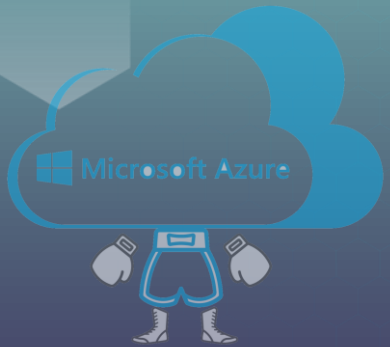
THIS IS THE WAY



AGE OF EMPIRES

A SONG OF ARCHITECTURE AND TECHNOLOGY

TABLE OF CONTENTS

01

OLD SOFTWARE ARCHITECTURE

Learn about the common software architecture before 2006

02

EMPIRES RISING

Learn what changed and enforced us to build a better products

03

PROBLEM & SOLUTION

Learn about the collision between the old software architecture and the change occurred at 2006

04

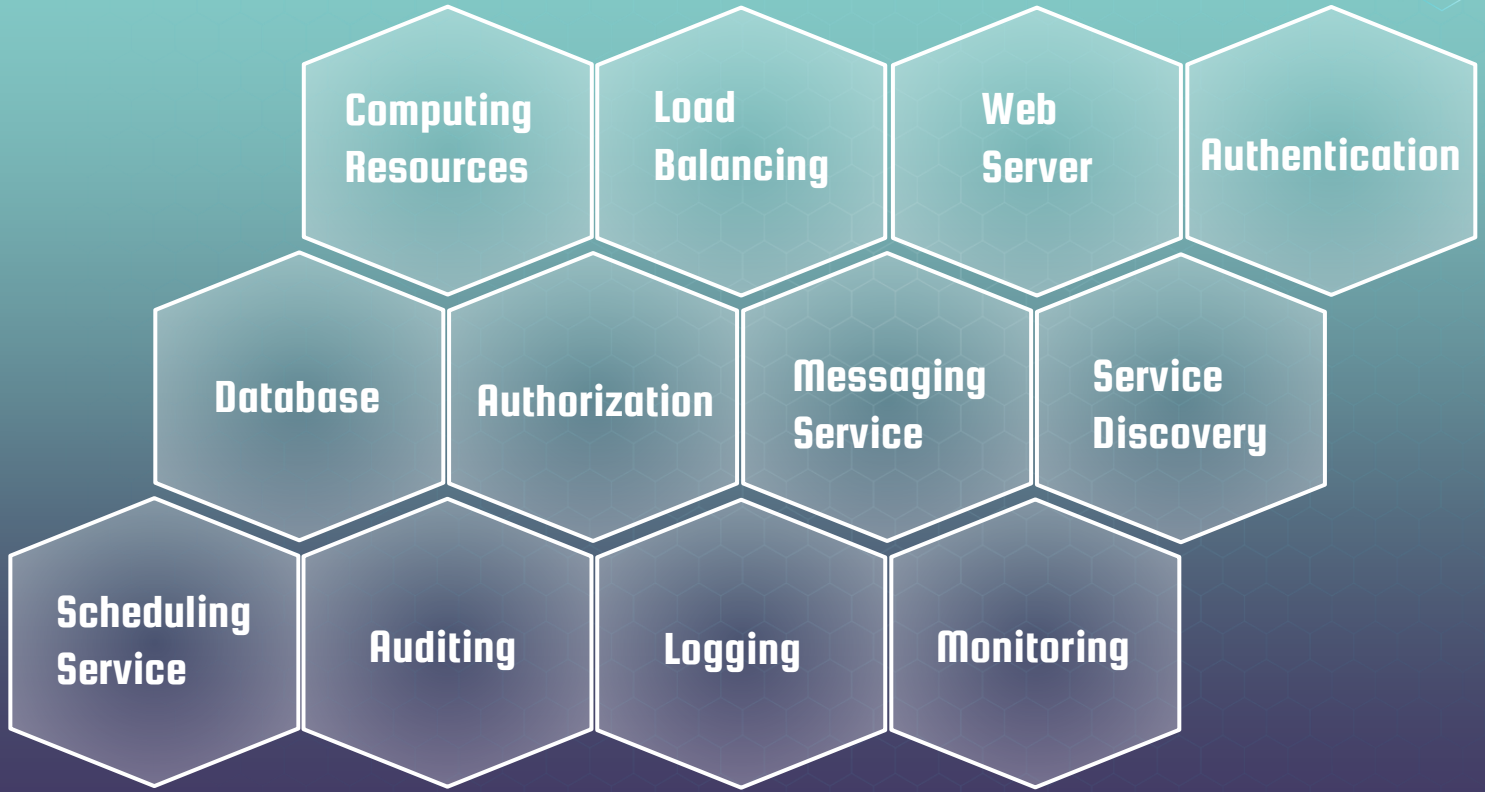
NEW SOFTWARE ARCHITECTURE

Learn about the common software architecture after 2006

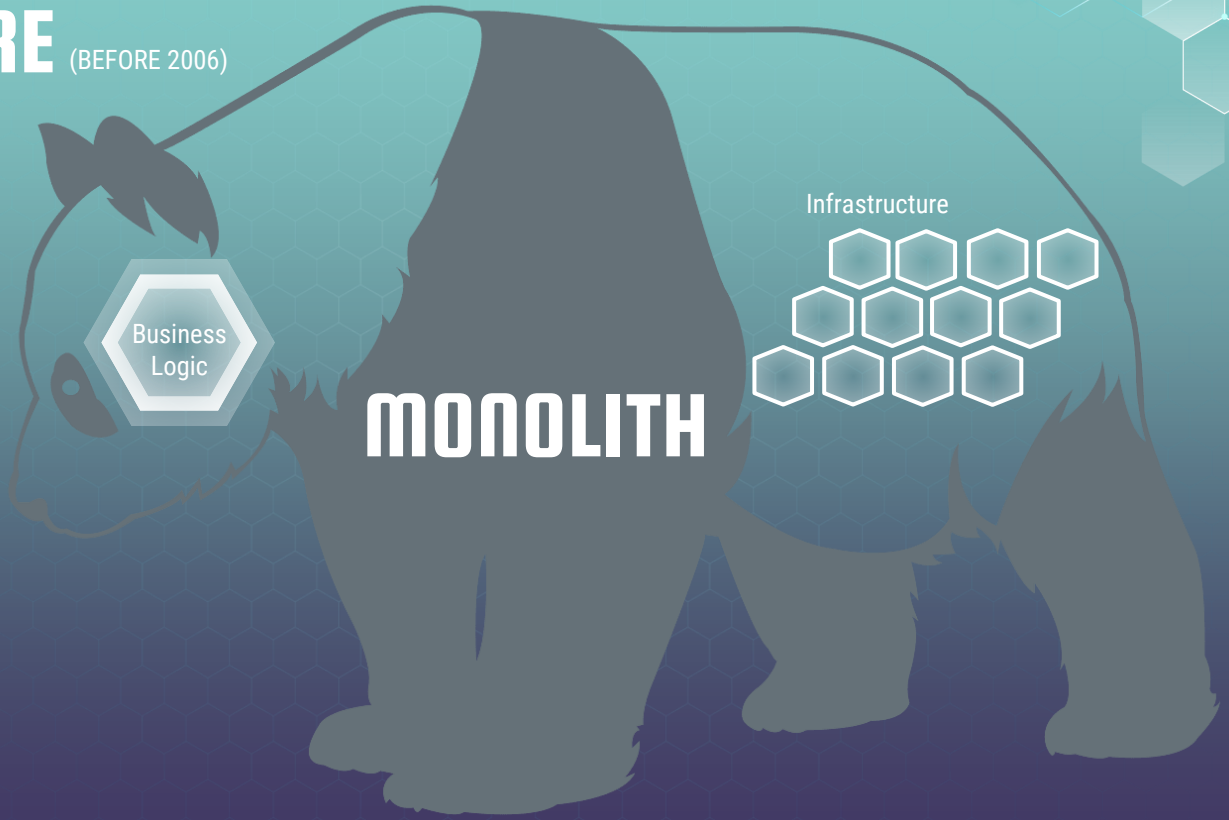
01

OLD SOFTWARE ARCHITECTURE

SOFTWARE NEEDS (BEFORE 2006)



