

Agile IT with MicroServices, Opportunities & Risks

Sam Rostam

srostan@shaw.ca

Cloud & Enterprise Integration Consultant & Trainer

Certified SOA & Cloud Architect

Certified Big Data Professional

MSc @SFU & PhD Studies – Partial @UBC

Topics

The context & the complexity ..

Digital Transformation & Modernization

The Layering principles

The Agile IT - What it is not!

Traditional vs. Distributed Computing

Service Oriented Computing – SOC

MicroService vs. SOA?

How does it relate to Mediation platforms/ESBs ?

Concept of Containers

What about the business processes?

What is a MUST for using MicroServices?

Digital Transformation

As established enterprises in all industries begin to evolve themselves into the successful **Digital Organizations** of the future they need to begin with the realization that the road to becoming a Digital Business goes through their IT functions.

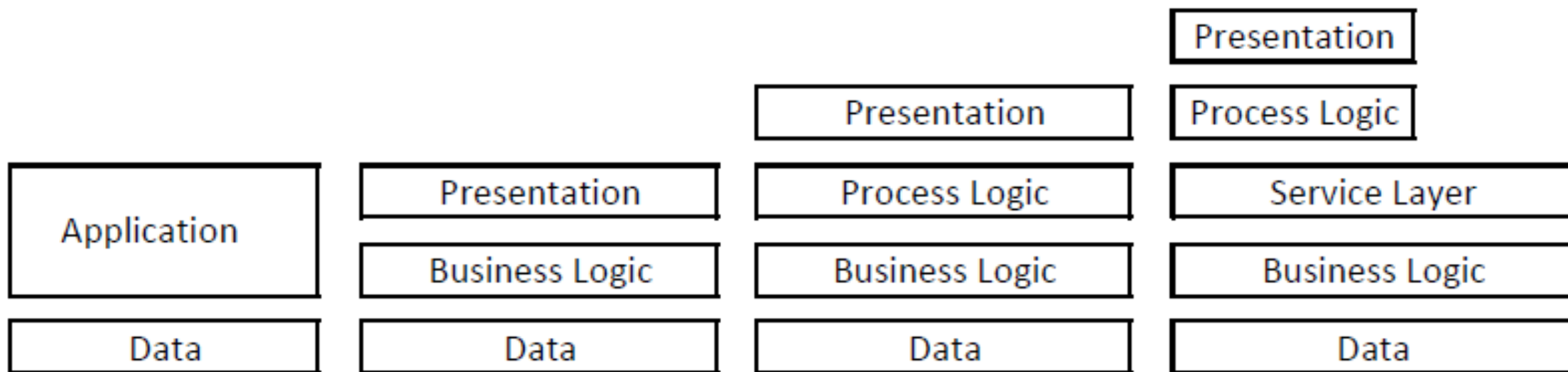
However, many of these incumbents are saddled with IT that has organizational structures, management models, operational processes, workforces and systems that were built to solve “turn of the century” problems of the past.

Many analysts and industry experts have recognized the need for a new model to manage IT in their Businesses and have proposed approaches to understand and manage a hybrid IT environment that includes slower legacy applications and infrastructure in combination with today’s rapidly evolving Digital-first, mobile-first and analytics-enabled applications.

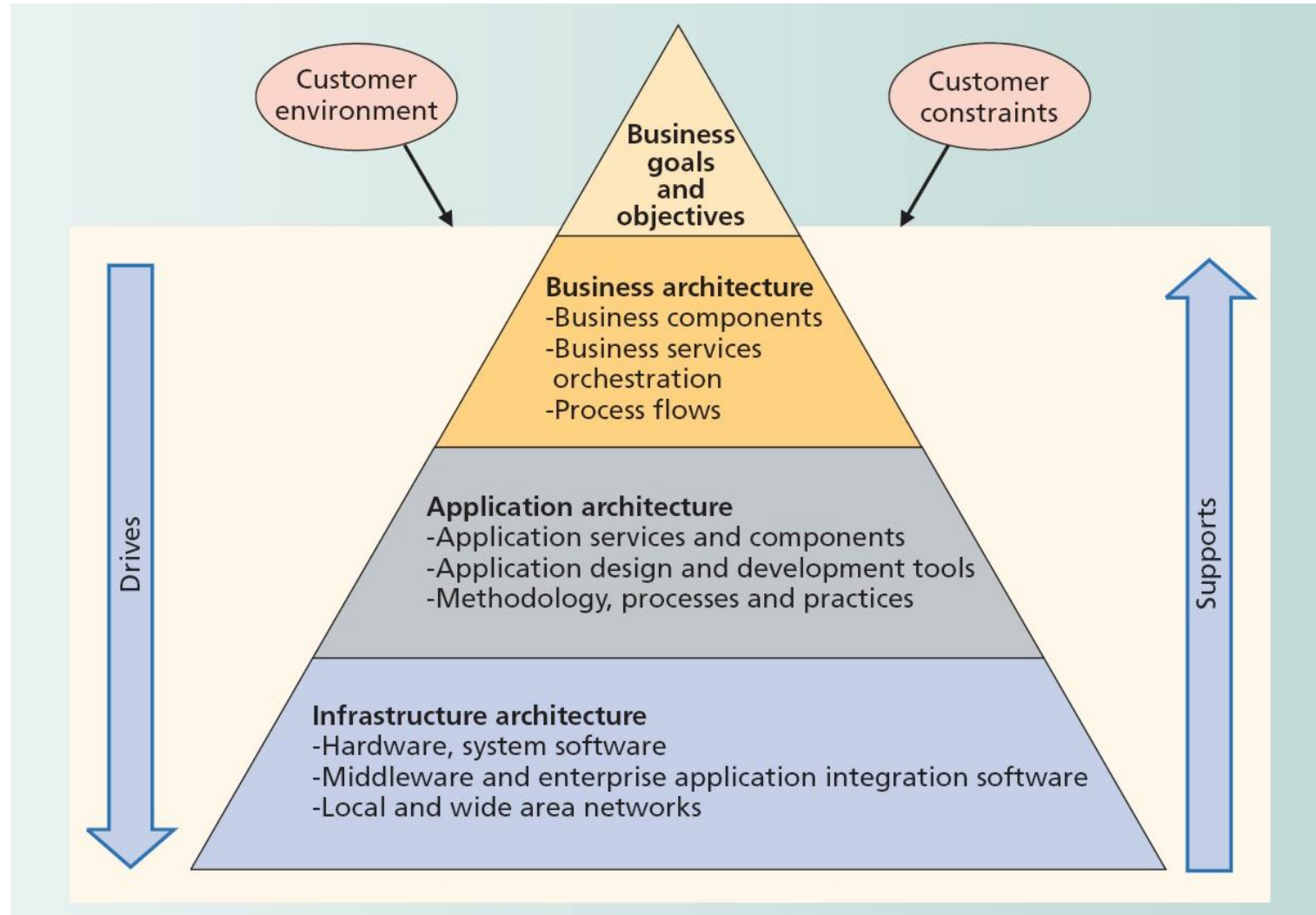
Digital Transformation requires building an ecosystem

