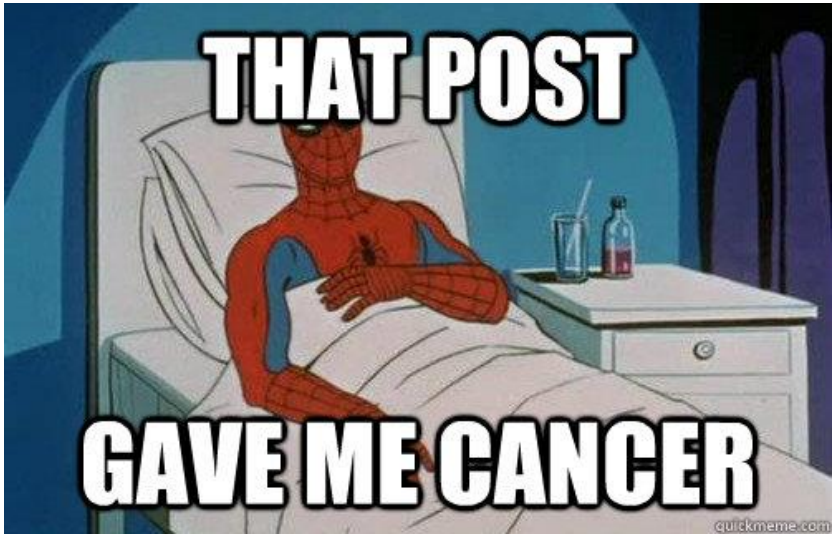
Service Oriented Computing

Joshua Tan

Alex Liu

Saeed Karimifard

Surgery Ward

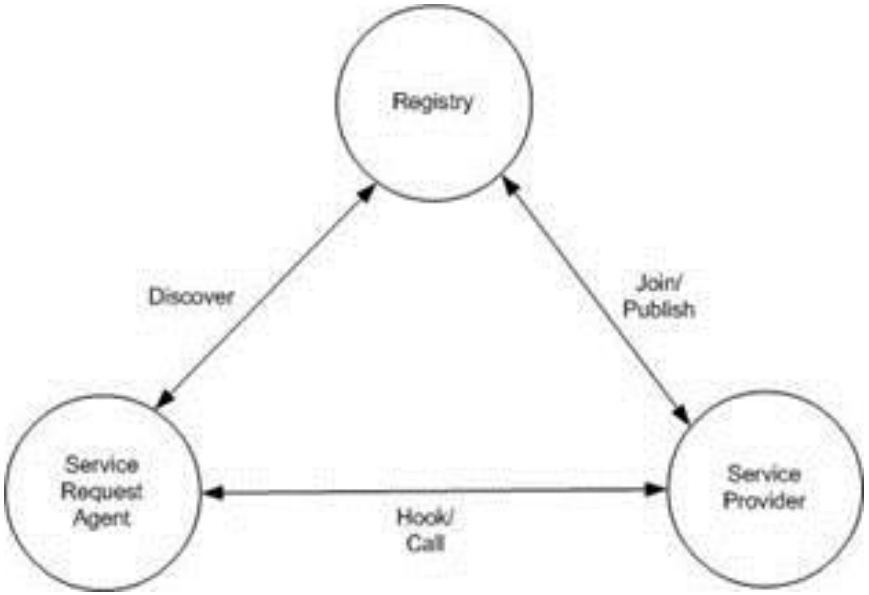


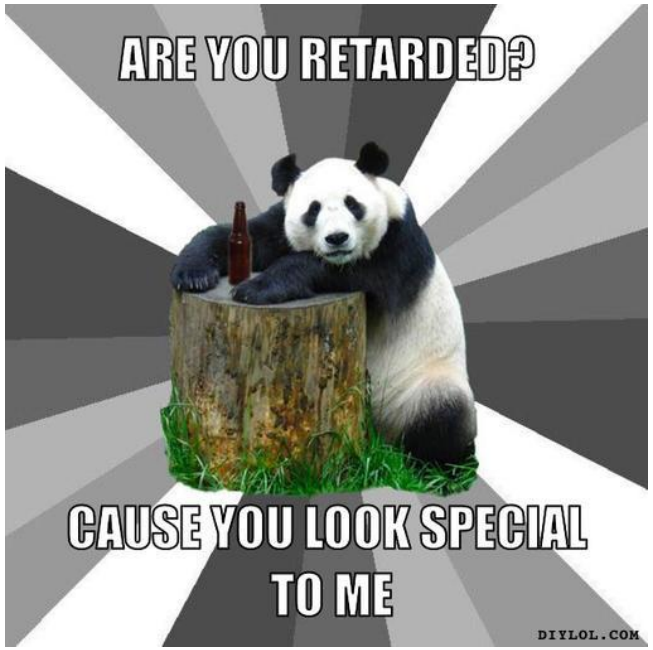
Surgery Ward

- Large-scale integration
- Various interconnected dependencies
- Requires:
  - –Effective communication
  - –Resource management

