

Arsitektur Teknologi Informasi

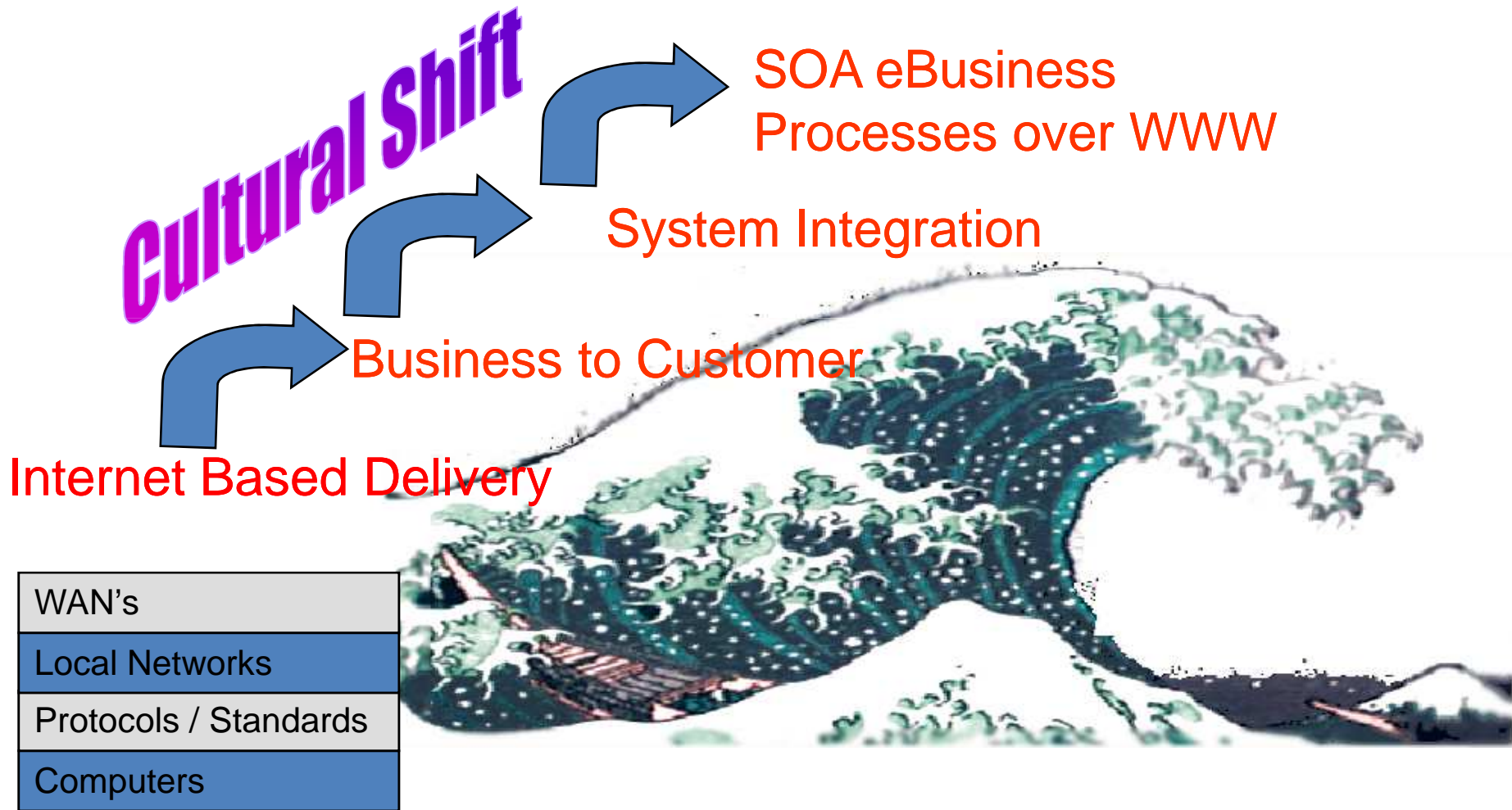
Services Oriented Architectures

Antonius Rachmat C, S.Kom, M.Cs

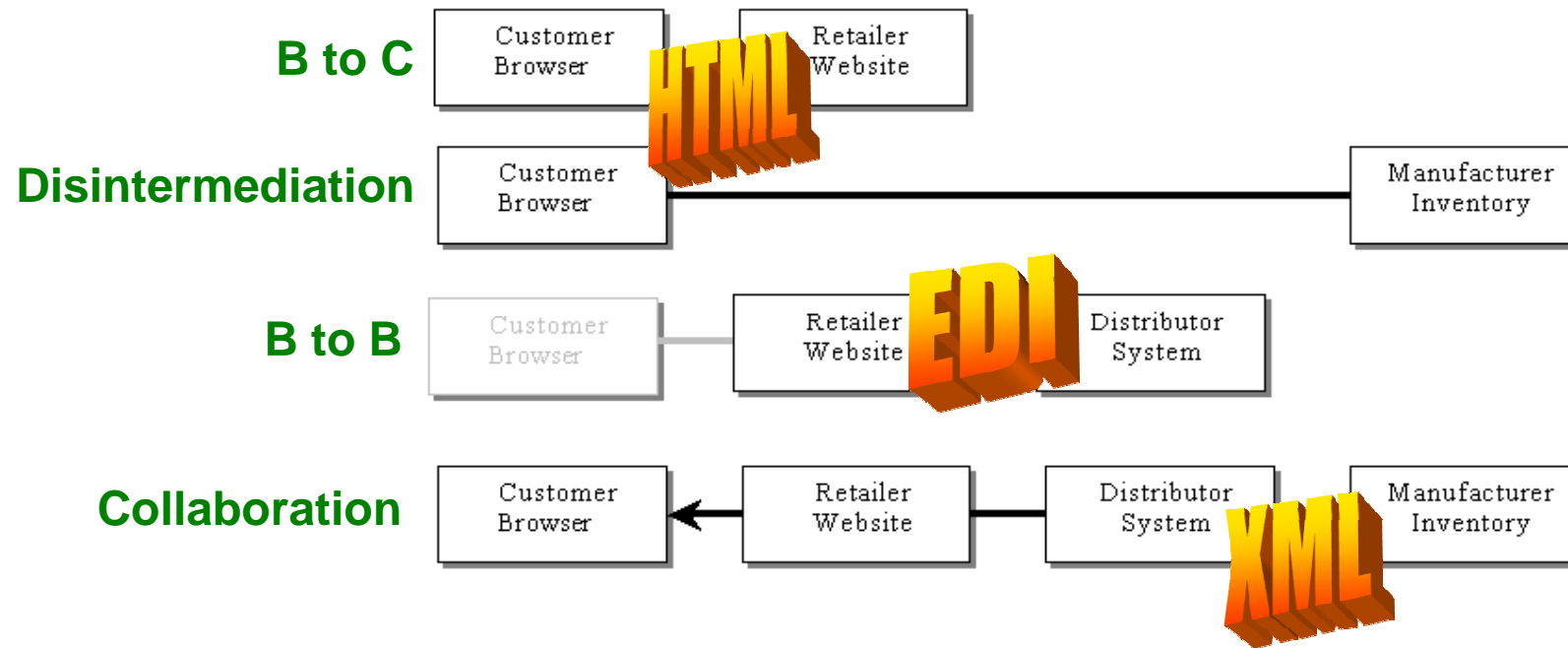
Tes Kecil I

- Bagaimana hasilnya?

