

معماری مایکروسرویس

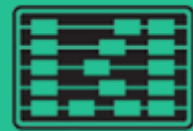
Microservices Architecture

امیر مهجوریان

مدیر فنی آزمایشگاه معماری سازمانی سرویس گرا
مدیر شرکت دانش بنیان کاریز سیستم پویا

۲۲ آبانماه ۱۳۹۸

سومین همایش ملی
پیشرفتهای معماری سازمانی
دانشگاه صنعتی شریف



انجمن کامپیوتر ایران



دانشگاه صنعتی شریف

سرفصل مطالب

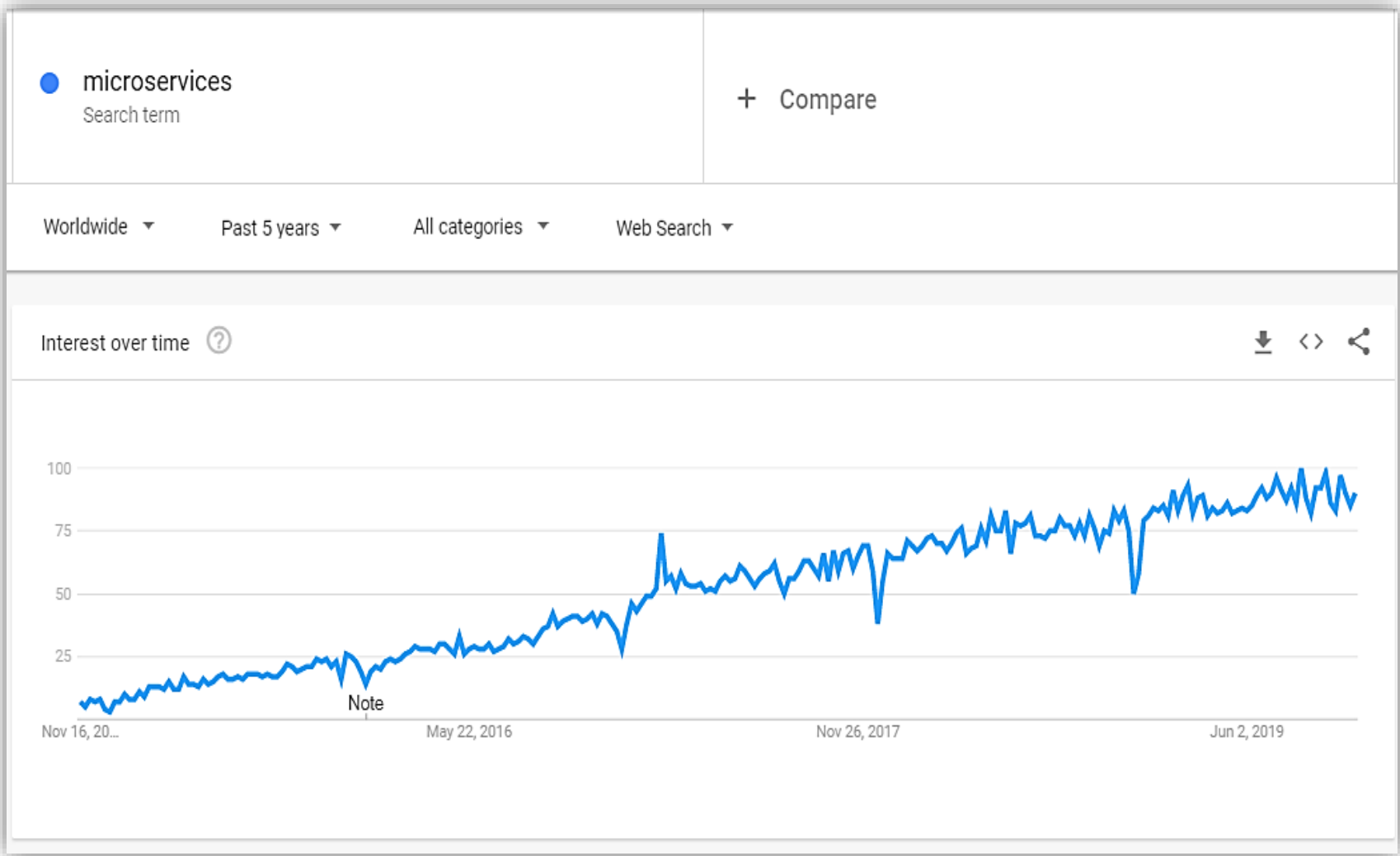
- مفاهیم، تعاریف و اصول معماری میکروسرویس
- تکنیک ها، فناوری ها و متدها
- مقایسه MSA با SOA و Web-Service
- میکروسرویس و معماری سازمانی سرویس گرا
- نمونه ای از تحلیل و طراحی میکروسرویس ها
- روند بازار میکروسرویس
- جمع بندی و تحلیل هزینه-فایده



Definition, principle and Concept

- اولین اشاره مستقیم به واژه "مایکروسرویس‌ها" به سال ۲۰۱۱ و در یک کارگاه معماری نرم‌افزار برمی‌گردد، اما داغ شدن این موضوع در طی سال‌های ۲۰۱۴ و ۲۰۱۵ بود؛ هم‌اکنون مایکروسرویس‌ها یکی از موضوعات جذاب در دنیای نرم‌افزار و معماری محسوب می‌شود و هرماه مقالات، کتاب‌ها و آرایه‌های جدیدی از آن منتشر می‌شود و در کنفرانس‌ها یا سمینارهای تجاری-علمی نیز علاقمندان زیادی را به خود جذب می‌کند؛ بر اساس گزارش‌های گوگل از میزان رشد جستجوی عبارات مرتبط با مایکروسرویس‌ها می‌توان به نقش محوری آن در معماری و توسعه سیستم‌ها پی برد.
- در دنیای تجاری نیز شرکت‌های مطرحی پیشگام پیاده‌سازی و استقرار آن بوده‌اند که از جمله می‌توان به شرکت‌های Uber، Netflix، Amazon، Ebay و Sound Cloud اشاره نمود.
- بر اساس تحقیقاتی متفاوتی که توسط Redhat، Forrester و Dimensional Research انجام شده است، بیش از ۷۰ درصد پرسش‌شوندگان اعلام کرده‌اند که برنامه‌ای برای توسعه و پیاده‌سازی معماری مایکروسرویس دارند.

رشد جستجوی مایکروسرویس در گوگل



2014

2019

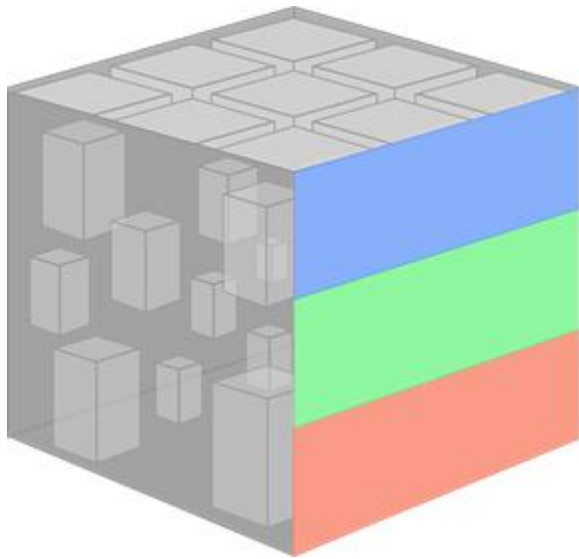
- سبک معماری مایکروسرویس رویکردی برای توسعه یک نرم‌افزار متشکل از تعدادی سرویس کوچک و مستقل است که هر سرویس به‌اتکاء منابع و زیرساخت خودش اجرا شده و از طریق پروتکل‌های سبک مبتنی بر HTTP با دیگران ارتباط دارد. این سرویس‌ها براساس قابلیت‌های کسب‌وکار طراحی و ساخته می‌شوند و بر بسترهای فناوری با زبان‌های برنامه‌نویسی مختلفی قابل استقرار هستند. این سرویس‌ها حداقل نیاز به مدیریت متمرکز را دارند و هر سرویس پایگاه داده مخصوص به خود را مدیریت می‌کند. (Martin Fowler)
- مایکروسرویس‌ها به‌صورت خلاصه سرویس‌های دانه‌ریز و خودمختاری هستند که با یکدیگر همکاری می‌کنند. هر سرویس باید بتواند مستقلاً تغییر کند بدون اینکه منجر به تغییر دیگر سرویس‌های مرتبط یا استفاده‌کنندگان از سرویس شود. (Sam Newman)
- معماری مایکروسرویس یک رویکرد مهندسی مبتنی بر شکست یک نرم‌افزار به ماژول‌های تک-کارکردی است که مستقلاً تولید و مستقر می‌شوند و با واسط‌های خوش‌تعریف با دیگر سرویس‌ها ارتباط دارند. این سرویس‌ها توسط تیم‌های کوچکی تولید و پشتیبانی می‌شوند که از تمام چرخه حیات سرویس پشتیبانی می‌کند (IBM)
- معماری مایکروسرویس از مجموعه‌ای از سرویس‌های خودمختار و کوچک تشکیل شده است که هر سرویس مستقل بوده و یک قابلیت کسب‌وکار را پیاده‌سازی می‌نماید (Microsoft)

- هر سرویس مسوول یک دامنه مشخص و به خوبی تعریف شده از سیستم (صورت مساله) است که مستقلا تولید (Build) و استقرار (Deploy) می یابد.
- هر سرویس از فناوری ها و ابزارهای مناسب خود بهره می برد و لزومی ندارد همه سرویس های یک سیستم از یک فناوری، زبان برنامه نویسی یا پلتفرم استفاده کنند.
- سرویس ها با واسطهای خوش تعریف و سبک با یکدیگر تعامل دارند، خروجی هر سرویس باید بتواند ورودی سرویس های دیگری قرار گیرد.
- هر سرویس مسوول مدیریت داده های خود است و می تواند از انواع ابزارهای DBMS استفاده نماید.
- اصول کلی معماری سرویس گرا در این معماری نیز صادق است.
- ترجیح استفاده از روش پیام رسانی غیرهمزمان (Asynchronous) نسبت به همزمان
- ترجیح استفاده از روش همکاری کاریگرافی (Choreography) نسبت به ارکستریشن



**MicroServices:
Do One Thing and
Do It Well**

معماری یک تکه

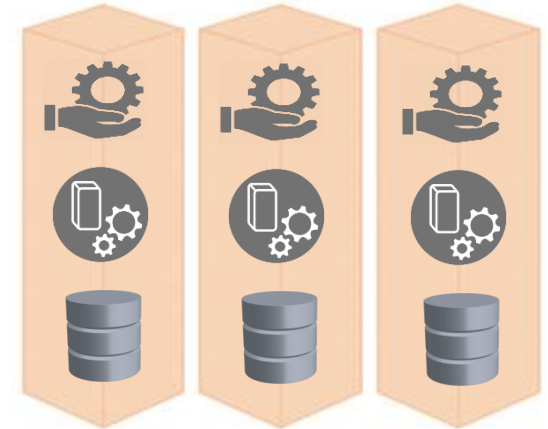


واسط کاربری

مولفه ها و محاسبات سیستم

مدیریت منابع داده

معماری مایکروسرویس



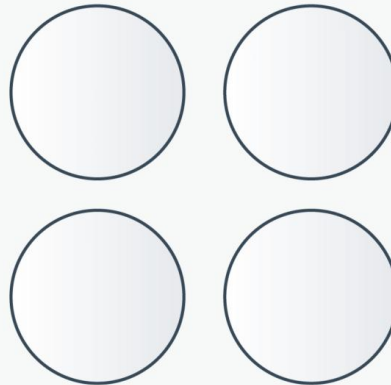
در سیستم با معماری یک-تکه (Monolithic) مجموعه مولفه‌ها-سرویس‌ها-داده‌ها چنان در هم آمیخته است که نمی‌توان بلوک‌های سازنده این سیستم‌ها را مستقلاً از هم جدا کرده و یا تغییر(جایجا) نمود.

Monolithic vs. SOA vs. Microservices



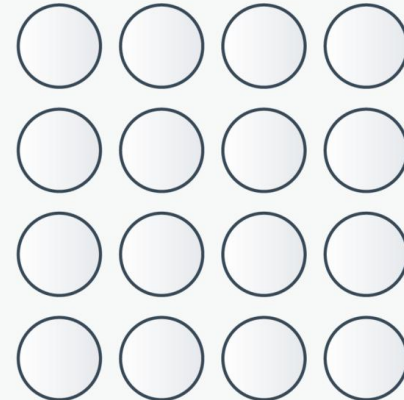
Monolithic

Single Unit



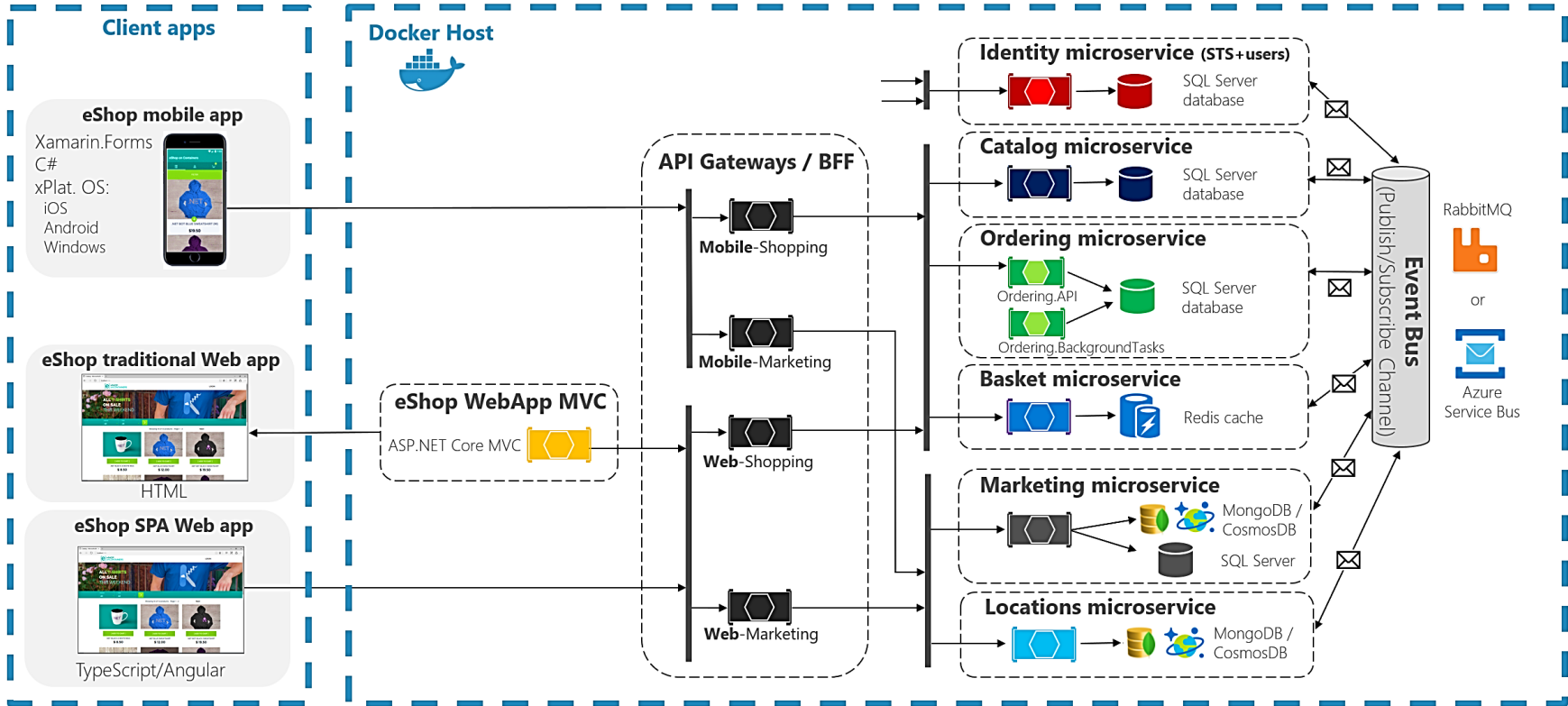
SOA

Coarse-grained



Microservices

Fine-grained



حق انتخاب فناوری/ابزار:

- در معماری مایکروسرویس حق انتخاب سبب متنوعی از فناوری‌ها-ابزارها برای تیم‌های طراحی و پیاده‌سازی مهیا است به صورتی که می‌توان در فرایند تولید یک سیستم، برای مایکروسرویس‌های مختلف از فناوری‌ها و ابزارهای مختلفی استفاده کرد بدون اینکه نگرانی از مشکلات بعدی یکپارچگی وجود داشته باشد.