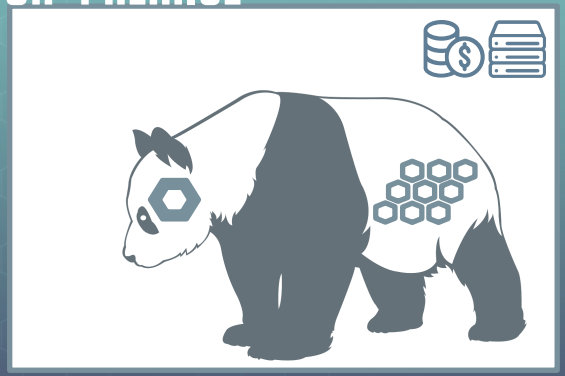
COMMON SOFTWARE ARCHITECTURE (BEFORE 2006)



SOFTWARE DELIVERY (BEFORE 2006)

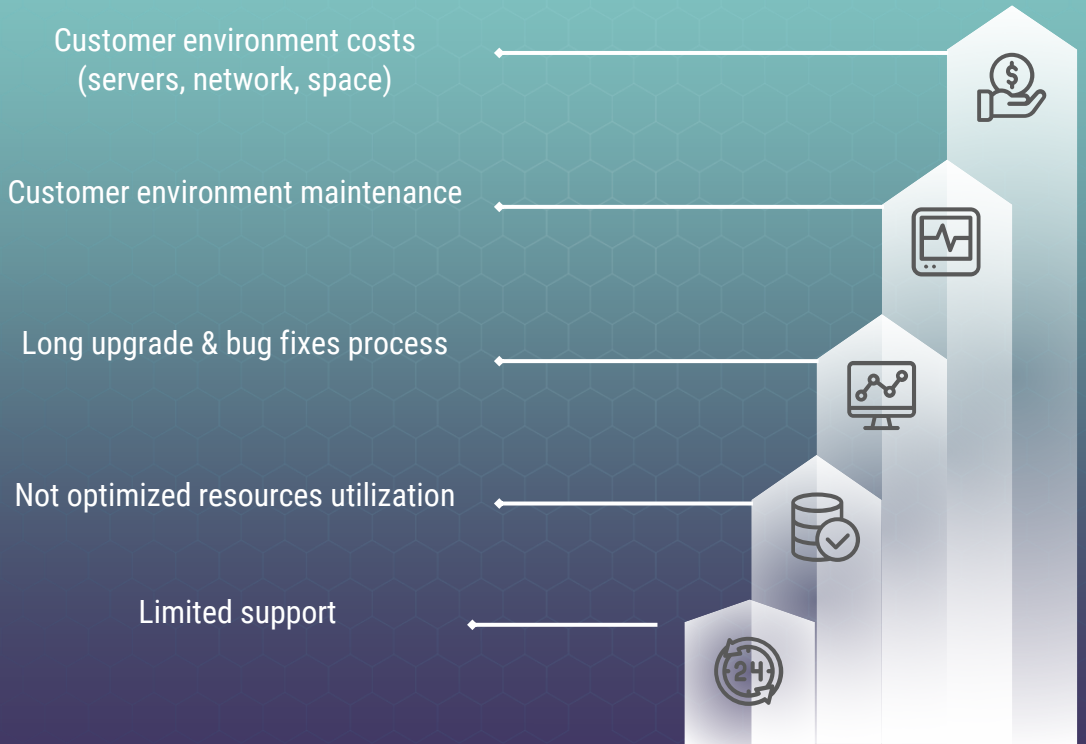
SOFTWARE

ON-PREMISE



SOFTWARE
UPDATE

On-Premise Disadvantages



02 EMPIRES RISING

THE BIG BANG

2006
Amazon
AWS

2010
Microsoft
AZURE

2013
Google
GOOGLE CLOUD PLATFORM

EMPIRES OFFERING

EMPIRES BUSINESS MODEL

PAY AS YOU GO

What you use is what you pay!