Service Oriented Computing (SOC)

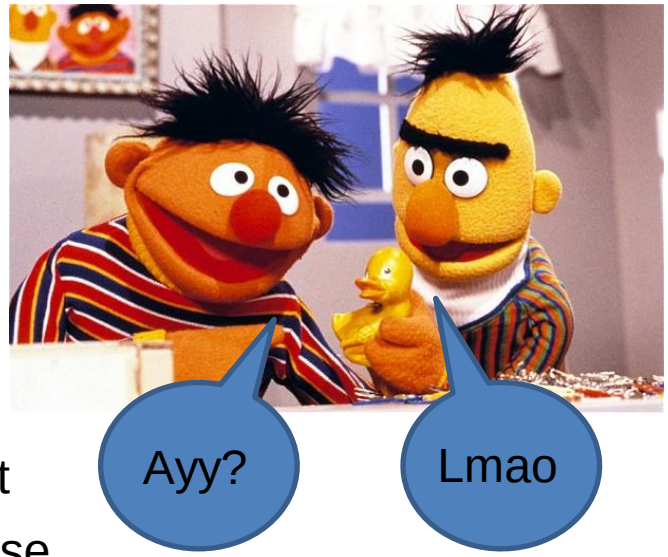
- System that emphasizes each component's autonomy and heterogeneity
- Modern approach to the open environment
- Modeled after Service Oriented Architecture





Service Oriented Architecture (SOA)

- Design pattern that stresses:
  - –Individualized components
  - –Cooperation between each part
  - –Through the use of standard interfaces
- Consists of these transactions:
  - –Service consumer issuing a service request
  - –Service provider returning a service response



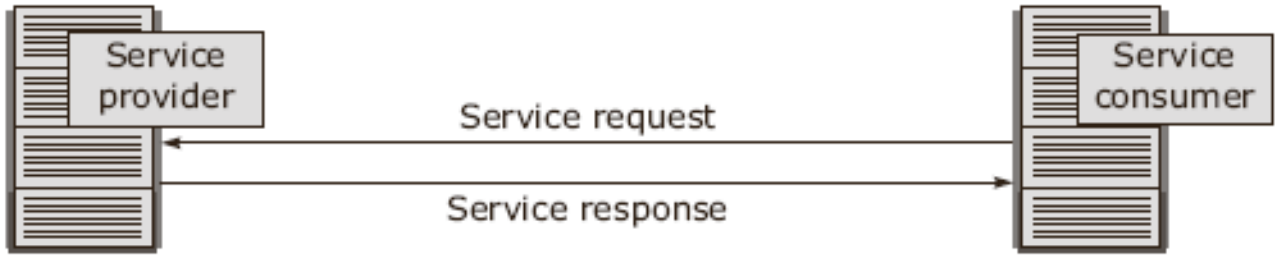
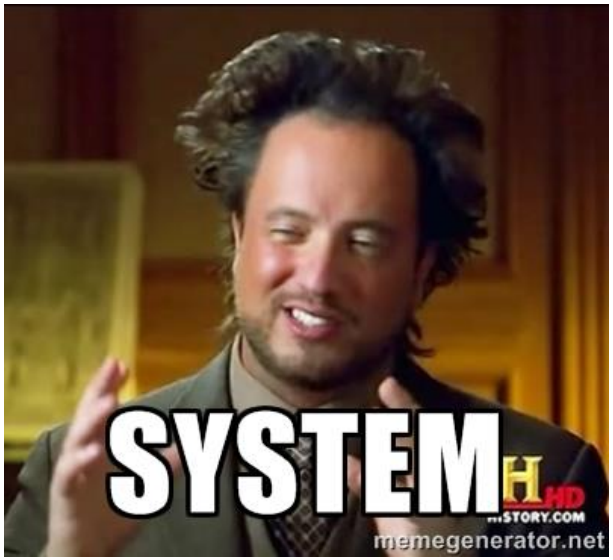