پایداری سیستم:

- یکی از ویژگی‌های اصلی سیستم‌های پایدار امکان ادامه فعالیت سایر سرویس‌ها (مایکروسرویس‌ها) در صورت از کار افتادن یک سرویس است (و در صورت نیاز جایگزینی سرویس از کار افتاده با نمونه مشابه یا پشتیبان)، این موضوع در معماری مایکروسرویس به دلیل خودمختاری و عدم وابستگی سرویس‌ها محقق می‌شود.

مقیاس پذیری بالا و هدفمند:

- امکان مقیاس پذیری سیستم در سیستم‌های یک تکه به صورت همه یا هیچ است، اما در معماری مایکروسرویس امکان مقیاس پذیری موثر به ازای هر سرویس دلخواه میسر است. بنابراین هر بخش از سیستم که بار (load) کاری بیشتری داشته باشد، متناسباً می‌تواند منابع پردازشی بیشتر نیز در اختیار گیرد و نیازی نیست برای همه مولفه‌های سیستم مقیاس پذیری یکنواخت انجام شود.

توسعه و تغییرات:

- امکان تغییر در منطق هر سرویس بدون نگرانی از تاثیرات منفی در سایر سرویس‌ها به دلیل خودمختاری سرویس‌ها ساده‌تر است، این موضوع البته برای تغییر منطق داخلی است و در صورتیکه واسط سرویس تغییر کند باید به سایر سرویس‌ها (استفاده‌کنندگان) اطلاع داده شود.

استفاده مجدد :

- به مانند معماری سرویس گرا یکی از اهداف اصلی از توسعه سرویس ها، امکان استفاده مجدد از سرویس های موجود برای ایجاد سرویس های جدید است.

تطابق با معماری سازمان سرویس گرا:

- سازمان سرویس گرا صرفا در مشتری-محوری و ارایه سرویس با کیفیت به مشتریان خلاصه نمی-شود بلکه موضوع مهم تر، معماری داخلی سازمان و نحوه چیدمان عناصر و منابع برای پویایی بیشتر و استقلال هر واحد از سایر واحدها است، معماری مایکروسرویس منطبق با معماری سازمانی سرویس گرا و تسهیل کننده آن است.

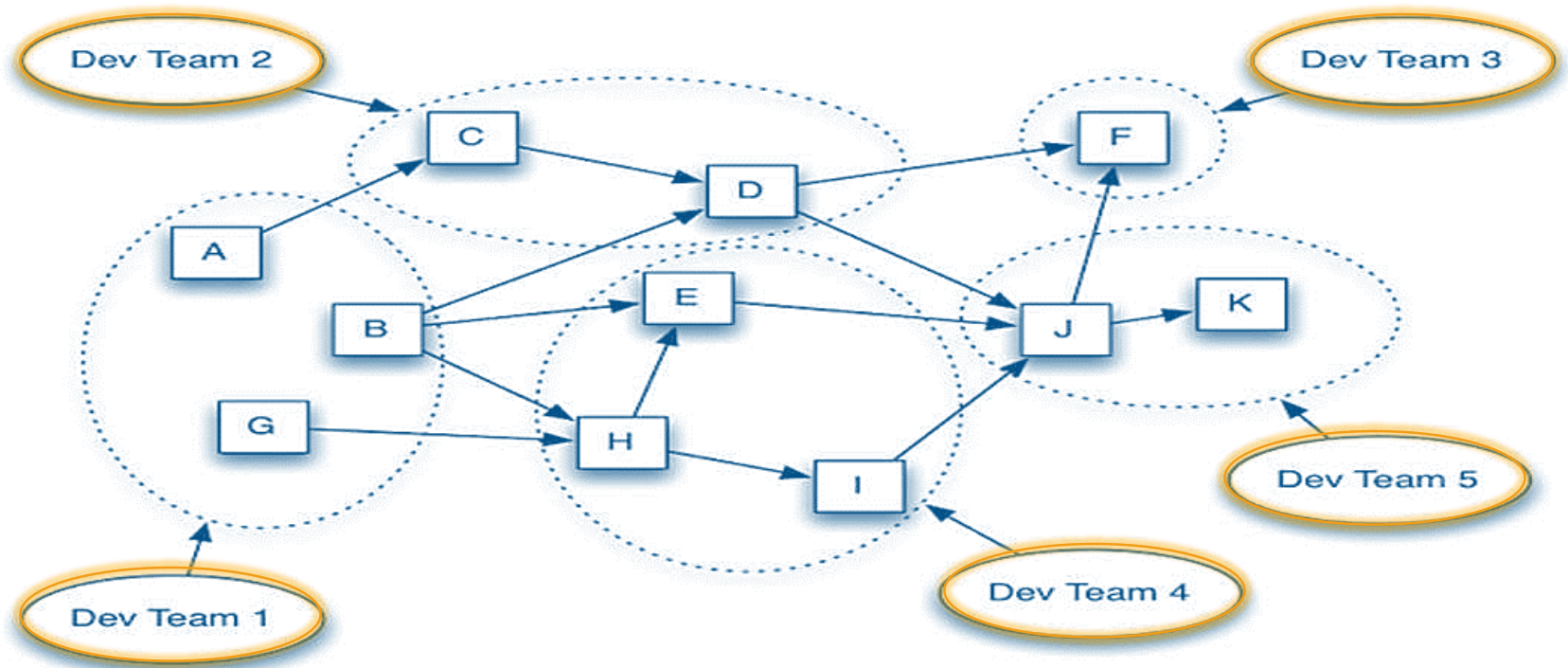
سهولت جابجایی و جایگزینی سرویس ها:

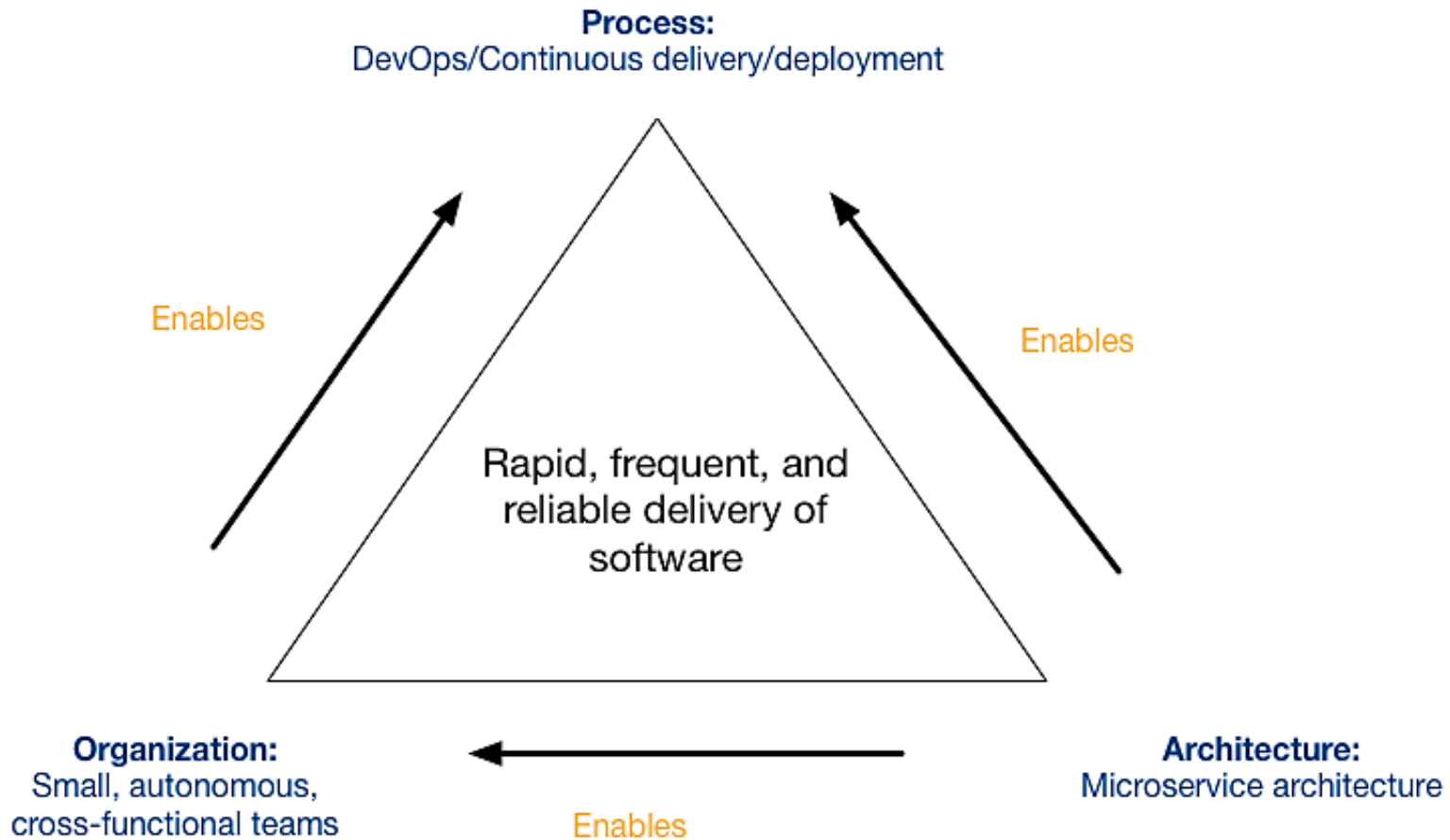
- با توجه به خودمختاری کارکردی و فناوری سرویس ها، به سادگی می توان یک سرویس را که عملکرد آن مناسب نبوده با نمونه بهتر جایگزین نمود یا یک سرویس را به تنهایی به یک محیط/سیستم دیگر منتقل نمود و از آن استفاده نمود



Technics, Methods and Technologies

- Domain Driven Design (DDD) is a software development approach first introduced by [Eric Evans](#)(2003). DDD requires a good understanding of the domain for which the application will be written. The necessary domain knowledge to create the application resides within the people who understand it — the domain experts.
 1. Start with a *ubiquitous language*, a common vocabulary that is shared between all stakeholders.
 2. Identify the relevant modules in the monolithic application and apply the common vocabulary to them.
 3. Define the domain models of the monolithic application. The **domain model** is an abstract model of the business domain.
 4. Define *bounded contexts* for the models. A bounded context is the boundary within a domain where a particular domain model applies.

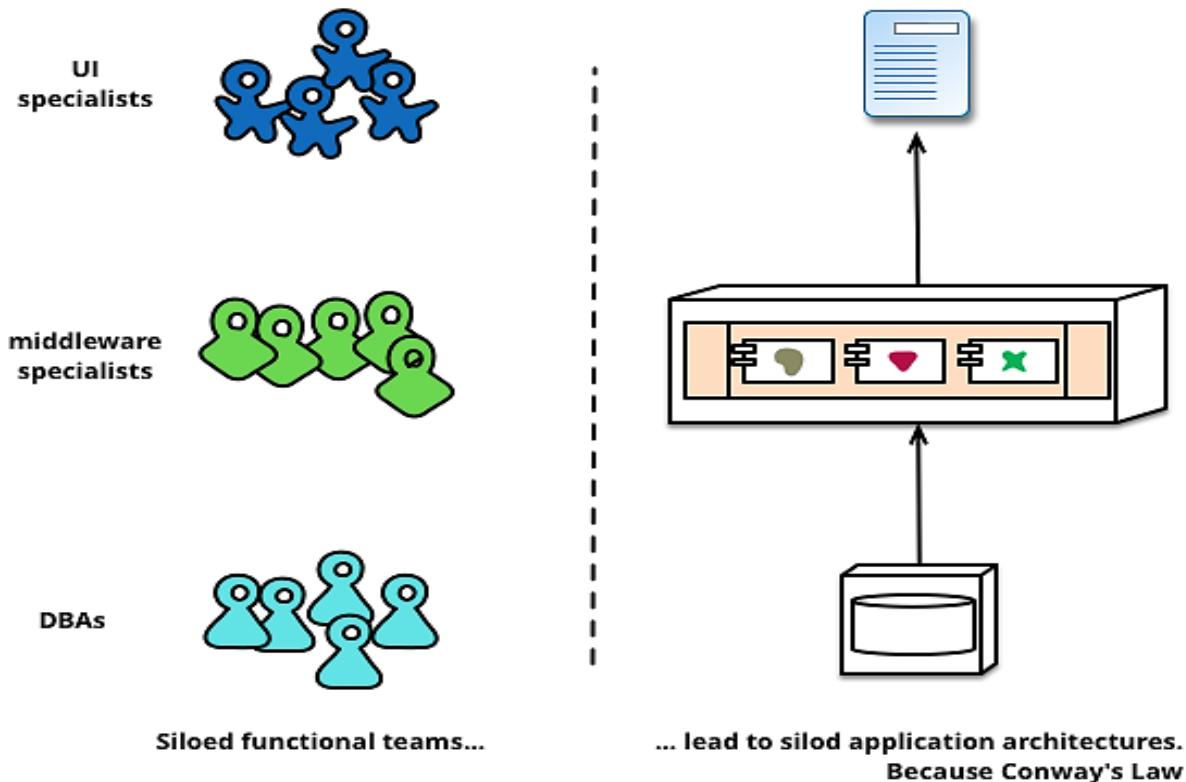




Chris Richardson:

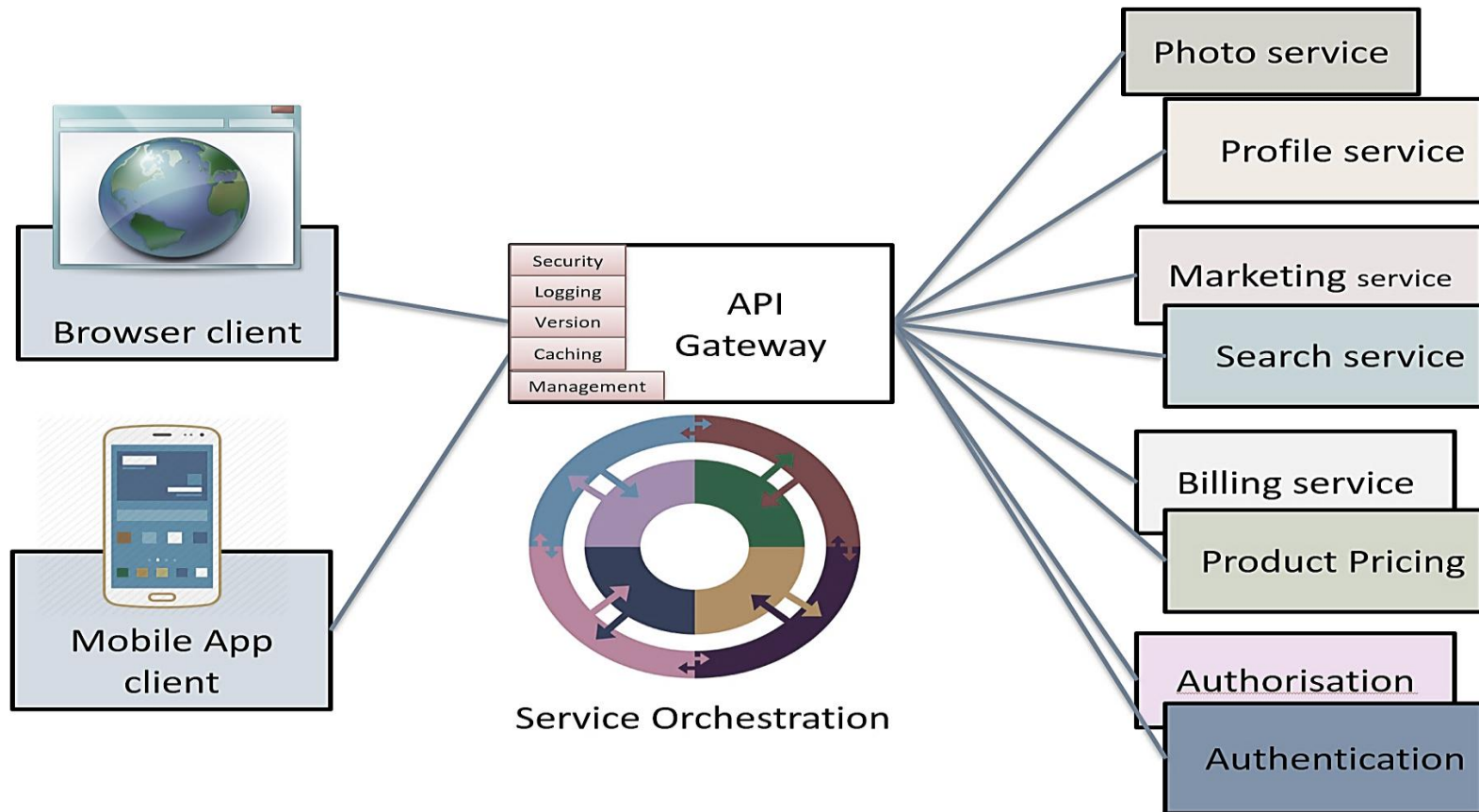
Understanding the Microservice Architecture Through Shapes

“Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.”