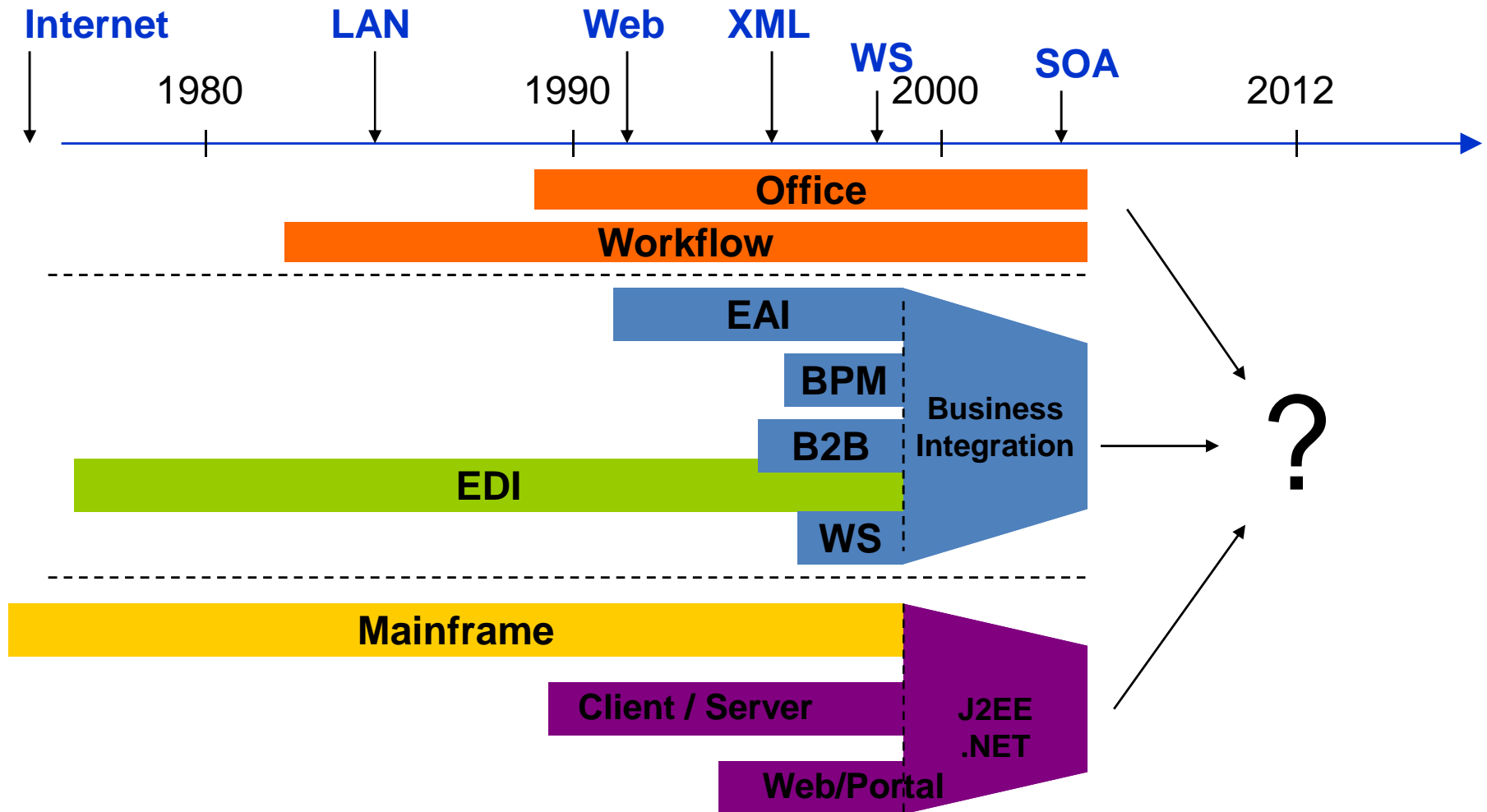
Business factors for Services



Business Interaction



Perkembangan SOA



Service Model expanded

- What if the two classes are **not on the same machine**? **Network!**
- How can one class **find** another? **Service Description**
- **How** can a class **send** its' information to the other class? **XML**
- Do the two classes have to be on the **same network**? **TCP/IP + HTTP + XML**

