Service Oriented Architecture

Service Oriented All Around Us

Lecture 2

Dr. Alessio Bonti

SOA Definition

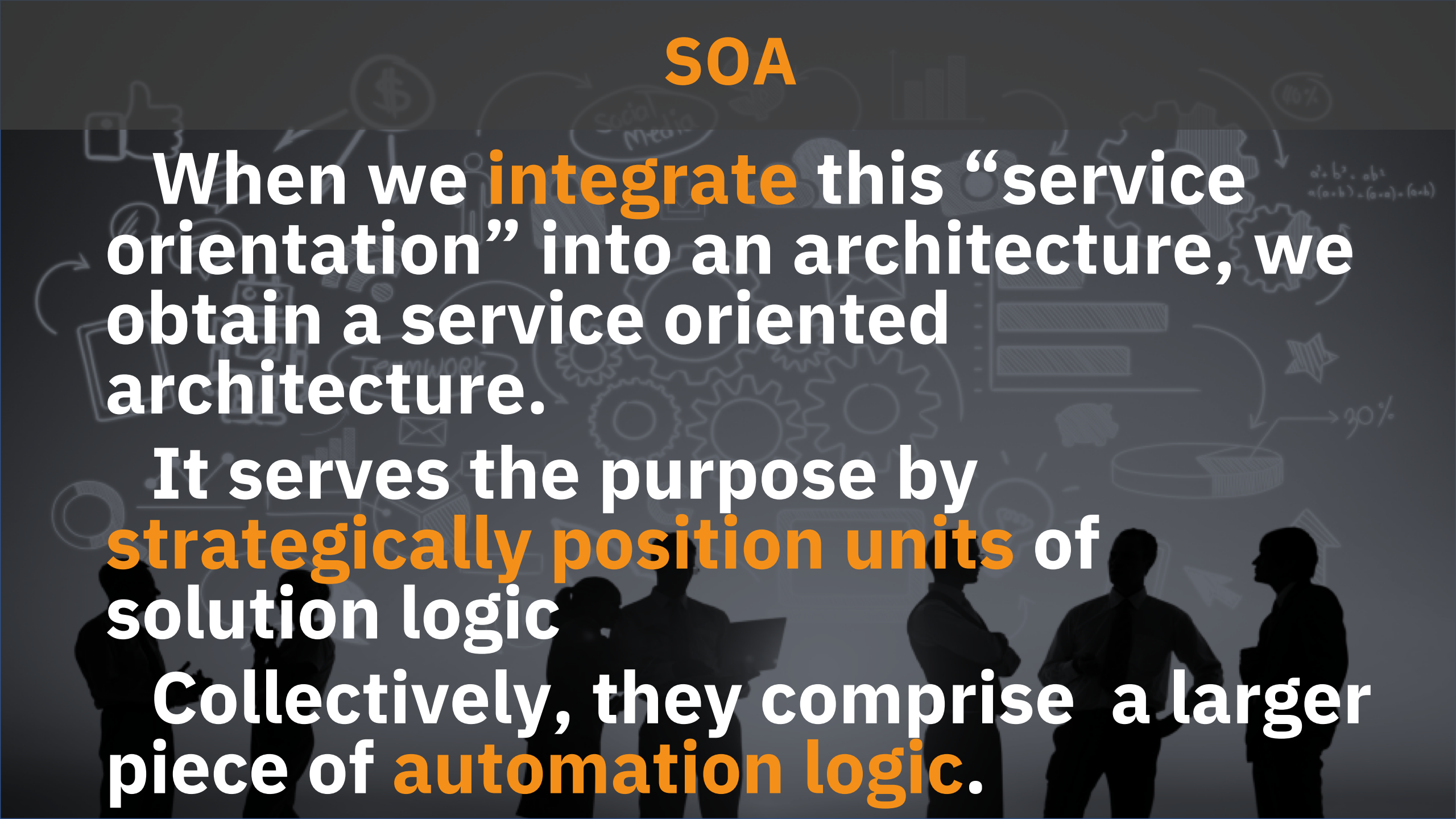
Businesses provide services, therefore they are service oriented.

SOA

When we integrate this “service orientation” into an architecture, we obtain a service oriented architecture.

It serves the purpose by strategically position units of solution logic

Collectively, they comprise a larger piece of automation logic.



In short SOA is

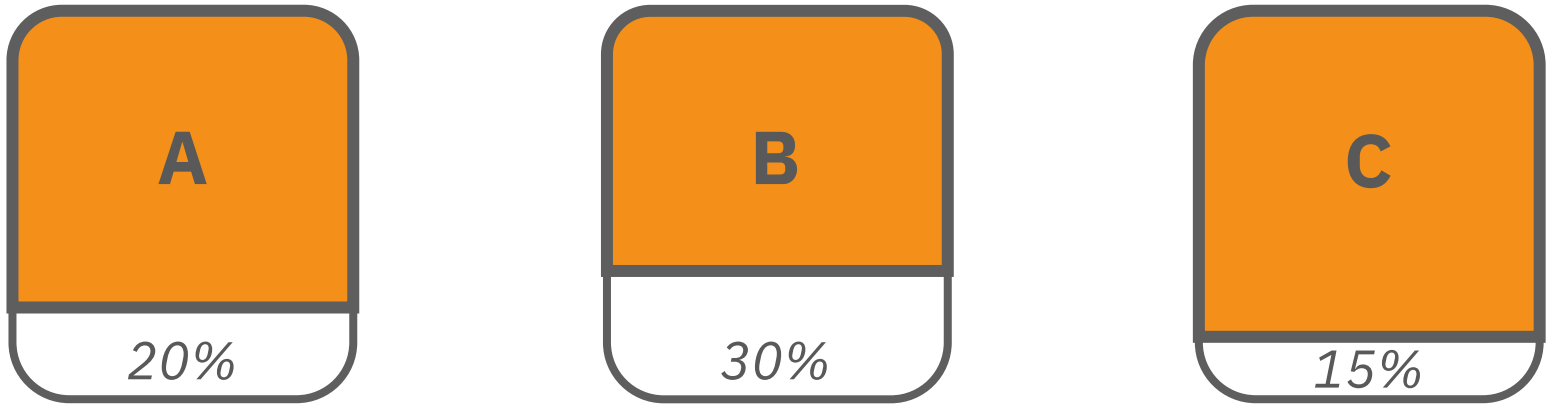
A distributed technology model with distinct characteristics in support to create service orientation