- Digital transformation is a strategic approach to IT that treats IT infrastructure and data as a potential product for customers.
- Digital transformation requires shifting perspectives and by looking at new ways to use data and data sources and looking at new ways to engage with customers.
- At the core it will include ..
 - Virtualization, public clouds, containers and orchestration, Data virtualization, memory caches and storage, multiple messaging protocols, various data formats from different data sources, management and deployment tools, test automation plus of course various applications & business process automations.

Architectural Layering



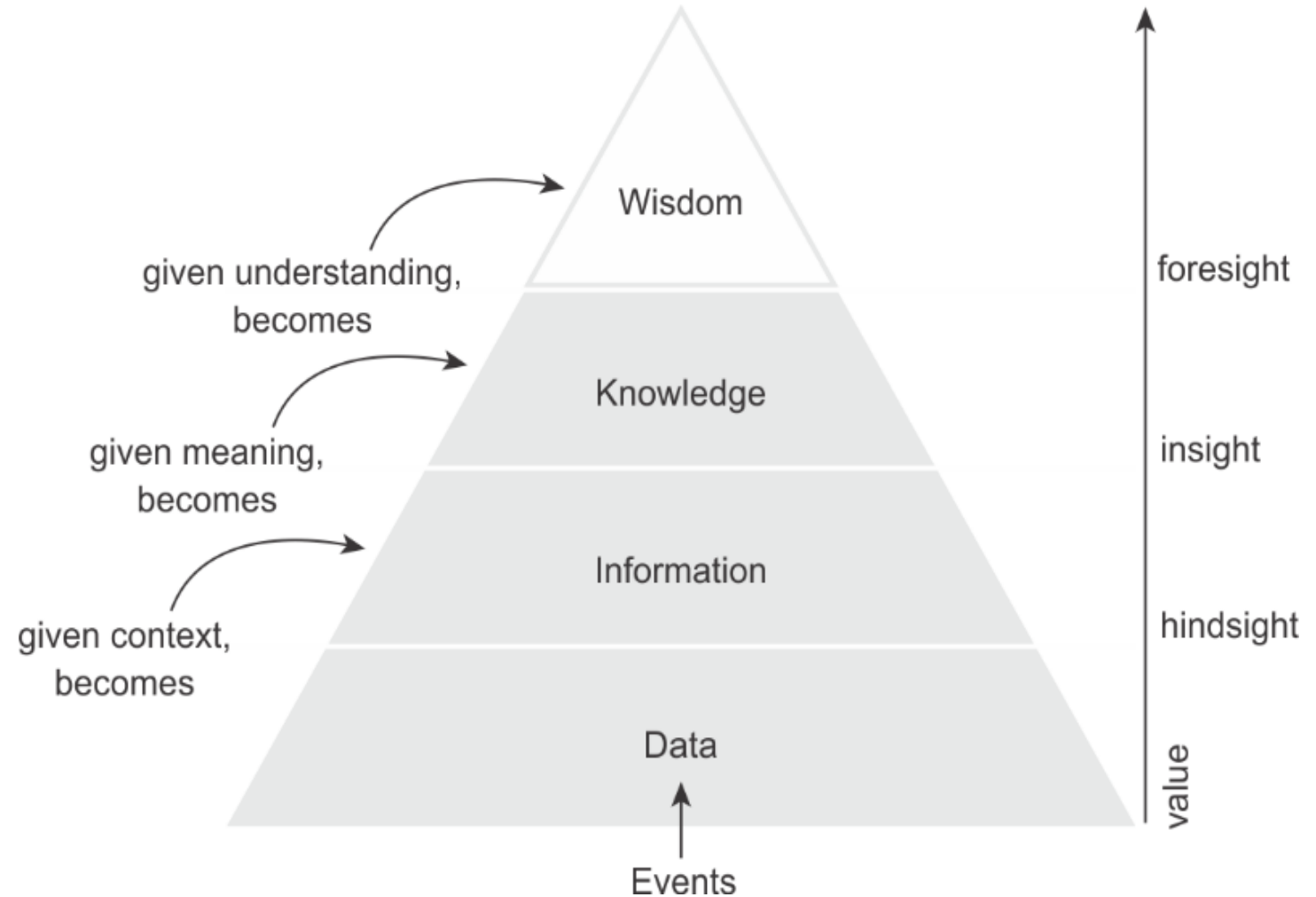
Architectures are Layered



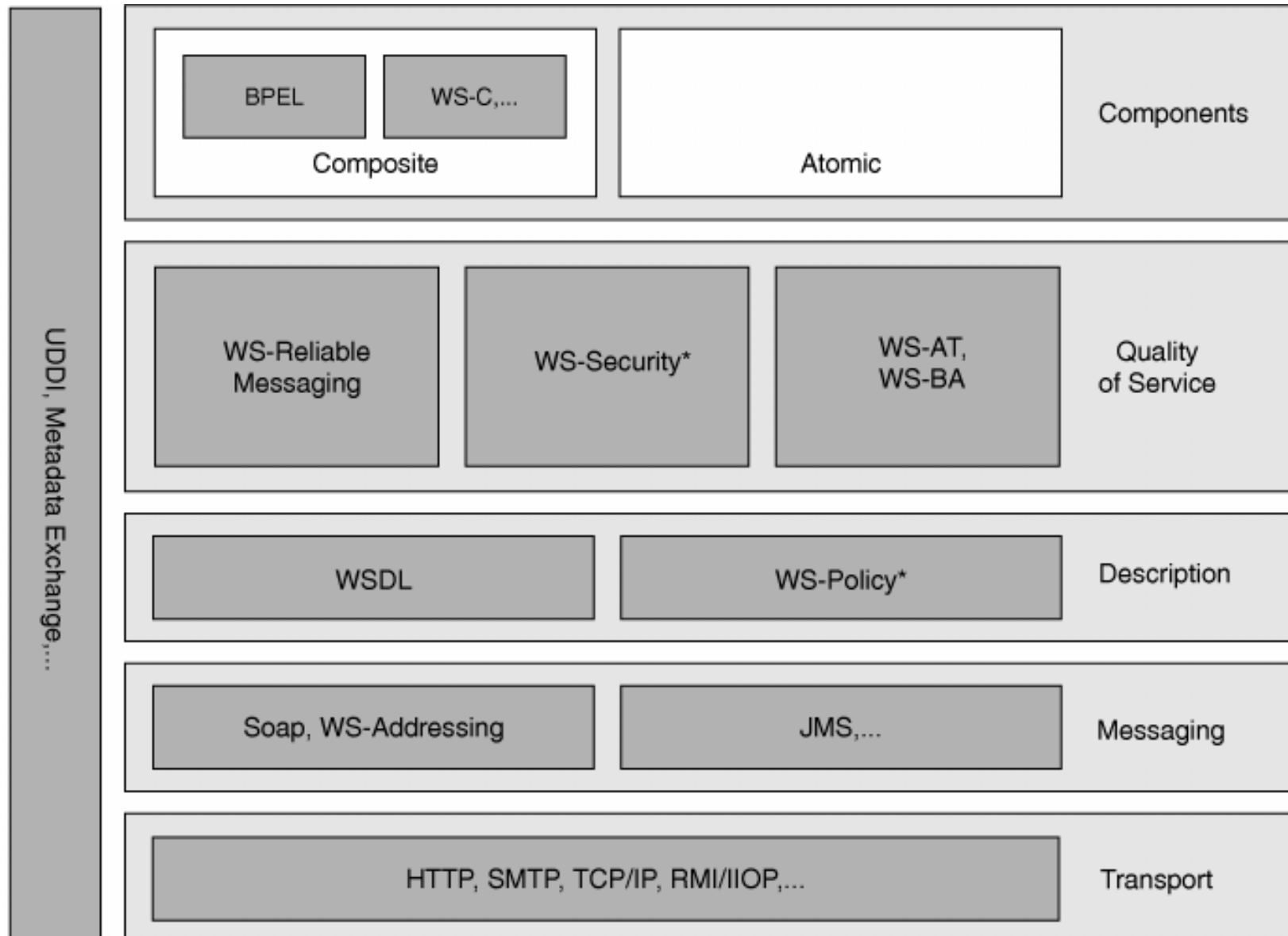
The Key - The Layering Principle

It is an accepted view that a business operates as a layered system.

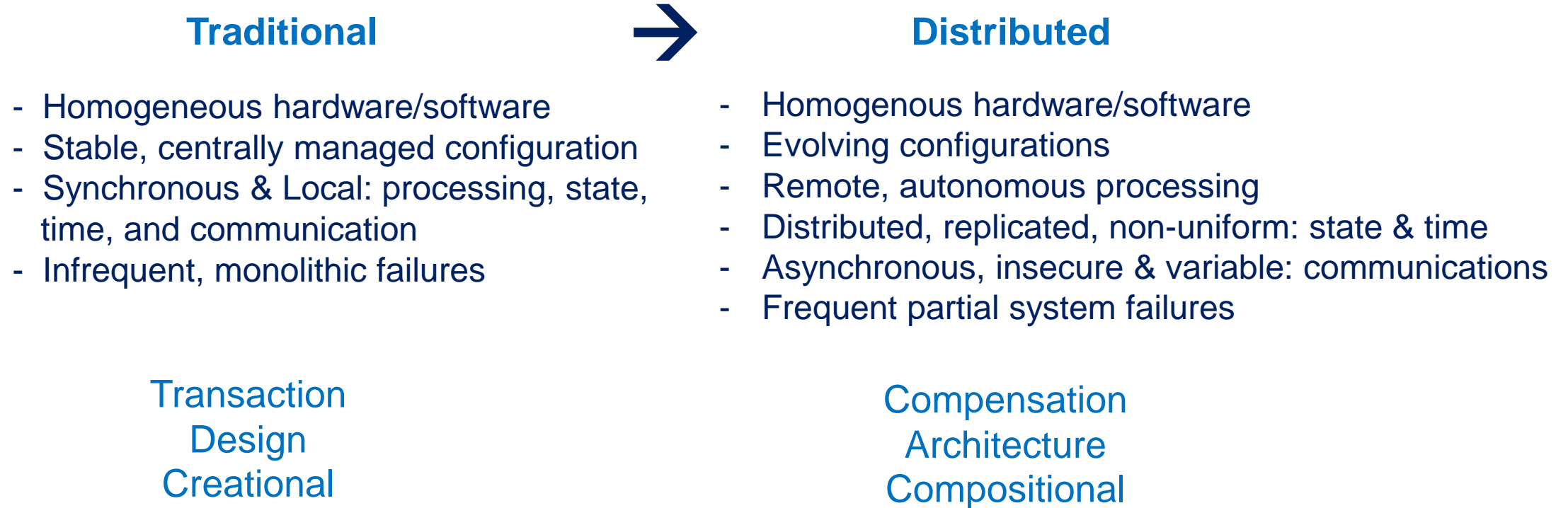
These three layers often exhibit a degree of independence from one another, nevertheless goals and objectives are influenced by and often defined by the layer above, in other words top-down.