Figure 13-15 Basic service-oriented architecture

SOC vs SOA

SOC



SOA

Services

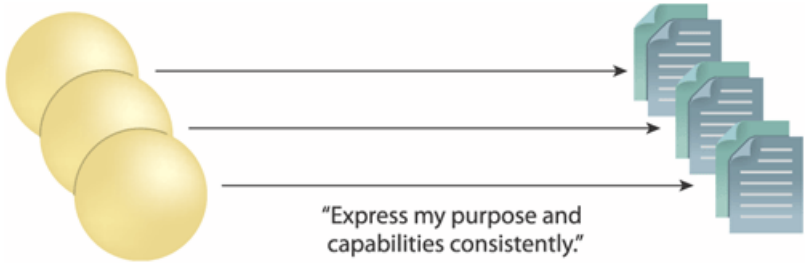
- Unassociated, loosely-coupled units that function cohesively
- e.g. submitting an online application, ordering tickets online, retrieving online bank statements

SOA Principles

- Standardized Service Contracts
- Loose Coupling
- Abstraction
- Reusability
- Autonomy
- Statelessness
- Discoverability
- Composability

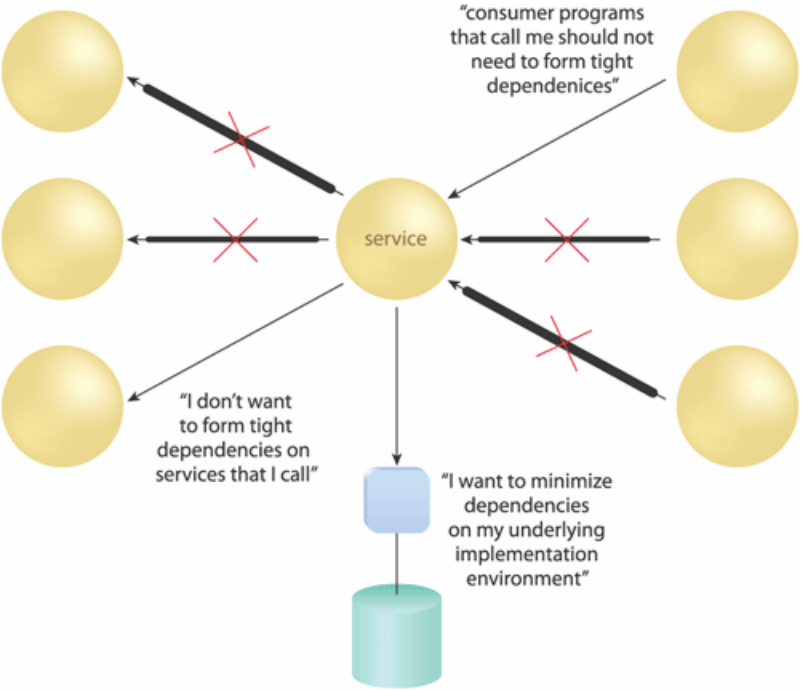
Standardized Service Contracts

- *Services within the same service inventory are in compliance with the same contract design standards*.



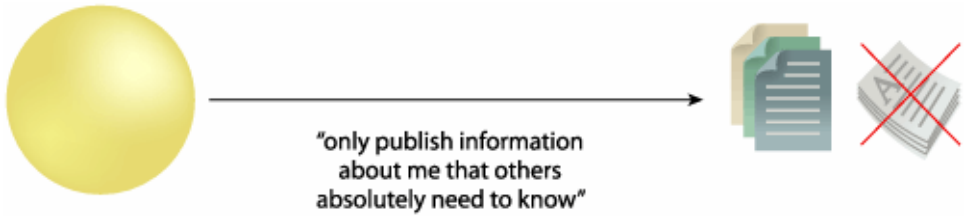
Loose Coupling

- *Service contracts impose low consumer coupling requirements and are themselves decoupled from their surrounding environment*