API Gateway:

- Decoupling / Routing/ Traffic Management/ Translation/Security



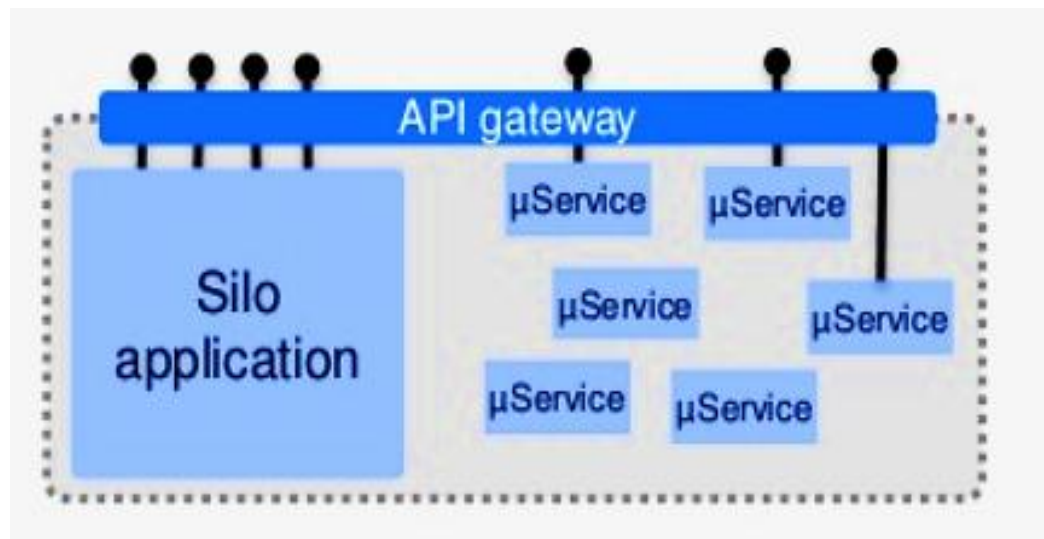
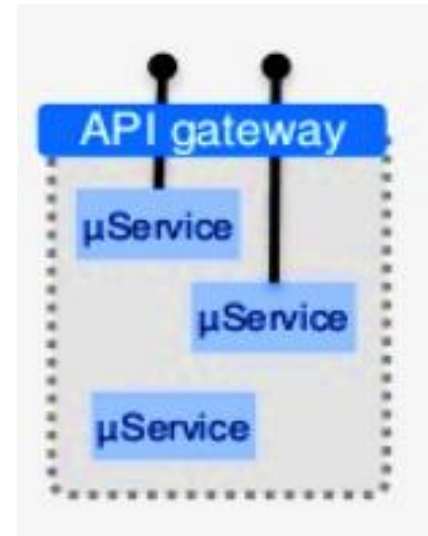
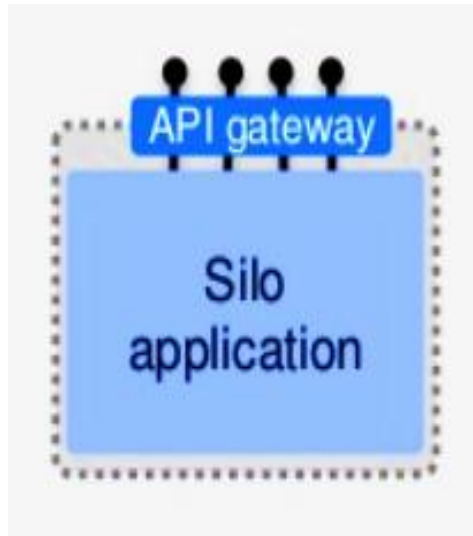
■ Pros

- Unification in one place of common concerns
- Separation of Concerns
- Micro-Services focus on business features
- API Gateway provides protection/common feature layer
- Minimize/Isolate services' change impacts

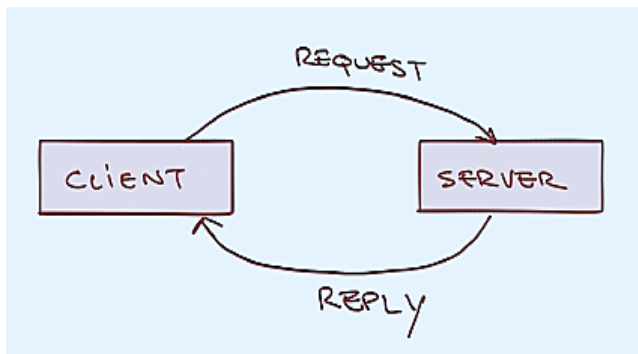
■ Cons

- Possibility of SPOF/bottleneck
- Performance tradeoff due to processing time in API Gateway and more network hops
- Need to manage routing rule or APIs
- Risk of management bottleneck

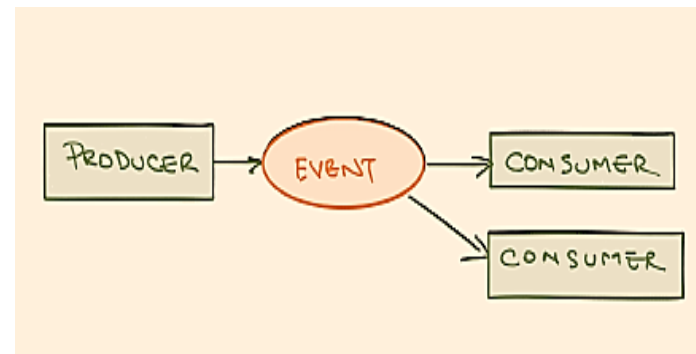
API Gateway is a good idea! For most Microservices-based applications, it makes sense to implement an API Gateway, which acts as a single entry point into a system.



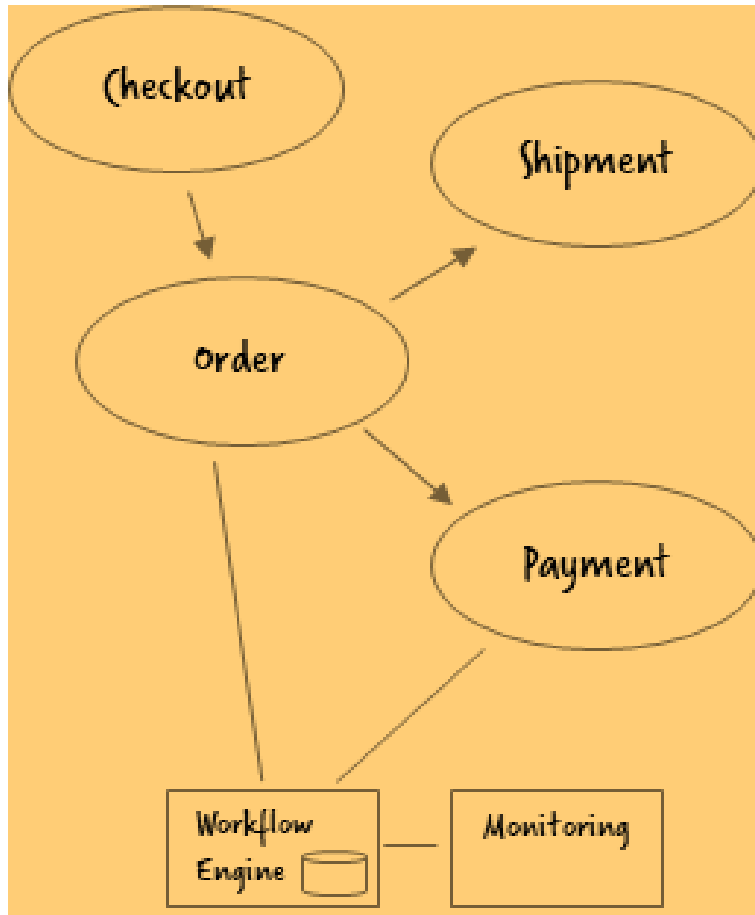
Request-reply and event-driven interaction



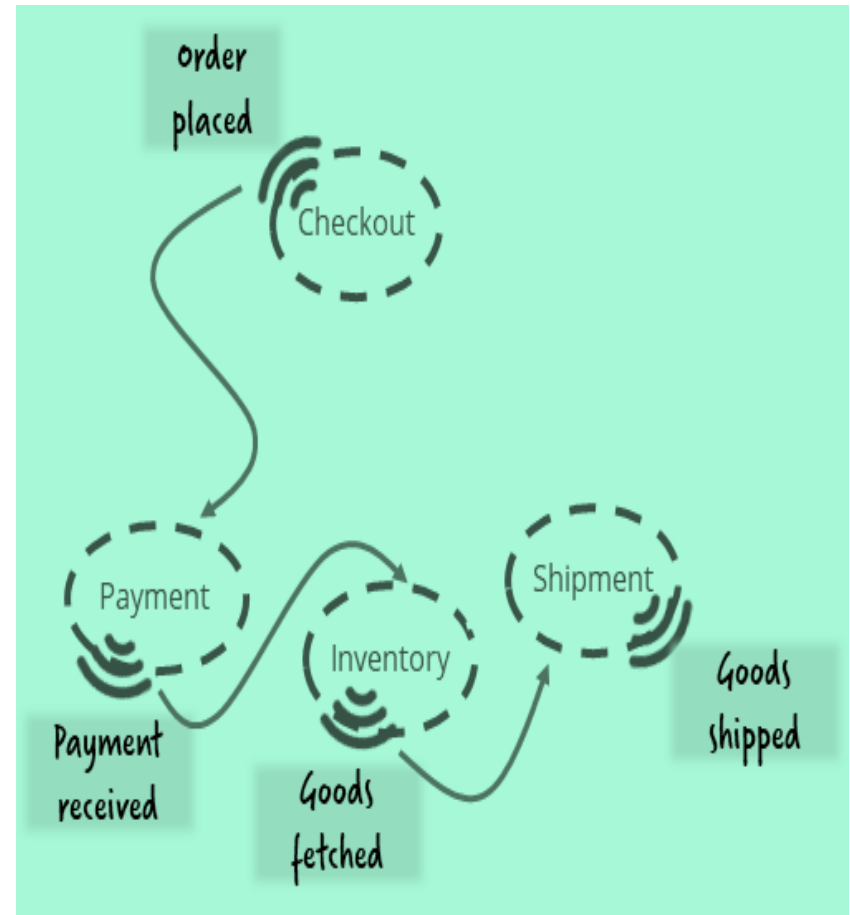
- Interaction: synchronous
- Highly coupled
- Simplified model
- Low tolerance to failure



- Interaction: asynchronous
- Decoupled
- Complex model
- High tolerance to failure



Synchronous



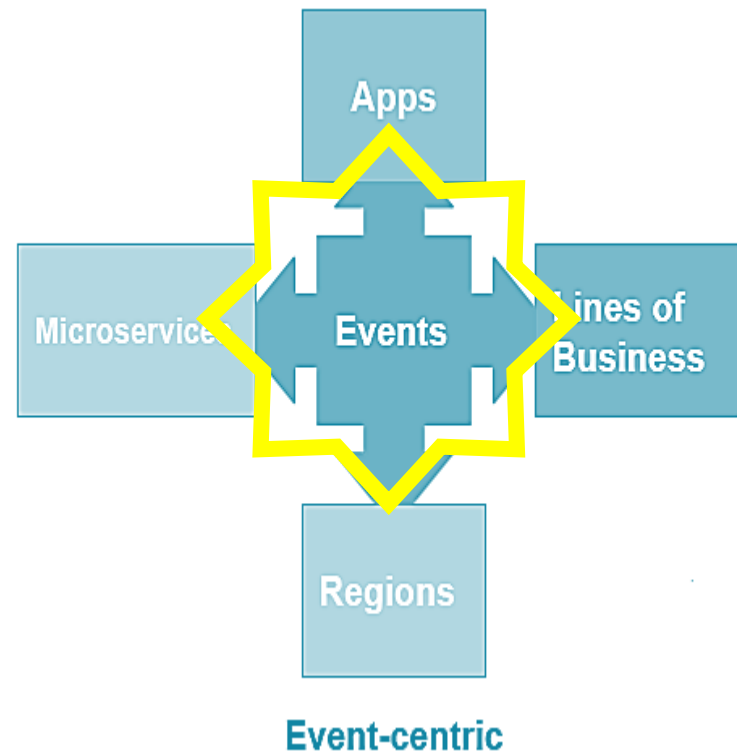
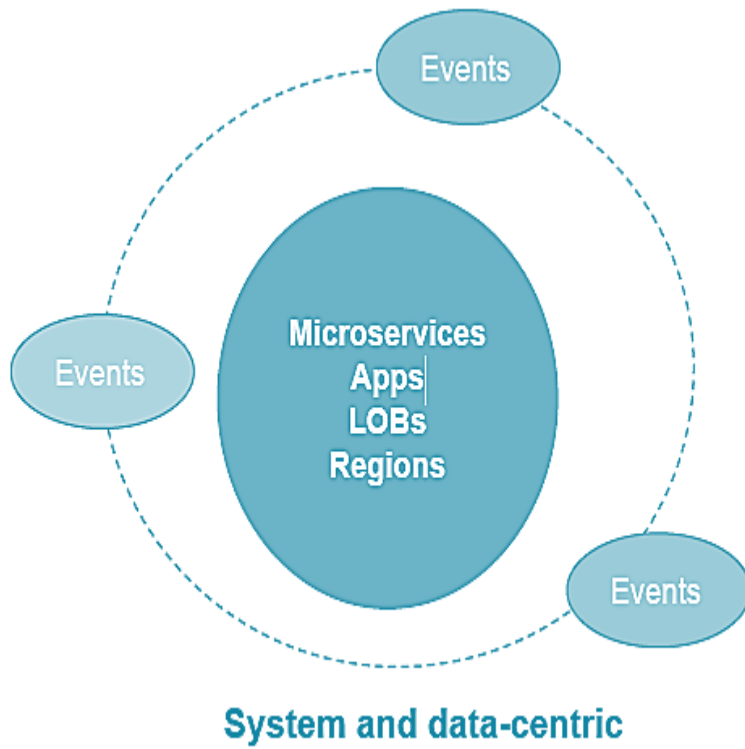
Asynchronous

Microservices

- A small problem domain
- Built and deployed by itself and Runs in its own process
- Uses the technology stack that is most suitable
- Integrates via well--known interfaces
- Owns its own data stores