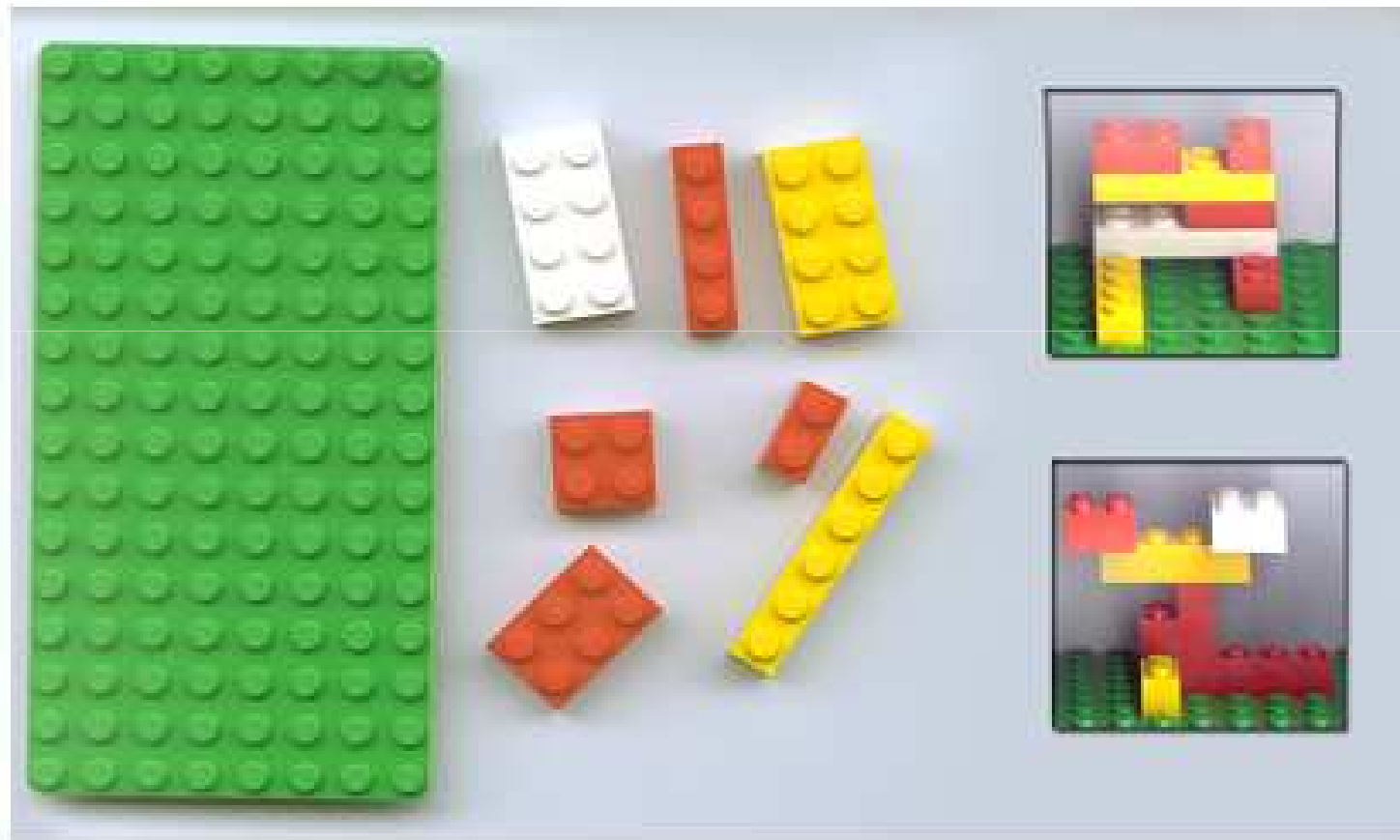
Service Oriented Architecture

- SOA adalah sebuah konsep software architecture yang mendefinisikan penggunaan **layanan** untuk mendukung kebutuhan pengguna software.
- Dari sisi prespektif bisnis:
 - A service-oriented architecture is a framework for **integrating business processes** and supporting IT infrastructure, standardized components—**services**—that can be **reused** and **combined** to address **changing** business priorities

Bentuk SOA

- SOA adalah sebuah arsitektur yang merepresentasikan **fungsi** dalam bentuk **layanan**
 - Mengapa **fungsi**?
 - Karena fungsi menunjukkan **abstraksi aktivitas** – sesuatu yang secara alami dilakukan oleh aplikasi/program, individu, dan organisasi
 - Mengapa **layanan**?
 - Karena layanan mengabstraksikan fungsi dan dapat menunjukkan bentuk **hubungan** yang bermakna antara 2 pihak yang berkomunikasi (requester dan provider)

SOA seperti puzzle



SOA dan Aplikasinya

- Ada dua arah pengembangan:
 - **Inward** → ke dalam institusi sendiri → integrasi sistem-sistem yang ada untuk membangun fungsionalitas yang lebih luas
 - Misal : untuk Supply Chain Management
 - **Outward** → memanfaatkannya sebagai perluasan sistem yang ada (external network, peluang bisnis, dsb)
 - Contoh: layanan pembuatan file PDF secara online

Penyebab SOA dan Tujuan SOA

- Pendorong berkembangnya SOA dari sisi bisnis:
 - Large scale **Enterprise** systems
 - **Internet** scale provisioning of services
 - Want to **reduce** the cost of doing business
- Tujuan
 - **Just-in-time integration** of applications by **discovering** and **orchestrating** network-available services

SOA dan Integrasi Aplikasi/Sistem

- SOA berfungsi sebagai **platform integrasi**:
 - SOA memisahkan antara **pesan/query/call** dengan **pemrosesan**
 - Pesan/query/call **distandardisasi** dan tidak dikaitkan dengan sebuah produk teknologi tertentu, sehingga bisa dikirimkan/diterima oleh siapapun
 - SOA memisahkan antara bagian **publik** dan bagian **privat**
 - Bagian **publik** dapat diakses oleh siapapun, berupa deskripsi tentang layanan yang ditawarkan
 - Bagian **privat** hanya bisa diakses oleh pemilik/penyedia layanan

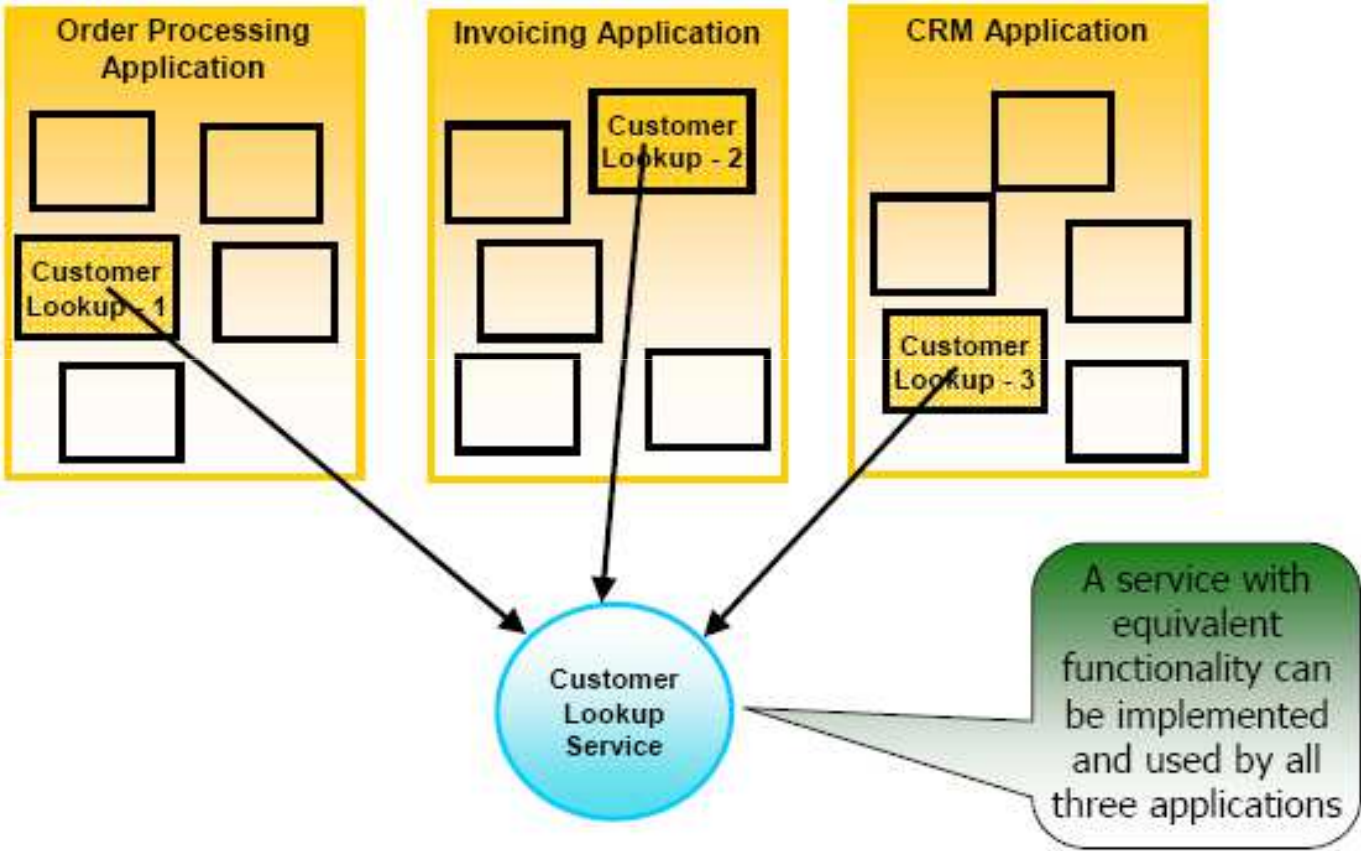
Sifat SOA

- SOA bersifat **behind the scence**,
 - SOA tidak terlihat secara langsung oleh client, SOA dihadapkan pada client melalui client UI
 - Digunakan untuk berkomunikasi antar aplikasi
- SOA merupakan suatu **service** yang “hanya menunggu” (**listen**) secara terus-menerus untuk digunakan.

