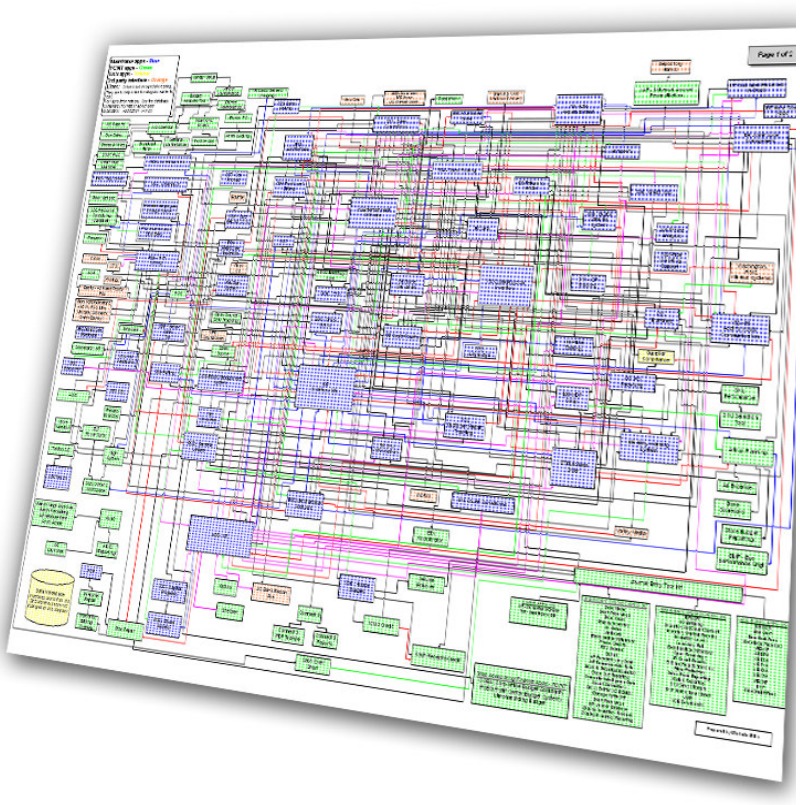
Value of SOA

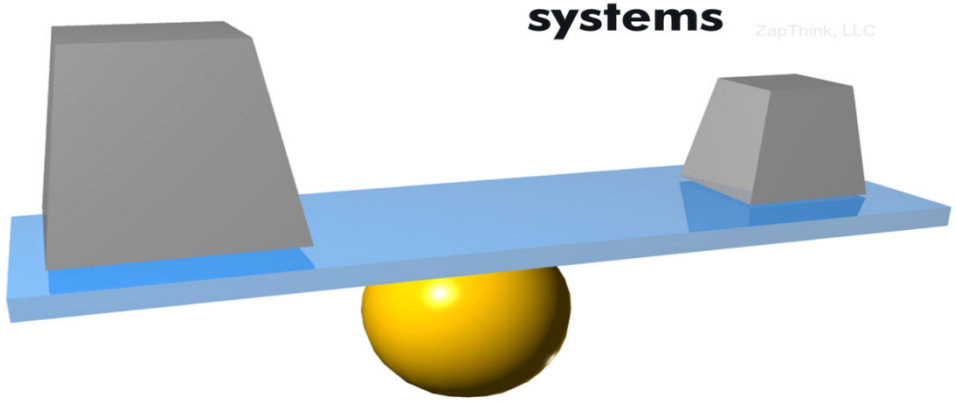
Prof. Vipul Dabhi
Department of Information Technology,
D. D. University.