Let’s imagine an enterprise architecture….

Imagine if applications were standalone

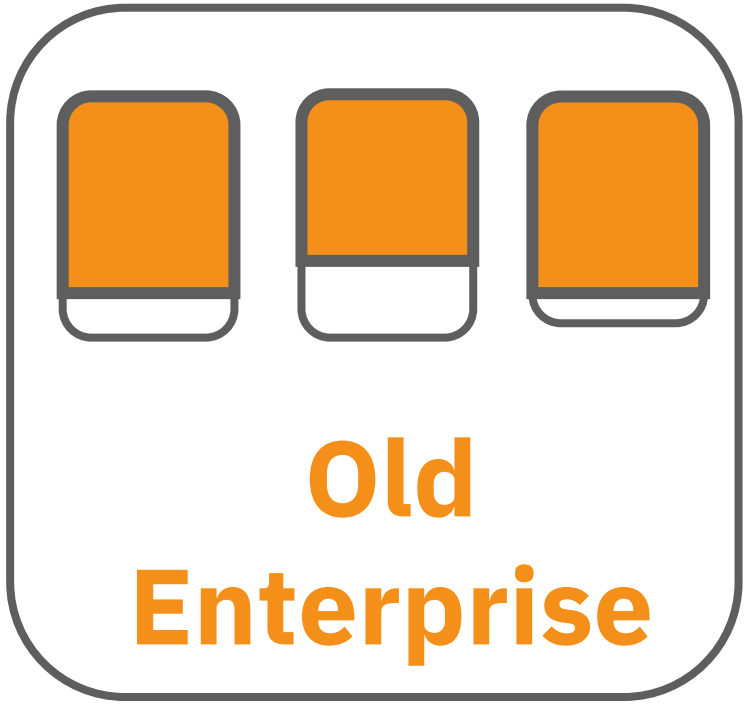
Imagine the orange part is redundant in all…

The impact would be extremely wasteful



Essentially we are running the same code over and over.

Enterprise without SOA

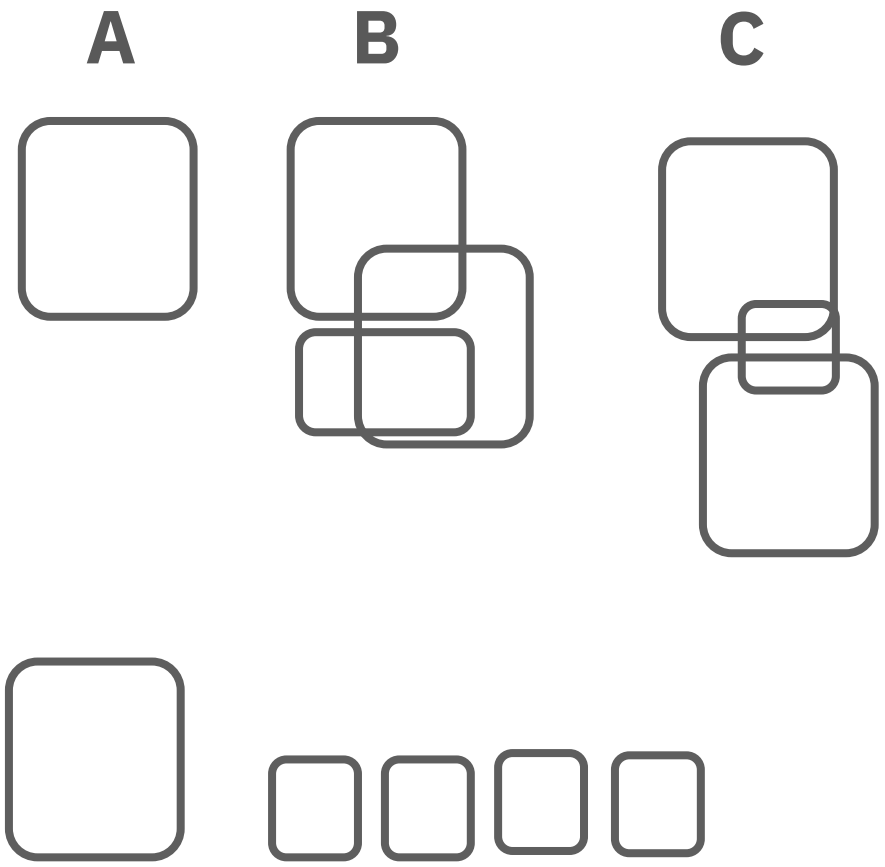


This makes it very difficult for the organization to grow

Growth of budge, resources allocation and size can become a huge drain on the overall organization.