Standards are Layered



Changing Systems Assumptions



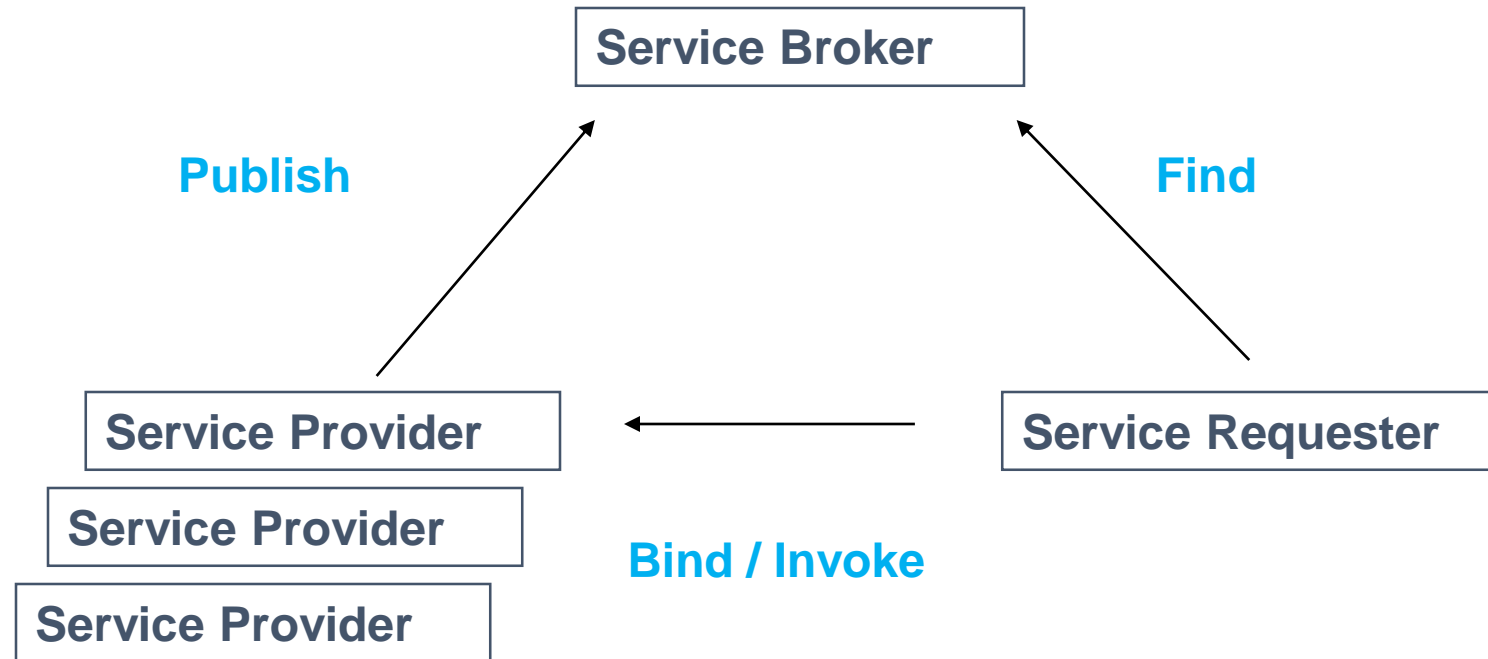
SOA & Service Oriented Computing

- In this open framework, services can be **shared** and reused across several business processes. The result is a highly **adaptive** environment, with lower costs for application development, **improved integration** and quicker deployments.
- SOA is not a new concept but a different approach to designing and building systems that are **flexible** and **adaptable** to support a dynamic business environment.
- “**Service-oriented computing**” is an umbrella term used to represent a distributed computing platform based on service-orientation.
- *As such, it encompasses many things, including its own design paradigm and design principles, design pattern catalogs, pattern languages, a distinct architectural model, and related concepts, technologies, and frameworks.*

Web Based Services

- SOAP Based :
 - <http://docs.oracle.com/javaee/6/tutorial/doc/bnayl.html>
- RESTful : .. as a simpler alternative to SOAP
 - Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web. In the REST architectural style, data and functionality are considered resources and are accessed using **Uniform Resource Identifiers (URIs)**, typically links on the Web.
 - REST defines a set of architectural principles by which you can design Web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP.

Basic Service-Oriented Model



Next Generation IT – Key Considerations

The current model for IT, which is built around delivering IT *Faster, Better* and *Cheaper*, is not adequate to deal with the increased volume, velocity and variety of change driven by the rapidly evolving digital landscape.

NTT Innovation Institute

Speed of operation and ‘fluidity’ are important prerequisites of next generation IT

Gartner

MicroServices architecture maps is a natural fit for implementing particularly data-driven & cross functional aspects of the systems, allowing bottom-up delivery of isolated bundles of functionality.

Leading to an Agile IT – The Behaviour

Agility isn't something you buy or do — it's something the entire organization must become.

Agility is desirable because it means greater responsiveness to the needs of the business. So one major source of greater agility is to identify and eliminate waste and duplications.

Business, development, and operations must understand the need for *continuous engagement* throughout the process and buy into and remain passionate about automation, measurement, and continuous improvement.

Understand how IT enables the business - Recognize and understand how the **business model** of any enterprise is implemented in practice.

Promote to Run IT as a Business - Understand the business model, operational processes and **metrics that measure the effectiveness** of IT in an enterprise.

Agility – The Challenges

- Application Integration
- Modernization Projects
- Service Orientation Paradigm – Misunderstood / Misused/Abused
- Lean Services – MicroServices
- Business Services vs. Data Services
- Bring both Approaches – Top Down / Bottom up!

Investing on any IT driven solution such as SOA or MicroServices requires a clear and well understood formal architectural blueprint that is implementation neutral.

Agility – Key Success Factors

- Define a Business Architecture
- Define the Service Inventory Architectures
- Define the Service Composition Architecture
- Define a set of Loosely Coupled & Autonomous Services
- Create an Service Catalog
- Use Centralized tools and processes to deliver services (both on premise & Cloud)
- Establish suitable Governance framework

Simplify & Streamline

- Use DevOps to ease the development, deployment, delivery and support processes
- Use containers to simplify deployment & cloud adoption
- Empower teams to work horizontally across the organization

Business Architecture – the purpose

Organizations need reliable and cost effective operations.

Business architecture provides the mechanism to clearly define how strategy, processes, business structure and staff can best be utilized to deliver reliable and cost effective operations.

With this clarity business can enable new functions and services, with the right resources and technology, effectively and efficiently.

Business architecture can help organizations analyze key value chains.

Value chains are the functions and services that yield the most economic value for the organization and provide the organization's customers with the necessary services, products and information to maintain customer loyalty.

Business Architecture Definitions

TOGAF describes enterprise architecture's purpose as supporting the business “by providing the fundamental technology and process structure for an IT strategy.”