Benefits of SOA

- **Better reuse of services**
 - Build new client functionality on top of existing Business Services
- **Well defined interfaces**
 - Make changes without affecting clients
- **Easier to maintain**
 - Changes/Versions are ok!
- **Platform Independence**
 - An enterprise can leverage its existing legacy applications that reside on different types of servers

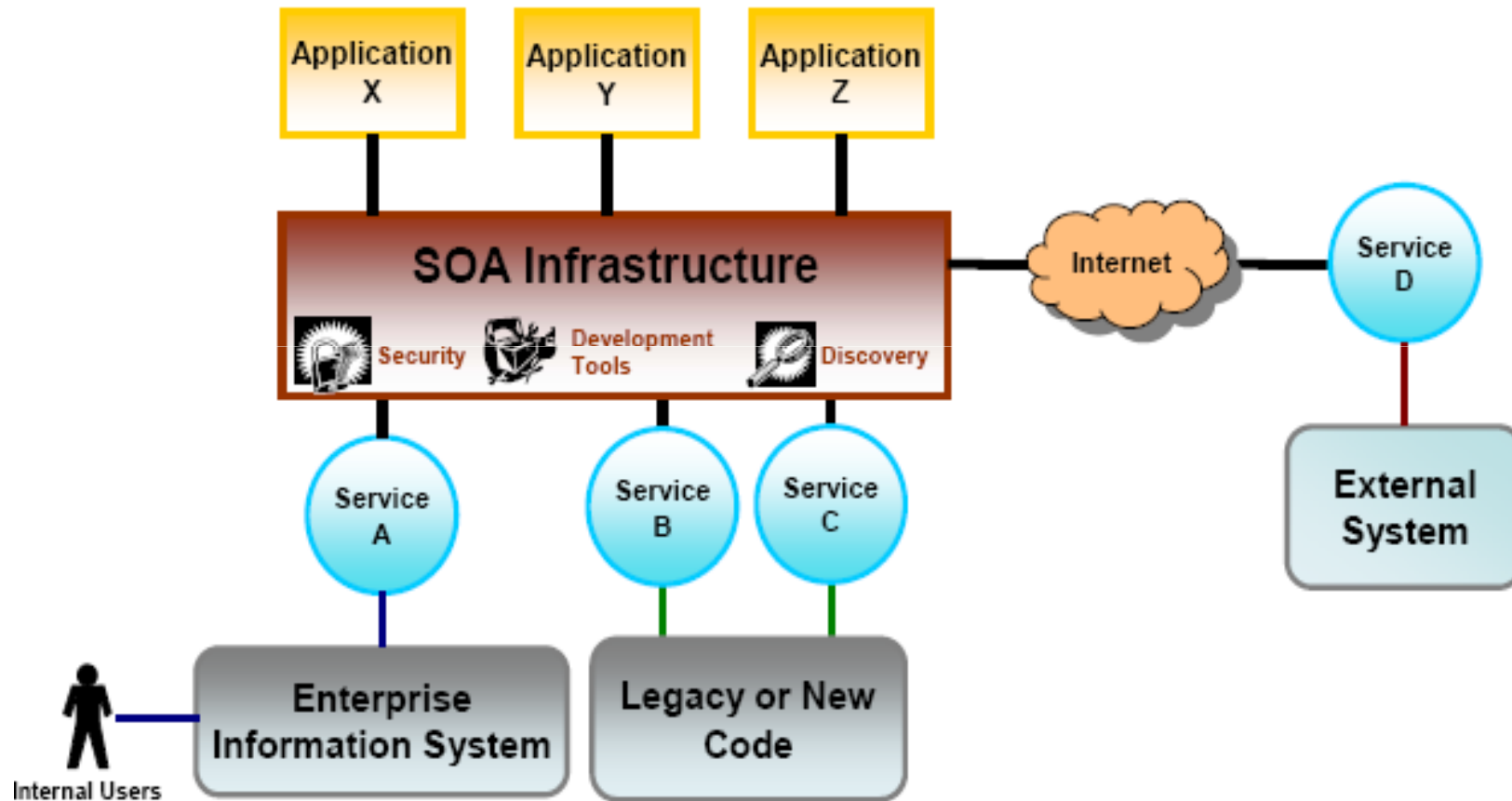
Reusability



Benefits of SOA (2)

- **Code Reuse**
 - the services can be **reused** in multiple applications
- **Location Transparency**
 - Web services are often published to a directory where consumers can look them up
- **Better scalability**
 - there can be **multiple** instances of the service running on different servers. This increases scalability
- **Higher availability**
 - Since the location of a service does not matter and you can have multiple instances of a service, it is possible to ensure high availability

Scalability

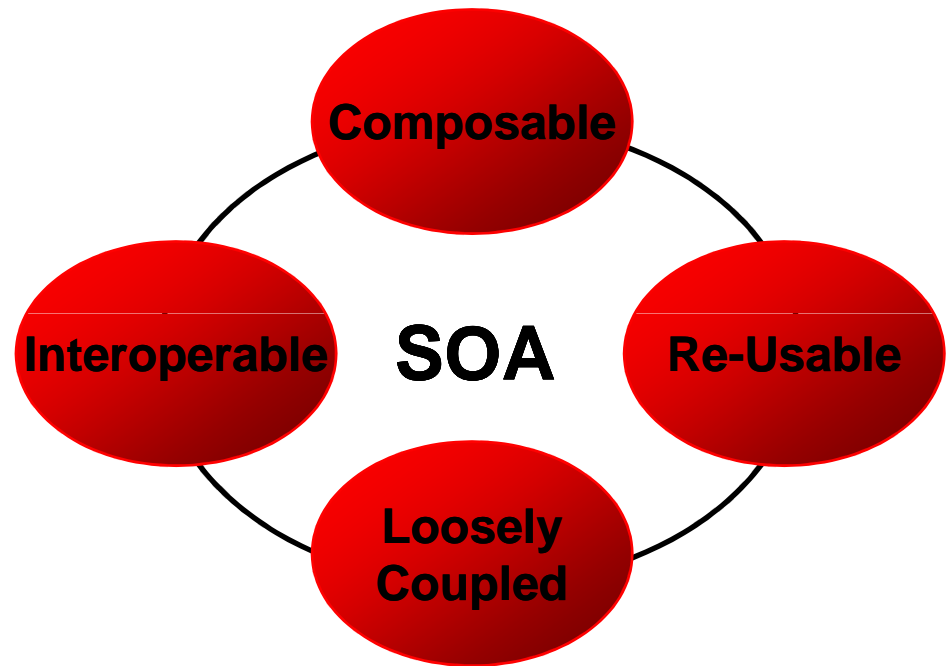


Prinsip-prinsip utama SOA

- **Standardized service contract** – by one or more service-description documents.
- **Service Loose Coupling**
- **Service Abstraction** – Beyond descriptions in the service contract, services hide logic from the outside world.
- **Service reusability**
- **Service autonomy** – Services have control over the logic they encapsulate.
- **Service statelessness** - Services minimize resource consumption by deferring the management of state information when necessary
- **Service discoverability** – Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted.
- **Service composability** – Services are effective composition participants.