This is just not scalable

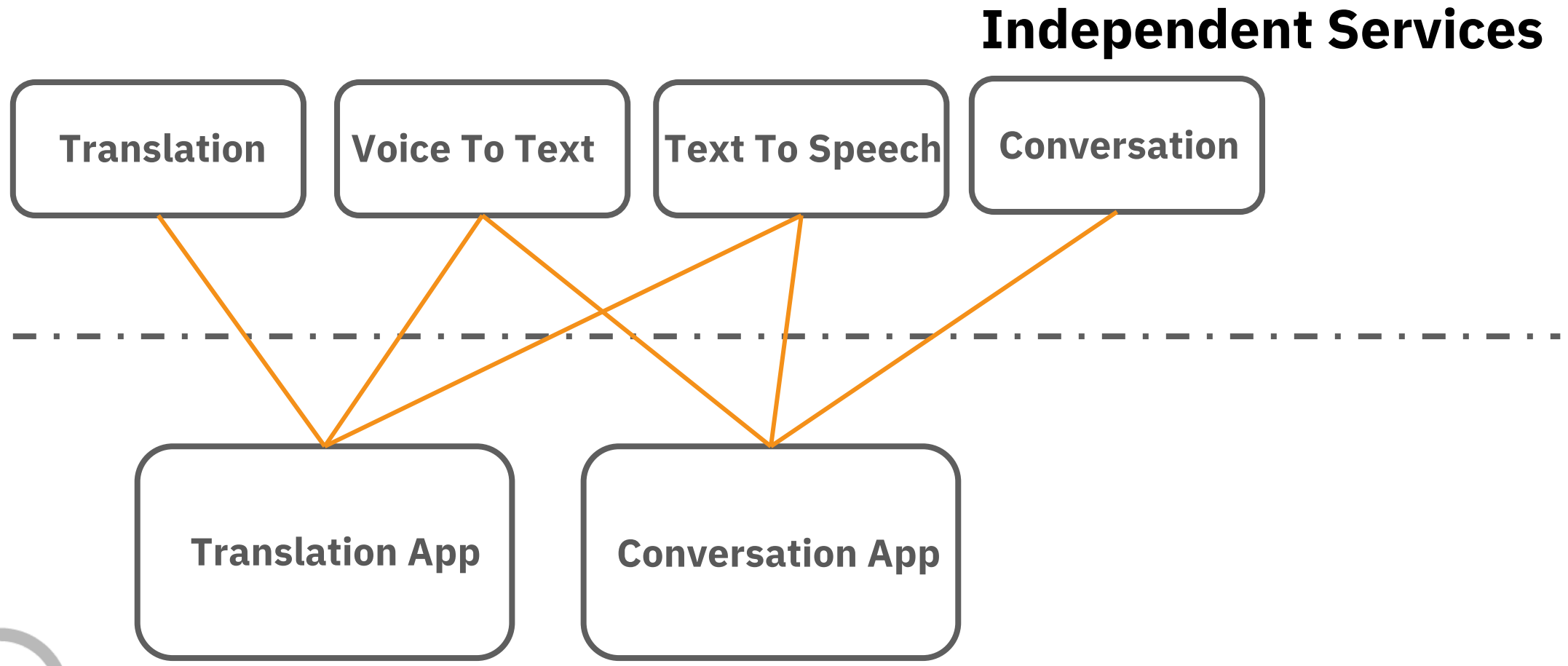
Object Orientation VS Services



If we look at the application model, object orientation suffers from being tightly coupled down to the logic.

Service composition allows instead for loosely coupled units, allowing for high level composition

Example of Service Oriented



Service orientation Requirements

- *Automation logic is distributed into units*
- *Loose coupling, no inter dependencies*
- *Each unit can grow and evolve independently*
- *Units maintain sufficient amount of commonality to allow interoperability*

Service orientation is a design paradigm

Intended for the creation of solution logic units.
Individually shaped for reutilization.

Solution Logic is designed in accordance with service orientation.
Units of service orientated logic are called services

Fundamental Characteristics

Business Driven

The technology should always reflect the business, it was born to be part of it, therefore it needs continuous evolution

Vendor Neutral

Allow the technology to be independent and ensure longevity, this will also improve interoperability

