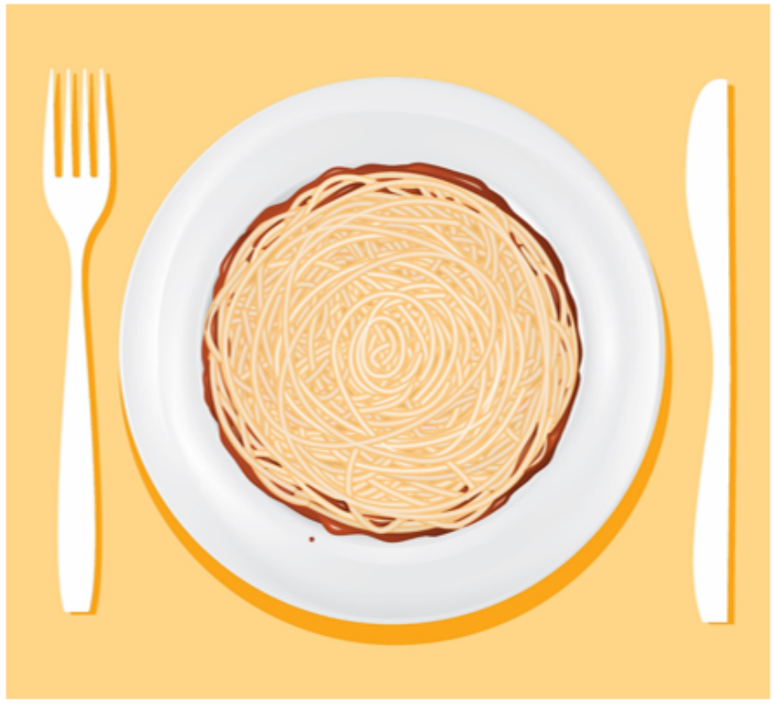
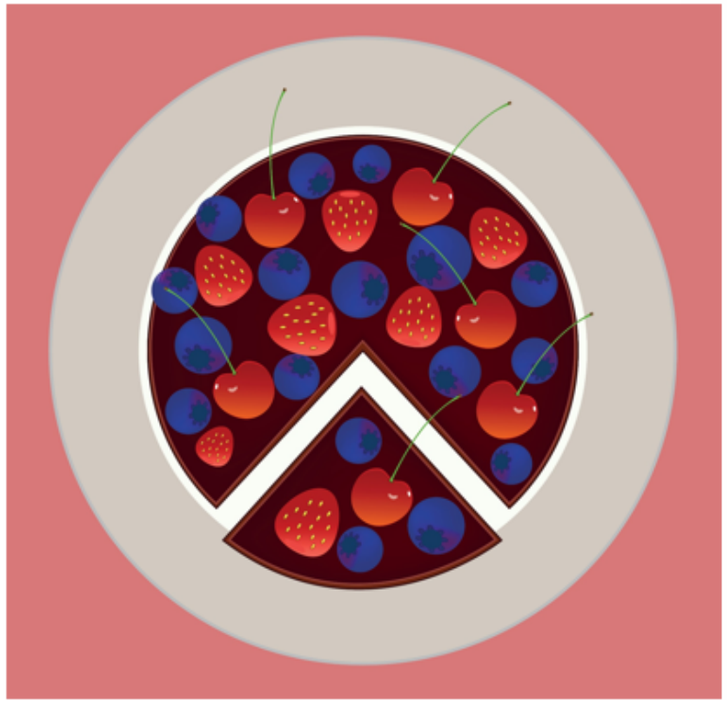
MICROSERVICE(S)

Santi Lertsumran #SysAdminDay

SERVICE EVOLUTION



With monolithic, tightly coupled applications, all changes must be pushed at once, making continuous deployment impossible.

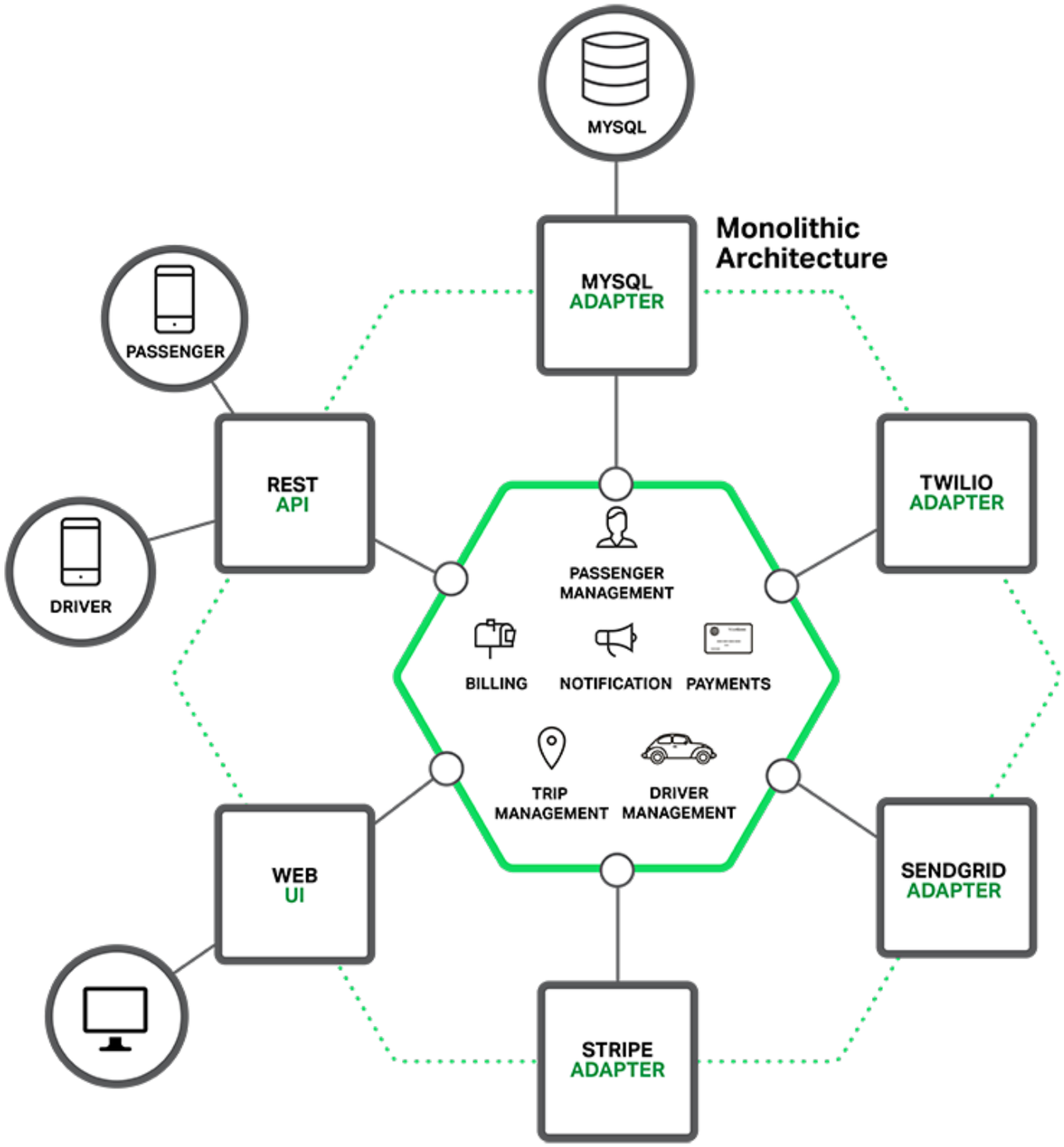


Traditional SOA allows you to make changes to individual pieces. But each piece must be carefully altered to fit into the overall design.

With a microservices architecture, developers create, maintain and improve new services independently, linking info through a shared data API.

