKET)

ES6

- GENERATORS
- PROXIES
- CLASSES

STUDIO JS: HANDS-ON

ES6 => ES5 MAP

- https://gist.github.com/ericholiveira/0eea43535cd78efbdd4a69bf795e9952

NEXT STEPS

- ✦SERVICES VERSIONING
- ✦BETTER DOCS
- ✦PLUGINS
- ✦SPREAD THE WORD

GITHUB

- GITHUB.COM/ERICHOLIVEIRA/STUDIO
- GITHUB.COM/ERICHOLIVEIRA/STUDIO-CLUSTER

THANK YOU

ERICH DE SOUZA OLIVEIRA
CTO - WINNIN.COM
TWITTER: @OLIVEIRA_ERICH
GITHUB:ERICHOLIVEIRA
WWW.ERICHOLIVEIRA.COM
GITHUB.COM/ERICHOLIVEIRA/STUDIO