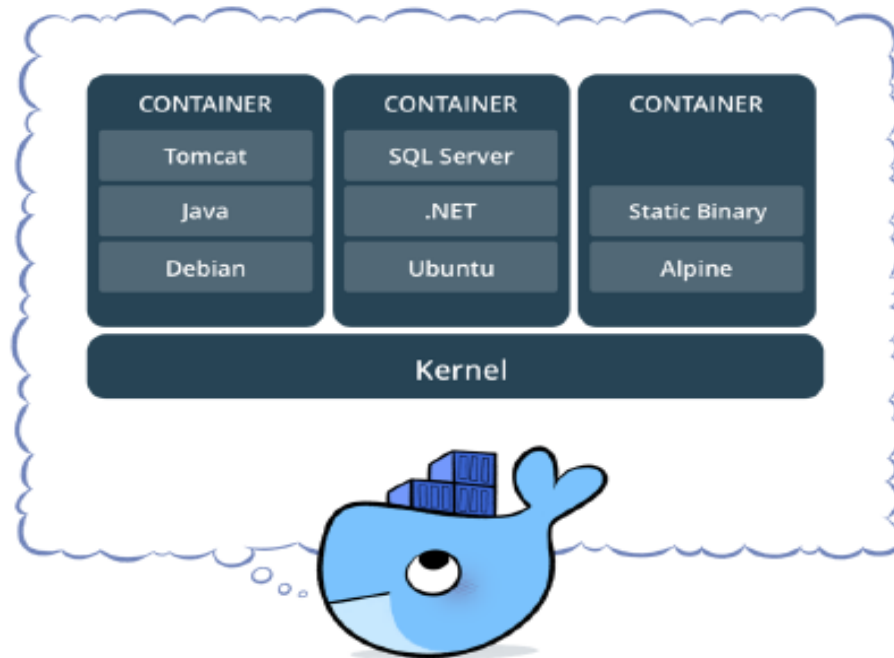
EVENT-DRIVEN

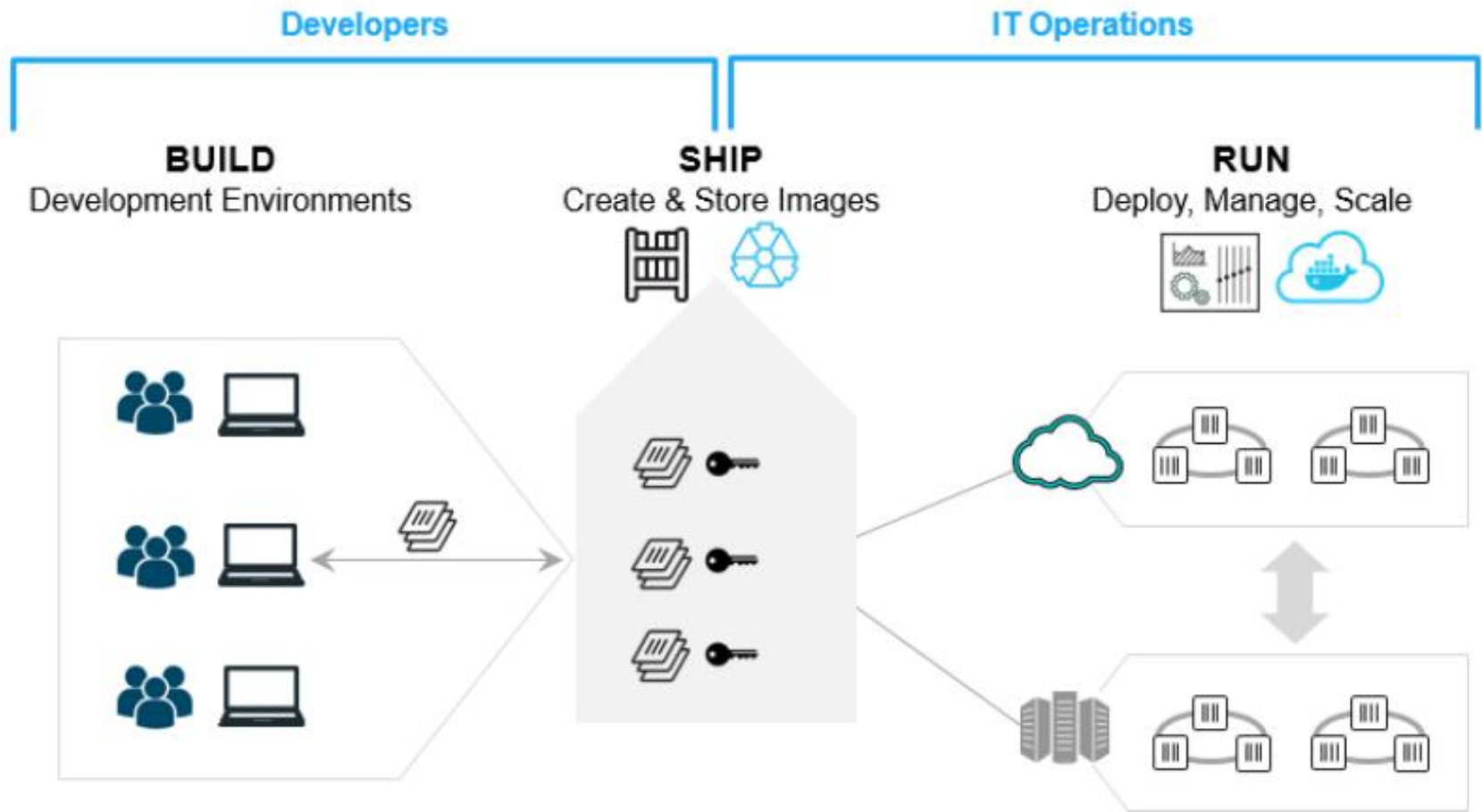
- EDA is aligned with the goals of domain-driven design (Enforce isolation and decoupling between bounded contexts)
- Asynchronous communication paradigm
- Message-driven – asynchronous and non-blocking
- Scalable – scaling out and embracing the network

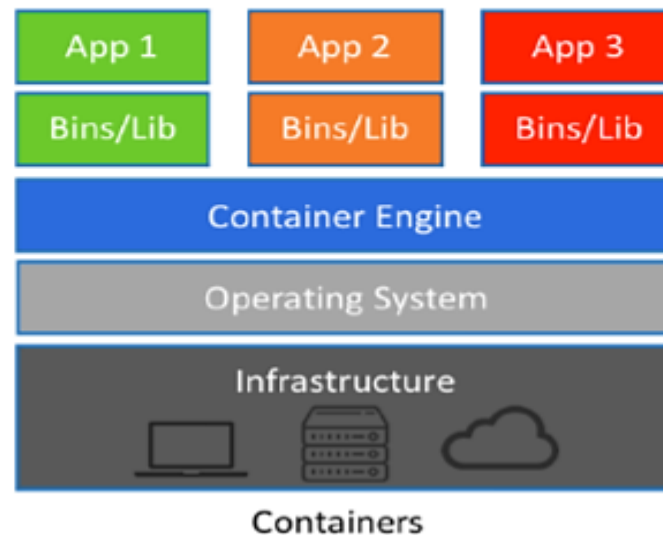
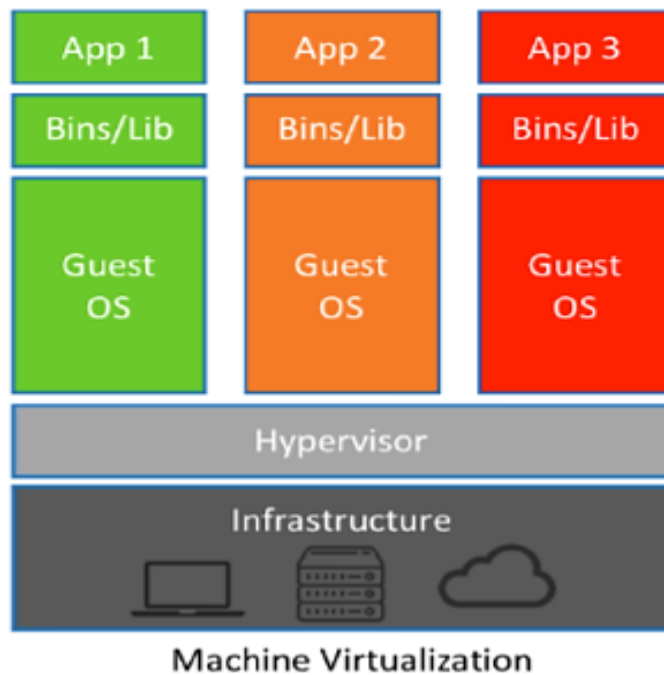


Docker is an open software development platform which help you to **Build**, **Ship** and **Run** your applications along with all its dependencies allowing **portability** among any system. Standardized packaging for software and dependencies via **Containers**.







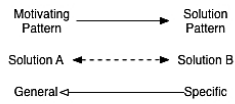
Virtual Machines

- Each VM runs its own OS
- Cannot run more than couple of VMs

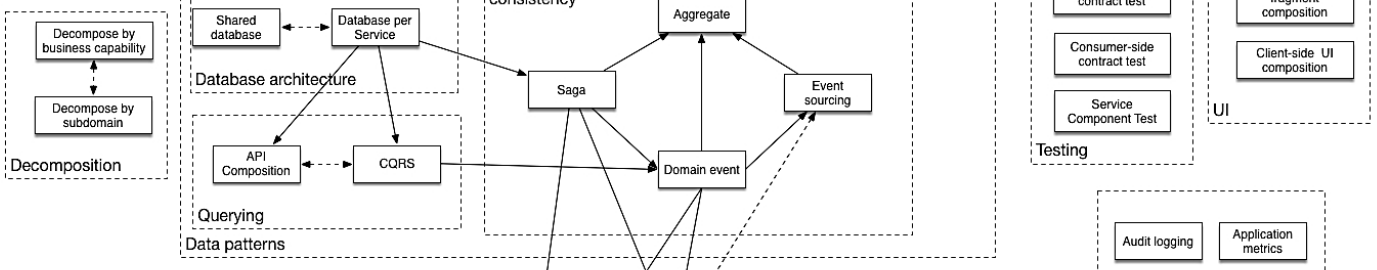
Docker

- Container is just a user space of OS
- Can run many Docker containers

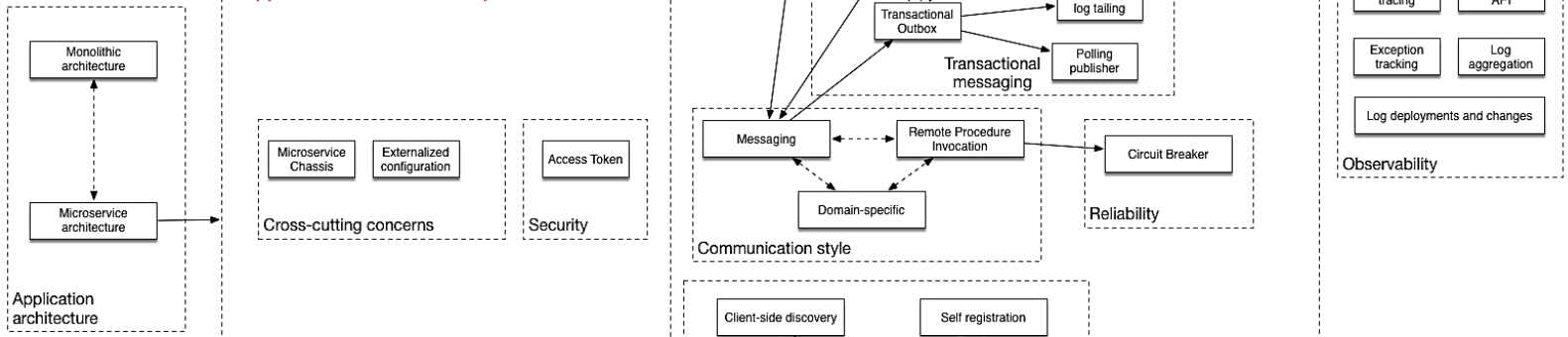
Microservices patterns



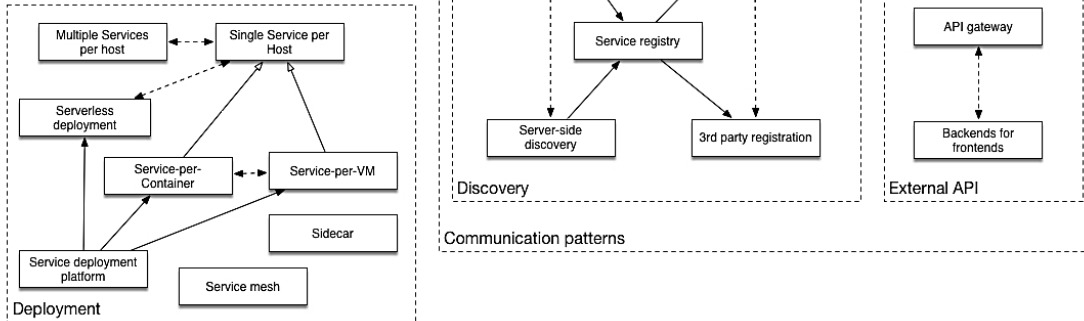
Application patterns



Application Infrastructure patterns



Infrastructure patterns



Microservice patterns

- Identity and Access Management
- Secure Communication Protocols
- Security Monitoring
- Service Discovery Mechanism
- ...

<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-204.pdf>

Security Strategies for Microservices-based Application Systems

NIST Special Publication 800-204 (August 2019)

- “An architecture pattern that allows you functionally decompose the software into a manageable and independent deployable unit”
- **Make EACH program/service do one thing WELL**

Microservices should be:

- Independent deployment.
- Independent Management
- Independent technology stack.
- Independent scalable.

Can you make a change to a service and deploy it by itself without changing anything else?

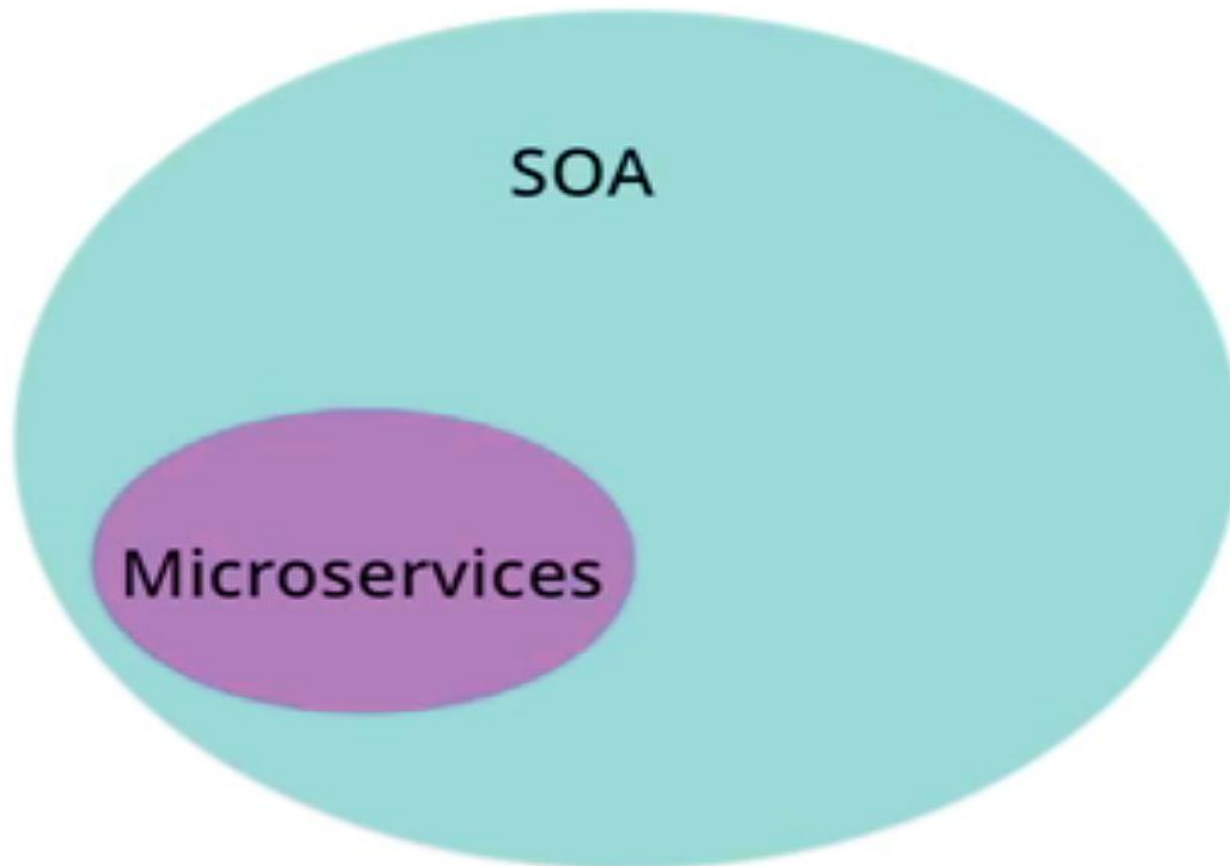
—Sam Newman, *Building Microservices* (O’Reilly)