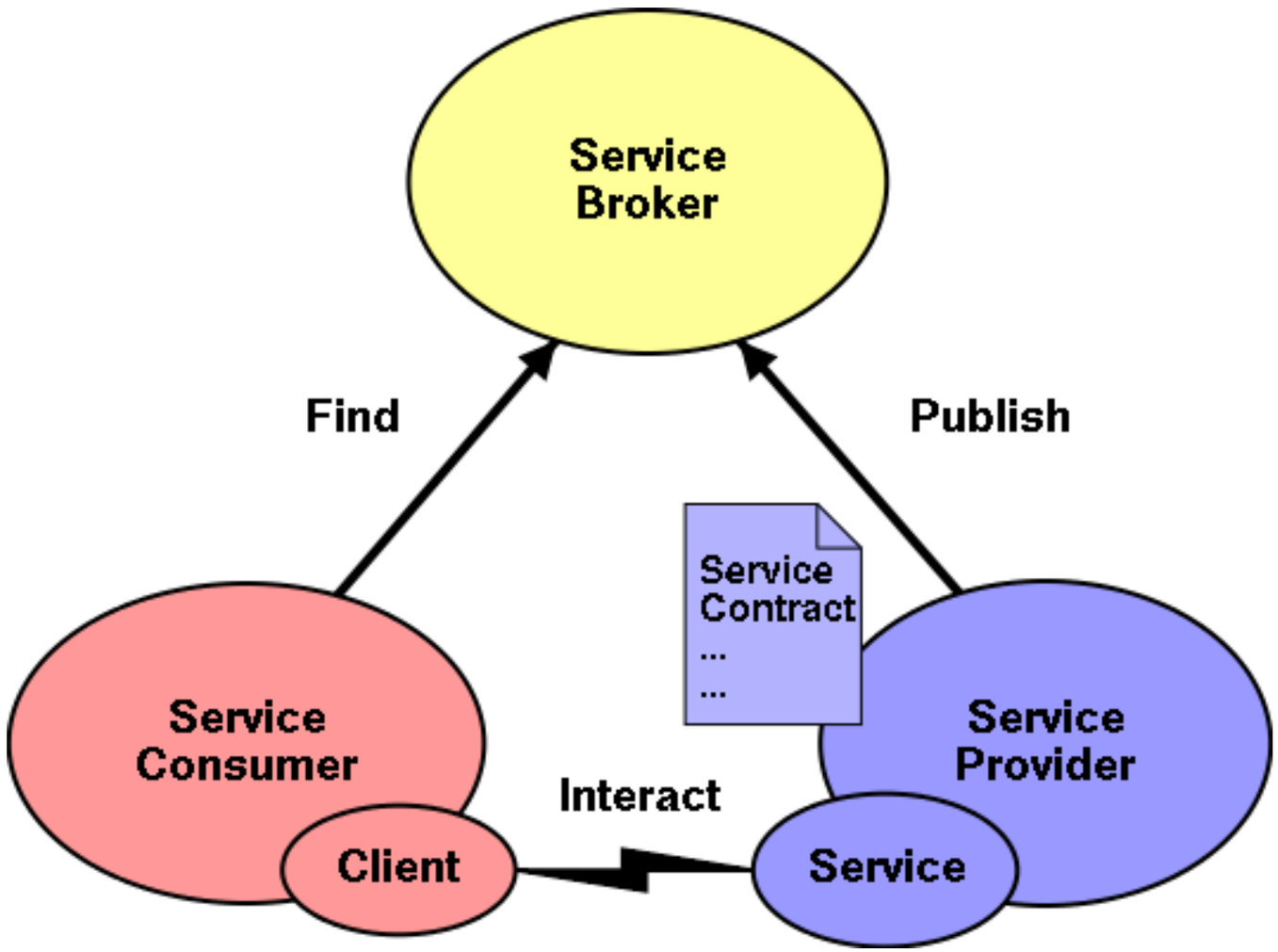
https://dzone.com/articles/what-are-microservices-actually

MONOLITHIC



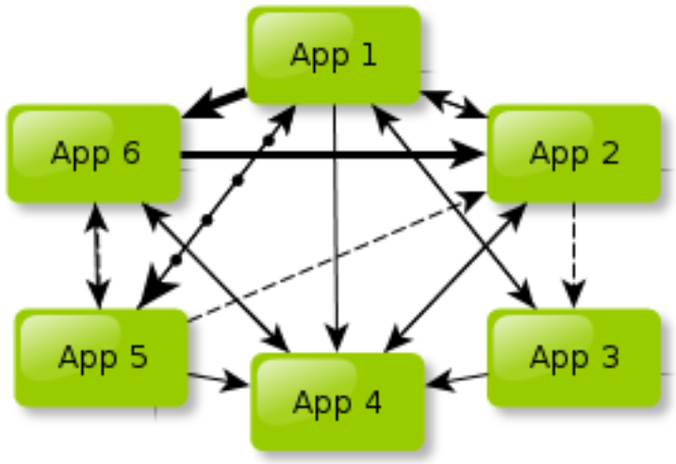
https://www.nginx.com/blog/introduction-to-microservices/