Implementation of SOA

- Services have **platform independent**, self describing interfaces (XML)
- Messages are **formally defined** (WSDL)
- Services **can be discovered** (UDDI)
- Services have **quality of service characteristics** defined in policies (SOAP)
- Services can be **provided on any platform** (HTTP)
- Services can be **secured** (WS-Security)



Komponen SOA

- Layanan / **Service**
- Penyedia layanan / **Provider**
- Pemakai layanan / Consumer / **Requester**
- Tempat penyimpanan / **Registry**
- Pesan / **query** / call

Beberapa Istilah dalam SOA

- **Service:** suatu fungsi yang menerima satu atau lebih request dan mengembalikan satu atau lebih response yang terdefinisi dengan baik dengan menggunakan interface yang standar.
 - Service is **self-contained**. That is, the service maintains its own state
 - Interface contract to the service is **platform-independent**
 - Service can be **dynamically located and invoked**
 - Pengguna service **dapat menentukan** service yang diperoleh untuk digunakan dalam application logic mereka.

Elements of a service

- **Header**
 - Name
 - Version
 - Owner
 - Type: presentation/process/business/data/intergration
- **Functional**
 - Service operation : method
 - Invocation: how to invoke (SOAP/REST)
- **Non functional**
 - Security constraints
 - Quality of service, etc.

Beberapa Istilah dalam SOA

- **Provider:** bagian dalam SOA yang menyediakan services
 - Terdiri dari ≥ 1 service
 - Harus dapat ditemukan oleh requester
 - Mendaftarkan dulu ke registry
- **Requester:** bagian dalam SOA yang mencari dan menggunakan services
 - Dapat menggunakan lebih dari 1 service
 - Harus dapat mencari provider
 - Mungkin mencari di registry

Beberapa Istilah dalam SOA

- **Registry:** tempat penyimpanan informasi provider-provider yang menyediakan berbagai services
 - Berupa layanan yang listen terus menerus
 - Bisa berbayar atau gratis
- **Query:** mekanisme invocation service
 - Berupa permintaan service yang bersifat standar
 - Menggunakan format khusus agar dapat dibaca oleh service

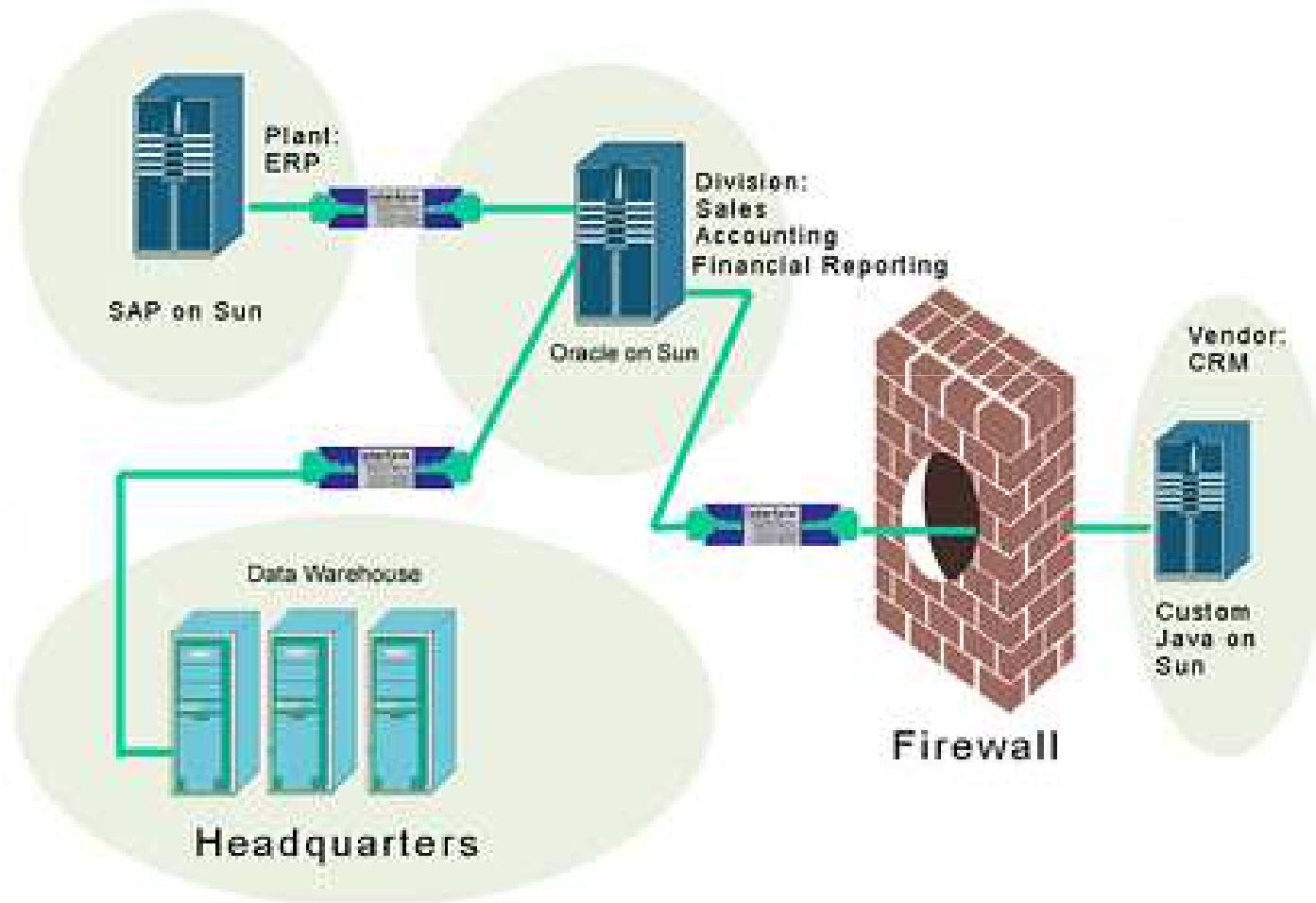
Build an SOA in 8 Steps

- (1) What **problem** are we trying to solve?
- (2) What aspects can be implemented as **services**? Old services? New services? Legacy wrappers?
- (3) **Track** services with registries and repositories
- (4) We need to **monitor** behavior, enforce **policies** & assess user satisfaction.
- (5) **Secure** the services. Using established standards, we need privacy, authentication, and authorization.