The GAO defines EA as:

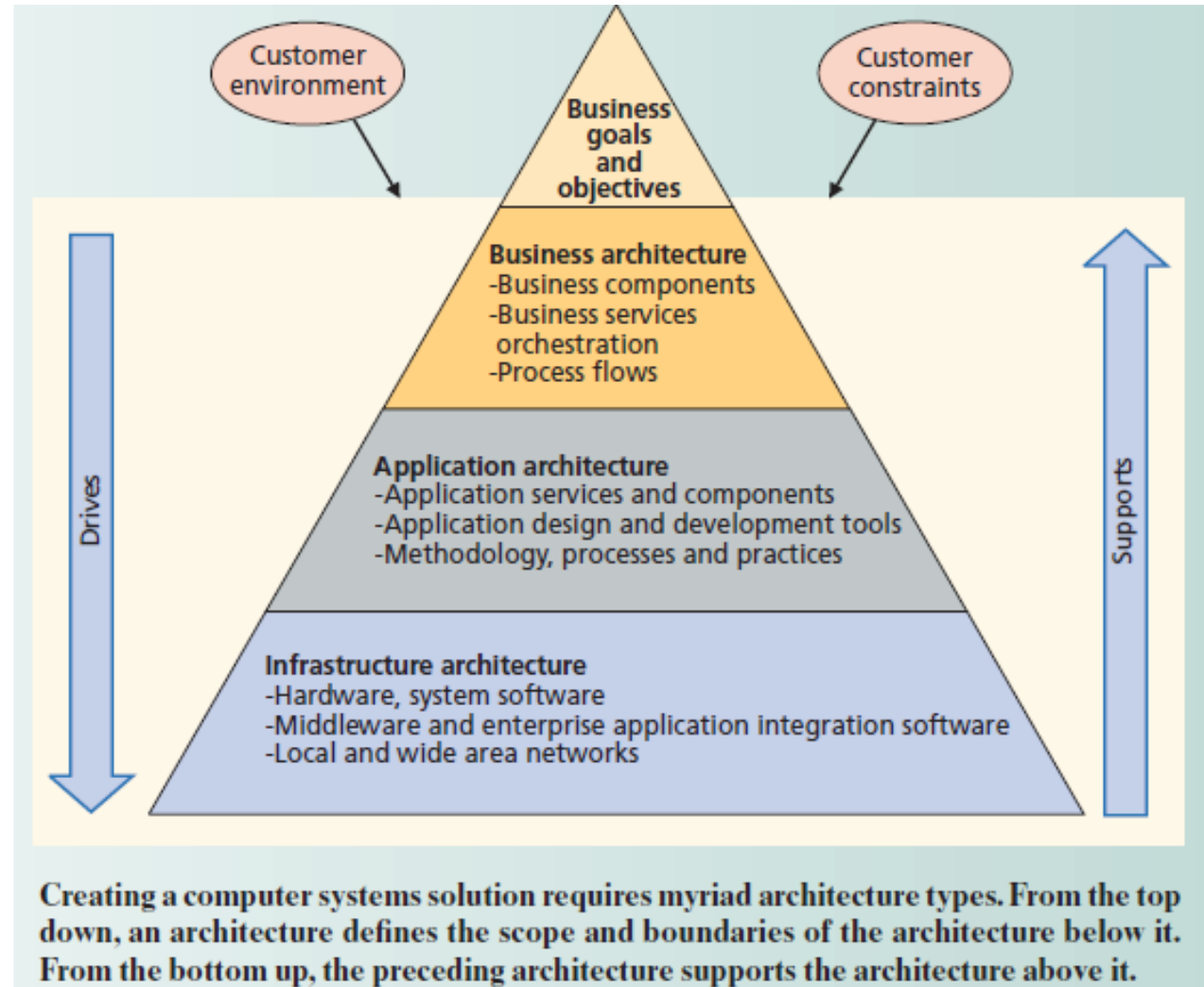
“a blueprint that describes an organization’s or a functional area’s current and desired state in both logical and technical terms, as well as a plan for transitioning between the two states.”

MIT’s CISR defines EA as

“the organizing logic for business processes and IT infrastructure reflecting the integration and standardization requirements of the firm’s operating model.”