Enterprise Centric

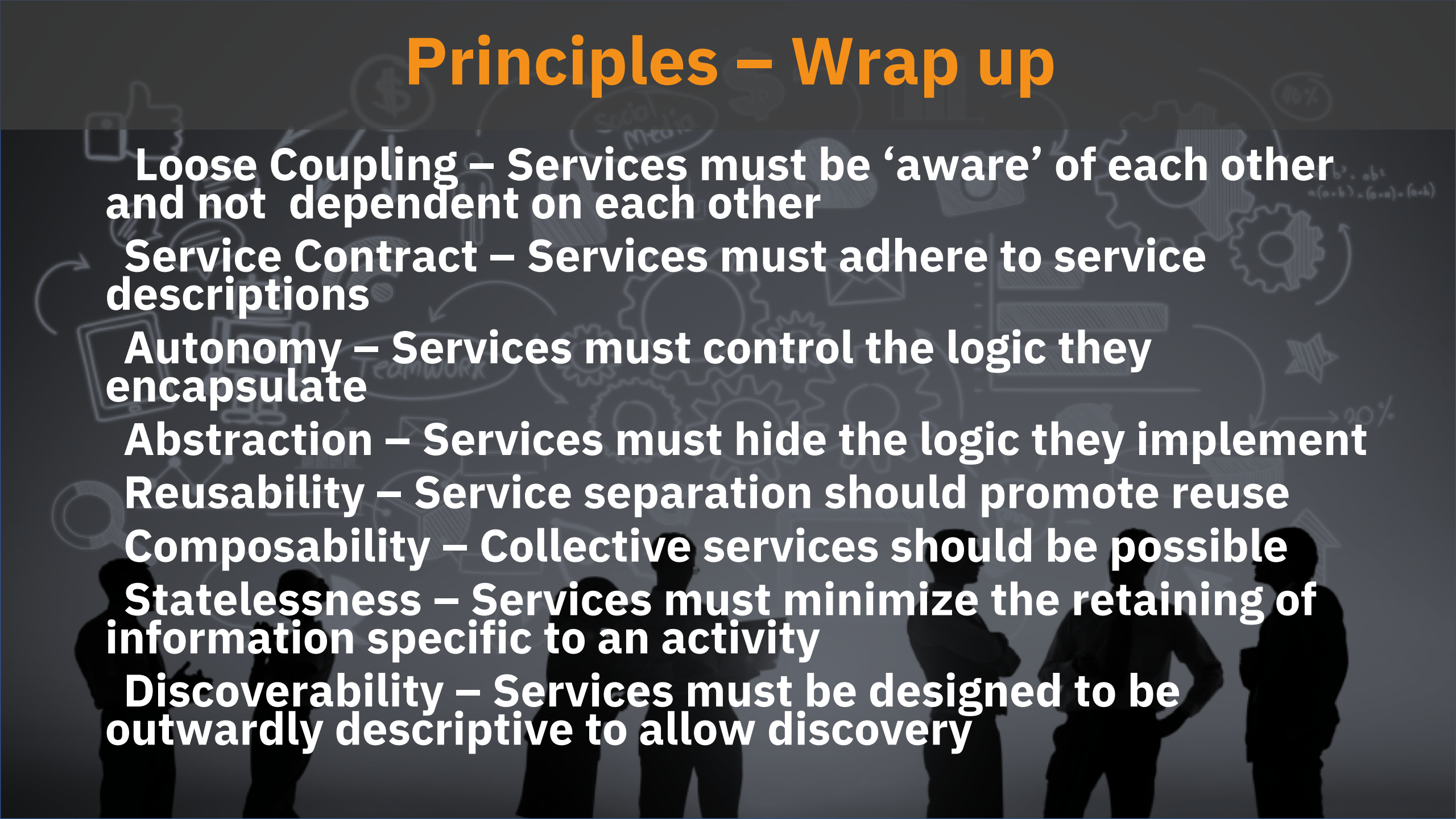
Supports agnostic services that are reused to participate in the automation of different business tasks and processes

Composition Centric

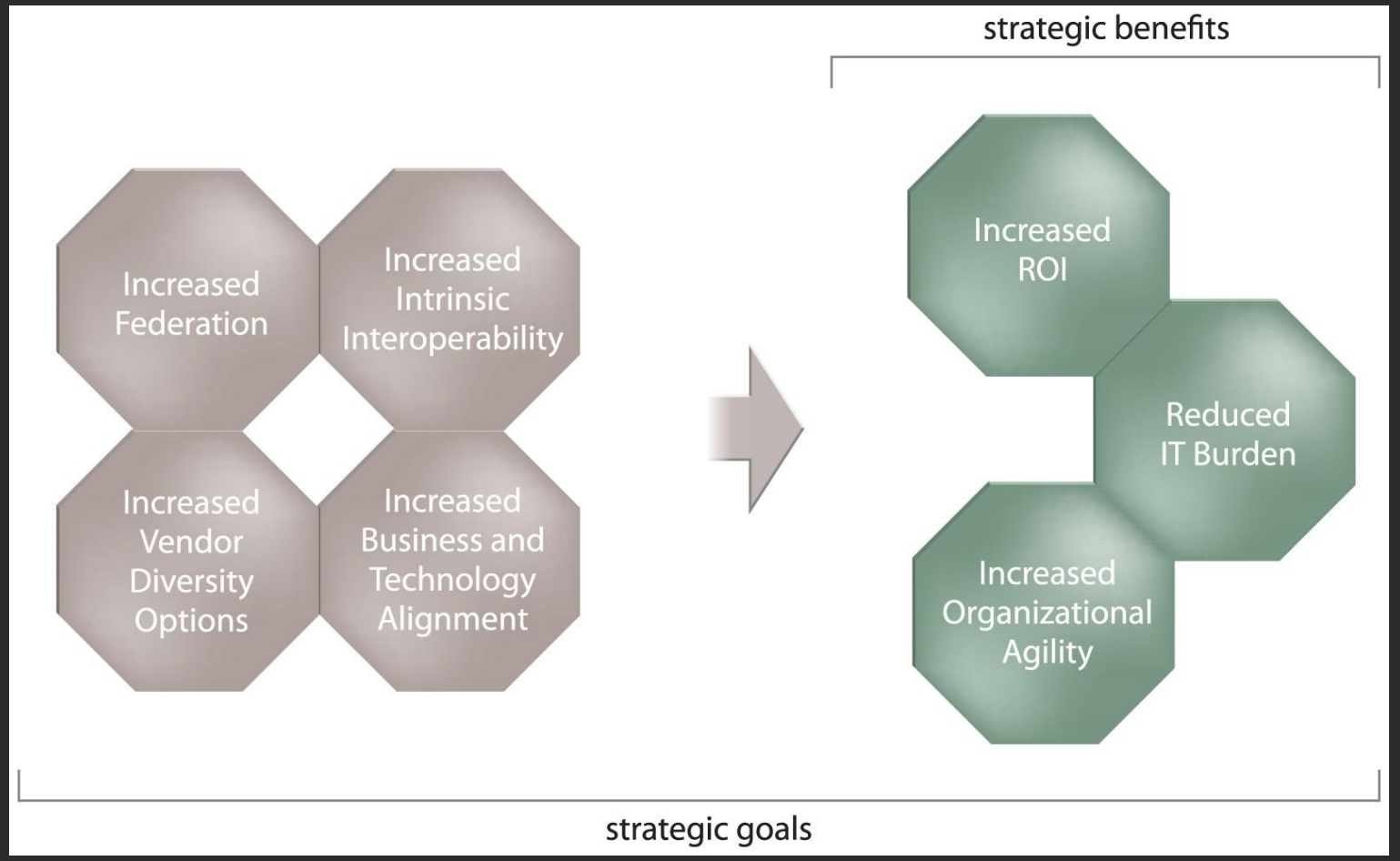
Being able to easily composed into new forms

