DISMANTLING THE MONOLITH

Jonathan Kaufman - @kauffecup - 2-25-15
MONOLITHIC APP
OK  SO
WHY IS THIS BETTER?
SMALL TASK
HIGHLY DECOUPLED
MODULAR
SCALABILITY
OK   SO
WHY IS THIS WORSE?
LATENCY
LOAD BALANCING
FAULT TOLERANCE
DEPLOYMENT
SERVICE

REGISTRY

HOST
PORT
NODE NAME
METADATA

CLIENT

ANY ___ SERVICES?
NETFLIX'S EUREKA

BITLY'S NSQ LOOKUPD

PLAIN OL' DNS

SENECA

SERF

SMARTSTACK

DOOZER

ETCD

CONSUL

SkyDNS

ZOOKEEPER
COMMUNICATE VIA MESSAGES
COMMUNICATE VIA MESSAGES

{
  cmd: "tweet",
  from: "kauffecup",
  text: "tweet town"
}
COMMUNICATE VIA MESSAGES

{
  cmd: "tweet",
  from: "kauffecup",
  text: "tweet town"
}
const Twitter = require('twitter');
const client = new Twitter({
    consumer_key, consumer_secret, access_token_key, access_token_secret
});

on('tweet', ({from, text}) => {
    client.post('statuses/update', {status: text});
});
ADD A DELAY

{
    cmd: "tweet",
    from: "kauffecup",
    text: "tweet town"
}
ADD A DELAY

{
  cmd: "tweet",
  from: "kauffecup",
  text: "tweet town",
  delay: 200
}
ADD A DELAY

{  
  cmd: "tweet",
  from: "kauffecup",
  text: "tweet town",
  delay: 200
}

TWITTER
{ 
  cmd: "tweet",
  from: "kauffecup",
  text: "tweet town",
  delay: 200
}
DEMO
The microservice style of architecture is not so much about building individual services so much as it is making the interactions between services reliable and failure-tolerant. [1]
REFERENCES