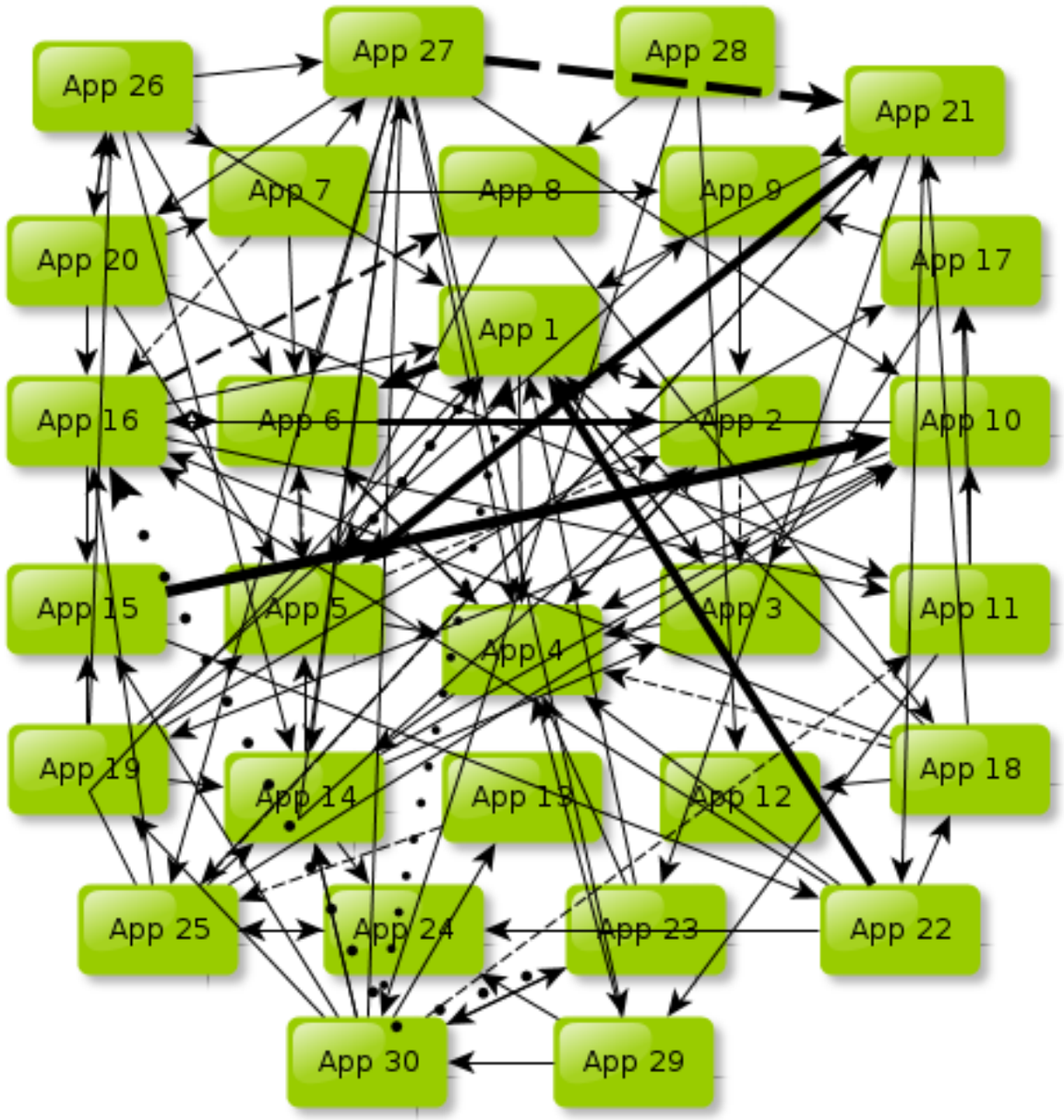
SOA



https://www.w3.org/2003/Talks/0521-hh-wsa/slide5-0.html

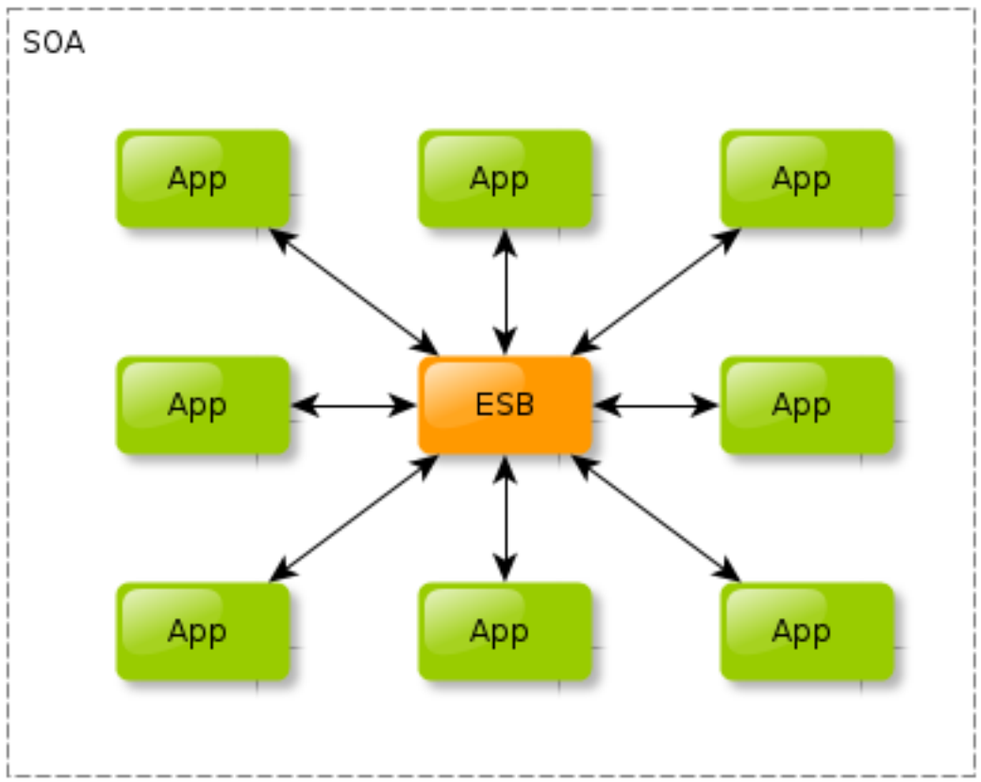
SOA?





