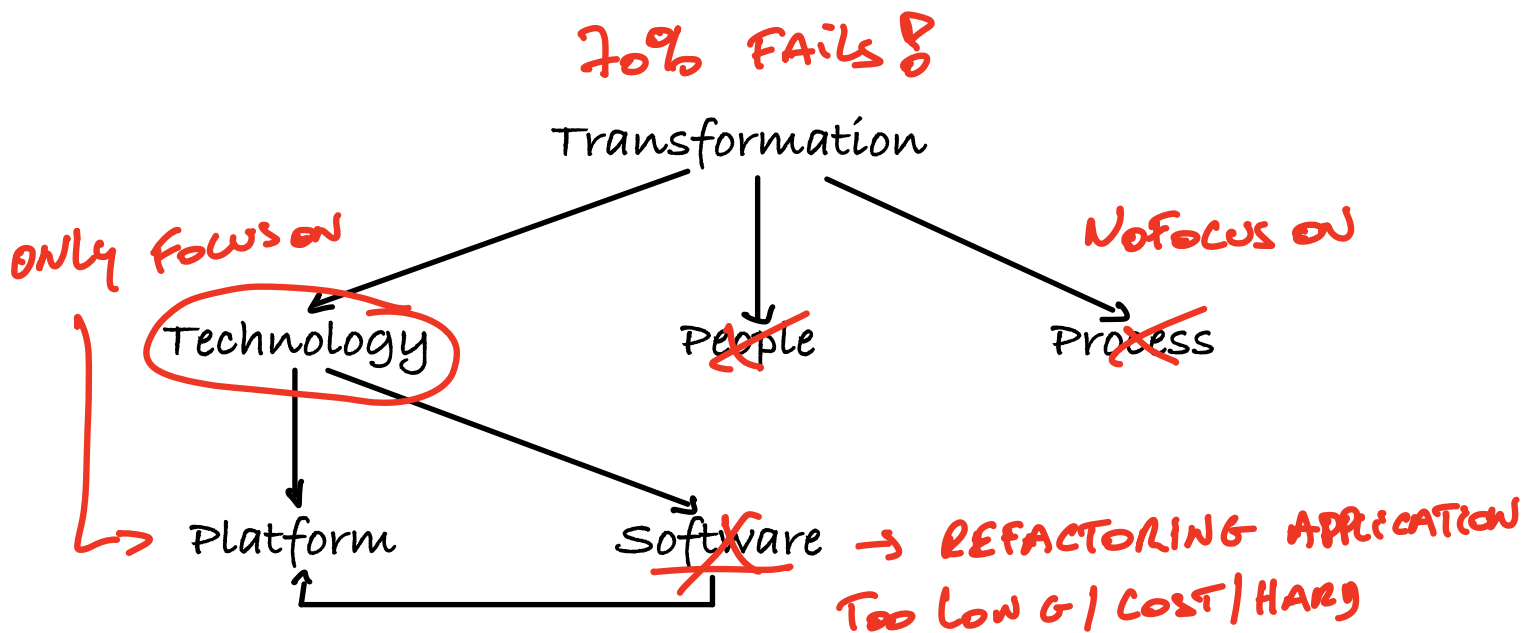


Refactoring and Modernizing at Your Own Pace

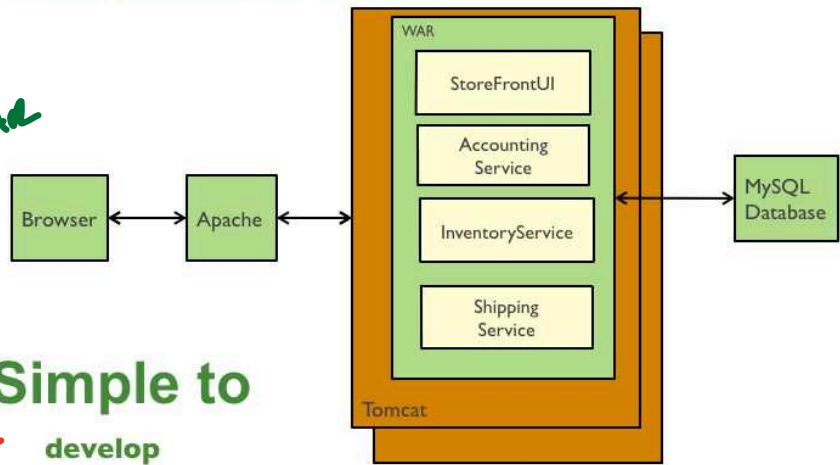
Romuald@  Red Hat



BCG FROM 30% TO 80% SUCCESS
<https://www.bcg.com/publications/2020/increasing-odds-of-success-in-digital-transformation>