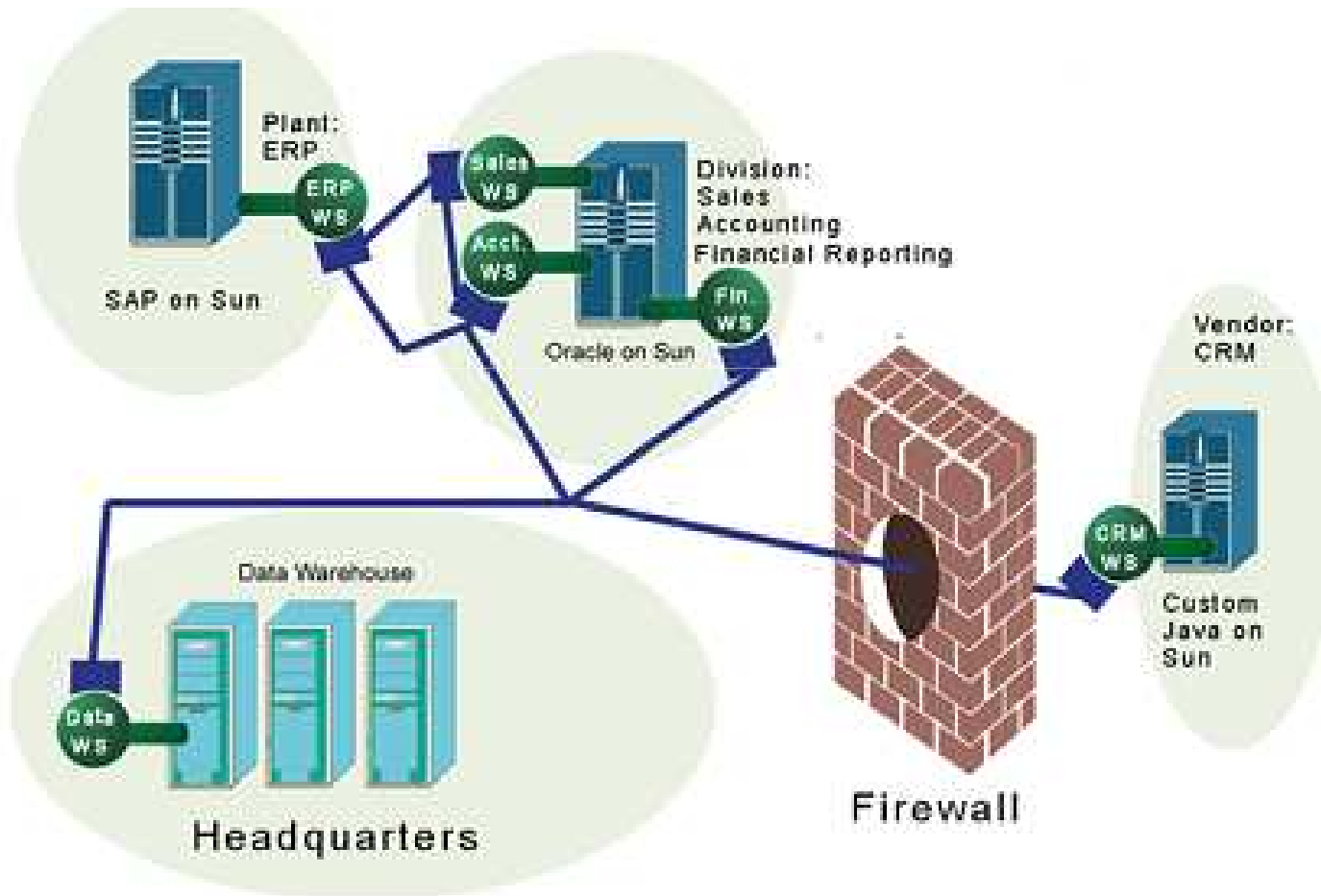
Build an SOA in 8 Steps

- (6) **Manage** the services. Are messages arriving on time? Is everything operating properly?
- (7) Virtualization through **mediation**. Are we free to move and change the services?
- (8) Design for **interoperability** through the adoption of **standards**.

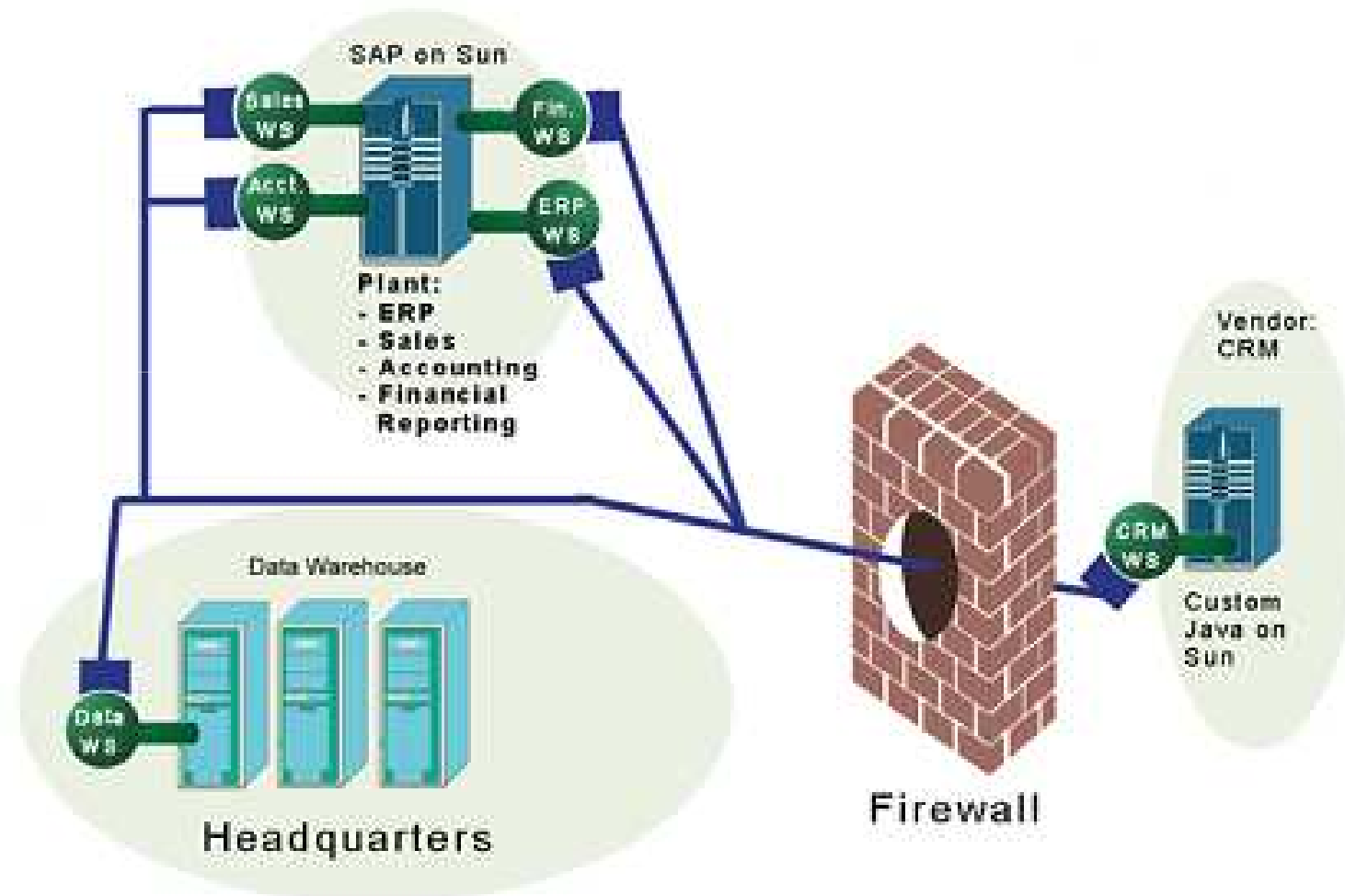
Non-SOA (Integration)