Principles – Wrap up

- Loose Coupling – Services must be ‘aware’ of each other and not dependent on each other
- Service Contract – Services must adhere to service descriptions
- Autonomy – Services must control the logic they encapsulate
- Abstraction – Services must hide the logic they implement
- Reusability – Service separation should promote reuse
- Composability – Collective services should be possible
- Statelessness – Services must minimize the retaining of information specific to an activity
- Discoverability – Services must be designed to be outwardly descriptive to allow discovery

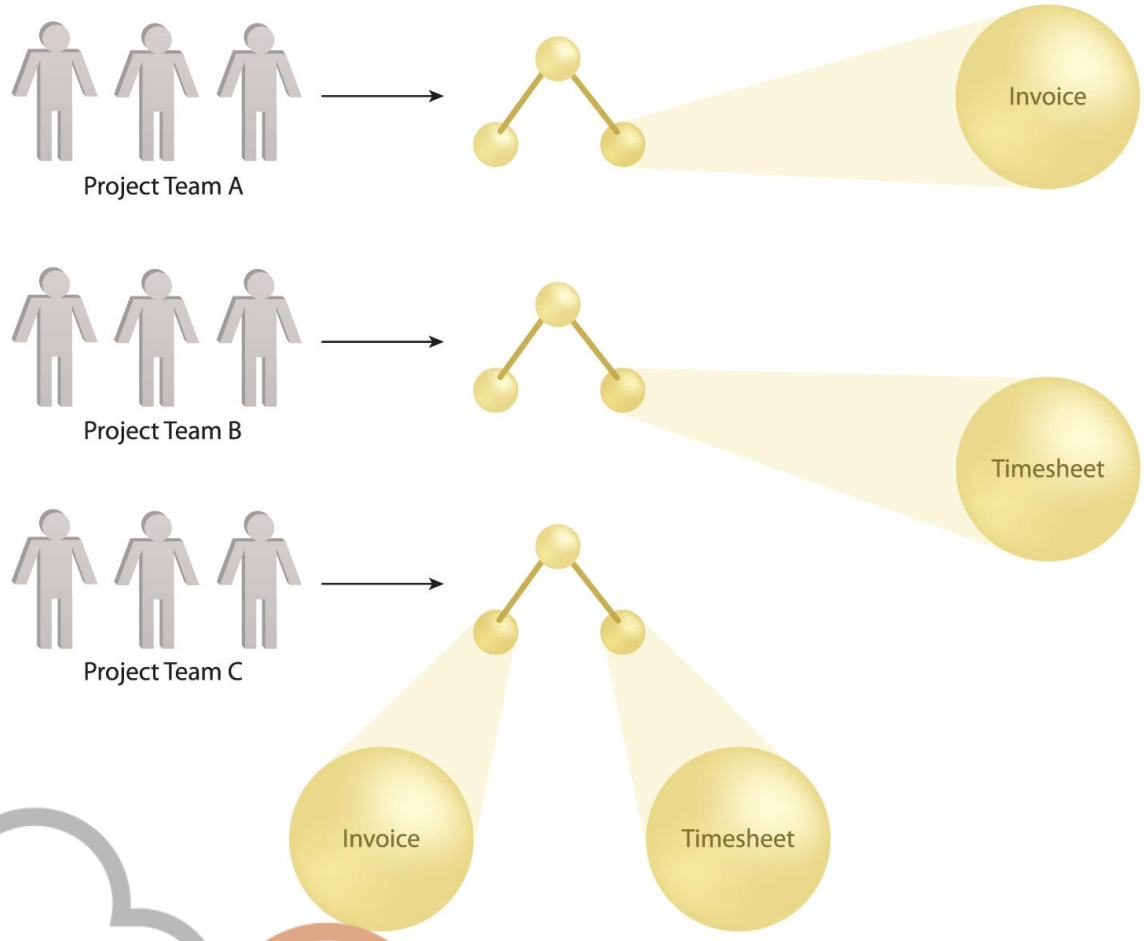


Strategic Goals and Benefits



Here are seven main goals, the strategic benefits are direct results of the first four being achieved