Monolithic Architecture

Traditional web application architecture



- 3 PROS**
- EASY TO START WITH
 - ONE STACK ROWLAND
 - EASY TO DEPLOY

- 3 CONS**
- ALL PROS LOST WHEN CODE BASE INCREASES
 - LOW G-TEAM COMMITMENT
 - NO CONTINUOUS DELIVERY / DEPLOY

Simple to

- develop
- test
- deploy
- scale

Source: <https://microservices.io/patterns/monolithic.html>

START

Virtual Machine

BUT WITH NO BENEFITS FOR LIFE CYCLE

built as container

~~deploy~~
USING PIPELINE BUT NOT CONTINUOUS

application
~~lift and shift~~

~~HA~~ CAN'T SURVIVE A RESCHEDULING FROM A SERVICE AND SLA PERSPECTIVES

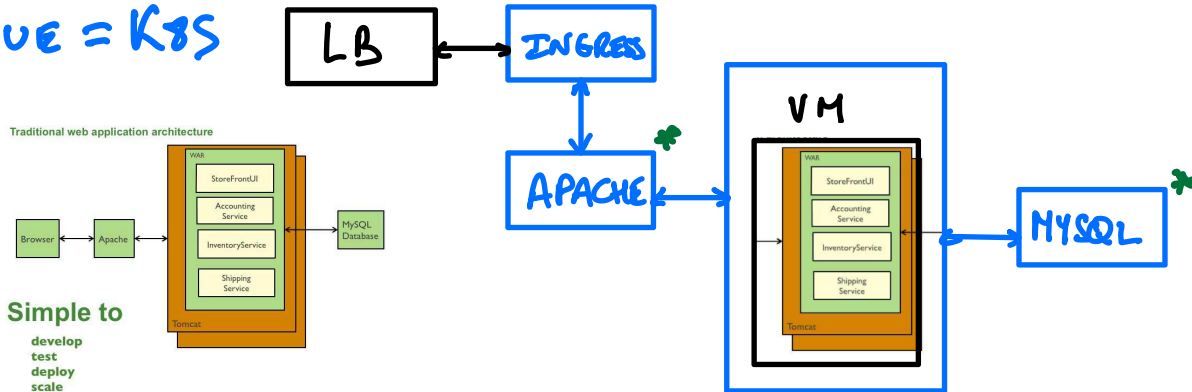
~~scaling~~
ONLY CPU/NET + MAYBE REDUCAS