03

PROBLEM & SOLUTION

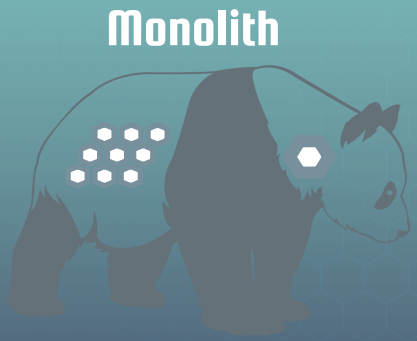
PROBLEM

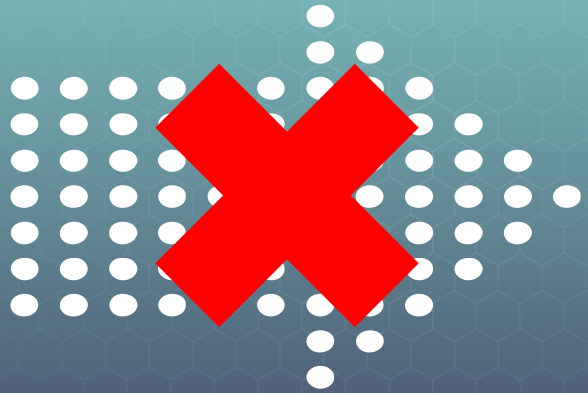
(BEFORE 2006)

Software is usually using monolith framework.
Monolith framework is heavy, it's like a toolbox, it tries to provide support & solution for any software need

Software uses ~30% of what monolith framework is offering

PROBLEM





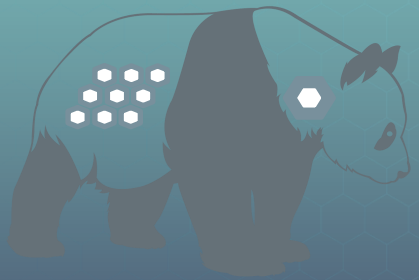
HUGE WASTE OF MONEY!!

SOLUTION

BREAK BEFORE YOU MAKE

MONOLITH TO MICRO SERVICES

Monolith



Micro Services

MICRO SERVICES CHALLANAGES

COMPLEXITY

Many individual services operate at one single project, version control, remote transactions

ORGANIZATION

Organizational change required. teams must be able to manage the entire lifecycle of service

SECURITY

Harder to maintain transaction safety, distributed communication goes wrong more likely

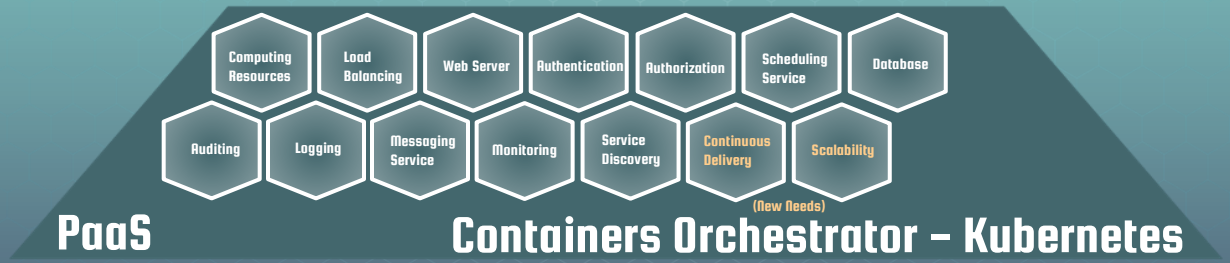
TESTING

Harder to test and monitor because of the complexity of the architecture

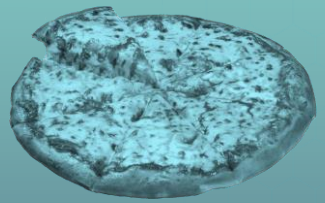
04

NEW SOFTWARE ARCHITECTURE

COMMON SOFTWARE ARCHITECTURE (2020)