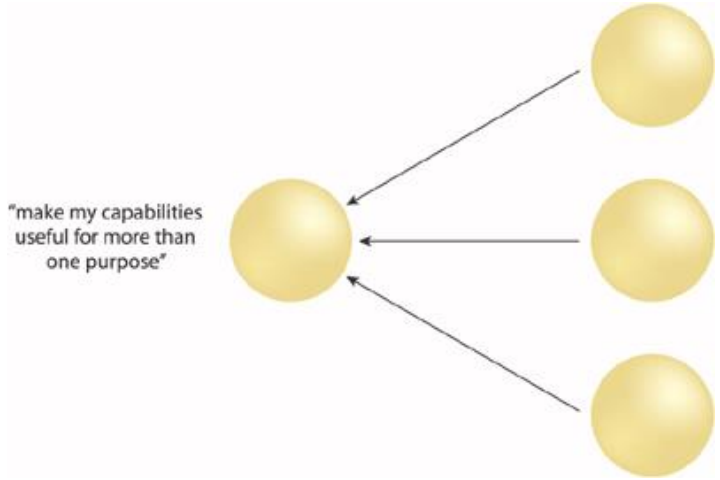
Abstraction

- *Service contracts only contain essential information and information about services is limited to what is published in service contracts*



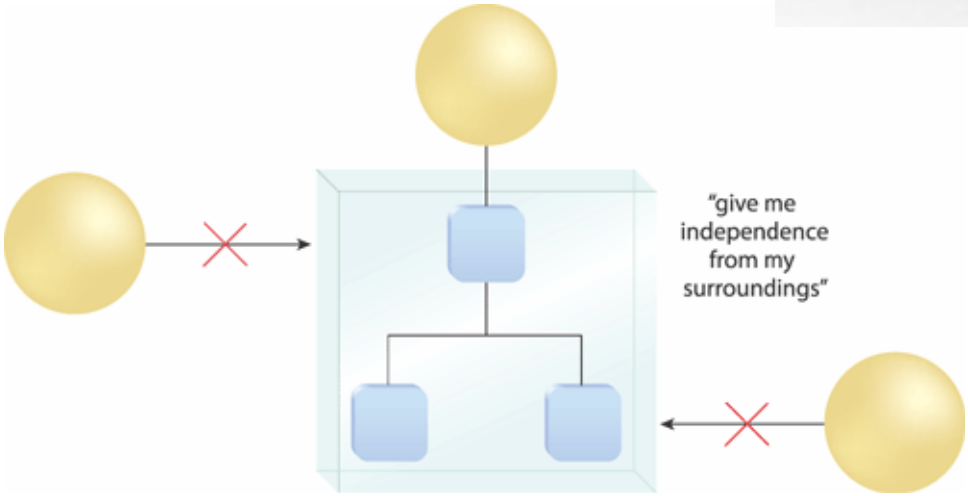
Reusability

- *Services contain and express agnostic logic and can be positioned as reusable enterprise resources*



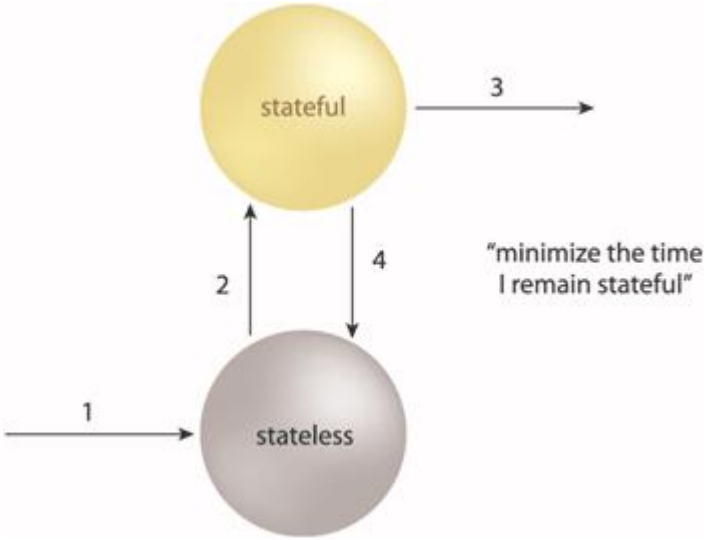
Autonomy

- *Services exercise a high level of control over their underlying runtime execution environment*