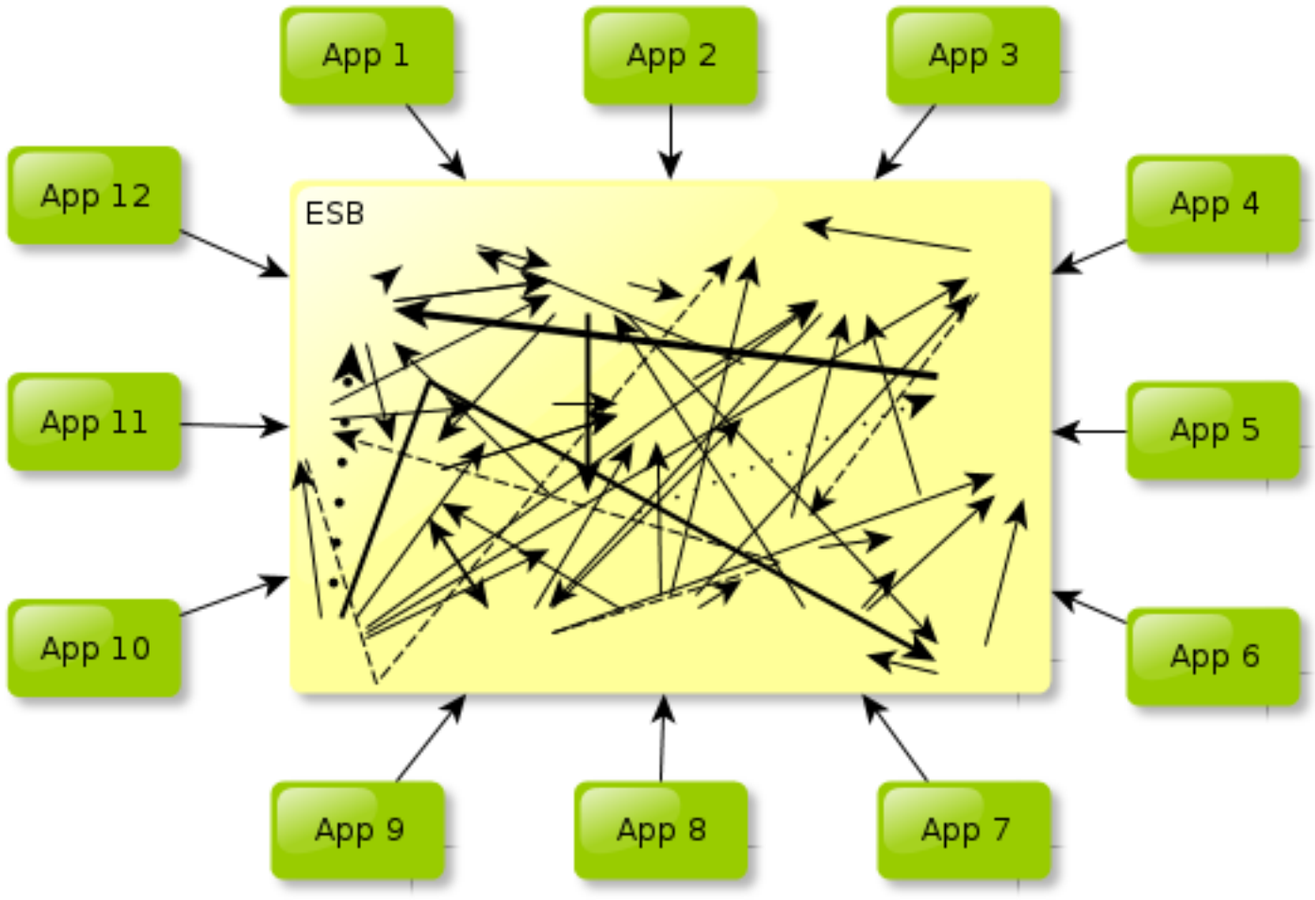
https://zato.io/docs/intro/esb-soa.html

SOA WITH ESB



https://zato.io/docs/intro/esb-soa.html

SOA WITH ESB

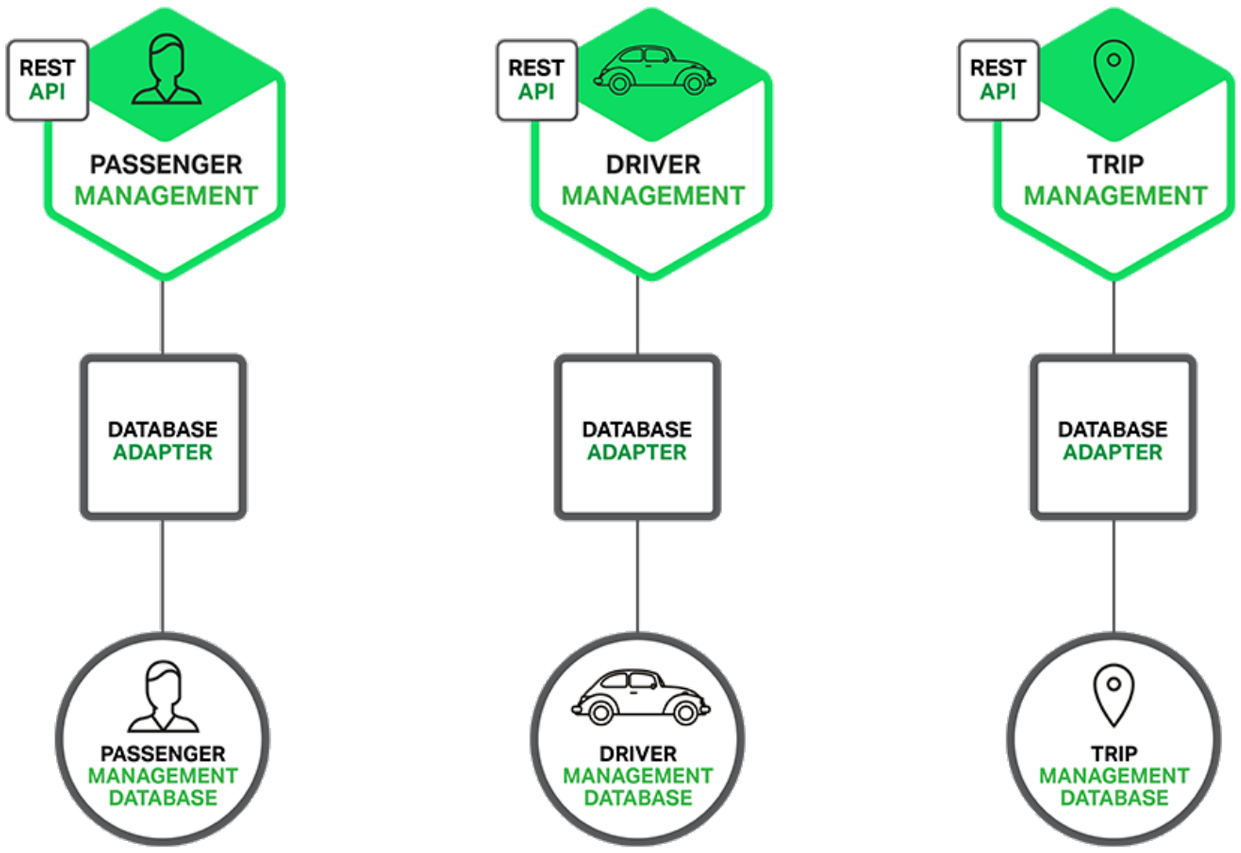


https://zato.io/docs/intro/esb-soa.html

MICROSERVICE(S)

MICROSERVICE(S)

NOT A SILVER BULLET



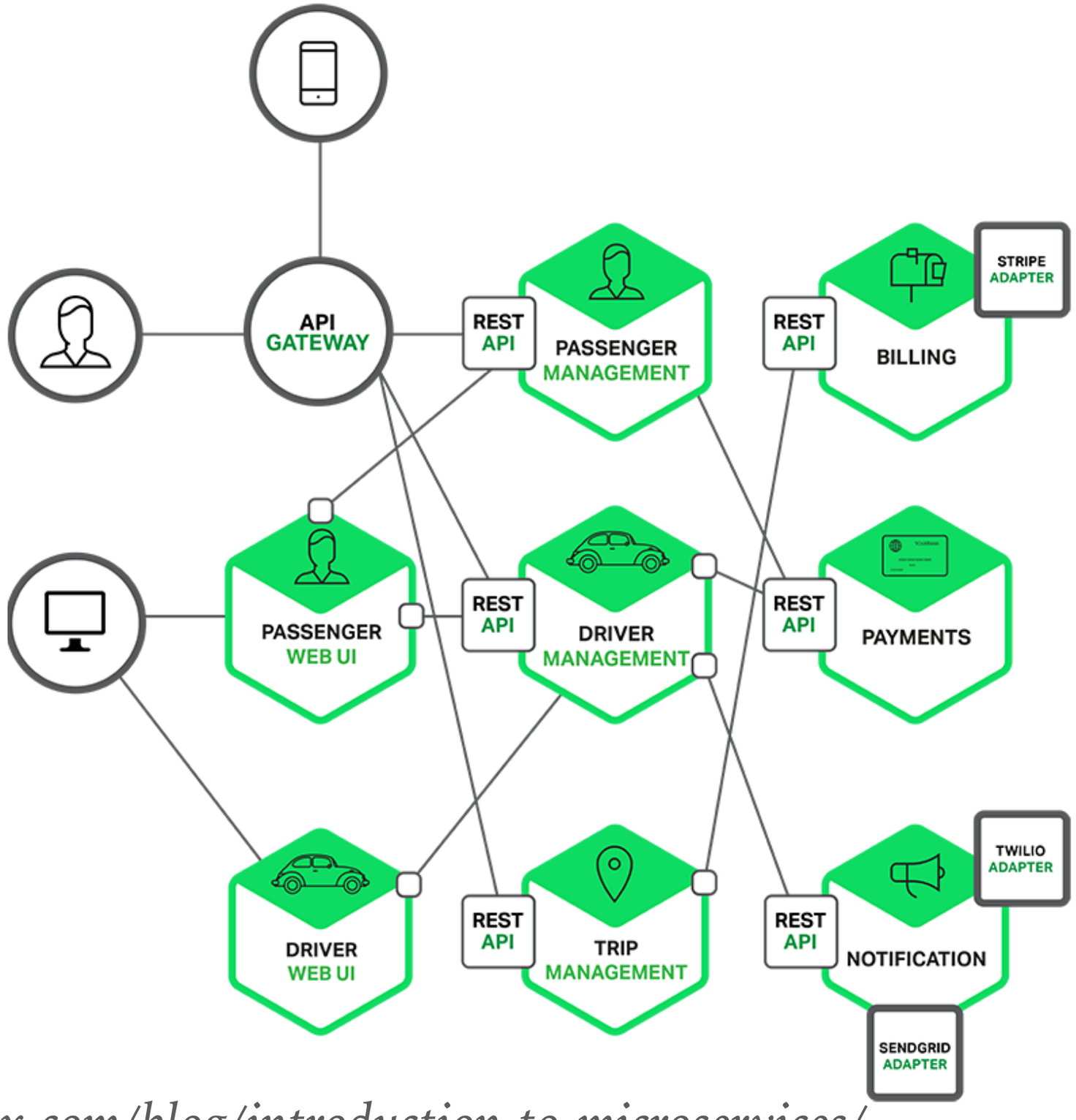
What (is)are microservice(s)?