Barriers to business flexibility and reusability

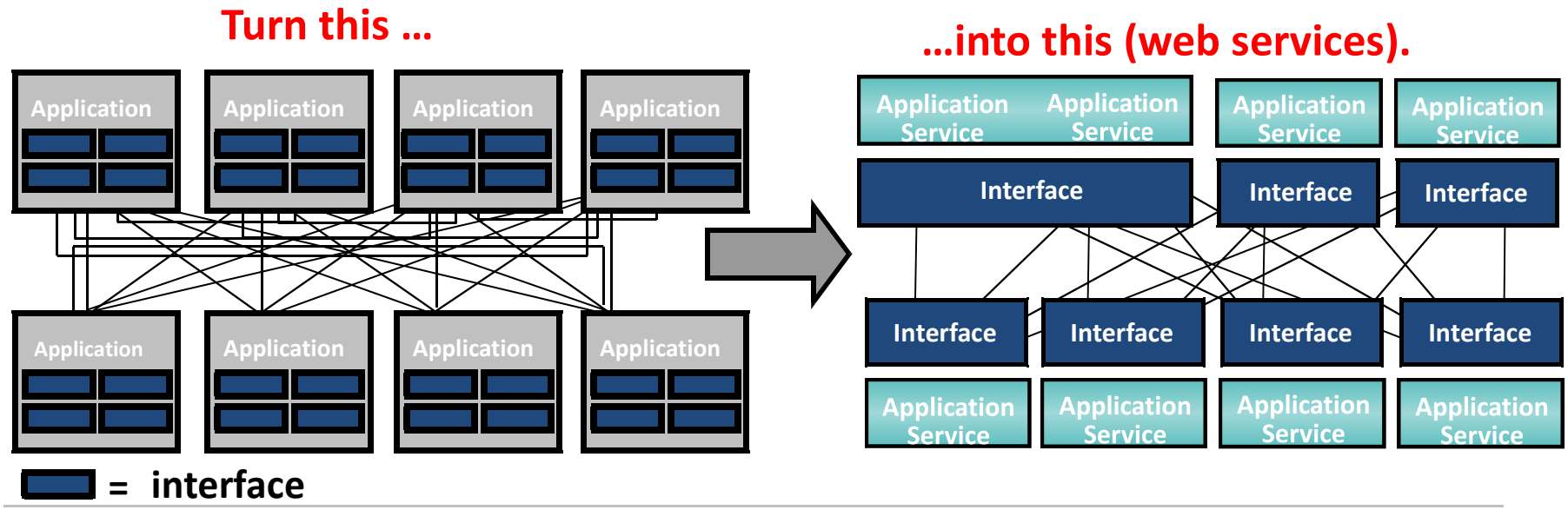
- **Lack of business process standards**
- **Architectural policy limited**
- **Point application buys to support redundant LOB needs**
- **Infrastructure built with no roadmap**