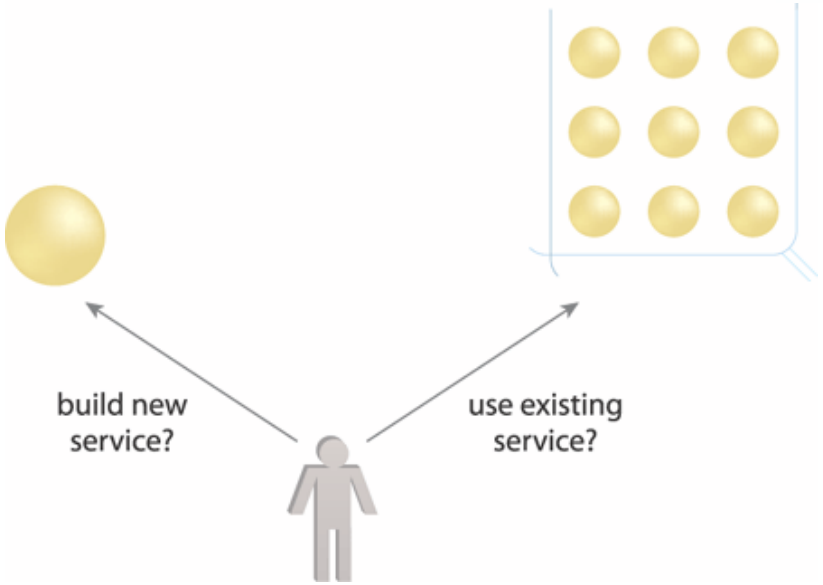
Statelessness

- *Services minimize resource consumption by deferring the management of state information when necessary*



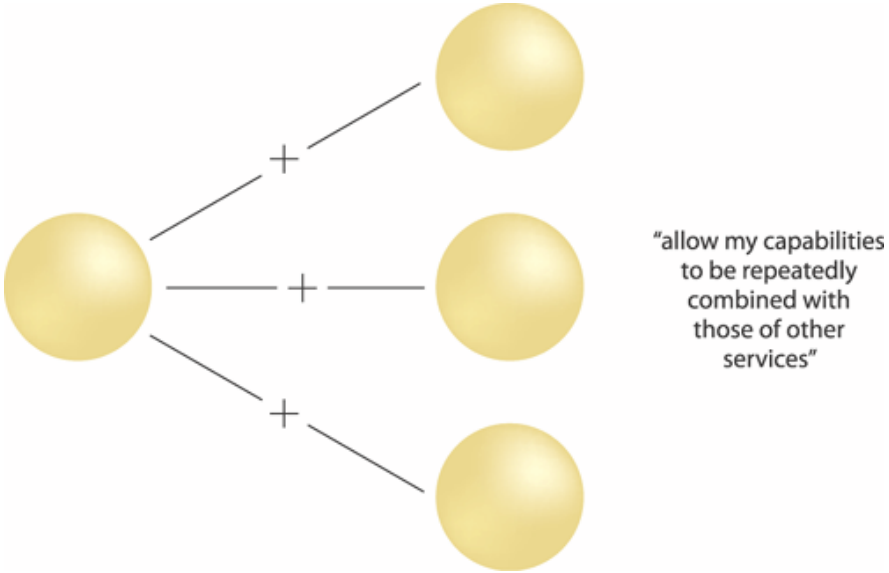
Discoverability

- *Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted*



Composability

- *Services are effective composition participants, regardless of the size and complexity of the composition*



"allow my capabilities to be repeatedly combined with those of other services"

Who is using SOC & SOA?