- small services
- running in its own process
- built around business capabilities
- communicating with lightweight mechanism
- Independently deployable
- bare minimum of centralized management

https://www.nginx.com/blog/introduction-to-microservices/

https://martinfowler.com/microservices/

MICROSERVICES



https://www.nginx.com/blog/introduction-to-microservices/

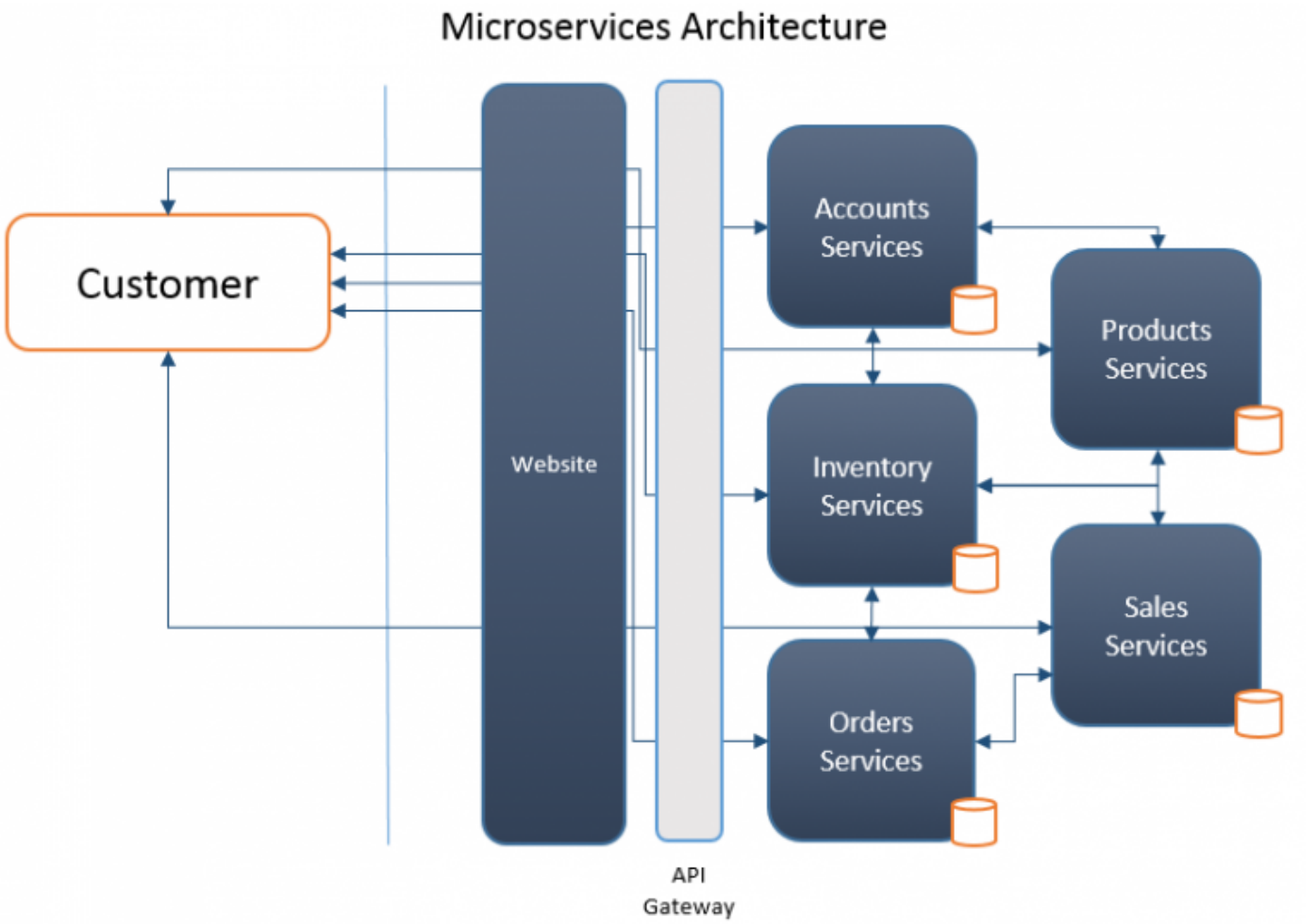
EMERGENCE OF MICROSERVICES | WHY NOW?

- Need to respond to change quickly
- Need for reliability
- Business domain-driven design
- Automated test tools
- Release and deployment tools
- On-demand hosting technology
- On-line cloud services
- Need to embrace new technology
- Asynchronous communication technology
- Simpler server side and client side technology

EMERGENCE OF MICROSERVICES | BENEFITS

- Shorter development times
- Reliable and faster deployment
- Enables frequent updates
- Decouple the changeable parts
- Security
- Increased uptime
- Fast issue resolution
- Highly scalable and better performance
- Right technology
- Enables distributed teams