70% of IT development budgets are spent on integrating different systems



Web Services decouples interfaces from their applications…

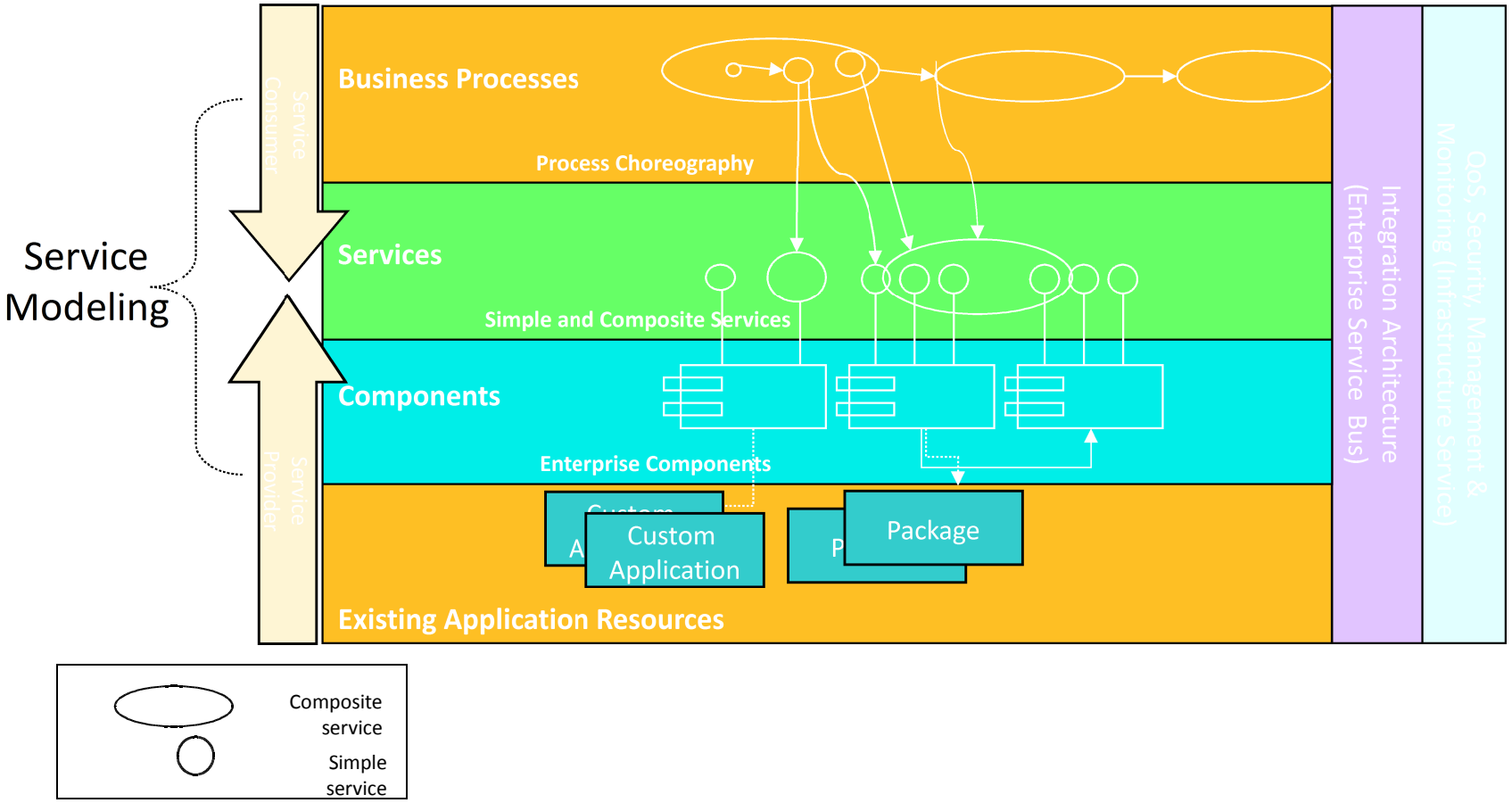


- ✓ Rich business abstractions describe the application interface
- ✓ Decouples the interfaces from the business applications
- ✓ The number and complexity of the interfaces is reduced
- ✓ Business applications and their interfaces become reusable

<u>But separate connection points still leaving bloated interfaces ….</u>

Service-Oriented Solutions – Architecture View

An SOA is composed of multiple layers.
At the heart of the SOA are Services, Components that realize services and Service Flows.



Achieving Business Agility

- Agile means able to move or change direction easily and quickly
- As the business transforms, the IT systems implementing transformations have to be agile enough to change quickly and cost effectively while still performing current business function.
- SOA provides business agility in three ways:
- ***Loosely coupled*** *services* are ones that no longer require the same technological implementation at each end of the connection. A simple mechanism connects applications regardless of the devices & location.
- ***Reuse*** The reuse of software, hardware, processes, code, services, and infrastructure provides some of the most measurable factors for an SOA return on investment (ROI) calculation
- ***Extensibility*** is defined as the ability to easily expand internal operations with new functions and to easily access organizations outside the enterprise.

Characteristics of a Service

- Services invoked through defined communication protocols
- Stress on interoperability and location transparency
- **Appear as a self-contained function**
- Use a **well-defined** interface
  - expose business functions
  - **hide underlying implementation** details
    - Loosely coupled
  - independent of any particular technology
- Services are not dependent on the context or state of other services.
  - Any dependencies between services are defined in terms of common business process, function and data models

Key principles of the SOA

1. **Loose coupling** - Services maintain a relationship that minimizes dependencies and only requires that they retain an awareness of each other.
2. **Service contract** - Services adhere to a communications agreement, as defined collectively by one or more service descriptions and related documents.
3. **Autonomy** - Services have control over the logic they encapsulate.
4. **Abstraction** - Beyond what is described in the service contract, services hide logic from the outside world.
5. **Reusability** - Logic is divided into services with the intention of promoting reuse.
6. **Composability** - Collections of services can be coordinated and assembled to form composite services.
7. **Statelessness** - Services minimize retaining information specific to an activity.
8. **Discoverability** - Services are designed to be outwardly descriptive so that they can be found and assessed via available discovery mechanisms.

SOA Benefits

- Saves money, time, and people
- Eliminates frustrations with IT
- Justifies IT investments
- Provides business executives with a clear understanding of what IT does and its value
- Eliminates IT's 6-6 answer (that is, the project will take 6 months and cost 6 figures)
- Provides a business and competitive differentiator

When a change in business process no longer requires a change to application programming logic, you have a successful SOA; your company has attained competitive business agility.

What will happen if you don’t adopt it?

An SOA could be the difference between the success and failure:

- Department, intra-company, or inter-company merger
- Acquisition
- Divestiture
- Product or service rollout
- Business partner, customer, or supplier addition
- Geographical expansion
- Competitive onslaught

When not to implement SOA?

- When you have a homogeneous IT environment
- When true real-time performance (nanoseconds response times) is absolutely critical
- When flexibility is not needed
- When tight coupling is needed
- If the organization isn’t ready for it