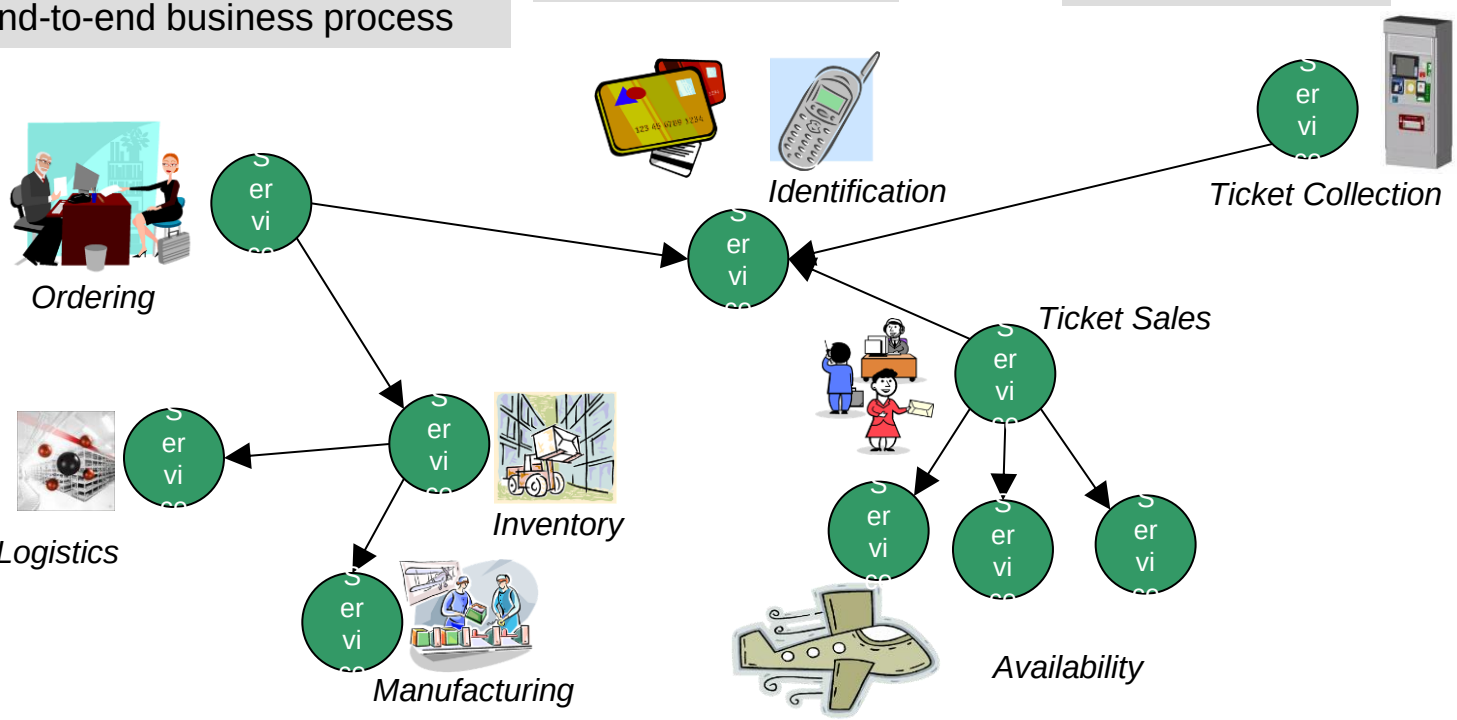
- *All major* computer corporations, including BEA, IBM, Microsoft, Oracle, HP, SAP, Intel, Cisco, Juniper, SAP, and Sun Microsystems, have moved towards the SOC paradigm.
- Furthermore, government agencies, such as the U.S. Department of Defense (DoD) and NASA, have adopted SOC.
- SOC is also being adopted by major computer users, including banks , retailers , airlines , travel agencies.

Why SOA?
To enable Flexible, Federated Business Processes

Enabling a virtual federation of participants to collaborate in an end-to-end business process

Enabling alternative implementations

Enabling reuse of Services

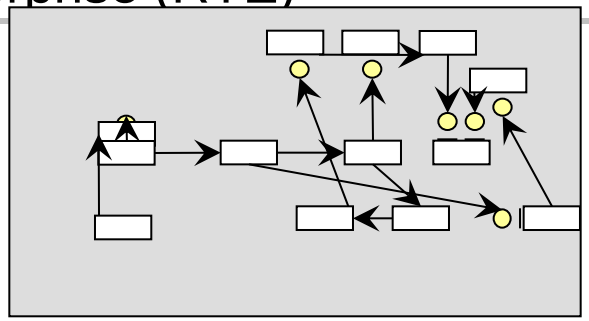


Enabling virtualization of business resources

Enabling aggregation from multiple providers

source:TietoEnator AB, Kurts Bilder

Why SOA? To enable Business Process Optimization and the Real Time Enterprise (RTE)



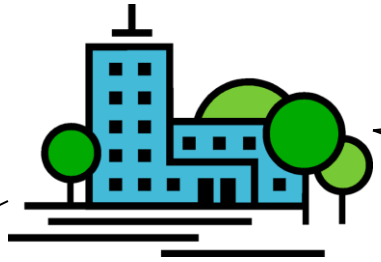
BPM Expressed in terms of Services Provided/Consumed

Seamless End to End Process

Service to Customers

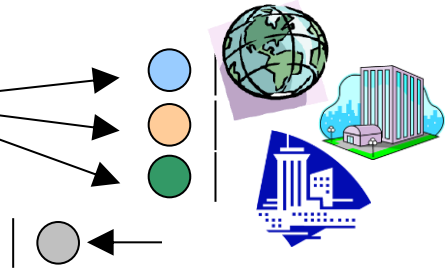
Smart Clients
Stores POS
Mobile
3rd Party Agents
Portal