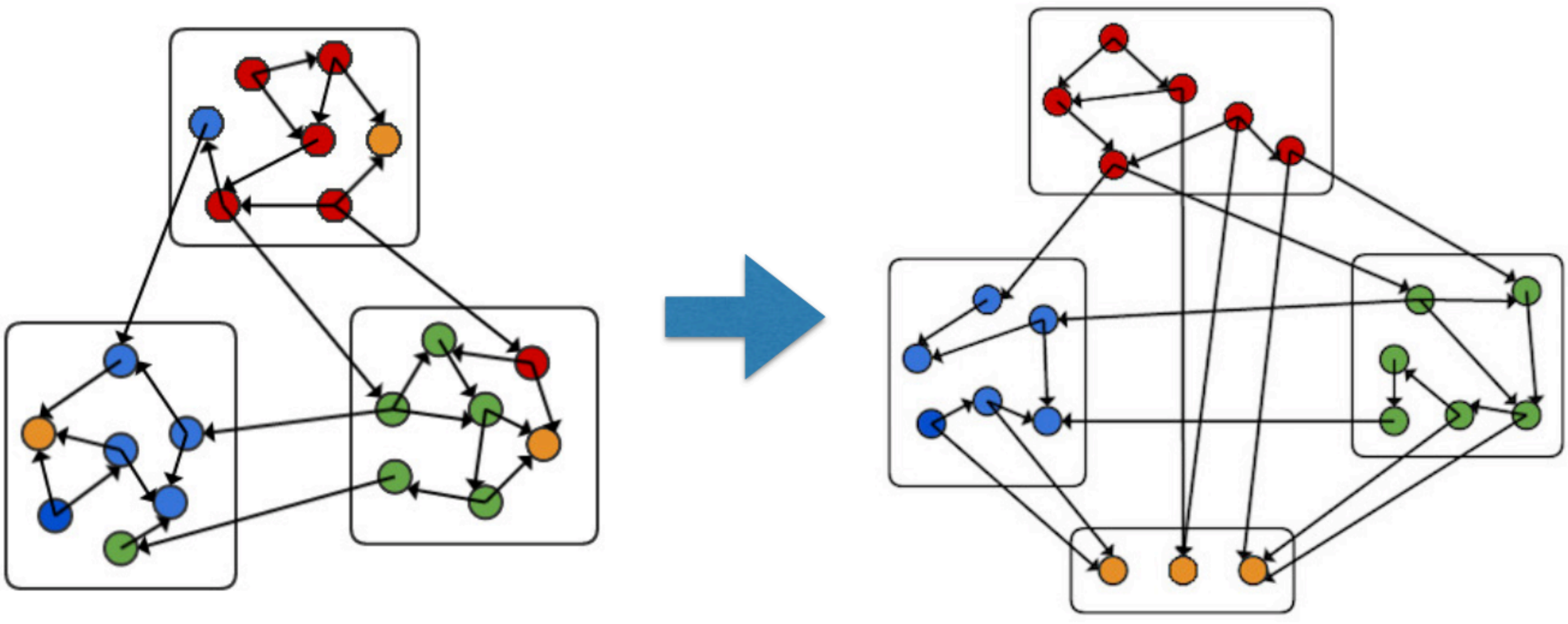
DESIGN PRINCIPLES | HIGH COHESION



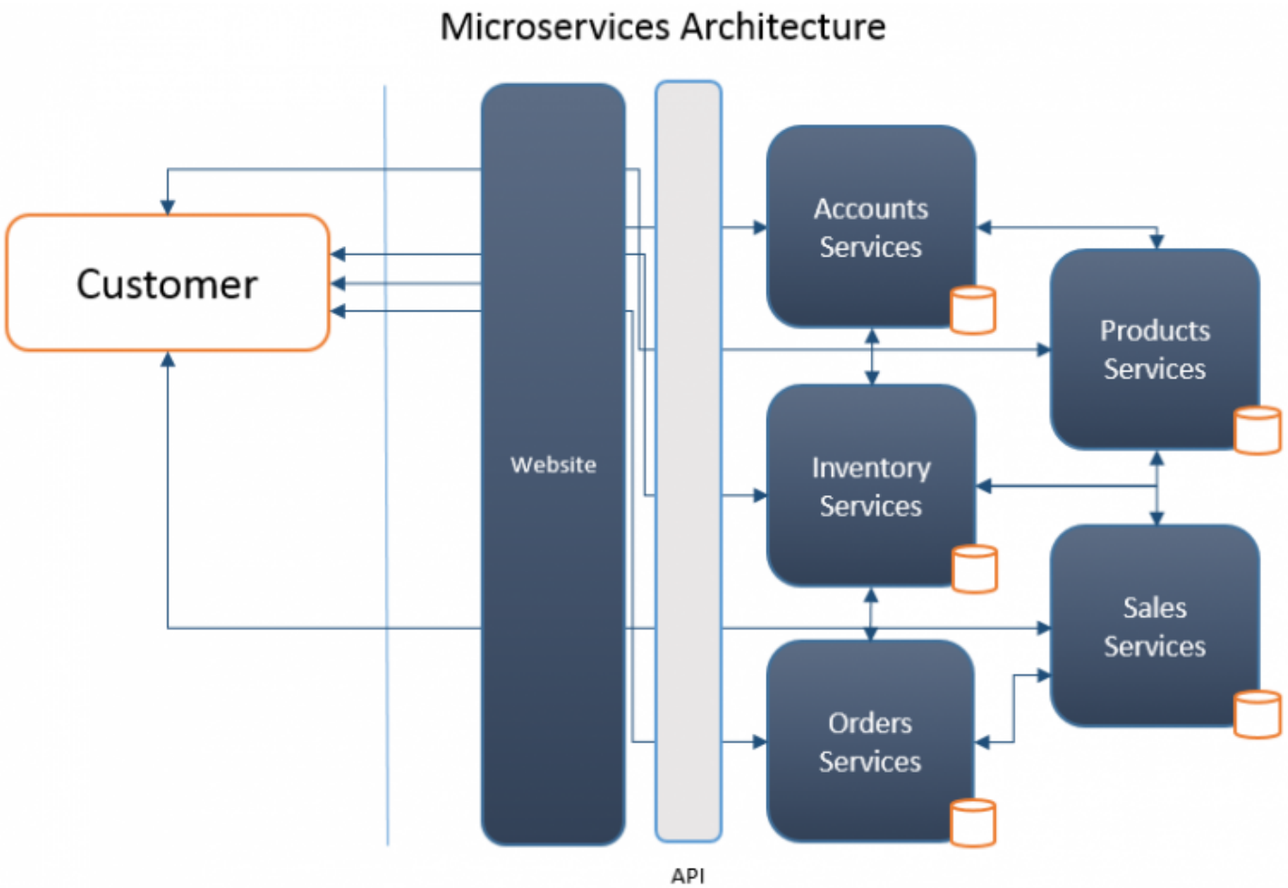
- Single focus
- Single responsibility
- Reason represents (A business function/domain)
- Easily rewritable code