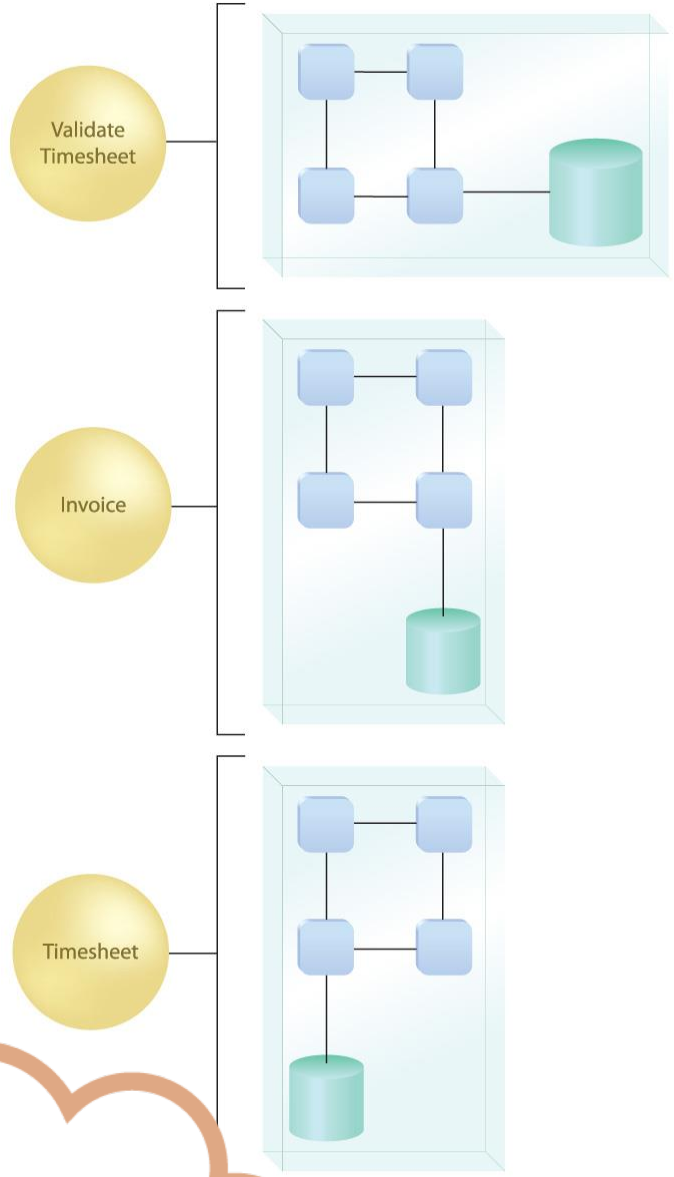
Intrinsic Operability



Services are designed to be intrinsically interoperable regardless of when and for which purpose they are delivered.

In this example, the intrinsic interoperability of the Invoice and Timesheet services delivered by Project Teams A and B allow them to be combined into a new service composition by Project Team C.

Increased Federation



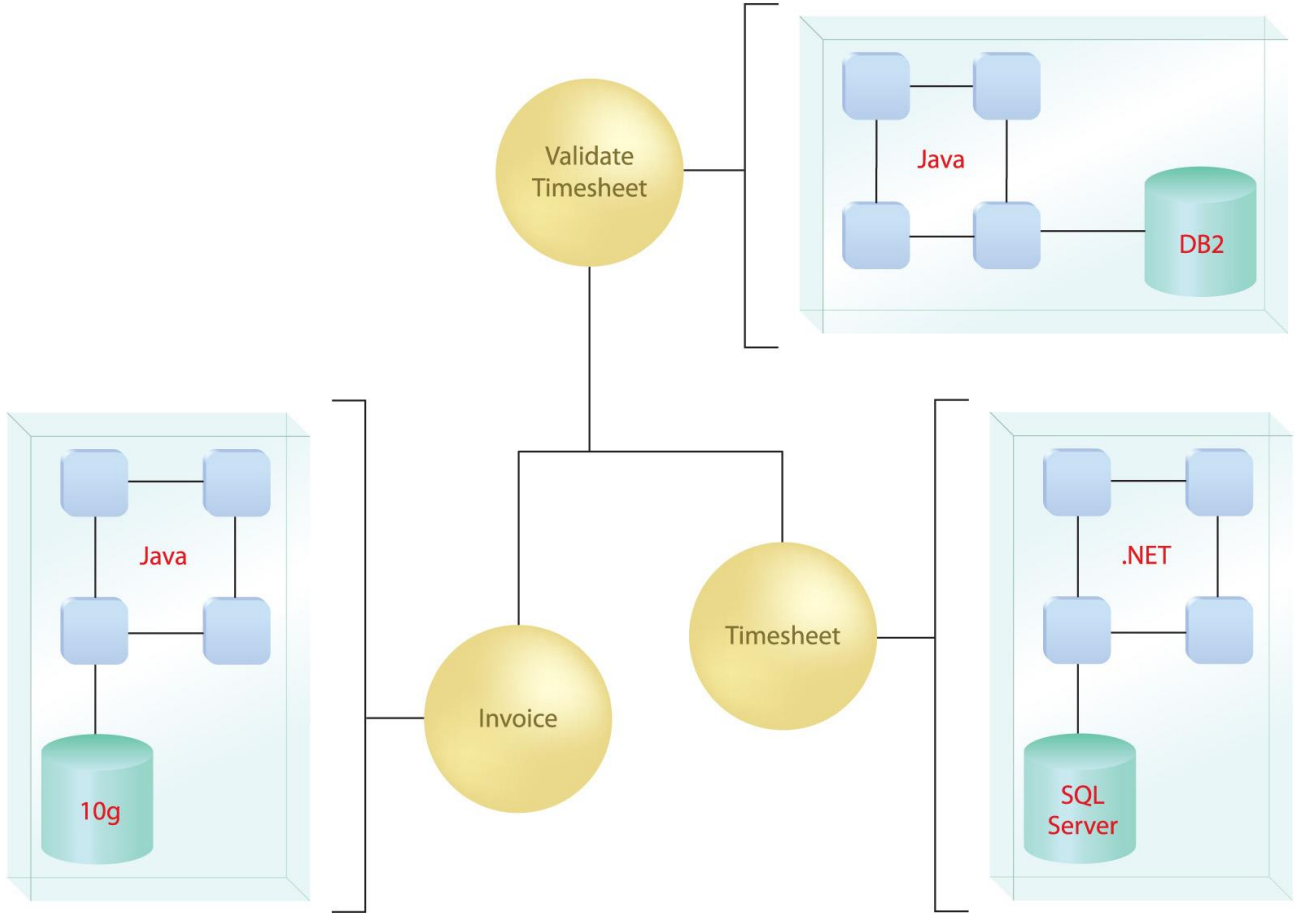
Federated IT environment: Resources and applications are united while maintaining their **individual autonomy**