SOA – Integration



Changing SOA (Integration)



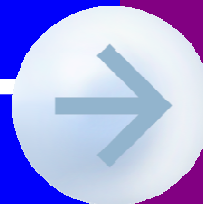
Shift From Application To A Service-Oriented Architecture

From

To

- **Function oriented**
- **Build to last**
- **Prolonged development cycles**

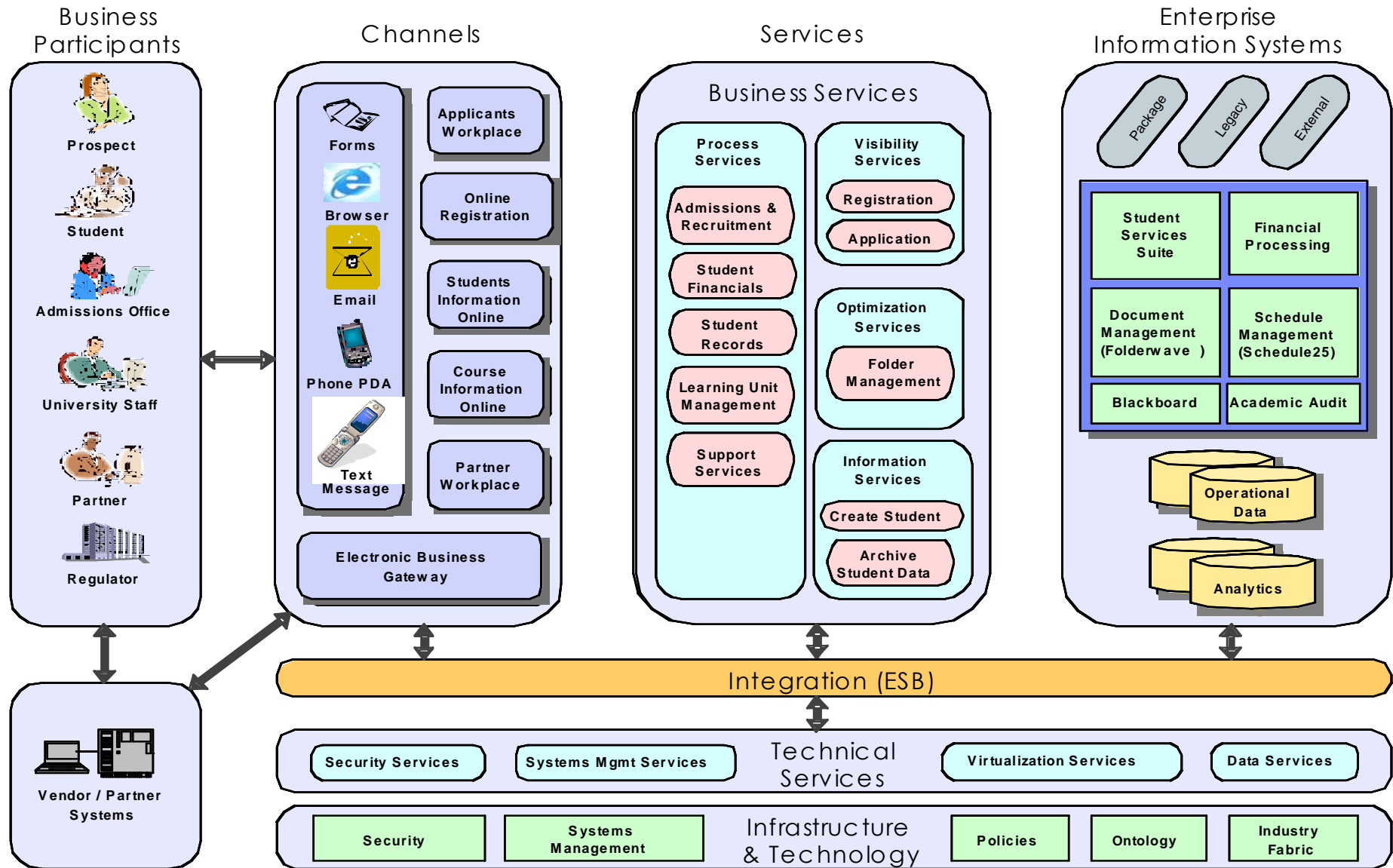
- **Coordination oriented**
- **Build to change**
- **Incrementally built and deployed**