Container

FINISH

OpenShift virtualization
Iteration 0

Blue = K8S



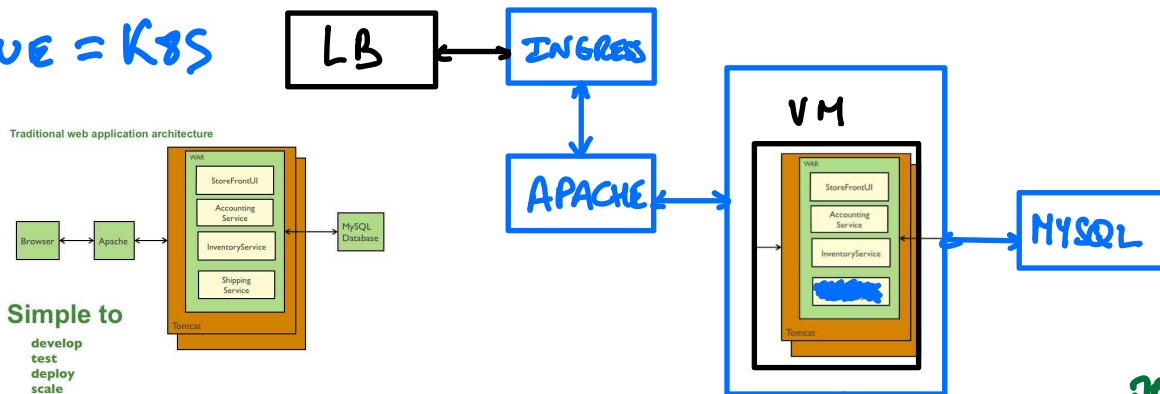
Simple to
develop
test
deploy
scale

MOVING TO UN IN K8S WITH UN FEATURE

- BEHAVE LIKE BEFORE
- ALREADY IN PLATFORM TO START REFACTORING
- DEPENDENCIES ARE NO ANYMORE*

OpenShift virtualization
Iteration 1

Blue = K8S



Simple to
develop
test
deploy
scale

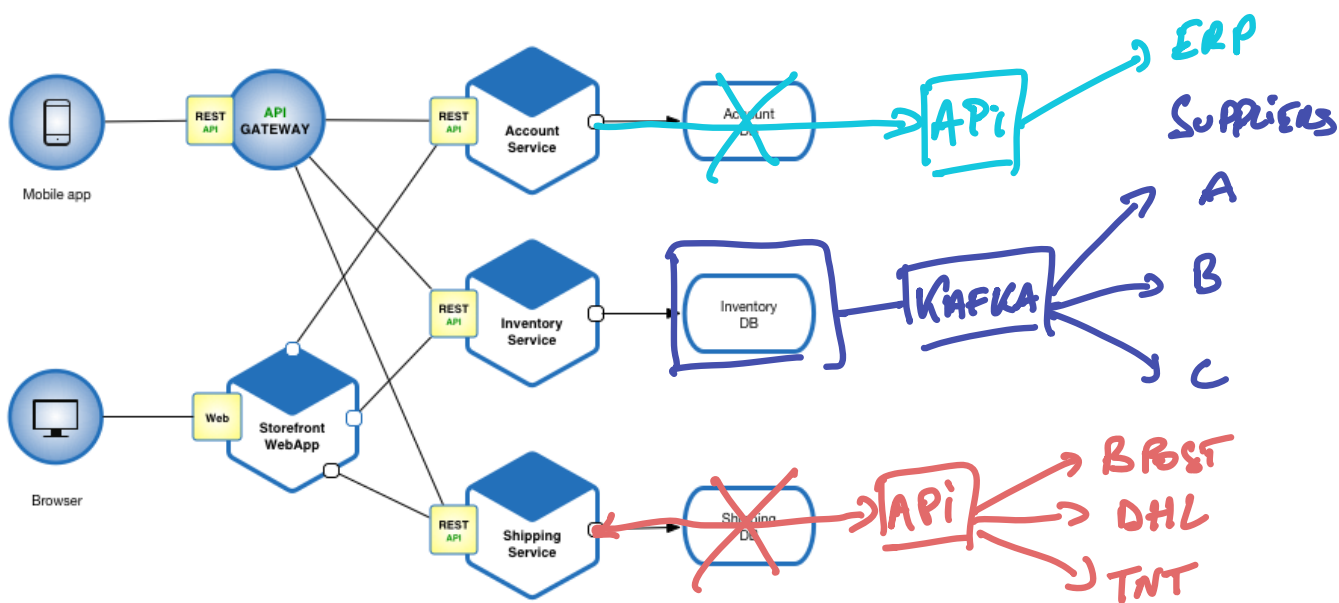
THIS COULD BE
USING INTEGRATION
SERVICES

REFACTORING ONE MODULE TO MICRO SERVICE
WITH ITS OWN DATA SERVICE*

IF SHIPPING GOES DOWN, IT DOESN'T
IMPACT THE REST

MAYBE ANOTHER
DB IS BEST *

OpenShift ~~virtualization~~
Final Iteration



ACHIEVING AN EVENT DRIVEN AND STATELESS ARCHITECTURE



Refactoring and Modernizing
at Your Own Pace