SOA increases federation through standardized and **composable** services

Example

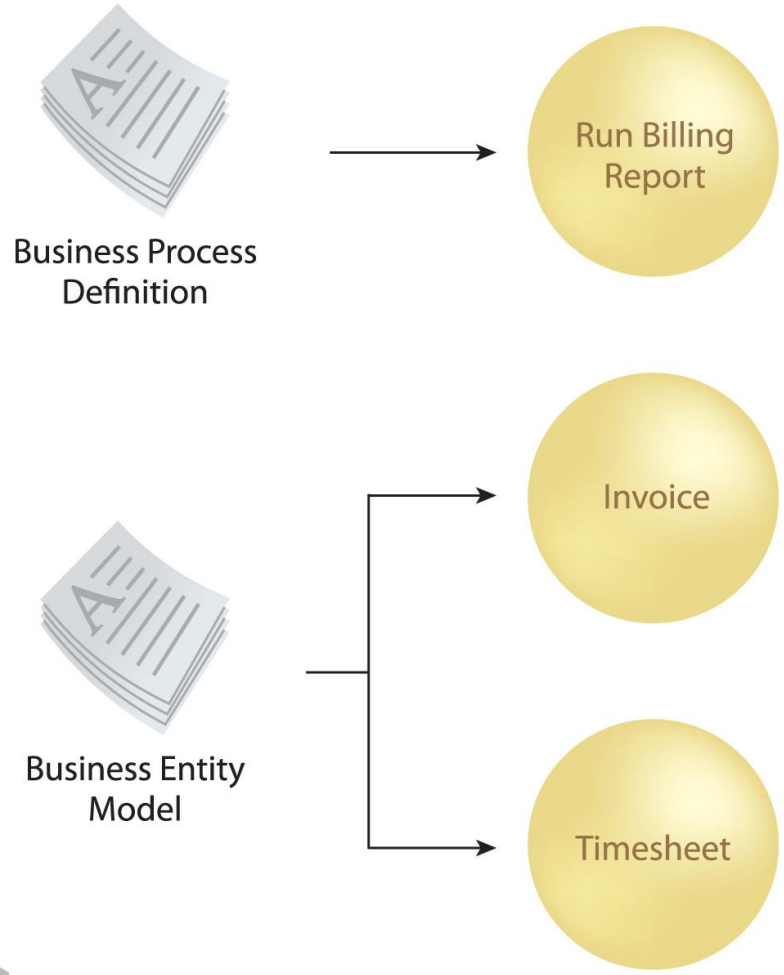
Three service contracts establishing a federated set of endpoints, each of which encapsulates a different implementation.

Increase Vendor Diversity



Implementation is **independent**, and **hidden** from the consumer.
We use strict inputs and outputs elaborated by what appears as a **black box** to the user.