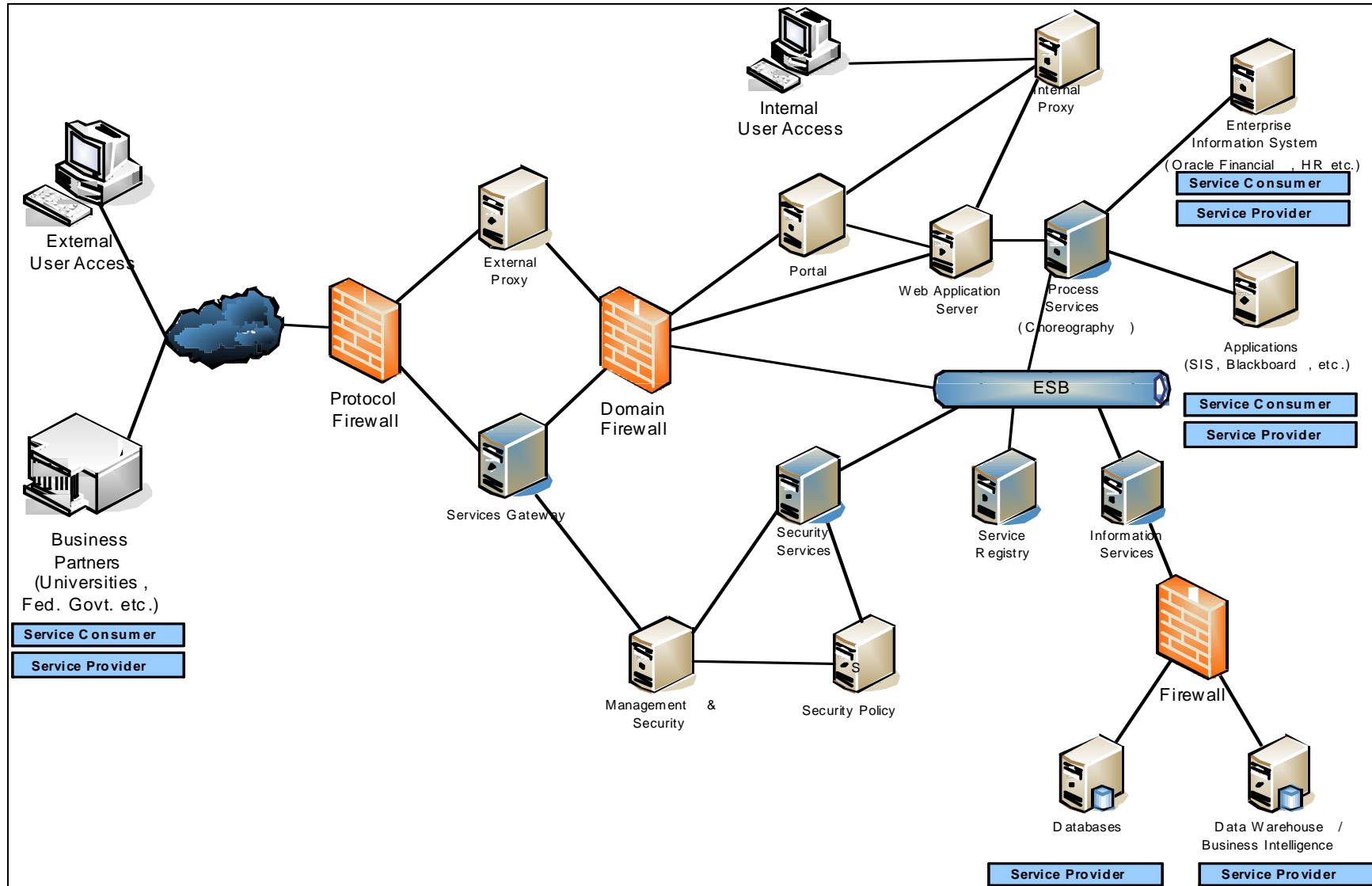
- **Application based solution**
- **Tightly coupled**
- **Function / Object oriented**
- **Known implementation**

- **Enterprise solutions**
- **Loosely coupled**
- **Message oriented**
- **Abstraction**

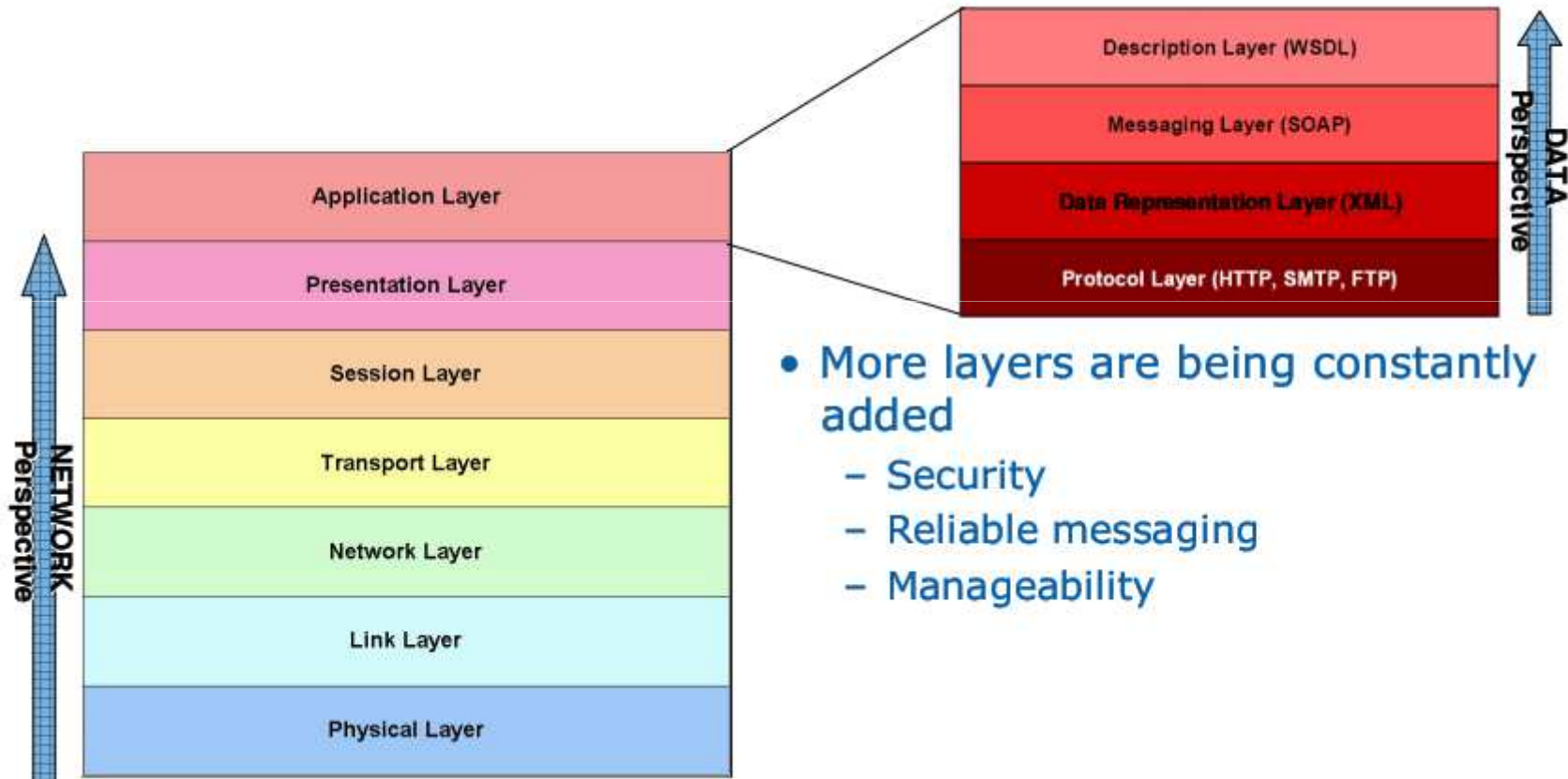
Enterprise View



IT Systems View



SOA at application layer of OSI



- More layers are being constantly added
 - Security
 - Reliable messaging
 - Manageability

SOA – Reference Architecture

Delivery Channels



Channel Mgmt

Service Mediation

Core Services



Integration Services

Resources



SOA challenges

- **Trust**
 - Data from a large number of services from different partners
- **Test**
 - All services work as designed?
- **Security**
 - Is the level of security is adequate?
- Continuous **updating, refinement and expansion**

NEXT

- Web Services (SOAP)