EXAMPLE:
PIZZA AS A SERVICE

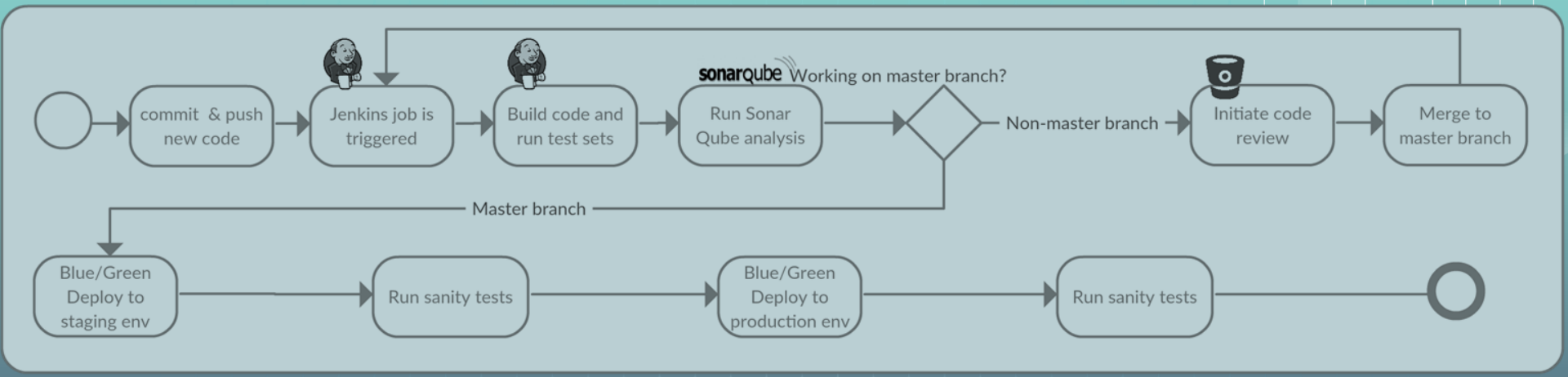


You Manage Everything

Traditional On-Premises Make at Home	Infrastructure As A Service Take And Bake	Platform As A Service Pizza Delivery	Software As A Service Dining Out
Dining Table	Dining Table	Dining Table	~~Dining Table~~
Drinks	Drinks	Drinks	~~Drinks~~
Electric/Gas	Electric/Gas	~~Electric/Gas~~	~~Electric/Gas~~
Oven	Oven	~~Oven~~	~~Oven~~
Fire	Fire	~~Fire~~	~~Fire~~
Pizza Dough	~~Pizza Dough~~	~~Pizza Dough~~	~~Pizza Dough~~
Tomato Sauce	~~Tomato Sauce~~	~~Tomato Sauce~~	~~Tomato Sauce~~
Toppings	~~Toppings~~	~~Toppings~~	~~Toppings~~
Cheese	~~Cheese~~	~~Cheese~~	~~Cheese~~
FLEXIBILITY / OPTIONS LEAST MOST	FLEXIBILITY / OPTIONS LEAST MOST	FLEXIBILITY / OPTIONS LEAST MOST	FLEXIBILITY / OPTIONS LEAST MOST

Vendor Manages Everything

SOFTWARE DELIVERY (2020)



Prerequisites

- Lifecycle per micro service
- Versioning mechanism managed by Jenkins
- Git repository per micro service
- IaaS for testing environment
- Releases collector for full deployment

4 Levels of software testing

- Unit/component tests
- Integration tests
- System tests
- Acceptance tests

Achievements

- Code reviews between DEVs (Bitbucket)
- Code contribution (Bitbucket)
- Code coverage by testing (SonarQube)
- Improve code quality (SonarQube)
- SaaS Readiness
- Continues Integration
- Continues Delivery
- Testing environment for QA

“IMAGINEER – ONE WHO HAS THE ABILITY TO IMAGINE AND THE SKILLS TO MAKE IT HAPPENED”

MICHAEL ELKABETZ

Jan 2020

https://www.shevanova.com

https://linkedin.com/in/michael-elkabetz

https://twitter.com/whyike

https://www.isyncya.com

https://www.appiryon.com

https://www.shevanova.com/peerfecto

http://www.cityeye.co.il