http://www.brunoarruda.com/introduction-to-microservices/

HIGH COHESION

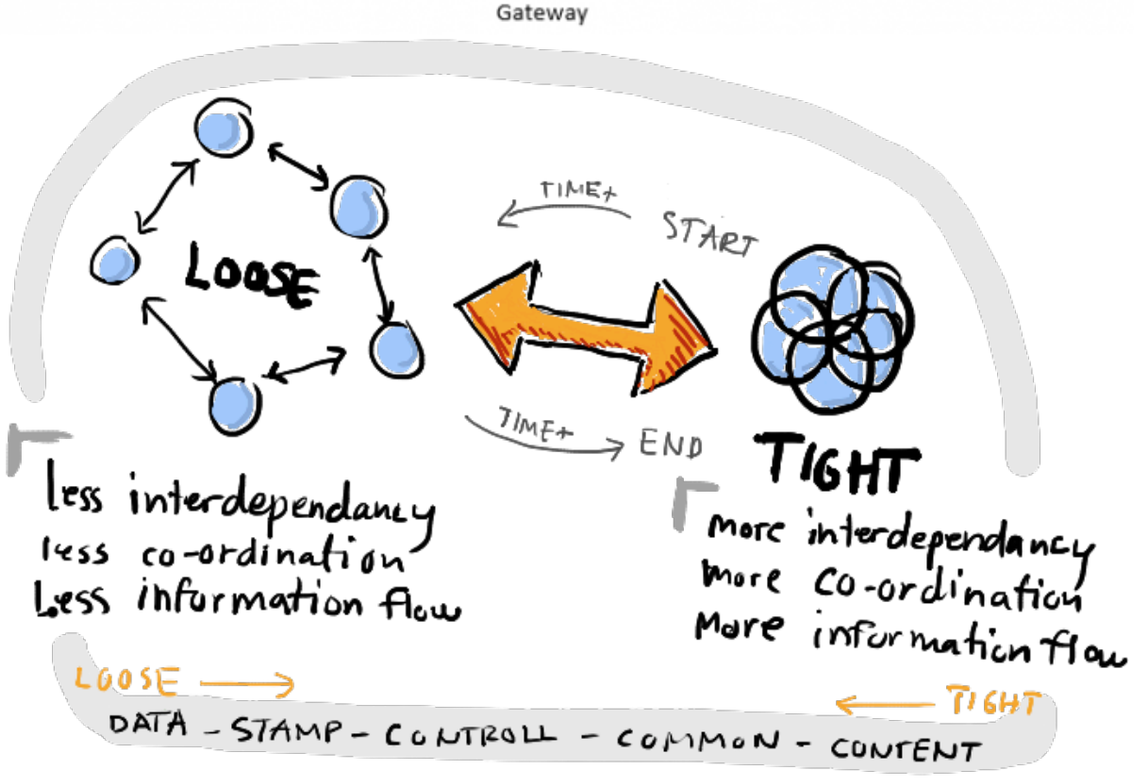


https://github.com/up1/course_microservice_kmitl



DESIGN PRINCIPLES | AUTONOMOUS

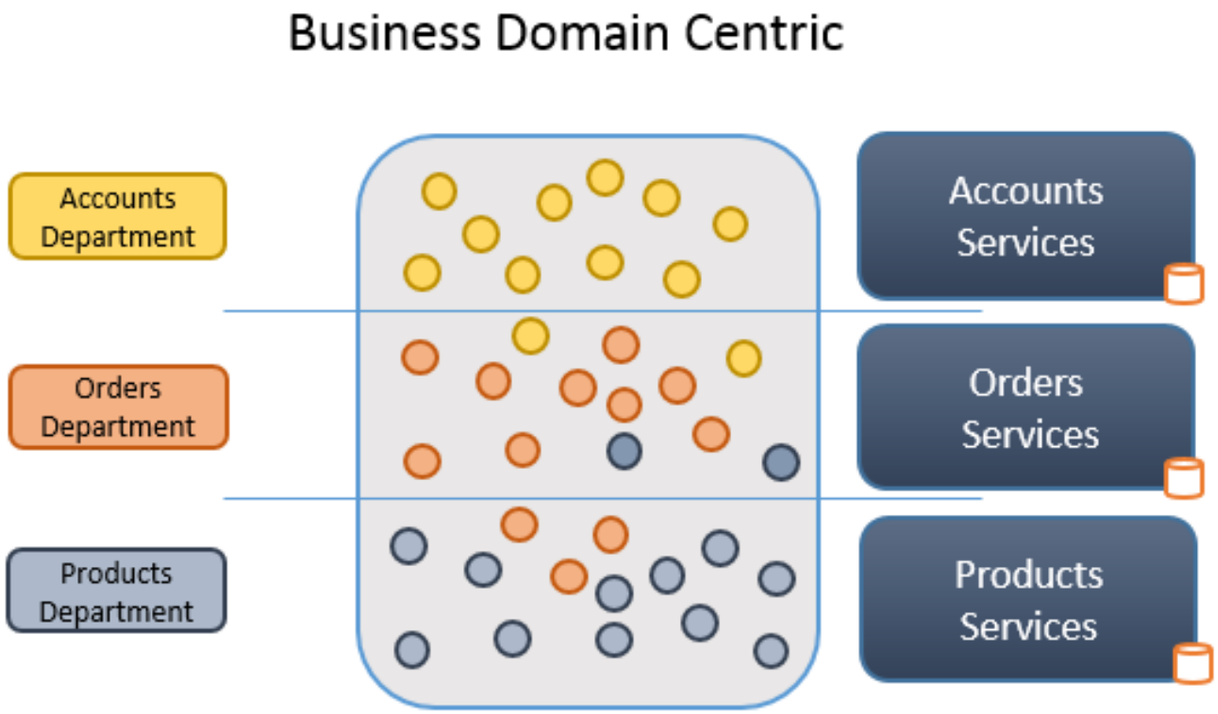
- Loose coupling
- Honor contracts and interfaces
- Stateless
- Independently changeable
- Independently deployable
- Backwards compatible
- Concurrent development



http://www.brunoarruda.com/introduction-to-microservices/

https://infomgmt.wordpress.com/2010/02/18/a-visual-respresentation-of-coupling/

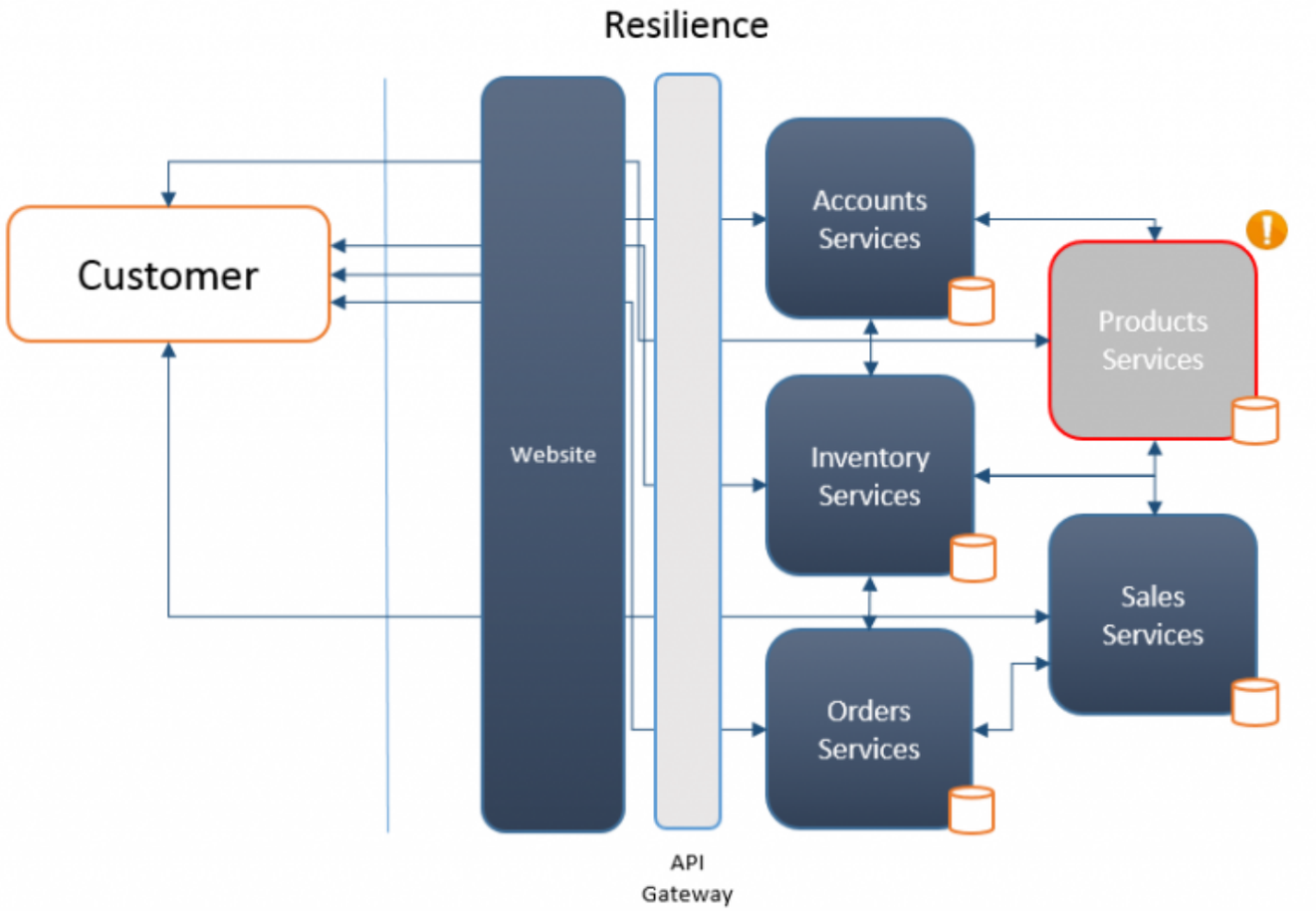
DESIGN PRINCIPLES | BUSINESS DOMAIN CENTRIC



- Service represents business function
- Scope of service
- DDD
- Shuffle code if required
- Responsive to business change

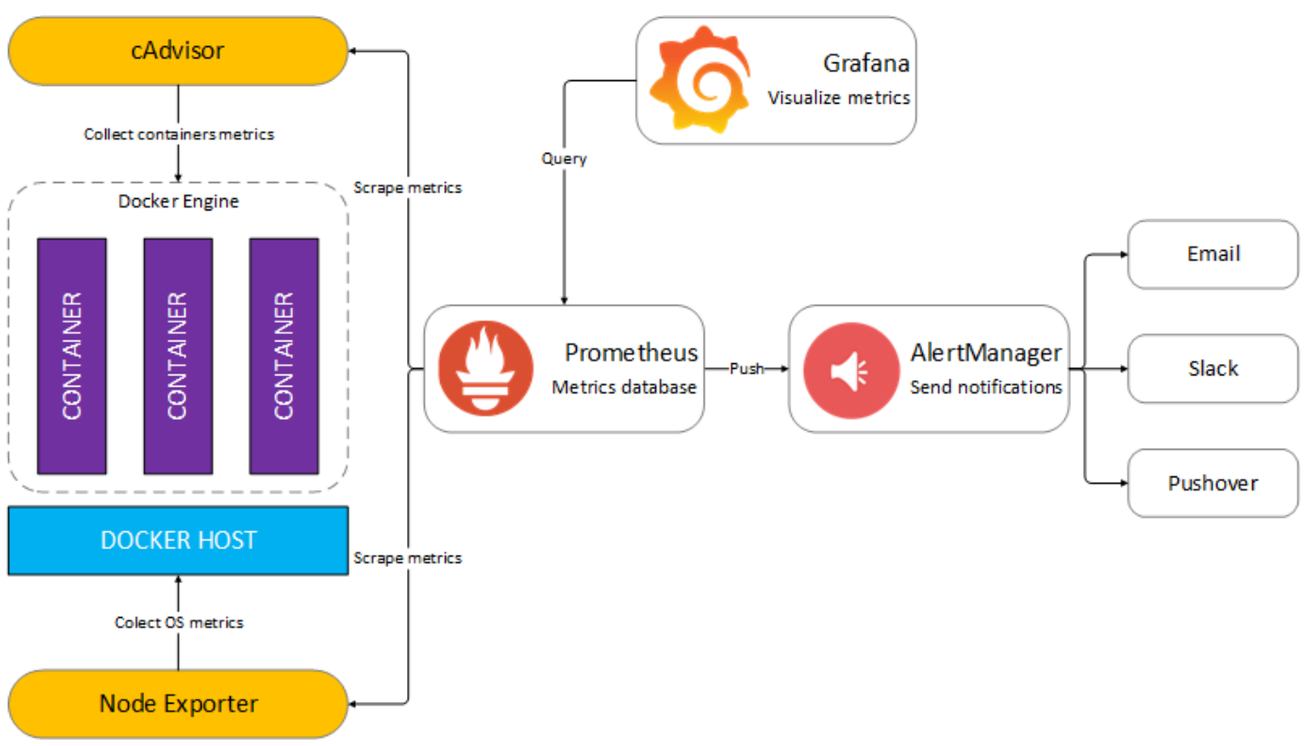
http://www.brunoarruda.com/introduction-to-microservices/