Enterprise



Internal Systems

Service from Multiple Suppliers

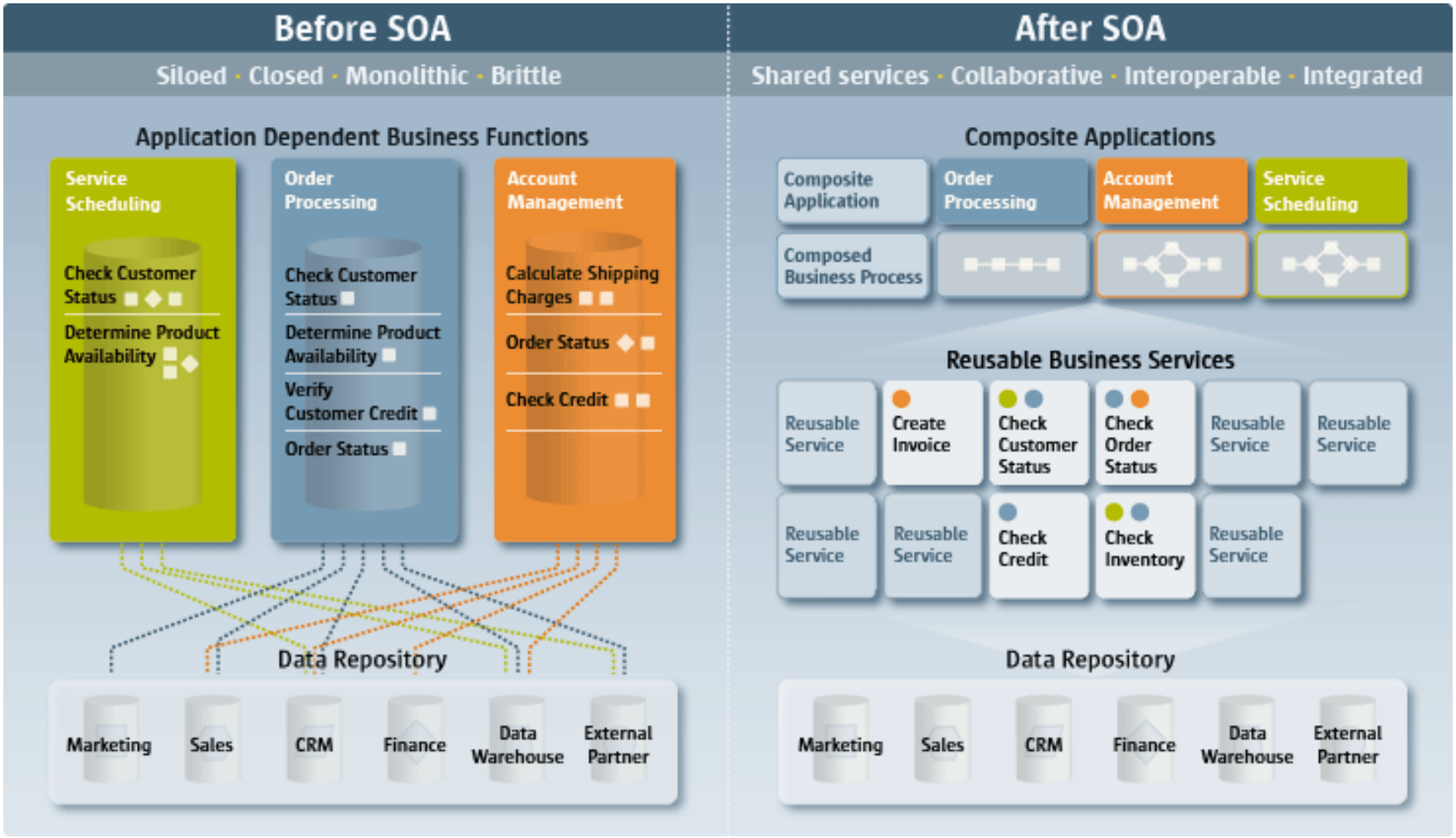


SOA Patterns: Single, Multi-Channel Service for consistency

source:TietoEnator AB, Kurts Bilder

SOA Pattern: Standardized Service provided by multiple suppliers

Before SOA – After SOA



source:IBM

Benefits of SOC

- Service-oriented computing enables new kinds of flexible business applications of open systems that simply would not be possible otherwise.

- Service-oriented computing improves the productivity of programming and administering applications in open systems.