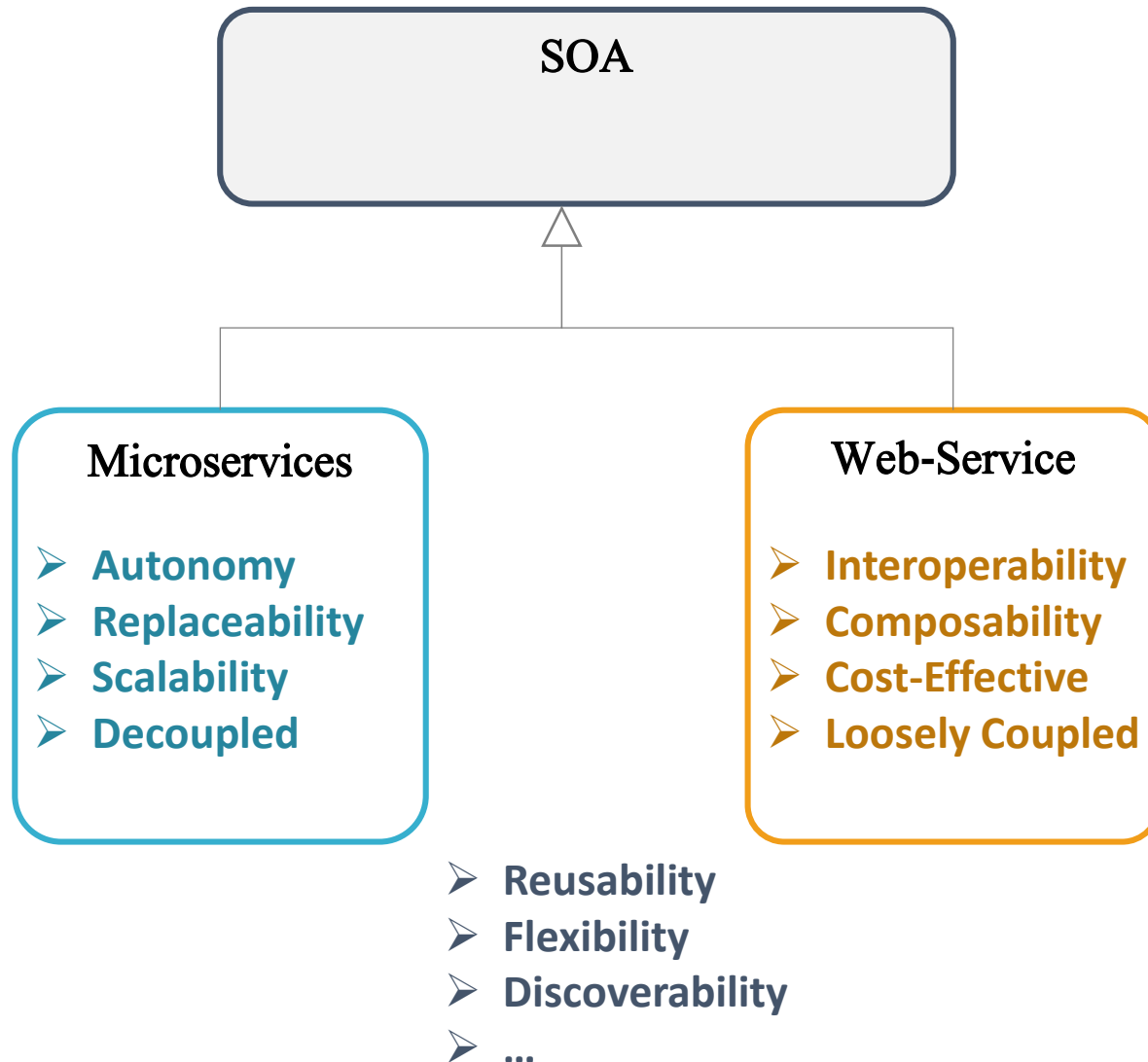


Microservices Vs SOA

Martin Fowler:

We should think about SOA as a superset of microservices.





- ❑ SOA is obsolete.
 - Not at all. Take a look to its primary goals. Is is actually an strong architecture to work with. It is very well aligned to today's needs.

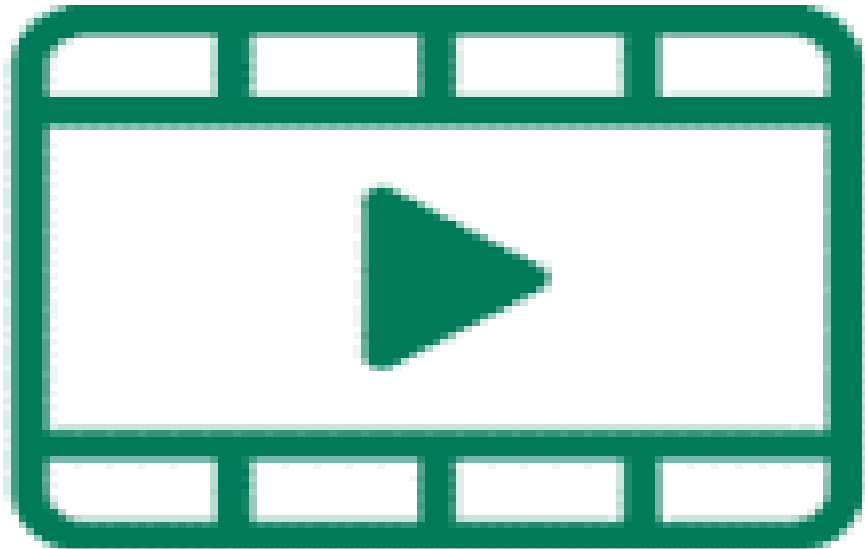
- ❑ SOA is very expensive
 - If you build componetes instead of services, it is definitely as expensive as a the traditional approaches

- ❑ Where is my ROI?
 - Again, that is on the Architects who “defines and design” services

- ❑ SOA is nothing more than a Service Bus
 - Enterprise Service Bus is a design pattern not a product

- ❑ Microservices architecture are better than service oriented architecture.
 - You may get surprised about the similarities of both architectures.





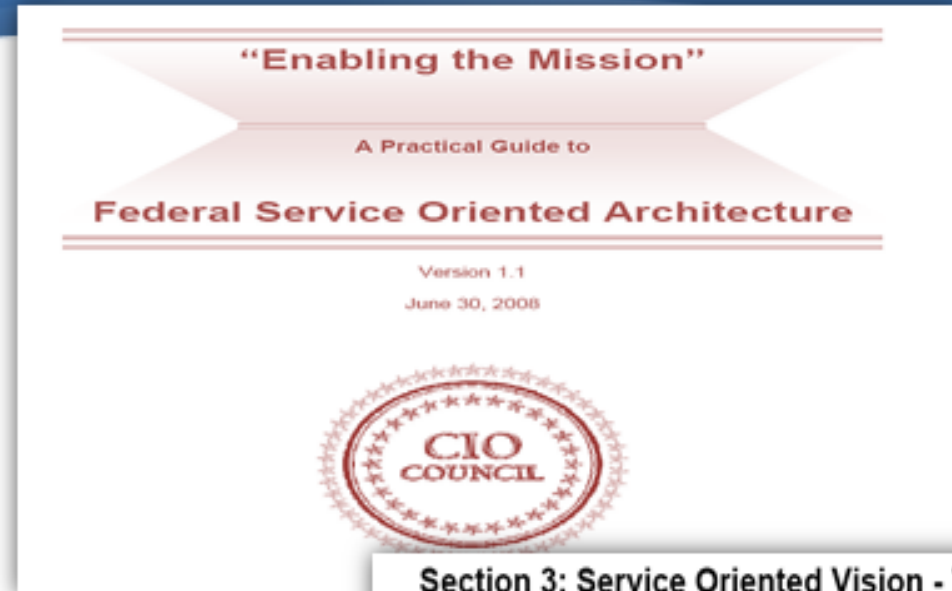


SOEA and Microservices

SOEA Realization: Looking behind

پارادایم سرویس گرایی در سازمان

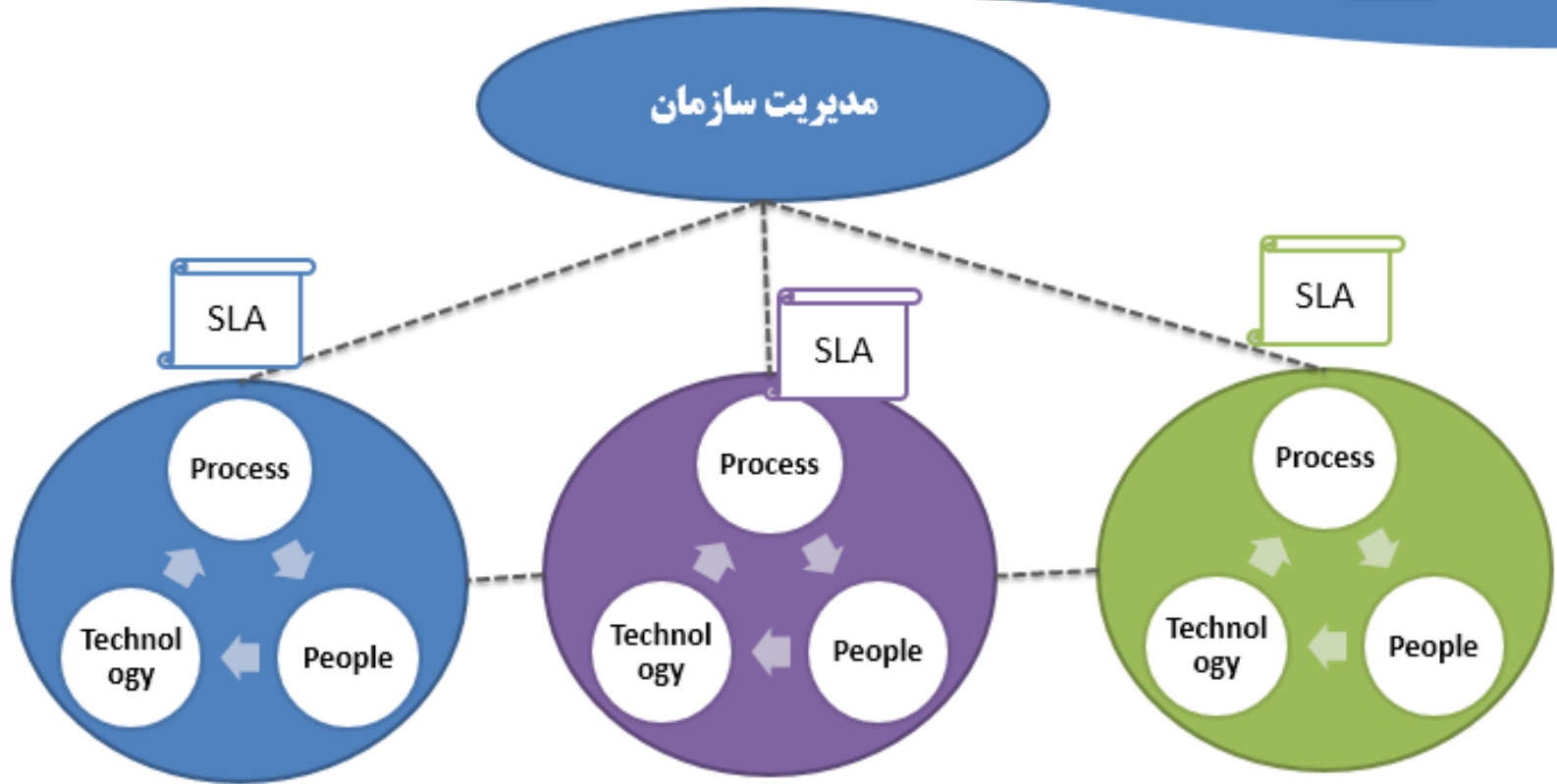




Section 3: Service Oriented Vision - The Target Architecture

- 3.1 Service-Oriented Enterprise (SOE)**
 - 3.1.1 Management, IT and Business are All Service Focused
 - 3.1.2 Sustaining SOE through Federated Governance
 - 3.1.3 Model Based Acquisition Processes
- 3.2 Service Oriented Architecture (SOA).....**
 - 3.2.1 Federal SDLC and EA are Integrated and Support SOA
 - 3.2.2 SOA and Interoperability Will be Well Established
 - 3.2.3 Standard Federal Government Services Will Emerge.....
 - 3.2.4 Model Driven Architecture Will Be Embraced.....
- 3.3 Service-Oriented Infrastructure**
 - 3.3.1 Service Management is Coordinated Throughout the Federal Government...
 - 3.3.2 Increased Collaboration with 3rd Parties.....
 - 3.3.3 Tools to Manage the SOI.....

ساختار سازمانی سرویس گرا



تأثیر معماری سازمانی سرویس گرا در : ساختار تشکیلاتی و مدیریت سازمان

ساختار تشکیلاتی و مدیریت سازمان

بانگاه نوین سرویس گرا

ساختار سازمانی سرویس محور

استقلال و خودمختاری نسبی واحدها

ارزیابی عملکرد مبتنی بر
سرویس (نتیجه)

تغاه به تغییرات به عنوان فرصت

تمرکز بر طراحی و بهبود معماری سازمان
(سرویس، فرایند، داده، سیستم، فناوری)

ساختار تشکیلاتی و مدیریت سازمان

بانگاه فرایند محور

ساختار سازمانی ماتریسی

وابستگی (زنجیره) بین واحدهای
سازمان

ارزیابی نتیجه فرایند

مدیریت و کنترل تغییرات

تمرکز بر مدیریت و بهبود فرایندها

ساختار تشکیلاتی و مدیریت سازمان

بانگاه وظیفه محور

ساختار سازمانی وظیفه محور

ارتباط عمودی (رئیس - مرئوس) بین
پستهای سازمان

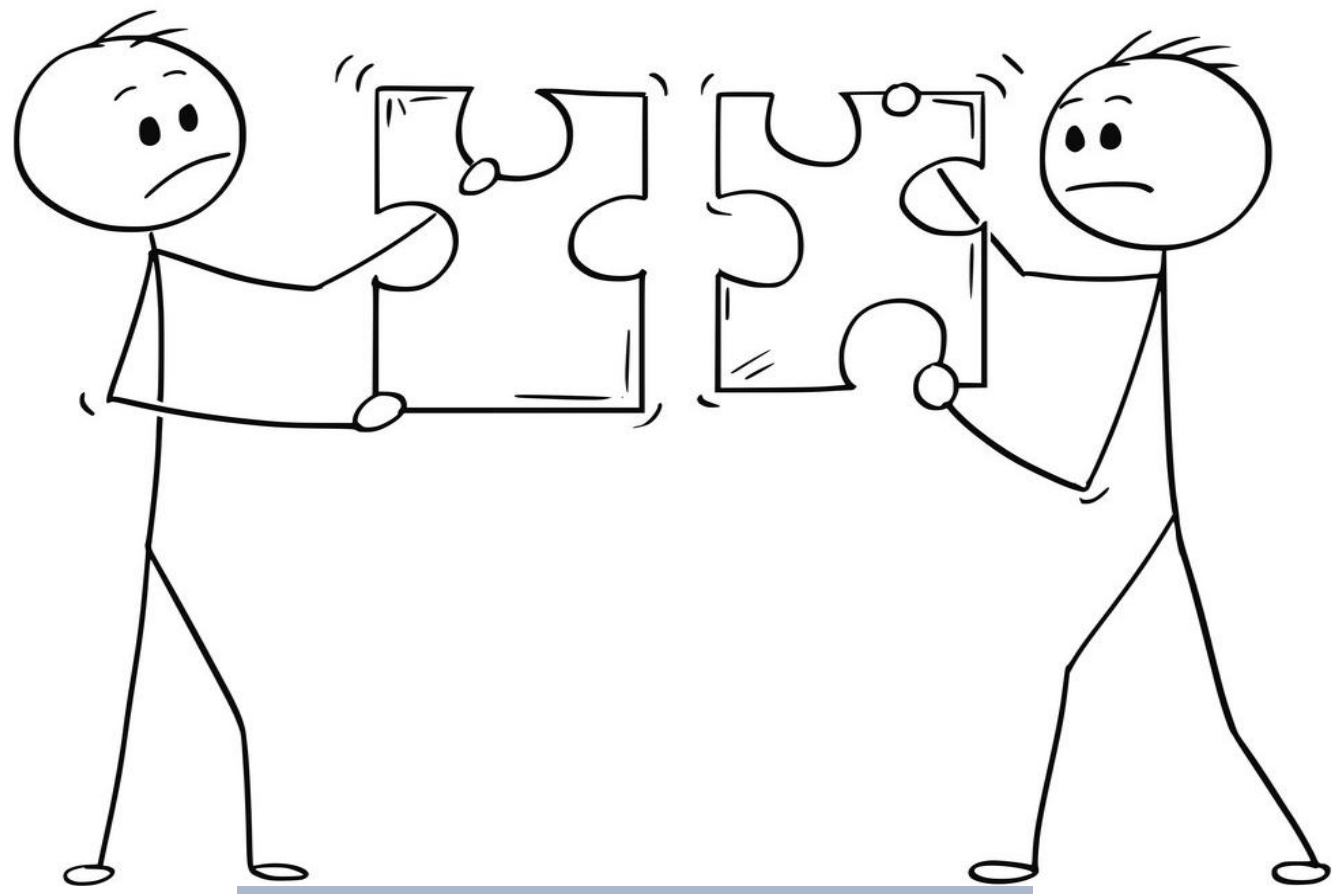
ارزیابی عملکرد پرسنل / واحد سازمانی

تغاه به تغییرات به عنوان تهدید

تمرکز بر مدیریت و بهبود عملکرد
واحدهای سازمانی / پرسنل

Web-Services

SOEA



Unmatched!