The Business Architecture – another View

*The system
solution
pyramid*

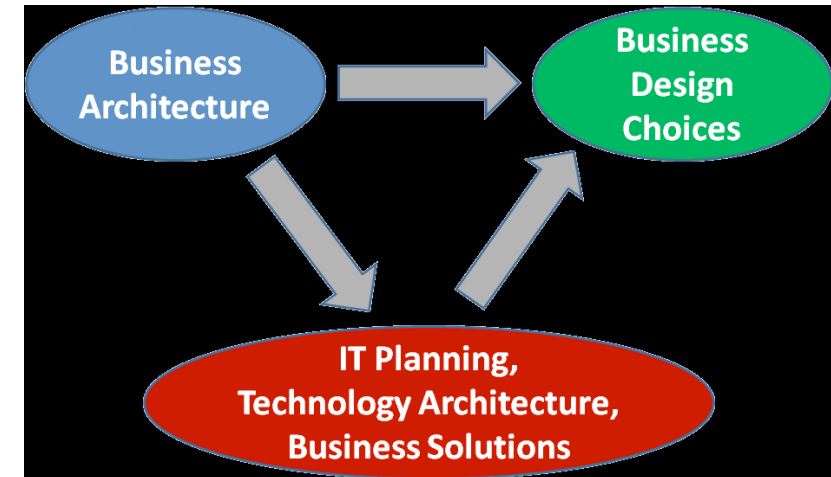


Business Architecture <> IT

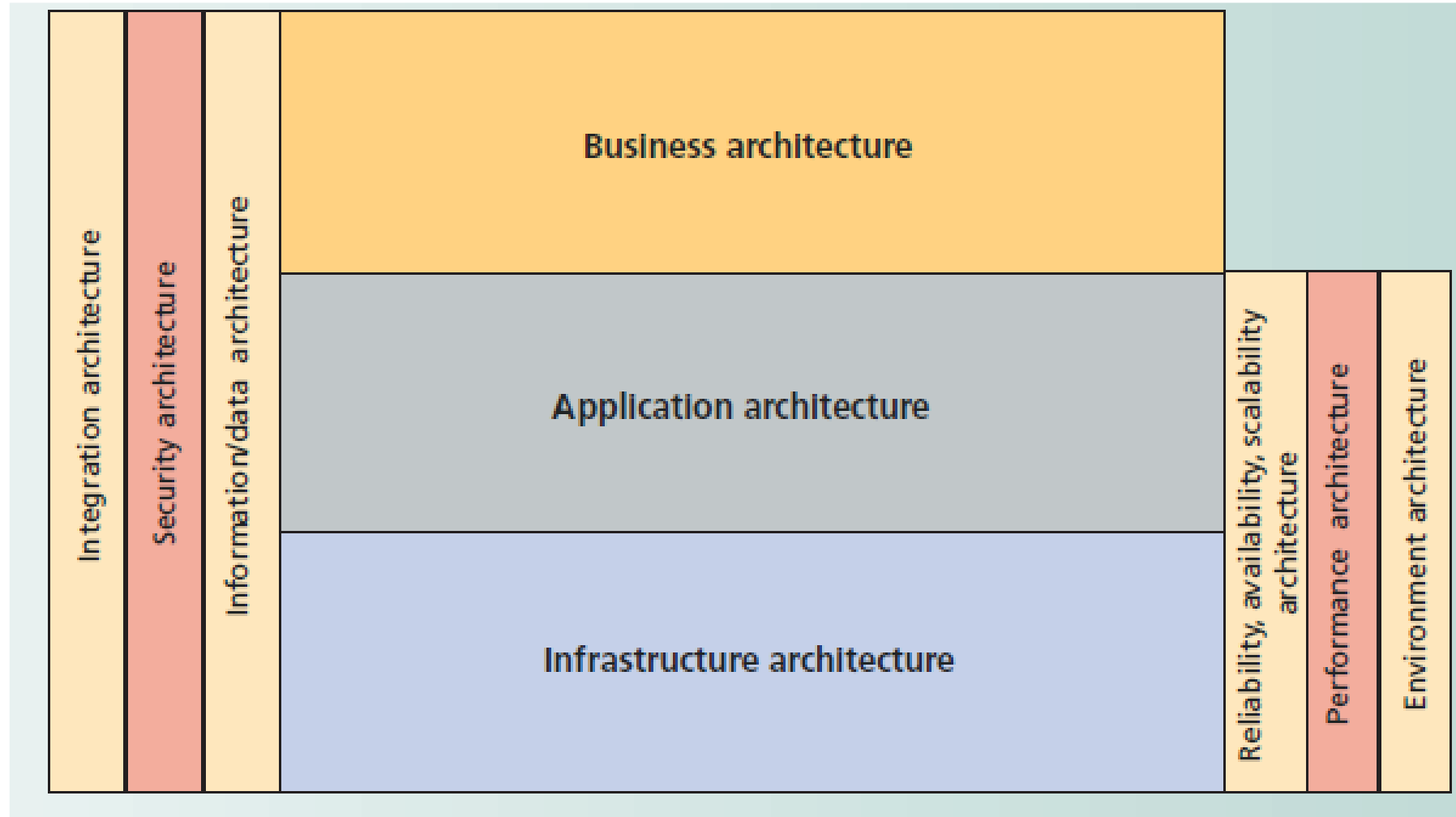
The relationship between business architecture and information technology is two-fold.

business architecture is a critical input to IT planning, technology architecture and business solution delivery.

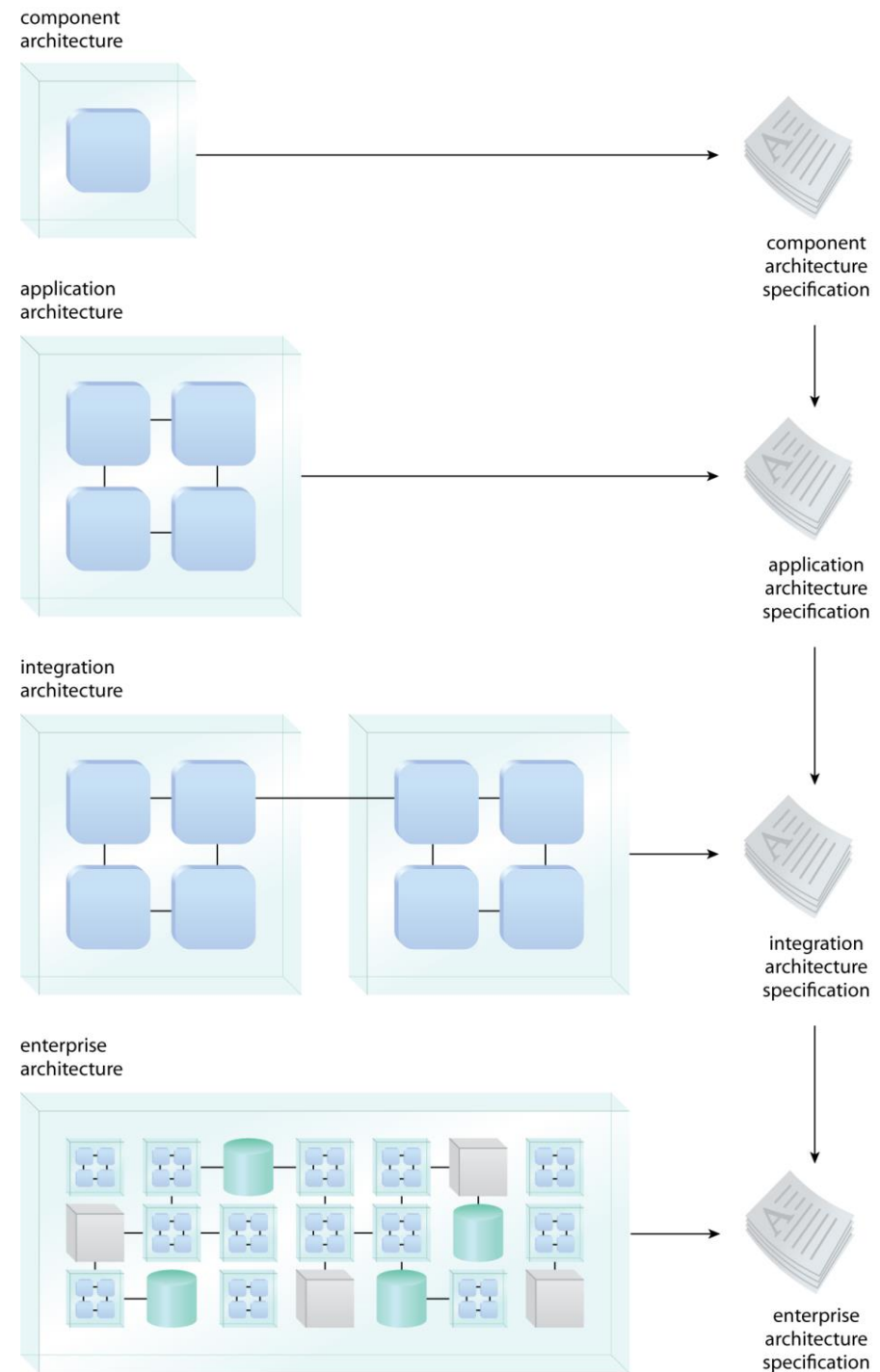
technology trends and IT capabilities influence business design choices in the realms of capabilities, value chains, processes, and channels.