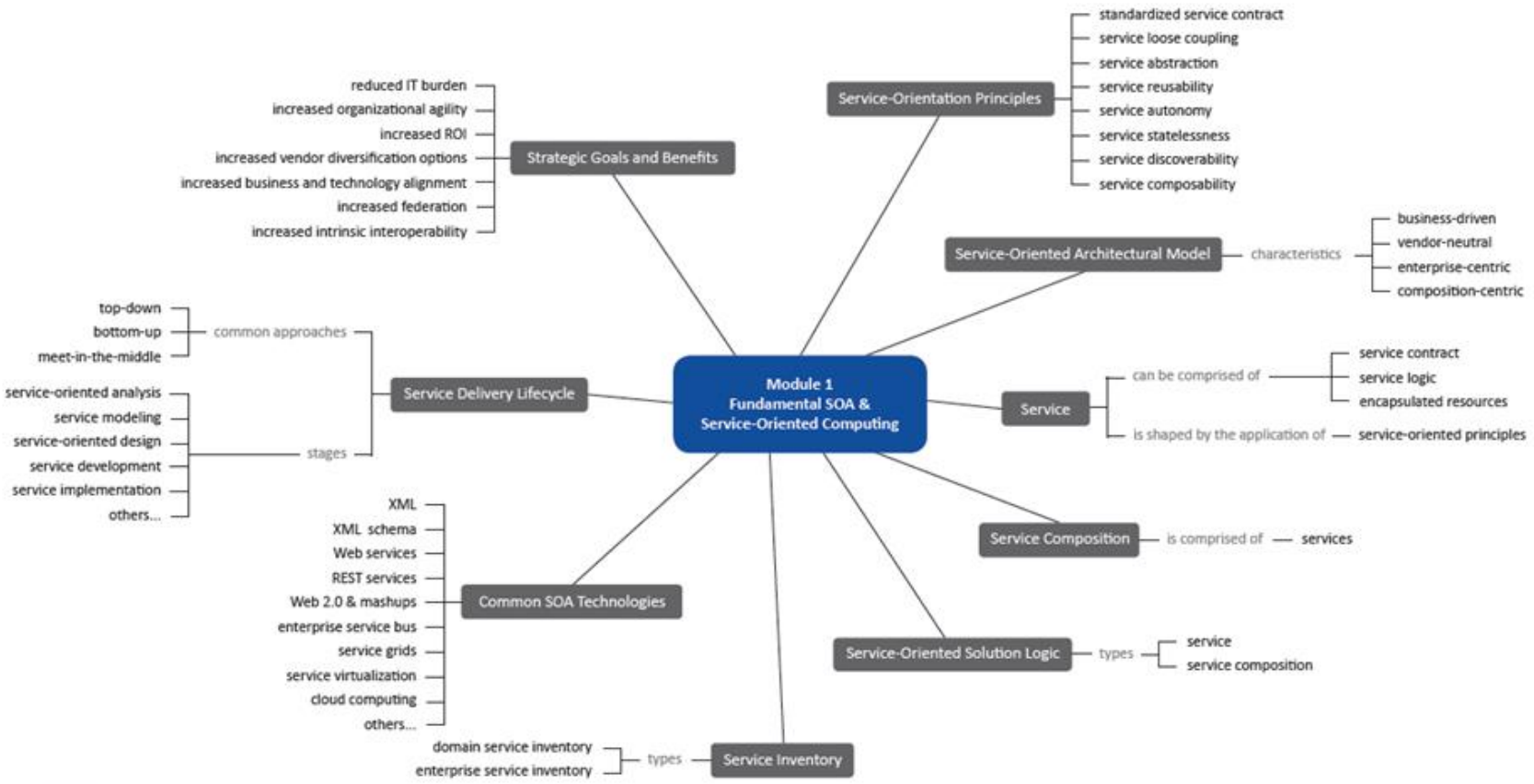
Grand Challenges

- Service Foundation
  - Dynamically (re)-configurable run-time architectures
  - End-to-end security solutions
- Service Composition
  - Composability analysis for repleceability, compatibility & conformance
  - Autonomic composition of services

Grand Challenges

- Service Management
  - Self-configuring services
  - Self-healing services
- Service Engineering
  - Design principles for engineering service applications
  - Flexible gap analysis techniques

SOC & SOA Overview



References

- S. Mahajan, S. Shah, “Distributed Computing”, Oxford University Press, 2011.
- T. Woods, “Service Oriented Architecture”
- M. P. Papazoglou, “Service Oriented Computing: a research roadmap”, International Journal of Cooperative Information Systems, Vol. 17, No. 2 (2008) 223–255.
- Singh & Huhns, “Service Oriented Compting”, chapter 5, July 2004

Q&A