A wide, teal-colored diagonal bar runs from the bottom-left towards the top-right of the page, serving as a background for the title text.

MSA Case Study (BIAN)

Financial Institutions



Software Vendors / Service Providers



BIAN & Academic Partners



Business Architecture

Service Domain Definition

Service landscape

Business Scenario

Business Capability Model

Wireframe Model

Information Architecture

Business Object Model

Asset Decomposition Model

Control Record

Application Architecture

Application Capabilities

Logical System

Message Format

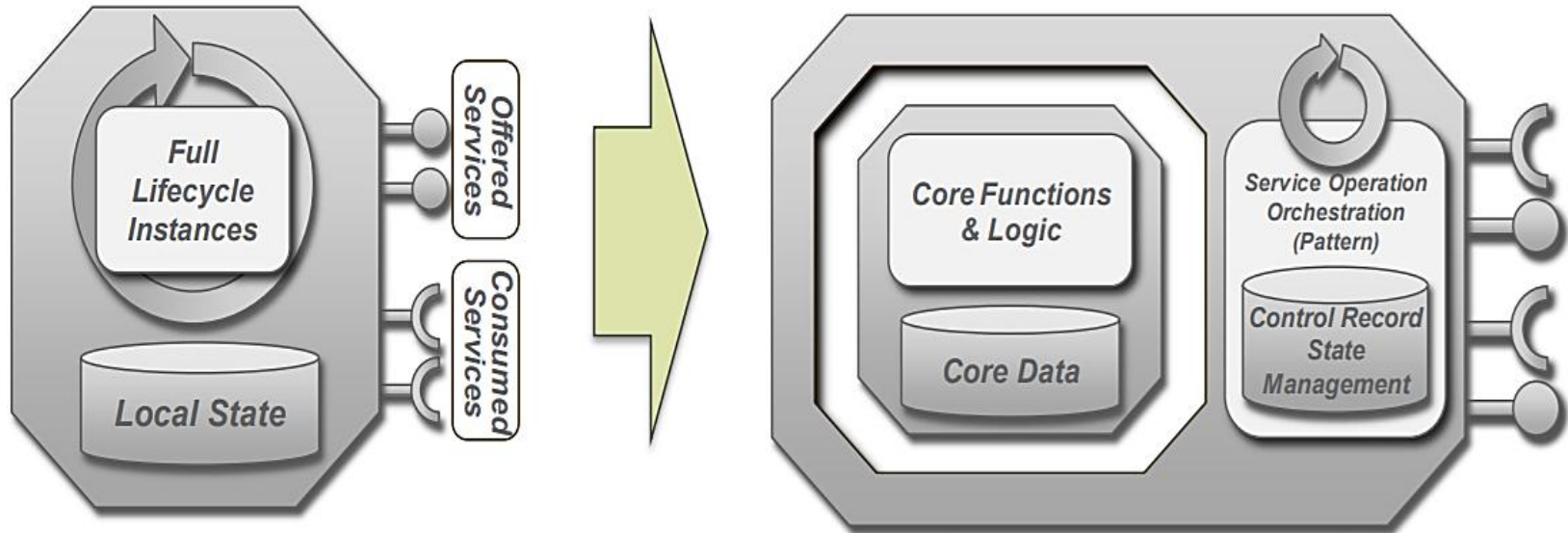
Open API

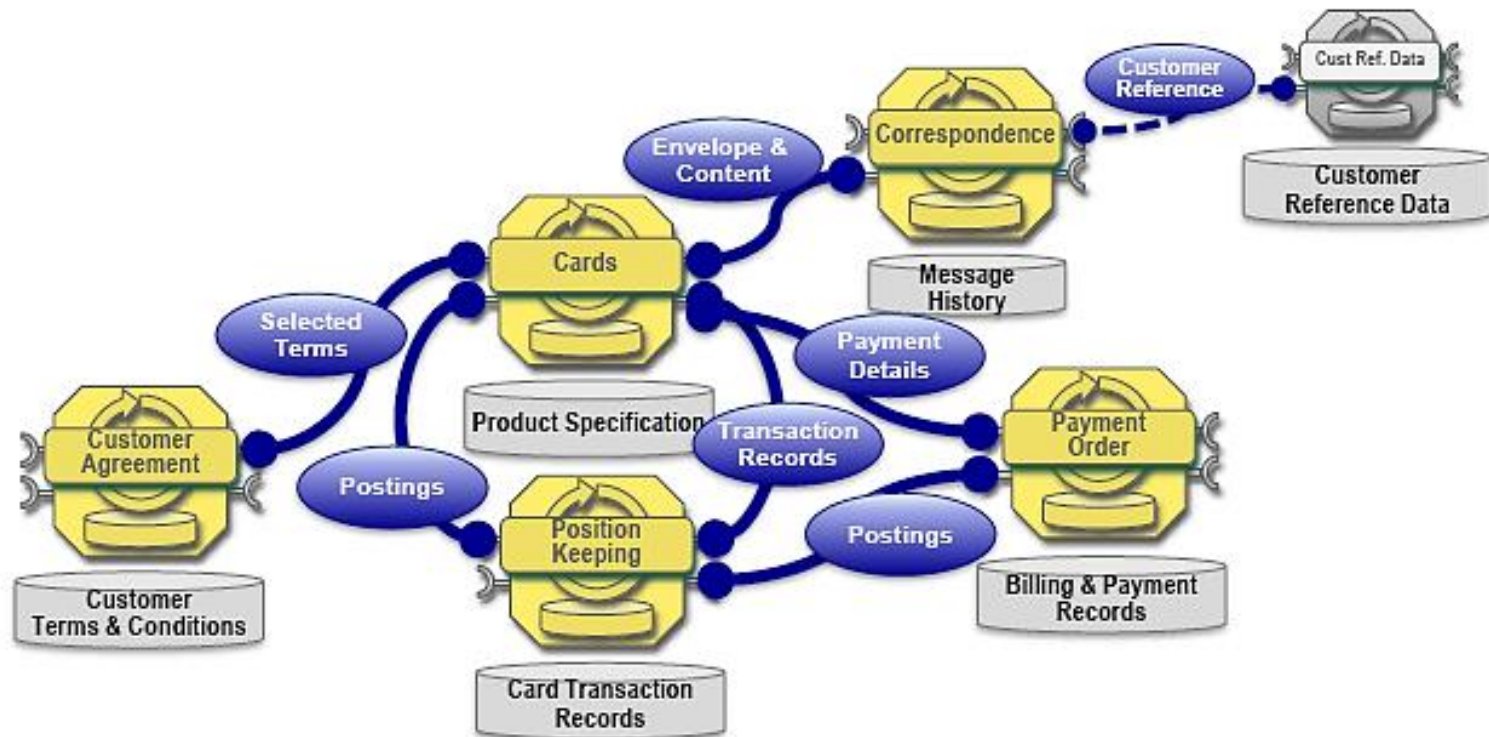
Technology Architecture

Service Protocol Standards

Reference Data	Sales & Service		Operations & Execution			Risk & Compliance	Business Support	
Party	Channel Specific	Marketing	Product Specific Fulfillment			Bank Portfolio & Treasury	IT Mngmt.	Finance
Party Data Mngmt. Syndicate Mngmt. Customer Profile	Branch Location Mngmt. Contact Center Mngmt. Branch Network Mngmt. E-Branch Mngmt. Adv. Voice Services Mngmt. ATM Network Mngmt. Contact Center Operations Branch Location Operations E-Branch Operations Adv. Voice Services Operat. ATM Network Operations Branch Currency Mngmt. Branch Currency Distribution Prod. Inventory Item Mngmt. Prod. Inventory Distribution Card Terminal Card Terminal Operation	Business Development Brand Mngmt. Advertising Promotional Events Prospect Campaign Mngmt. Prospect Campaign Design Customer Campaign Mngmt. Customer Campaign Design Customer Surveys	Loans & Deposits	Investment Mngmt.	Trade Banking	Corporate Treasury Analysis Corporate Treasury Asset Securitization Asset & Liability Mngmt. Bank Portfolio Analysis Bank Portfolio Administration Stock Lending/Repos	IT Systems Direction IT Stds & Guidelines Systems Administration Development Environment System Deployment Production Release System Deployment Systems Operations Platform Operations Systems Help Desk Systems Assurance Internal Network Operation	Financial Statements Financial Control Financial Compliance Enterprise Tax Administration
External Agency			Loan Leasing Current Account Deposit Account Corporate Current Account Consumer Loan Corporate Loan Corporate Deposits Corporate Lease Merchandising Loan Fiduciary Agreement Savings Account	Investm. Portfolio Planning Investm. Portfolio Analysis Investm. Portfolio Mngmt. eTrading Workbench	Letter of Credit Bank Guarantee Trade Finance Credit Mngmt. Credit Facility Project Finance Limit & Exposure Mngmt. Syndicated Loan Cash Mngmt. & Account Services Direct Debit Mandate Check Debit Cheque Lock Box Factoring	Models	Human Resource Mngmt.	
Information Provider Admin Syndicate Mngmt. Interbank Relationship Mngmt. Correspondent Bank Relationship Mngmt. Correspondent Bank Data Mngmt. Sub Custodian Agreement Product Service Agency Product Broker Agreement Contractor/Supplier Agreement		Sales	Wholesale Trading	Cards	Corp. Financing & Advisory Services	Market Risk Models Financial Inst. Valuation Models Gap Analysis Credit Risk Models Liquidity Risk Models Economic Capital Business Risk Models Customer Behavior Models Fraud Models Credit/Margin Management Production Risk Models Operational Risk Models Contribution Models	Human Resources Direction Employee Assignment Employee Data Management Employee/Contractor Contract Employee Certification Employee Evaluation Employee Payroll and Incentives Travel and Expenses Employee Access Employee Benefits Workforce Training Recruitment	
Market Data	Cross Channel	Customer Mngmt.	Market Operations	Consumer Services		Market Risk Models Financial Inst. Valuation Models Gap Analysis Credit Risk Models Liquidity Risk Models Economic Capital Business Risk Models Customer Behavior Models Fraud Models Credit/Margin Management Production Risk Models Operational Risk Models Contribution Models	Non IT & HR Enterprise Services	Knowledge & IP Mngmt.
Information Provider Operation Market Information Mngmt. Financial Market Analysis Financial Market Research Quant Model Market Data Switch Admin Market Data Switch Ops Financial Instr. Ref Data Mngmt. Counterparty Administration Public Reference Data Mngmt. Location Data Mngmt.	Party Authentication Transaction Authorization Point of Service Servicing Event History Servicing Activity Analysis Sales Product	Prospect Campaign Execution Prospect Mngmt. Lead/Opportunity Mngmt. Customer Campaign Execution Customer Offer Sales Planning Underwriting Commission Agreement Commissions Product Matching Product Expert Sales Support Product Sales Support Sales Product	Trading Book Oversight Trading Models Dealer Workbench Quote Mngmt. Suitability Checking Credit Risk Operations Market Making ECM / DCM Program Trading Traded Position Mngmt. Market Order Market Order Execution	Credit/Charge Card Card Authorization Card Capture Card Billing & Payments Merchant Relations Merchant Acquiring Card Network Participant	Corporate Finance M&A Advisory Corp. Tax Advisory Public Offering Private Placement	Legal Compliance Internal Audit Security Advisory Security Assurance Approved Supplier Directory Procurement Company Billing & Payments Fixed Asset Register		Mngmt. Manual Intellectual Property Portfolio Knowledge Exchange
Product Management			Market Operations			Business Analysis	Buildings, Equipment and Facilities	Corporate Relations
Product Design Product Deployment Product Training Product Quality Assurance Discount Pricing Product Directory Special Pricing Conditions		Customer Relationship Mngmt. Customer Prod./Service Eligibility Customer Agreement Sales Product Agreement Customer Access Entitlement Customer Behavioral Insights Customer Credit Rating Account Recovery Customer Event History Customer Reference Data Mgmt Customer Precedents Customer Proposition	Mutual Fund Admin. Hedge Fund Admin. Unit Trust Admin. Trade Confirmation Matching Order Allocation Settlement Obligation Mngmt. Securities Dlvry & Receipt Mngmt. Securities Falls Processing Trade/Price Reportg. Custody Administration Corporate Events Financial Instrument Valuation	Corporate Trust Services Remittance Currency Exchange Bank Drafts & Trvl. Checks Brokered Product Consumer Investments Consumer Tax Handling Consumer Advisory Services Trust Services Service Product		Segment Direction Product Portfolio Customer Portfolio Branch Portfolio Channel Portfolio Competitor Analysis Market Research Market Analysis Contribution Analysis	Property Portfolio Site Operations Site Administration Equipment Administration Equipment Maintenance Utilities Administration Building Maintenance	Corporate Communications Corporate Alliance/Stakeholder Corporate Relationship Regulatory and Legal Authority Investor Relations
		Servicing	Cross Product Operations			Regulations & Compliance	Business Command & Control	Business Direction
		Servicing Issue Customer Case Mngmt. Case Root Cause Analysis Customer Case Card Case Customer Order Payment Order	Payments	Account Mngmt.	Operational Services	Guideline Compliance Regulatory Compliance Compliance Reporting Regulatory Reporting Fraud/AML Resolution Financial Accounting	Organization Direction Business Unit Financial Analysis Business Unit Financial Operations Business Unit Accounting Business Unit Direction Business Unit Management	Corporate Strategy Corporate Policies Product & Services Direction Business Architecture Continuity Planning
			Payments Execution Financial Message Analysis Financial Gateway Correspondent Bank Cheque Processing Central Cash Handling ACH Fulfillment Card eCommerce	Position Keeping Reward Points Account Accounts Receivable Account Reconciliation Counterparty Risk Position Mngmt. Fraud Detection Transaction Enqine	Issued Device Admin Issued Device Tracking Disbursement Open Item Mngmt. Leasing Item Administration Dunning Customer Billing Rewards Points Awards & Red.		Document Mngmt. & Archive	Document Services Correspondence

- The building block of the BIAN SOA is the Service Domain – it is a conceptual specification of a functional partition.
- A critical aspect of the Service Domain's definition is to ensure effective encapsulation.





- BIAN Service Domain Designs are Suited to **Highly Distributed Architecture**
- With service based designs each Service Domain 'encapsulates' its own data.