Various Architecture Types



Established Architecture Types



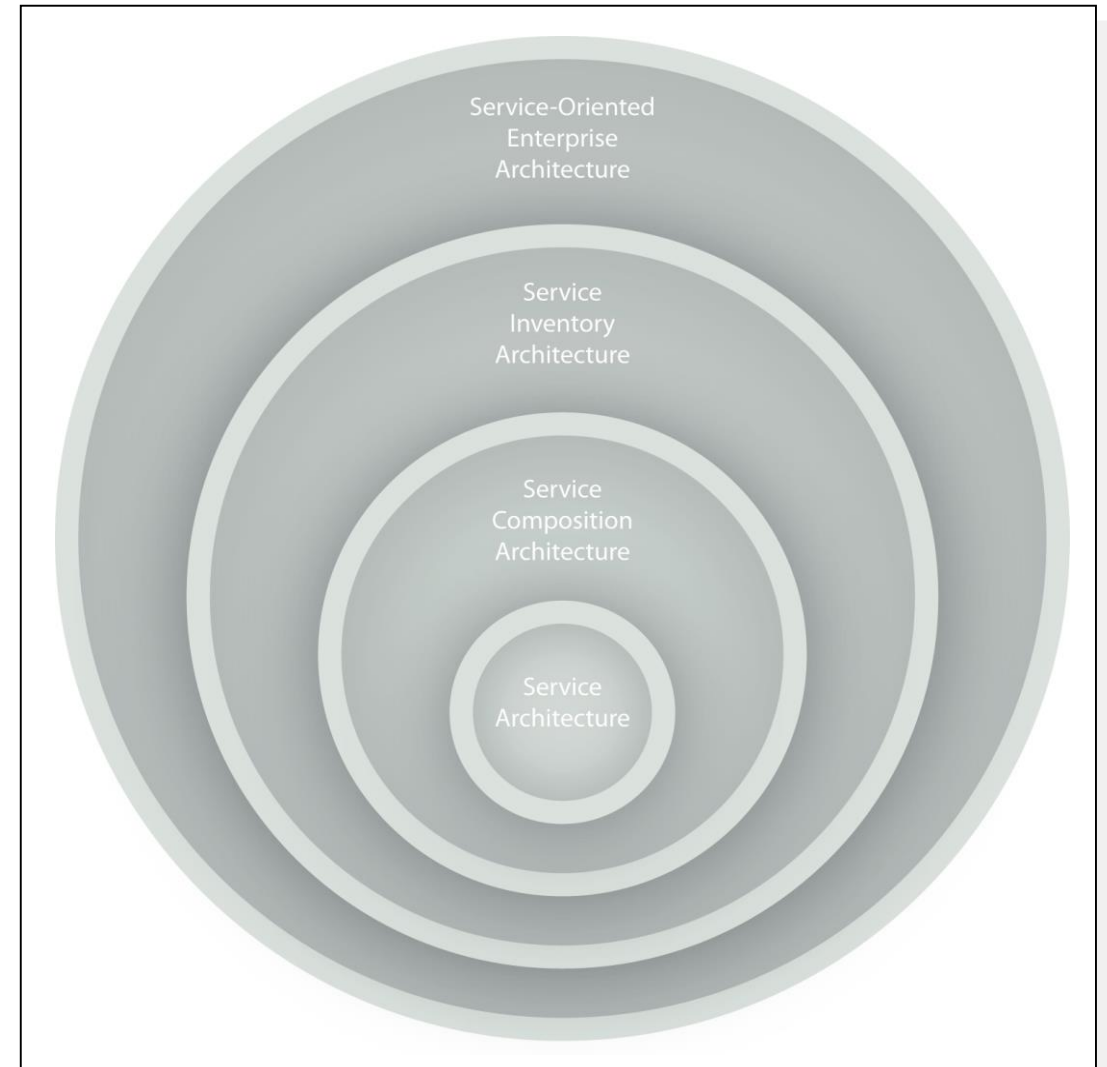
SOA Architecture Types

Service Architecture – The architecture of a single service.

Service Composition Architecture – The architecture of a set of services assembled into a service composition.

Service Inventory Architecture – The architecture that supports a collection of related services that are independently standardized and governed.

Service-Oriented Enterprise Architecture – The architecture of the enterprise itself, to whatever extent it is service-oriented.



SOA – Strategic Goals & Benefits

Strategic goals and benefits supported by service-orientation:

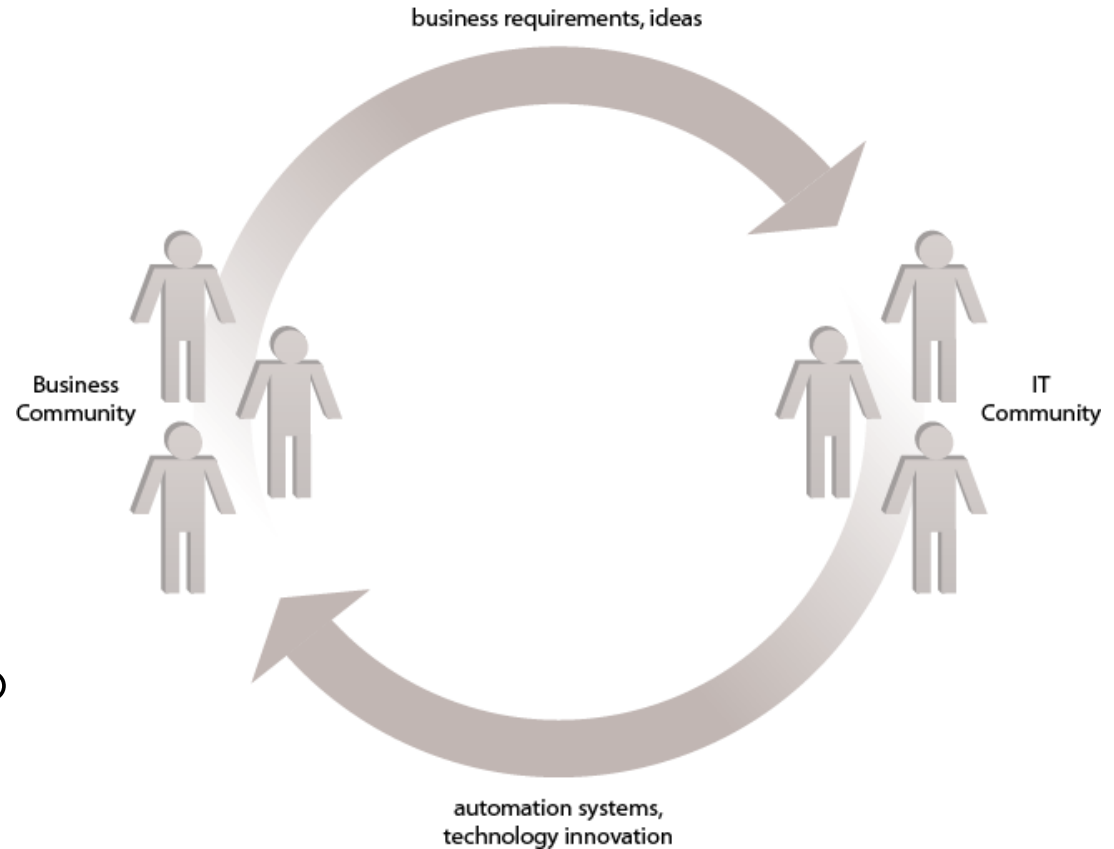
- Increased Intrinsic Interoperability
- Increased Federation
- Increased Vendor Diversification Options
- Increased Business and Technology Alignment
- Increased ROI
- Increased Organizational Agility
- Reduced IT Burden

Supported by comprehensive & widely adopted industry practitioners as catalogued under SOA Design Patterns
<http://soapatterns.org/>

Increased Agility & Reduced IT-Burden

The essence of service-orientation is to establish an IT environment that inherently is **capable of accommodating change**.

This is in contrast to continue building siloed application which has proven to results is duplication and bloated IT.

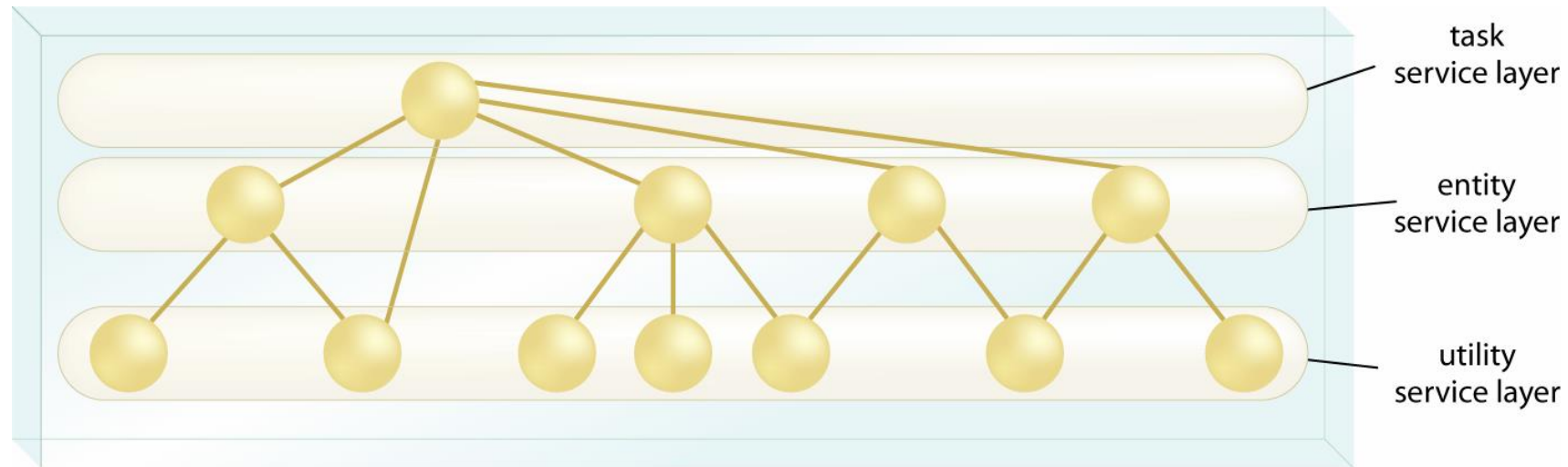


Service Layers

There are Four service Layers including :

- Utility Services
- Entity Services
- Task Services
- Orchestrated Task Service

They provide abstraction layers and assist with managing the impact of change.



MicroService & SOA – Misunderstood & Misused

Although SOA is a move in the right direction, a more mature model, called microservices architecture, has evolved.

The services within the architecture use a common messaging framework, like REST API, which they use to communicate with each other without difficult data conversion transactions or additional integration layers.

Microservices architectures are similar to SOA patterns in that services are specialized and loosely coupled, but microservices are broken down even further. A microservices architecture defined services in a very clear way: A single, clear purpose, Well defined parameters, Polyglot implementation.

This lightweight architecture helps optimize distributed or cloud resources and supports dynamic scalability for individual services.

SOA Design Patterns ..

Asynchronous Queuing

Atomic Service Transaction

Compensating Service Transaction

Concurrent Contracts

Contract Centralization

Data Format Transformation

Data Model Transformation

Decoupled Contract

Domain Inventory

Exception Shielding

Enterprise Service Bus

Event-Driven Messaging

Intermediate Routing

Legacy Wrapper

Logic Centralization

Messaging Metadata

Official Endpoint

Orchestration

Policy Centralization

Process Abstraction

Process Centralization

Protocol Bridging

Redundant Implementation

Reliable Messaging

Rules Centralization

Schema Centralization

Service Agent

Service Broker

Service Data Replication

Service Façade

Service Normalization

State Repository

Validation Abstraction

http://soapatterns.org/design_patterns/overview

ESB - An Enabler Platform for MicroServices

ESB is a Compound Pattern ..
typically consisting of:



Data Format Transformation

Data Model Transformation

Protocol Bridging

Service Broker

Event-Driven Messaging

Intermediate Routing

Asynchronous Queuing

Reliable Messaging

Atomic Service Transaction

Q/A & Discussions