DESIGN PRINCIPLES | RESILIENCE



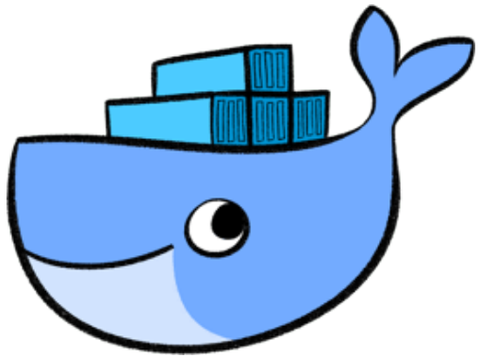
- Embrace failure
  - Another service
  - Specific connection
  - Third-party system
- Degrade functionality
- Default functionality
- Multiple instances
  - Register on startup
  - Deregister on failure
- Types of failure
  - Exceptions\Errors
  - Delays
  - Unavailability

DESIGN PRINCIPLES | OBSERVABLE



- System Health
  - Status
  - Logs
  - Errors
- Centralized monitoring
- Centralized logging
- Why
  - Distributed transactions
  - Quick problem solving
  - Quick deployment requires feedback
  - Data used for capacity planning
  - Data used for scaling
  - What actually used
  - Monitor business data

https://stefanprodan.com/



DESIGN PRINCIPLES | AUTOMATION

- Tools to reduce testing
  - Manual regression testing
  - Time taken on testing integration
  - Environment setup for testing
- Tools to provide quick feedback
  - Integration feedback on check in
  - Continous Integration
- Tools to provide quick deployment
- Why
  - Distributed system
  - Multiple instances of services
  - Manual integration testing too time consuming
  - Manual deployment time consuming and unreliable

MICROSERVICES ECOSYSTEM

SEQUOIA

MICROSERVICES ECOSYSTEM

DEVELOPER TOOLS

SOURCE
Atlassian
GitHub
Gitlab

CONTINUOUS INTEGRATION
Atlassian, CloudBees, JFrog, Codeship, CircleCI, Werker, Shippable

CONTAINER REGISTRY
Docker
Amazon
Google

SERVICE DISCOVERY & PLANNING
Docker, Kubernetes, Hashicorp

SERVICE OPTIMIZATION
Force12.io

SECURITY & COMPLIANCE
Illumio, Apcera, CloudPassage, Banyan, Twistlock, Redlock, Palo Alto Networks, StackRox, Conjur, Scalock

INTER-SERVICE COMMUNICATIONS
Confluent, Tensyr, Rabbit (Pivotal), Hystrix, Thrift, Finagle, NATS, gRPC

API MANAGEMENT
Mulesoft, Akana, Apigee, Runscope, Kong, WSO2, 3Scale, Mashery

MONITORING | LOG ANALYSIS
Wavefront, DataDog, Sysdig, Elastic, Logentries, Nagios, Gencore, App Dynamics, SumoLogic, Runscope, New Relic, SignalFX, Splunk

ORCHESTRATION
Docker, Mesosphere, Apcera, Kubernetes, HashiCorp

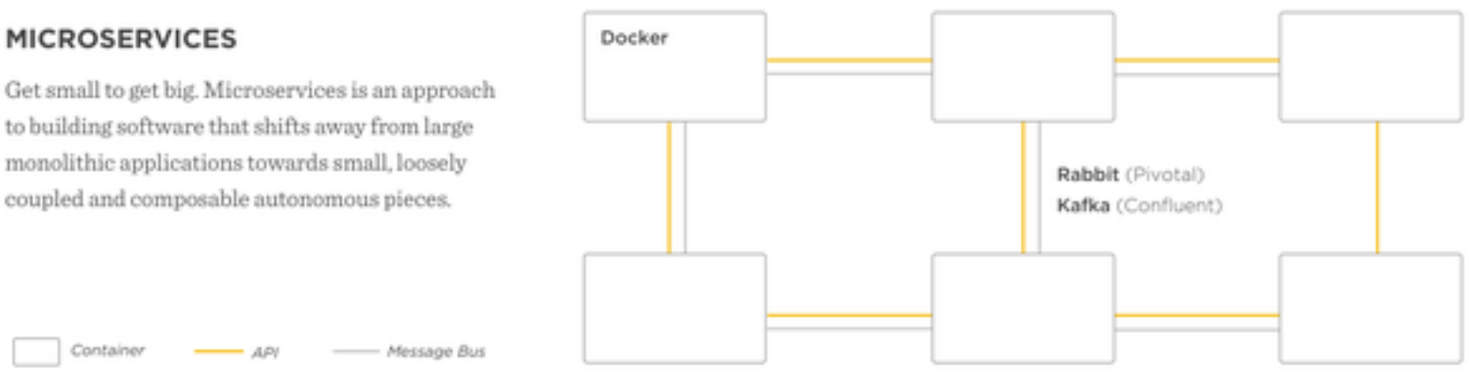
REGISTRATION
Zookeeper, CoreOS

PLATFORM MANAGEMENT
Docker, Mesosphere, AppFormix, Nirmata, Rancher, Stack Engine (Oracle), Apcera, Flexiant, Containership, ManageIQ, Kubernetes

LOAD BALANCING
NGINX, Datawire, Buoyant, HAProxy, Traefik

MICROSERVICES

Get small to get big. Microservices is an approach to building software that shifts away from large monolithic applications towards small, loosely coupled and composable autonomous pieces.



OPERATING SYSTEM
Linux, UNIX, Windows, CoreOS, OpenStack, Mesosphere

DATA CENTER

INFRASTRUCTURE AUTOMATION
HashiCorp, Ansible (Red Hat), Puppet, SaltStack, Chef

DATABASE & DATA MANAGEMENT
ClusterHQ, Minio, MongoDB, Crate.io, Cockroach

NETWORK
Cumulus, Docker, Big Switch, Weaveworks, FBOSS, Calico, OpenSwitch

CONVERGED INFRASTRUCTURE
Ceph (Red Hat), Datawise, Springpath, Portworx

PLATFORMS
OpenShift, Joyent, Cloud Foundry, Deis, Docker

PUBLIC CLOUD
AWS, DigitalOcean, Azure, VMware, IBM, Google

Includes both companies and open source projects

https://twitter.com/mcmiller00/

Version 1.2 | 01.18.16

Q/A