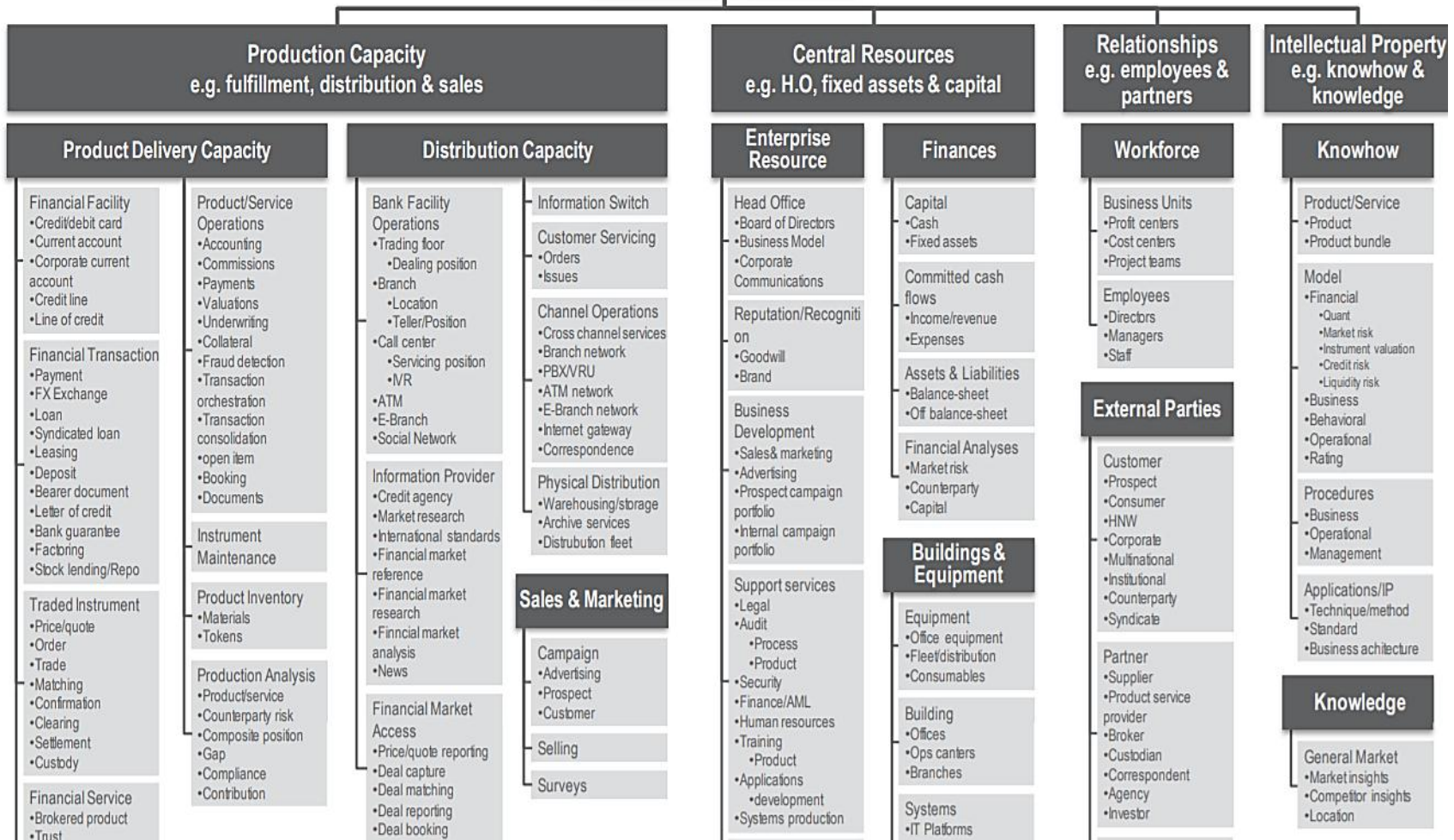
Functional pattern of Service Domain



Functional Pattern	Brief Definition	Information Profile			
		Generic Artifact	Definition/Description	Behavior Qualifier Type	Behavior Qualifier Type Description
DIRECT	Define the strategy	Strategy	The purpose and mission for the enterprise including its competitive positioning and bases for competing in the market	Goals	A collection of goals and objectives for the enterprise and its main divisions
MANAGE	Oversee activity	Management Plan/Charter	The management and oversight while running an operational unit of an enterprise	Duties	A collection of one or more responsibilities or tasks under management
ADMINISTER	Administer activity	Administrative Plan	The clerical support for an operational unit/function of an enterprise	Routines	A collection of one or more clerical routines that are to be followed to administer the operational unit/function
OPERATE	Operate facility	Operating Session/ Facility	The operation of a technical/automated facility employed/provided by an enterprise	Functions	The collection of operational services/functions offered by the operational facility
PROCESS	Process work	Procedure	The performance of a supporting office activity within the enterprise (not product/service fulfillment specific)	Worksteps	The main worksteps to be followed in the execution of the procedure
REGISTER	Register details	Directory	A registry of items recording key reference information and properties relating to each	Properties	The properties/reference details recorded in the registry for items
DESIGN	Design solutions	Specification	A specification of a product or service offering covering all aspects required for its use	Aspects	The main design elements/views making up the overall specification
DEVELOP	Execute projects	Development Project	A discrete or bounded effort with a defined remit and intended purpose/outcome	Deliverables	A collection of one or more deliverables that may be further defined in terms of an approach to be followed to create them
ASSESS	Test compliance	Assessment	A formal evaluation or test of a subject against a predefined set of properties or performance criteria	Tests	A collection of one or more tests' evaluations that can be made to certify a subject
MAINTAIN	Maintain resources	Maintenance Agreement	A service to provide maintenance and repair to operational capabilities/technology	Tasks	A collection of tasks needed to support maintenance and repair work
TRACK	Log events	Log	A mechanism to track and record specific events and if necessary maintain associated derived/accumulated values	Events	A collection of the events/transactions recorded by the log
ANALYSE	Analyse activity	Analysis	A service to apply specific types of analysis against a set of provided data related to an item or activity	Algorithms	A collection of models/calculations/algorithms that can be applied to a subject or activity
MONITOR	Measure resources	Measurement	A mechanism to track and report on the state or dynamic property of some item or activity	Signals	A collection of information feeds/measures that can be used to track the status of one or more items/entities
AGREE TERMS	Govern activity	Agreement	A service to apply specific laws and/or rules to define the terms and conditions that govern a business service or activity	Terms	A collection of terms (within some jurisdiction) that can be selected and configured to define a contract/agreement
ENROLL	Register members	Membership	A registry of entities that qualify for membership to a group with a recognised business purpose or categorization	Clauses	A collection of clauses that govern the eligibility for membership
ALLOCATE	Allocate resources	Allocation	A service to track the availability and allocate business resources (staff and/or facilities) on request	Assignments	A collection of one or more specific assignments of inventory allowing for different allocation types and states
FULFILL	Fulfill agreement	Fulfillment Arrangement	The fulfillment of a financial facility, including customer initiated and internally triggered actions	Features	The product features/services available with a financial facility
TRANSACT	Execute transactions	Transaction	The execution of a financial transaction	Tasks/Steps	The sub-tasks involved in the execution of the financial transaction

Asset type decomposition

Asset Types



Enterprise Resource

- Head Office
 - Board of Directors
 - Business Model
 - Corporate Communications
- Reputation/Recognition
 - Goodwill
 - Brand
- Business Development
 - Sales & marketing
 - Advertising
 - Prospect campaign portfolio
 - Internal campaign portfolio
- Support services
 - Legal
 - Audit
 - Process
 - Product
 - Security
 - Finance/AML
 - Human resources
 - Training
 - Product
 - Applications
 - development
 - Systems production

Finances

- Capital
 - Cash
 - Fixed assets
- Committed cash flows
 - Income/revenue
 - Expenses
- Assets & Liabilities
 - Balance-sheet
 - Off balance-sheet
- Financial Analyses
 - Market risk
 - Counterparty
 - Capital

Buildings & Equipment

- Equipment
 - Office equipment
 - Fleet/distribution
 - Consumables
- Building
 - Offices
 - Ops centers
 - Branches
- Systems
 - IT Platforms

Relationships e.g. employees & partners

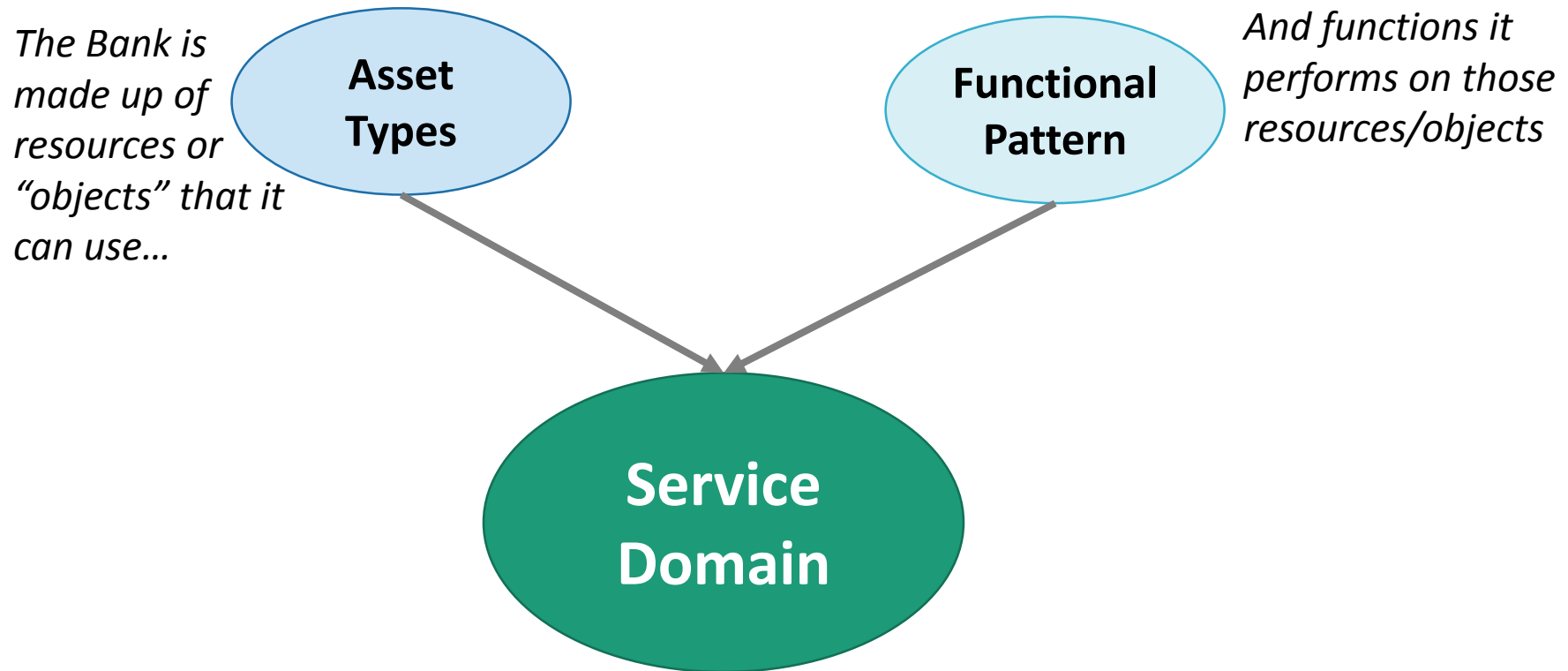
- Workforce**
 - Business Units
 - Profit centers
 - Cost centers
 - Project teams
 - Employees
 - Directors
 - Managers
 - Staff

External Parties

- Customer
 - Prospect
 - Consumer
 - HNW
 - Corporate
 - Multinational
 - Institutional
 - Counterparty
 - Syndicate
- Partner
 - Supplier
 - Product service provider
 - Broker
 - Custodian
 - Correspondent
 - Agency
 - Investor

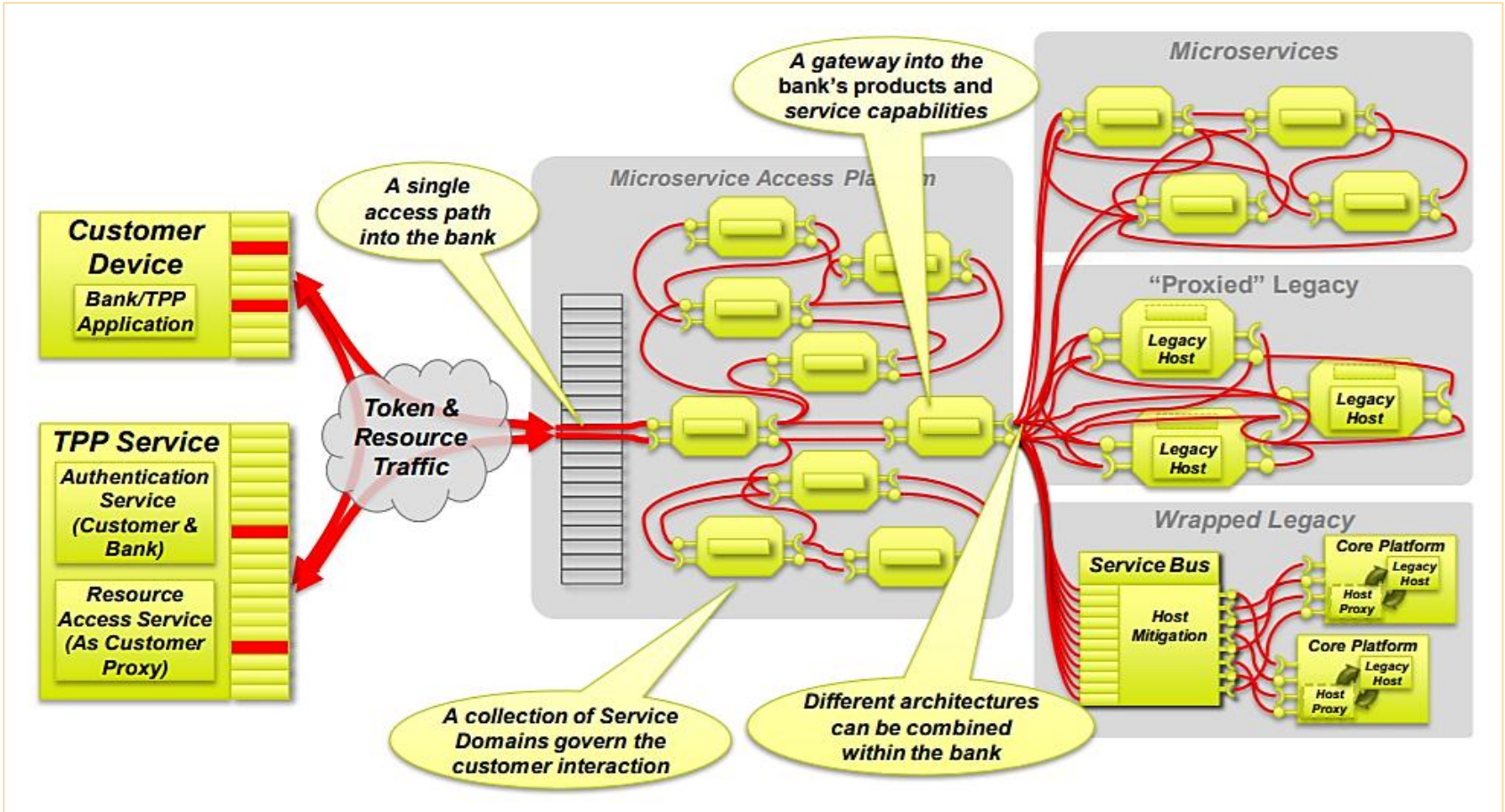
Intellectual Property e.g. knowhow & knowledge

- Knowhow**
 - Product/Service
 - Product
 - Product bundle
 - Model
 - Financial
 - Quant
 - Market risk
 - Instrument valuation
 - Credit risk
 - Liquidity risk
 - Business
 - Behavioral
 - Operational
 - Rating
 - Procedures
 - Business
 - Operational
 - Management
 - Applications/IP
 - Technique/method
 - Standard
 - Business architecture
- Knowledge**
 - General Market
 - Market insights
 - Competitor insights
 - Location



A Service Domain is a discrete and **'elemental'** business capability that exacts or creates value by **"doing something to something"**

Implementation Architecture



Let's talk about Microservices MARKET!