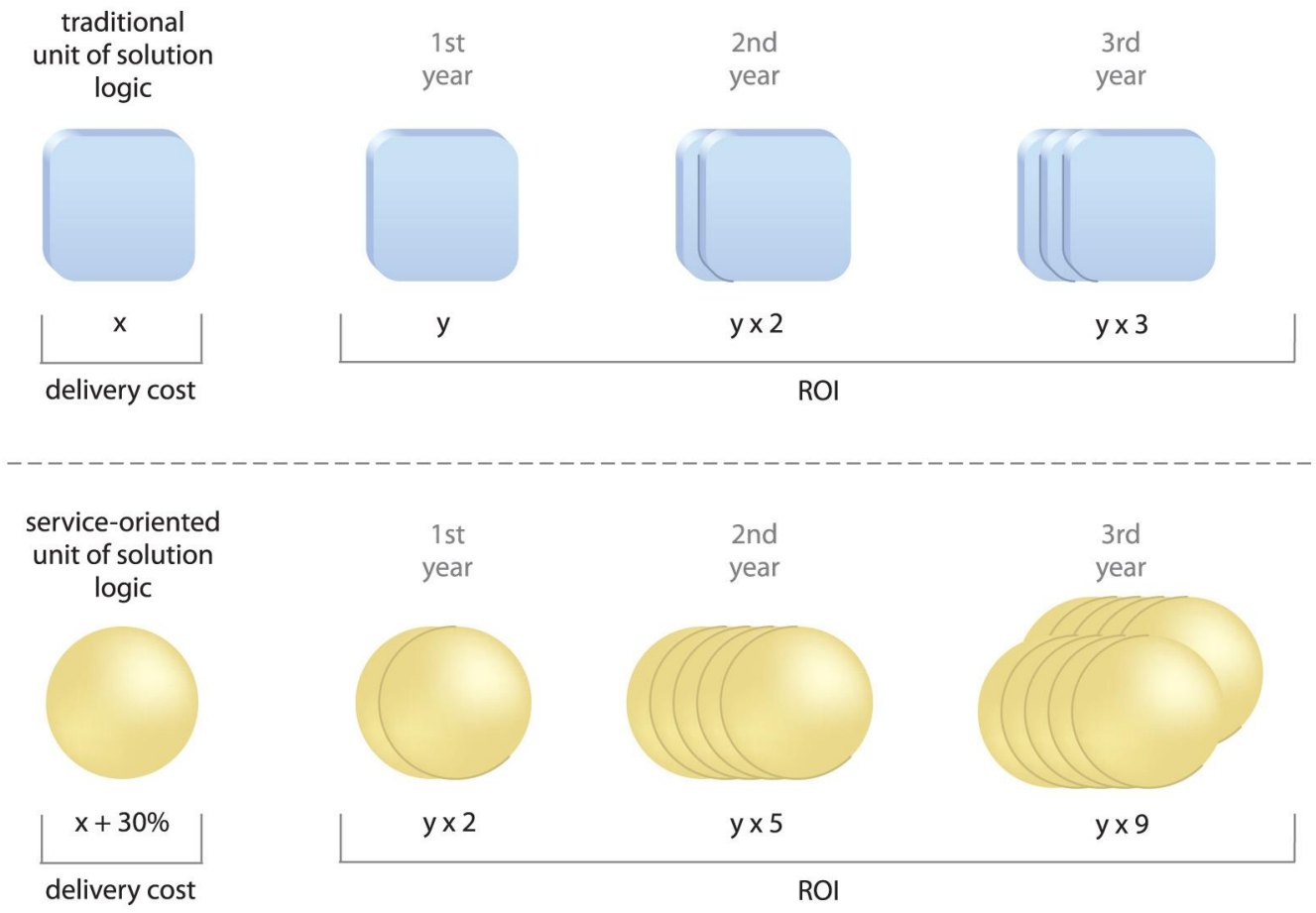
Increased Business and Technology Alignment



Services with business-centric functional contexts are carefully **modeled to express** and encapsulate corresponding **business models and logic**.

Once implemented, business models remains as a form of physical service

Increased Business and Technology Alignment



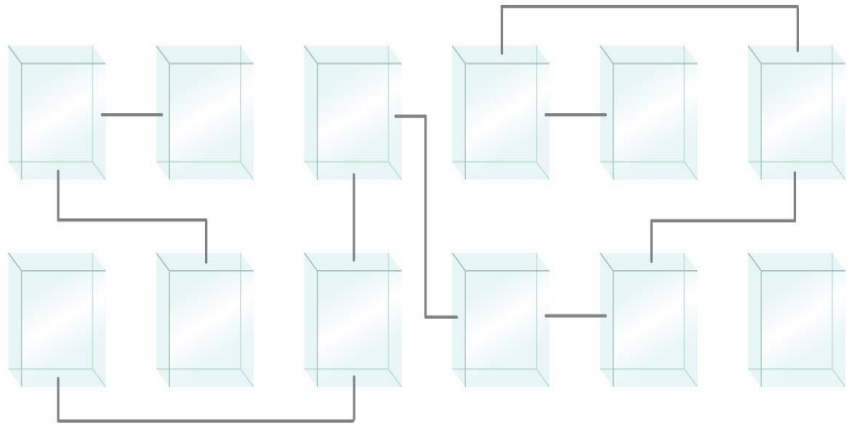
Any agnostic service can be **repurposed** many times to automate different business processes as part of SOA solution

Upfront expense is **invested** into every piece of solution considered as an IT assets for the purpose of **repeatable**

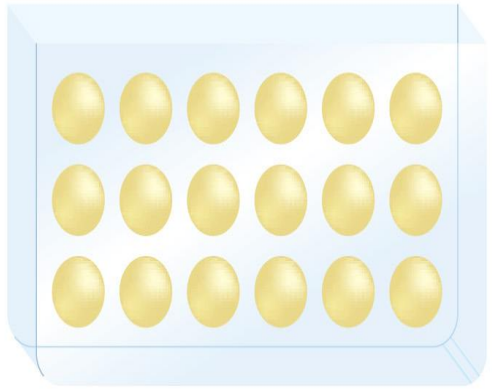
An example of the types of formulas being used to calculate ROI for SOA projects. More is invested in the initial delivery with the goal of benefiting from increased subsequent reuse.

Decreased IT Burden

enterprise with an inventory of integrated applications



the same enterprise with an inventory of services



Reduced Waste and Redundancy

Reduced Size and operational cost

Take Home

SOA is a distributed technology architectural model with distinct characteristics in support of realizing service-orientation

Service orientation is achieved by the application of service-orientation principles to the definition of services

The SOA ‘target state’ is an inventory of agnostic services that can be composed to achieve a diverse set of business process automation

SOA Promotes organizational agility and better alignment between technology and business processes

Readings

Thomas Erl, *Principles of Service Design* - Chapter 3 Service Oriented Computing and SOA (hardcopy available in library)

M Rosen, *Applied SOA : Service-oriented Architecture and Design Strategies* - Chapter 1 Realizing the promise of SOA (eBook available through library)

Any Questions?