	Close	Net chg	% chn
son ll	10.94	-.97	-7.9
exstarB	4.64	-.40	

Category	Value
Money Markets	0.67
6-Month CD	0.77

microservices
Search term

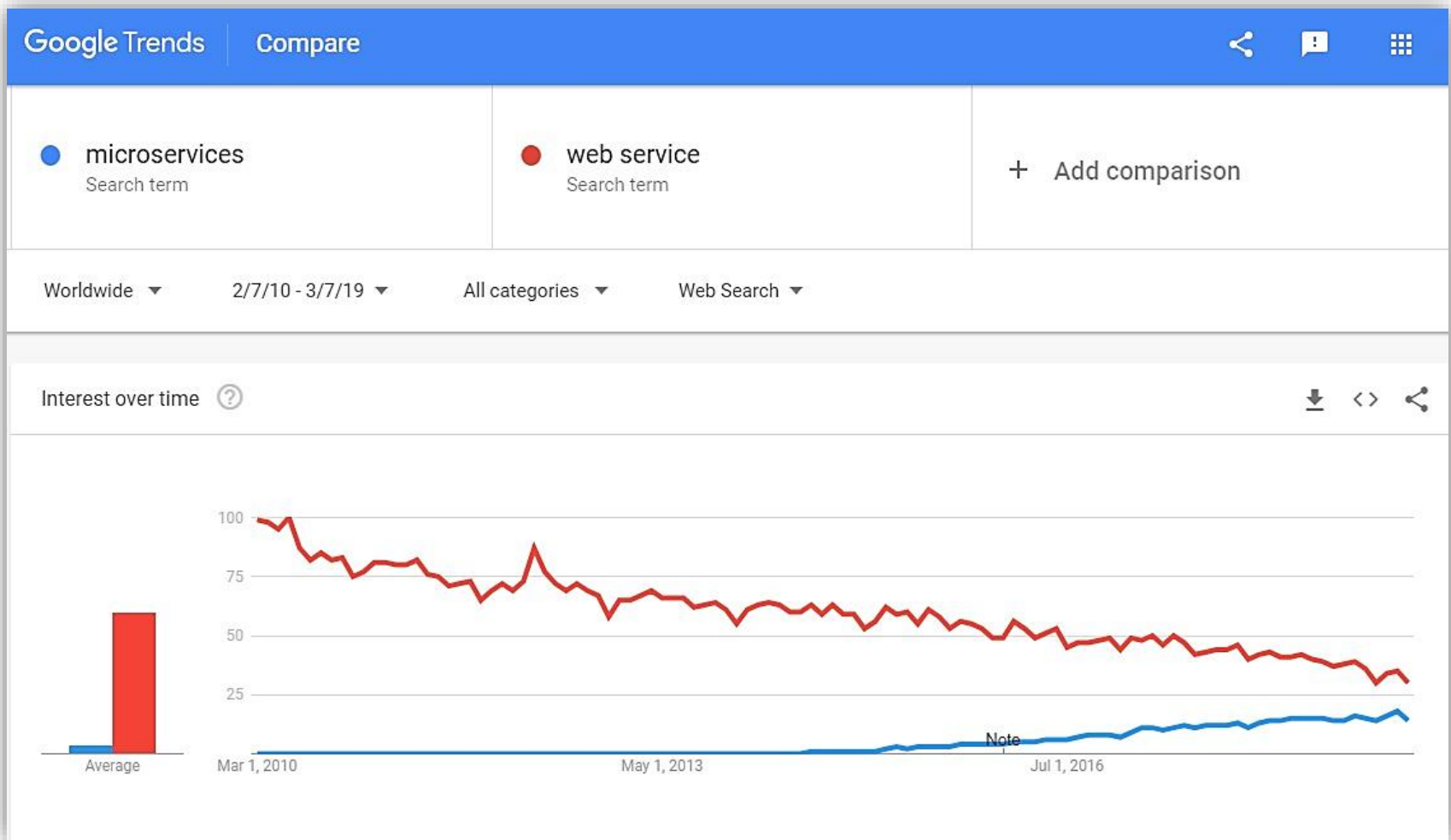
+ Compare

Worldwide ▾ Past 5 years ▾ All categories ▾ Web Search ▾

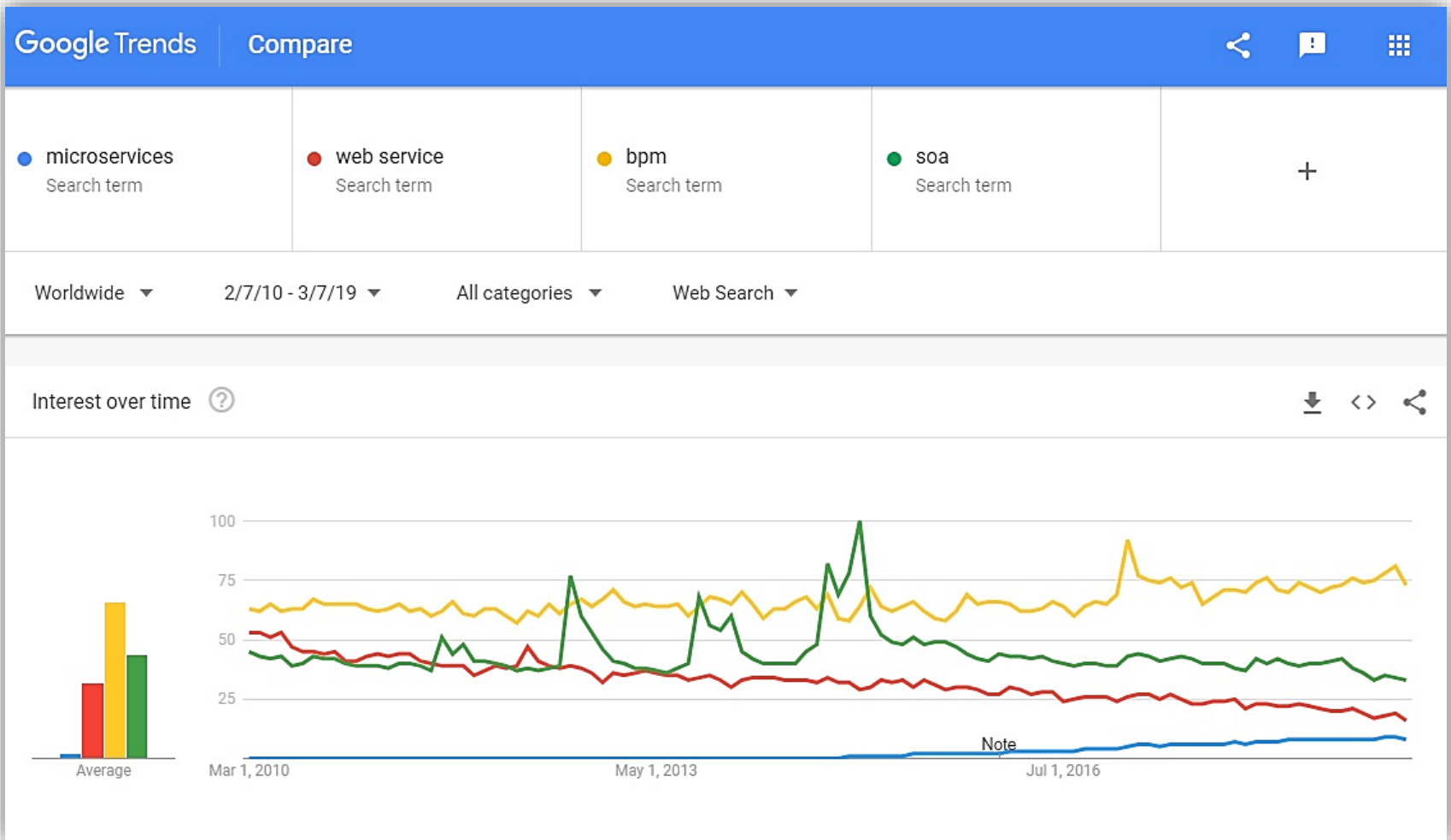
Interest over time



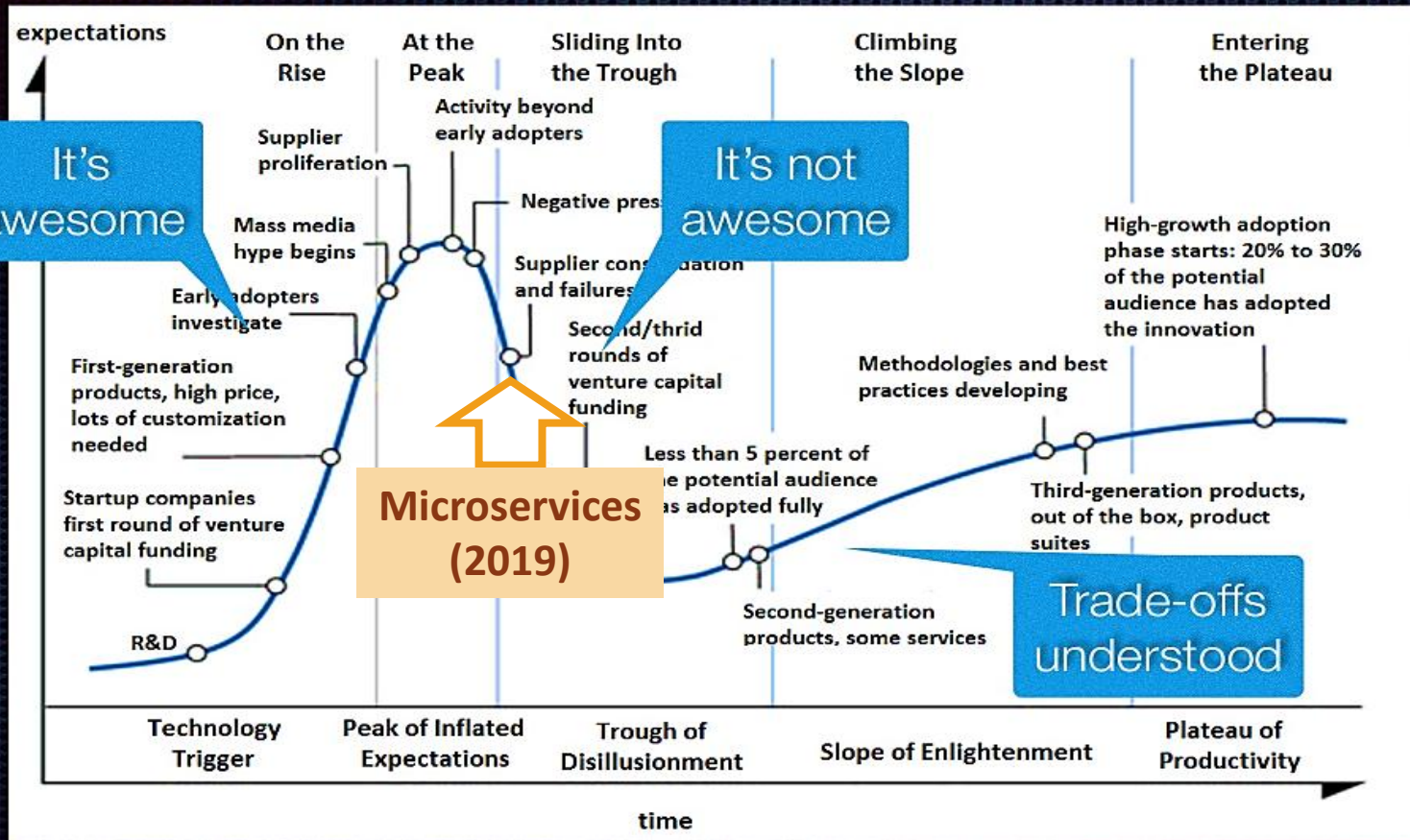
Web Service and Microservices



BPM and SOA still on Top



Gartner Hype Cycle



Conclusion

Architectural Challenges

- Testing, logging, monitoring, security, versioning – all become much harder
- Distributed System is hard to maintain
- A lot of duplicated effort since each team is independent and goal is to minimize dependencies

Organizational Challenges

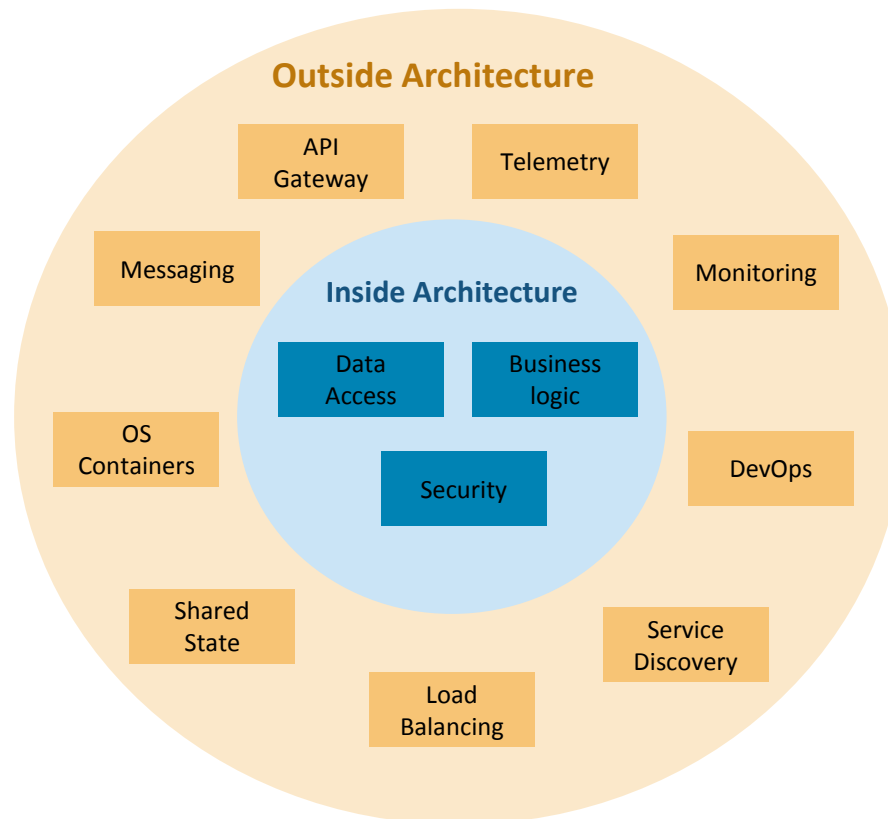
- Most organizations are organized around horizontal technology layers – need to build small product-focused teams
- Much higher skills required
- Many developers will not want to do production support

Delivery Challenges

- Microservices don't eliminate complexity – they simply moved it and often add to it
- Monolithic applications allow architects to deal with complexity in one body of code
- Microservices force the move to distributed computing

Services are simple, surrounding architecture is not

- Building services is rather straightforward using one of the multiple frameworks currently available
- However, there is a relatively large amount of concerns and capabilities, some of them unique to microservices, that should be addressed before we decide to build microservices



“Microservices removes the need for an Enterprise Service Bus”



Don't confuse the Integration pattern with the Architecture style

“Microservices solves the problems of SOA”



Don't confuse improper SOA deployments as problems with SOA

“Companies like Netflix and LinkedIn use microservices, so we should too”



Netflix and LinkedIn are in the platform business. Is that your business too?

“We must choose microservices, or SOA”

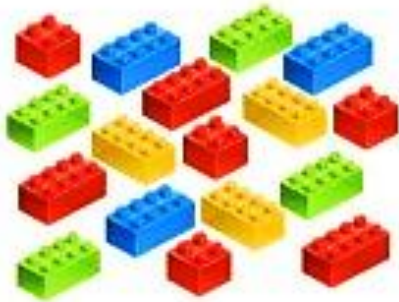


MSA is a realization of SOA

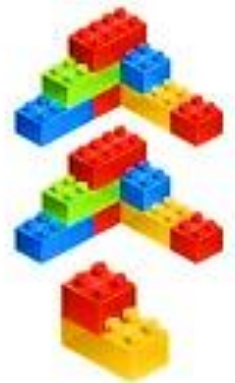
Microservices(Service-Oriented) Paradigm!



Legacy Apps



**SOA
(Microservices)**



New Apps

- Martin Fowler: <https://martinfowler.com/>
- Chris Richardson: <https://microservices.io>
- Building Microservices, Sam Newman (2015)
- Microservices for the Enterprise, Kasun Indrasiri & Prabath Siriwardena (2018)
- Vertically Integrated Architectures, Jos Jong (2019)
- Building Microservices with ASP.NET Core, Kevin Hoffman (2017)
- Building Evolutionary Architectures: Support Constant Change, Neal Ford, Rebecca Parsons, and Patrick Kua (2017)
- <https://developer.ibm.com/technologies/microservices>
- <https://www.redhat.com/en/topics/microservices>
- <https://aws.amazon.com/microservices/>
- <https://docs.oracle.com/en/solutions/learn-architect-microservice/index.html>
- <https://wso2.com>
- <https://camunda.com/learn/whitepapers/microservices-and-bpm>

باتشکر از وقت و حوصله شما عزیزان امیر مهجوریان

آزمایشگاه معماری سازمانی سرویس گرا

دانشگاه شهید بهشتی - دانشکده مهندسی و علوم

کامپیوتر - طبقه دوم - اتاق ۲۱۲

تلفن: ۲۲۴۰۹۶۰۹ ، ۲۲۴۲۴۵۷۲

<http://soea.sbu.ac.ir>



mahjorian@ieaf.ir , mahjorian@karizsystem.ir